PAPER

A Novel Lambertian-RBFNN for Office Light Modeling

Wa SI^{†a)}, *Nonmember*, Xun PAN^{†b)}, *Student Member*, Harutoshi OGAI^{†c)}, and Katsumi HIRAI^{††d)}, *Nonmembers*

SUMMARY In lighting control systems, accurate data of artificial light (lighting coefficients) are essential for the illumination control accuracy and energy saving efficiency. This research proposes a novel Lambertian-Radial Basis Function Neural Network (L-RBFNN) to realize modeling of both lighting coefficients and the illumination environment for an office. By adding a Lambertian neuron to represent the rough theoretical illuminance distribution of the lamp and modifying RBF neurons to regulate the distribution shape, L-RBFNN successfully solves the instability problem of conventional RBFNN and achieves higher modeling accuracy. Simulations of both single-light modeling and multiple-light modeling are made and compared with other methods such as Lambertian function, cubic spline interpolation and conventional RBFNN. The results prove that: 1) L-RBFNN is a successful modeling method for artificial light with imperceptible modeling error; 2) Compared with other existing methods, L-RBFNN can provide better performance with lower modeling error; 3) The number of training sensors can be reduced to be the same with the number of lamps, thus making the modeling method easier to apply in real-world lighting systems.

key words: illumination modeling, RBFNN, lambertian function

1. Introduction

In lighting control systems, accurate data of artificial light are essential for the illumination control accuracy and energy saving efficiency. Although accurate and real-time illumination data can be achieved by using illumination sensors, this approach can only provide information in limited points (sensor positions) and is difficult to apply in large offices. By comparison, illumination models enable the illumination coefficients at any position inside the office to be calculated. However, luminosity is very sensitive to numerous factors such as building materials, furniture position, room decorations, etc., which makes illumination modeling a very difficult problem.

Lambertian function [1] is a widely used model for the illumination pattern of LED lamps and is used to provide illumination coefficients in some lighting systems [2], [3]. However, due to its extremely high calculation cost, the

lighting control systems using this model always neglect reflections of light occurring in the room, which need to be accounted for in the overall illumination in practice.

Many programs for lighting simulation (such as DIALux) have been developed that enable rough simulation for lighting projects and control strategy without real establishment of lighting systems [4]. When using these simulators to estimate the illumination coefficients in a lighting system, a virtual office model that simulates the real-office environment should be built, and then virtual sensors are placed in the model to collect the data. However, it is always difficult to simulate a true office with high accuracy by simulators (for example, it is difficult to simulate the texture of walls, windows, etc.). Thus, the problem of lighting simulators is not the ability to build a reasonable and detailed office model, but the ability to simulate a specific real office.

An ideal solution is to find a modeling method depending on accurate sensor data. Miki et al. developed such an illuminance distribution measurement system to visualize illuminance distribution provided by lighting system [5]. In this system, illumination sensors are placed with certain interval inside the office to measure the illuminance. Then, the cubic spline function is used for the interpolation between measurement points. However, this system needs too many sensors for big offices (interval of 0.5 m). Moreover, the cubic spline function also differs from the real distribution of illuminance.

As a popular method for solving real-world industrial problems concerning functional prediction and system modeling, artificial neural networks (ANN) are being applied to many lighting systems. Wang et al. have developed a holistic and scalable ANN model that can represent the complex relationship between dimming ratios of lamps and illuminance in target positions [6]. By training the ANN model using data from sensors in working tables (target positions), given the inputs of all the dimming ratios of lamps, the outputs will be the illuminance in target positions. However, the model can only calculate illuminance in limited positions (positions of training sensors), if the layout of any lamp or table is changed, training of ANN must be conducted again. Dong et al. employed hierarchical radial basis function (HRBF) network to implement coarse-to-fine modeling of illumination environment, so that the illuminance on continuous working plane can be calculated by measurements of limited sensors [7]. The lighting system then calculates the difference between real and target illuminance, and use a linear light transport model [8] to calculate how

Manuscript received October 6, 2015.

Manuscript revised March 5, 2016.

Manuscript publicized April 18, 2016.

[†]The authors are with Graduate School of Information, Production and Systems, Waseda University, Kitakyushu-shi, 808–0135 Japan.

^{††}The author is with Hakutsu Technology Ltd., Kitakyushu-shi, 808–0135 Japan.

a) E-mail: wsi8756@gmail.com

b) E-mail: xunpan@ruri.waseda.jp

c) E-mail: ogai@waseda.jp

d) E-mail: Khirai@hakutsu-tech.co.jp

DOI: 10.1587/transinf.2015EDP7411

to adjust the dimming ratio of the lamps. Since the HRBF model only calculates the integrated illuminance environment by using real-time sensor data, and cannot provide any information about contributions from each lamp to the target plane, it can only be used in feedback control.

In our previous research [9], we developed an illumination modeling method for lighting control, which can model the illumination distribution inside the office. The method uses data from illumination sensors to train Radial Basis Function Neural Networks (RBFNN). This method can be used to provide detailed illumination contribution from both artificial and natural light sources for lighting control algorithms. However, this method may provide totally wrong modeling result under certain training condition. Since this modeling failure cannot be predicted and avoided effectively, this instability is a fatal drawback that makes RBFNN modeling unsuitable for real-world applications.

This research is aiming at developing a modeling method for office lighting that 1) realizes overall modeling of both lighting coefficients and the illumination environment for an office by using small number of training sensors; 2) provides accurate and stable modeling data for lighting control systems. To fulfill these targets, we propose a novel Lambertian-RBFNN by adding some pre-knowledge of illumination pattern of lamps to RBFNN structure. Section 2 briefly introduces conventional RBFNN and its application for illumination modeling. Then, RBFNN is improved to be the proposed L-RBFNN and applied to illumination modeling in Sect. 3. In Sect. 4, simulations by DIALux are conducted to verify the performance of L-RBFNN in both single light modeling and office light modeling. Comparisons with other modeling methods are made and the results are analyzed. Finally, Sect. 5 makes the conclusions.

2. RBFNN Modeling for Office Lighting

2.1 Application of RBFNN to Illumination Modeling

RBFNN is a single hidden-layer feedforward neural network and is proved to be an effective universal approximator for non-linear functions in various applications [10]–[12]. Figure 1 shows the typical structure of RBFNN, which contains 2 layers: a radial basis layer and a linear layer. The radial basis layer has U neurons. Each neuron is a radial basis function (RBF), C_u is the center of neuron u and has the same dimension with the input $\mathbf{x}$, b_1 is the bias that is a given constant and affects the shape of the RBF. Each neuron calculates the distance between its center and the input vector, and then multiplies the distance by bias b_1 ; the result is then transferred by the RBF ($rb(x) = e^{-x^2}$). The linear layer multiplies the output of radial basis layer (a vector of size U) by the weight vector $\mathbf{w}$, and then adds the result by constant bias b_2 .

Thus in general, RBFNN can be represented as:

$$g(\mathbf{x}) = \sum_{u=1}^U w_u \times e^{-(b_1 \times \|\mathbf{x} - \mathbf{C}_u\|_2)^2} + b_2 \quad (1)$$

Fig. 1 Typical structure of RBFNN.

Fig. 2 Structure of single RBFNN for lighting coefficients.

To model the office lighting, the artificial illuminance (without daylight) E_i in any position i inside the office can be seen as the summation of lighting contribution from all the lamps:

$$E_i = \sum_j^n \alpha_{ij} \times d_j \quad (2)$$

where d_j is the dimming ratio of lamp j , and α_{ij} is the lighting coefficient, which is the illuminance contribution to target position i from lamp j when its dimming ratio is 100%.

Thus the illumination modeling problem can be solved by modeling the illumination from each lamp so that the illuminance in target positions can be calculated by given the lamps' dimming ratios. Our approach is to use single RBFNN to represent each lamp's illuminance distribution with maximum power and integrate them to be a hierarchical RBFNN to represent the overall artificial illuminance in the office.

Since the working tables have a standard height, we simplify the 3-D space in the office to be a 2-D working plane. Figure 2 shows the structure of single RBFNN for each lamp (lamp j). The input $\mathbf{x}$ is a 2 dimensional vector $[x_i, y_i]$ representing a position in the working plane, and the output is the lighting coefficient α_{ij} .

Figure 3 shows the structure of the hierarchical RBFNN. The inputs include the position $[x_i, y_i]$ and dimming ratios for all the lamps $[d_1, d_2, \dots, d_n]$. The output is the artificial illuminance in position i .

Fig. 3 Structure of hierarchical RBFNN for office light modeling.

2.2 Training of RBFNN

To collect the training data for RBFNN modeling method, the illumination contribution from each lamp to m different positions (sensors) in the working plane should be measured. For each lamp, at least one training sensor should be placed right under the lamp (where maximum illumination is rendered) in order to train the network correctly. We denote $\mathbf{tp}_i$ as the position of the i th training data and te_i as the illuminance of the i th training position.

A simple and effective training method for RBFNN is to set neuron centers the same with the training data, which means that $U = m$, $\mathbf{C} = \mathbf{tp}$, and then use te to calculate $\mathbf{w}$ and $b2$ by solving linear problem.

So when implementing the training, we initialize the RBF network with m (which is same as pair number of the training data) neurons, whose centers ($\mathbf{C}$ in Eq. (5)) are the same as the input training data. And then set $b1$ to be 0.8326, which equals to the square root of $\ln(2)$ so that when the distance between input and the center of neuron is 1, the output of the radial layer will be 0.5. After $b1$ and $\mathbf{C}$ are determined, we can get the output of the RBF layer. Then we can get the weights and bias ($\mathbf{w}$ and $b2$ in Eq. (5)) in the linear layer by solving a linear problem.

Totally, n RBFNNs $\{net_1, net_2, \dots, net_n\}$ are needed to model the illumination distributions of n lamps. The training data is an $n \times m \times 3$ matrix, which contains the information of positions of m training sensors (inputs) and the illumination contribution from each lamp to these sensors (outputs). Once all the networks are trained, training sensors are no longer needed.

3. A Novel L-RBFNN for Illumination Modeling

Although proved to be effective in learning unknown functional relationships, RBFNN still suffers from the disadvantage of “non-transparency” [13]. In our application of RBFNN to illumination modeling, it is found that RBFNN may provide totally wrong modeling results under certain training conditions. For example, when modeling the same illumination distribution, the results maybe totally different if we change one training sensor or add another training sen-

sor. Since all these training sensors are under same lighting scenario and do not have measuring error, this modeling failure cannot be predicted or avoided effectively. This instability will be fatal in real-world applications.

To overcome drawback of “non-transparency”, a good approach is to incorporate the pre-knowledge into the design of ANN [14], [15]. Thus to improve the performance of RBFNN in illumination modeling, we develop a novel L-RBFNN by employing two pre-knowledge, which are:

- 1) The theoretical LED illuminance pattern can be approximated by generalized Lambertian function;
- 2) As the distance with lamp increases, the illuminance approaches 0 lx.

3.1 Lambertian Neuron

Generally, the theoretical illumination coefficient α_{ij} , which is the illuminance from lamp j (at the location (x_{lj}, y_{lj})) to target position i (at the location (x_i, y_i)) in the working plane (at a distance h from the plane where lamps are located), can be calculated by the Lambertian function:

$$\alpha_{ij} = A \left(1 + \frac{(x_i - x_{lj})^2 + (y_i - y_{lj})^2}{h^2} \right)^{-(\mu+3)/2}$$

$$A = \frac{(\mu + 1)A_0}{2\pi h^2}$$

$$\mu = -\frac{\ln(2)}{\ln(\cos(\phi_{1/2}))}$$
(3)

where A_0 is the luminous flux of the light and $\mu > 0$ is the Lambertian mode, which is related to the semi-angle of the light beam at half power $\phi_{1/2}$. Lamps with $\phi_{1/2}$ higher than 90° will have same μ .

Thus we add a new Lambertian neuron in the RBFNN to represent the main illuminance pattern of the lamp. In the Lambertian neuron, the distance between its center $\mathbf{C}_0$ and the input vector is calculated and the result is than transferred with a simplified Lambertian function:

$$l(x) = \left(1 + \frac{x^2}{h^2} \right)^{-(\mu+3)/2}$$

$$\mu = -\frac{\ln(2)}{\ln(\cos(\phi_{1/2}))}$$
(4)

3.2 Modified RBF Neurons

Since the illuminance approaches 0 lx when the distance with lamp increases, $b2$ should be 0 and is no longer needed for all the neurons. Moreover, in order to approximate the illuminance pattern more accurately, $b1$ is no longer a constant but a vector $\mathbf{b}$ of size U (each b_i for each neuron i) trained during the training procedure.

Thus together with the Lambertian neuron, the structure of L-RBFNN can be depicted as Fig. 4, and L-RBFNN can be represented as:

Fig. 4 Structure of L-RBFNN for lighting coefficients modeling.

$$f(\mathbf{x}) = w_0 \times \left(1 + \frac{\|\mathbf{x} - \mathbf{C}_0\|_2^2}{h^2} \right)^{\frac{\ln(2)}{\ln(\cos(\phi_{1/2}))} - 3} + \sum_{u=1}^U w_u \times e^{-(b_u \times \|\mathbf{x} - \mathbf{C}_u\|_2)^2} \quad (5)$$

3.3 Training of L-RBFNN

To train the L-RBFNN, besides the $n \times m \times 3$ matrix that contains the information of positions of m training sensors (inputs) and the illumination contribution from each lamp to these sensors (outputs), the distance between working plane and each lamps h is also needed. (We use only one h since typically the lamps are always mounted in same plane. But the feature of L-RBFNN also allow different h for the lamps.)

For each lamp, the training procedure of one L-RBFNN is described as follows:

- 1) Train the Lambertian neuron.
 - i) Firstly, find the training point with maximum illuminance, which is supposed to be the position of the lamp.
 - ii) Then set the center of Lambertian neuron $\mathbf{C}_0$ the same with the point's position (input).
 - iii) Set the weight of Lambertian neuron w_0 the same with the point's estimation (output) because the maximum output of simplified Lambertian function is 1.
 - iv) If the $\phi_{1/2}$ of the lamp is known, directly set the $\phi_{1/2}$ in Lambertian neuron.
If not, calculate MSE between training measurement and neuron output by using different $\phi_{1/2}$ ranged from 1° to 90° . Then, set $\phi_{1/2}$ in Lambertian neuron by the optimal $\phi_{1/2}$ with minimal MSE.
- 2) Train RBF neurons.

Until the MSE (Mean Squared Error) between network output and training measurement reaches the target set by user, repeat the following steps:

- i) Build a new RBF neuron, find the training point with maximum error and set its position as the new RBF neuron's center (no matter if it has been chosen as another neuron's center).
- ii) Set the weight of the new RBF neuron w_u the same with the error of the selected training point.
- iii) Calculate MSE by using different b_u ranged from 0 to higher value until the MSE begins to increase. Then, set b_u for the u th RBF neuron by the optimal b_u with minimal MSE.

From the steps above we can conclude that the Lambertian neuron represents a rough and theoretical illuminance distribution of the lamp, and the RBF neurons compensate the differences caused by reflection, shadows, etc. Thus this ANN structure reflects the prior knowledge of lights (position and some physical parameters), and has higher transparency than the conventional RBFNN.

4. Verification by Simulations

To evaluate the performance of modeling methods, we use illumination simulating software DIALux [4] to collect training data and verifying data (detailed information will be introduced in each subsections). DIALux is a software for illumination simulating, and can provide accurate modeling for office structure and lighting environment considering various factors. Since our objective is to check the modeling performance of overall illuminance distribution by information of limited points, DIALux models can provide enough information. Moreover, compared with real office environment, DIALux can provide repeatable simulation environment for comparison of different methods and will not be affected by measuring error.

To evaluate the simulation results, we mainly focus on the following aspects:

- Difference with the verifying data (Δ)
The verifying data (illuminance in different positions) are collected in DIALux in 64×64 positions in the office models. $\mathbf{E}'$ is the vector that contains all the DIALux measurements in 4096 positions while $\mathbf{E}$ is the vector containing the illuminance in these positions calculated by modeling methods. Δ is the vector containing all the absolute differences between verifying data and modeling results.

$$\Delta_i = |E_i - E'_i|, \quad i = 1, 2, \dots, 4096 \quad (6)$$

Δ_{max} and Δ_{min} are the maximum and minimum value in Δ respectively while mean difference with the verifying data Δ_{mean} is the average value of Δ .

- Normalized Difference (Δ')

- When analyzing modeling for office lighting system, Δ' is the vector containing all the absolute differences between verifying data and modeled data normalized by the verifying data.

$$\Delta'_i = \Delta_i / E'_i, \quad i = 1, 2, \dots, 4096 \quad (7)$$

- When analyzing modeling for single lamp, Δ'_{mean} is calculated by: $\Delta'_{mean} = \frac{\Delta'_{mean}}{E'_{max}}$ (E'_{max} is the maximum value in E'). This is because when modeling single lamp, some values in E' are 0.

Δ'_{max} and Δ'_{min} are the maximum and minimum value of Δ' respectively while Δ'_{mean} is the average value of Δ' .

Research has shown that people are insensitive to environment illuminance change within 20% and are willing to accept up to 30% change [16]. As for desk-surface illuminance, it is also shown that people can only perceive change within +6% to -8% of current illuminance [17]. Thus to evaluate our modeling methods, we use relatively strict targets, which are 6% and 8% for the modeling to make sure the error cannot be perceived by human eyes. And our goal of modeling accuracy is to achieve unperceivable difference with verification data, which is a normalized mean difference lower than 6% to 8%.

For comparison against proposed L-RBFNN, besides the conventional RBFNN, we also check the following three methods:

- Lambertian function modeling method (LF)
We use the Lambertian function (Eq. (3)) to calculate the illuminance in working plane by given $\phi_{1/2}$ and A_0 (luminous flux).
- Normalized Lambertian function modeling method (NLF)
In Eq. (3), we set A directly by the maximum sensor measurement, which means that we only use Lambertian function to represent the shape of distribution curve.
- Cubic spline function interpolation method (CSI)
We use cubic spline function to interpolate between the data of training sensor.

4.1 Performance of L-RBFNN in Single Light Modeling

To evaluate the performance of different modeling methods for single lamp, we use three totally different lamps listed in Table 1. The lamps are mounted in the middle of a $6.1m \times 6.1m$ DIALux office model. Figure 5 shows the illuminance distribution of the 3 lamps in the office's working plane (DIALux data).

As for the interval of training data, we choose 3 training scenarios which are:

- 1) Interval of 0.5 m (which are suggested for CSI method in [5])
169 sensors are needed for collecting the training data.
This training scenario provides precise data for all the

Table 1 3 different lamps for simulations.

No.	lighting type	$\phi_{1/2}$	h	luminous flux
1	Indirect Lighting	$>90^\circ$	1.53 m	8840 lm
2	Direct Lighting	40°	1.95 m	4320 lm
3	Direct Lighting	160°	2.08 m	2160 lm

Fig. 5 Illuminance distribution of the 3 lamps in the office's working plane.

Fig. 6 Changing of MSE during the L-RBFNN training procedure.

modeling methods but needs too many sensors.

- 2) Interval of 1 m
49 sensors are needed for collecting the training data.
- 3) Interval of 1.5 m
25 sensors are needed for collecting the training data.

4.1.1 Training Procedure of L-RBFNN

To train the L-RBFNN, our target MSE is set to be 0.001. We test L-RBFNN modeling methods by training procedure with and without given $\phi_{1/2}$. Figure 6 shows some examples of the changing of MSE during the training. The results show that:

- 1) The iteration number needed for both training procedures increases as the training data amount increases.
- 2) The 2 training procedures need approximately the same iteration time, but giving $\phi_{1/2}$ slightly increase the converging speed.

Table 2 Trained $\phi_{1/2}$ in L-RBFNN.

Lamp	Actual $\phi_{1/2}$	Equivalent $\phi_{1/2}$	Interval	Learned $\phi_{1/2}$	Difference of $\phi_{1/2}$
1	> 90°	90°	0.5 m	89°	1°
			1 m	89°	1°
			1.5 m	88°	2°
2	40°	40°	0.5 m	36°	4°
			1 m	37°	3°
			1.5 m	40°	0°
3	160°	90°	0.5 m	90°	0°
			1 m	90°	0°
			1.5 m	90°	0°

Fig. 7 Changing of L-RBFNN output during the training procedure.

- Although the iteration time becomes very large when the size of training data is large, both methods can reach the MSE less than 1 within 100 iterations, Which means that the L-RBFNN can achieve high modeling accuracy within very few training iterations.

Table 2 shows the $\phi_{1/2}$ in Lambertian neuron trained without given $\phi_{1/2}$ of the lamps. Nearly half of the 9 training scenarios have no difference with the actual $\phi_{1/2}$ value and the difference are all lower than 4°. The comparison results prove that the L-RBFNN is capable of learning the $\phi_{1/2}$, which is essential for the shape of lamp's distribution curve, thus proves that L-RBFNN structure can reflect the prior knowledge of some physical parameters of lights.

Figure 7 shows the changing of L-RBFNN outputs during the training procedure for Lamp 2 with data interval of 0.5 m (no given $\phi_{1/2}$). As can be seen from the figure,

Table 3 Modeling of different methods using different data intervals compared with verifying data.

Method	Interval	Lamp 1		Lamp 2		Lamp 3		Average	
		Δ_{mean}	Δ'_{mean}	Δ_{mean}	Δ'_{mean}	Δ_{mean}	Δ'_{mean}	Δ_{mean}	Δ'_{mean}
LF	n/a	65 lx	18.7%	38 lx	8%	9 lx	9.1%	37 lx	11.9%
NLF	n/a	10 lx	2.8%	16 lx	3.3%	4 lx	3.8%	10 lx	3.3%
CSI	0.5 m	5 lx	1.4%	5 lx	1%	1 lx	1.1%	4 lx	1.2%
	1 m	7 lx	2.1%	6 lx	1.2%	1 lx	1.2%	5 lx	1.5%
	1.5 m	8 lx	2.2%	13 lx	2.8%	2 lx	1.9%	8 lx	2.3%
RBFNN	0.5 m	322 lx	93.3%	183 lx	38.5%	48 lx	48.3%	184 lx	60%
	1 m	6 lx	1.8%	9 lx	1.9%	1 lx	1.1%	6 lx	1.6%
	1.5 m	7 lx	2%	13 lx	2.8%	1 lx	1.4%	7 lx	2.1%
L-RBFNN given $\phi_{1/2}$	0.5 m	5 lx	1.3%	5 lx	1%	1 lx	1%	4 lx	1.1%
	1 m	6 lx	1.6%	6 lx	1.2%	1 lx	1.2%	4 lx	1.3%
	1.5 m	7 lx	2%	11 lx	2.4%	1 lx	1.3%	6 lx	1.9%
L-RBFNN no $\phi_{1/2}$	0.5 m	5 lx	1.4%	5 lx	1%	1 lx	1%	4 lx	1.1%
	1 m	6 lx	1.7%	6 lx	1.2%	1 lx	1.2%	4 lx	1.4%
	1.5 m	7 lx	2.1%	11 lx	2.4%	1 lx	1.3%	7 lx	1.9%

Fig. 8 Modeled distribution by all the discussed methods for Lamp 2.

the training procedure firstly builds a Lambertian neuron, which represents the theoretical shape by Lambertian function. Then RBF neurons are trained to regulate the shape to approximate the training data distribution.

4.1.2 Performance of L-RBFNN Compared with Other Methods

Table 3 shows the performance of L-RBFNN compared with other modeling methods. Figure 8 shows one example of the modeled distribution of all the discussed methods for Lamp 2. Figure 9 shows one example (Lamp 3, Interval 1 m) of the modeled distribution and the difference with DIALux data by all the discussed methods. From the simulation results, we can see that:

- No matter whether the $\phi_{1/2}$ is given during the training, L-RBFNN is able to provide good modeling results. (Thus we only consider L-RBFNN trained with given $\phi_{1/2}$ in the following simulations.) L-RBFNN has higher

Fig. 9 Modeled distribution of Lamp 3 and the difference with DIALux data by all the discussed methods.

accuracy in almost all the analyzed aspects compared with other methods.

- 2) As a theoretical modeling method, LF has relatively higher error than other methods. Using sensor data to normalize the LF result (NLF method) can greatly improve the LF modeling performance. However, it is still not as good as other methods. In Fig. 9, it is quite obvious that LF methods ignore the reflection from the walls. This difference will be much bigger when modeling lamps near the walls.
- 3) CSI method can provide results almost as good as the L-RBFNN method. However it is too strict with the training data position.
- 4) Generally, for most methods requiring training data (CSI

Fig. 10 Example of modeling failure by conventional RBFNN (Lamp 2, Interval 0.5m, position: $y = 3.05m$, $x \in [0m, 6.1m]$).

and L-RBFNN), more data will improve the modeling accuracy. But they can also provide very good results by small number of data.

- 5) Under certain training conditions, the conventional RBFNN may provide modeling results totally different with the desired distribution. Figure 10 shows the detailed comparison of one line (position: $y = 3.05m$, $x \in [0m, 6.1m]$) in Fig. 8's scenario. We can see that although the RBFNN modeling results also fits the data from training sensors, it provides wrong modeling results without the constraints of pre-knowledge. By introducing constraints of pre-knowledge in L-RBFNN, this problem is perfectly solved.

4.2 Performance of L-RBFNN for Office Light Modeling

To evaluate the performance of the proposed L-RBFNN for office lighting system, we use a DIALux office model ($5.4m \times 3.6m$) with 9 lamps. The lamps are doubled Lamp 3 discussed in last subsection.

As proved in last subsection, L-RBFNN is able to model single lamp accurately and needs at least one training sensor data for each lamp. Thus for an office with n lamps, at least n training sensors should be measured. It is also intuitive that measurements on the edge of working plane may also improve the modeling accuracy by enlarge the coverage of training data. Thus we test the following two training scenarios for both conventional RBFNN and L-RBFNN:

- 1) n training sensors under every lamps.
- 2) $n + 4$ training sensors, which include 4 sensors in the 4 corners.

Figure 11 shows the 2-D view and training sensors' positions of the office model.

Fig. 11 2-D view and training sensors' positions in the office model.

Table 4 Mean difference with DIALux data (Δ_{mean}) by using different modeling methods.

Lamp	LF	NLF	CSI	RBFNN		L-RBFNN	
				n sensors	$n+4$ sensors	n sensors	$n+4$ sensors
1	44.1 lx	11.9 lx	7.4 lx	9.9 lx	12.7 lx	9.6 lx	9.0 lx
2	44.5 lx	8.2 lx	7.2 lx	10.0 lx	9.3 lx	8.3 lx	7.2 lx
3	46.6 lx	7.9 lx	8.5 lx	9.0 lx	7.6 lx	6.5 lx	6.1 lx
4	40.1 lx	13.3 lx	10.1 lx	7.7 lx	6.7 lx	9.2 lx	7.4 lx
5	40.7 lx	13.3 lx	13.8 lx	7.3 lx	6.6 lx	7.5 lx	5.7 lx
6	44.4 lx	11.5 lx	7.1 lx	7.3 lx	6.5 lx	6.8 lx	5.6 lx
7	32.6 lx	11.5 lx	7.2 lx	22.9 lx	5.1 lx	6.1 lx	4.9 lx
8	33.9 lx	13.0 lx	7.9 lx	7.5 lx	5.8 lx	5.2 lx	4.9 lx
9	35.4 lx	9.6 lx	10.5 lx	5.8 lx	6.1 lx	4.8 lx	5.0 lx
Average	40.3 lx	11.1 lx	8.9 lx	9.7 lx	7.4 lx	7.1 lx	6.2 lx

4.2.1 Performance of Single L-RBFNN for Lighting Coefficients

Table 4 shows the mean difference with DIALux data (Δ_{mean}) by using different modeling methods. The results by ANN methods are listed in the last 4 columns. The table also lists the modeling results by compared methods. (By using CSI method, area out of the 3×3 training data range can not be calculated due to the strict data requirement. These checking points are not considered when calculating the Δ_{mean} for CSI.) Figure 12 shows examples of modeling results by using all the methods for lamp 7 (9 training sensors). We can concluded that:

- 1) L-RBFNN can provide better modeling results than all other methods almost in every cases. Except 2 cases:
 - i) For lamp 1, CSI is slightly better (0.5%) than L-RBFNN result. However, as explained before and shown in Fig. 12, area out of the 3×3 training data range can not be modeled by CSI, which means that more than half of the office (51.1% of the measuring points) cannot be modeled by CSI. Moreover, the average performance of L-RBFNN is still better than CSI even if the invalid area is ignored.
 - ii) For lamp 4, conventional RBFNN is slightly better (0.3%) than L-RBFNN result. However, besides that L-RBFNN has better average performance, it also has the merits of stable performance.
- 2) Generally, adding training data in the corners leads to

Fig. 12 Example of modeling results (Lamp 7, 9 training sensors) by all methods.

better modeling results (for RBFNN, only case 1 and 9 are exceptions; for L-RBFNN, only case 9 is an exception). But only n training sensors are enough to provide satisfying results, especially for L-RBFNN.

- 3) Figure 12 shows another typical example of modeling error by conventional RBFNN without adding pre-knowledge. Although the mean difference with DIALux is only 22.9 lx and may not be considered as “failure” like the example in Figs. 8 and 10, the distribution is obviously not good (especially the area near the wall). While with the modification of RBF neurons, the L-RBFNN provides much better distribution.

4.2.2 Performance of Hierarchical L-RBFNN for Office Light Modeling

Since the superiority against other modeling methods is already verified in the previous subsections, we will mainly focus on the modeling performance of proposed L-RBFNN later. As shown in Fig. 13, we design 4 scenarios for simulations to verify the performance of hierarchical L-RBFNN. In Scenario 1, all the lamps are 100% on. In Scenario 2, the lamps' dimming ratios increase from 30% to 80% from left to right. Scenario 3 and 4 have randomly generated dimming ratios.

Figure 14 shows the modeled illumination distribution

Fig. 13 Dimming ratios for lamps in 4 lighting scenarios.

Fig. 14 Modeled illumination distribution of L-RBFNN trained by 9 sensors and 13 sensors.

compared with DIALux data by L-RBFNN using 9 and 13 training sensors. Table 5 lists the modeling performance of hierarchical L-RBFNN in which $Rate_6\%$ and $Rate_8\%$ represent the proportion of measuring points with difference higher than 6% and 8% among all the measuring points. Figures 15 and 16 show the difference and normalized difference of L-RBFNN results trained by 9 and 13 sensors respectively. We can conclude that:

- 1) The L-RBFNN is able to achieve an average mean difference of 3.7% (trained by 9 sensors) and 2.6% (trained by 13 sensors), which are much lower than the threshold of human perception. Thus only n sensors can successfully train the L-RBFNN to achieve considerable modeling accuracy.

Table 5 Modeling performance of hierarchical L-RBFNN.

Scene	Trained by	Δ_{min}	Δ_{max}	Δ_{mean}	Δ'_{min}	Δ'_{max}	Δ'_{mean}	$Rate_6\%$	$Rate_8\%$
1	9 sensors	0.1 lx	215.9 lx	33.7 lx	0%	25.7%	3.7%	22.9%	7.4%
	13 sensors	0.1 lx	80 lx	23.3 lx	0%	11.5%	2.6%	10.4%	0.9%
2	9 sensors	0.1 lx	100.6 lx	16.3 lx	0%	28.3%	3.6%	17.2%	9%
	13 sensors	0 lx	43.2 lx	10.6 lx	0%	10.6%	2.3%	9.4%	1.3%
3	9 sensors	0 lx	90.3 lx	13.8 lx	0%	26.3%	3.8%	24.9%	9.9%
	13 sensors	0 lx	30.8 lx	9.4 lx	0%	11.1%	2.7%	13.9%	2.2%
4	9 sensors	0 lx	111.7 lx	17.7 lx	0%	22.2%	3.7%	28%	5.2%
	13 sensors	0 lx	49 lx	13.1 lx	0%	11.6%	2.8%	13.6%	2.2%
Mean	9 sensors	0.1 lx	129.6 lx	20.4 lx	0%	25.6%	3.7%	23.3%	7.9%
	13 sensors	0 lx	50.8 lx	14.1 lx	0%	11.2%	2.6%	11.8%	1.7%

Fig. 15 Difference between L-RBFNN results (trained by 9 sensors) and verifying data.

- 2) The measuring points with high difference are mainly near the edge of the room, thus by adding the training sensors, the rate of L-RBFNN results with perceivable modeling difference is greatly reduced ($Rate_6\%$ from 23.3% to 11.8%, $Rate_8\%$ from 7.9% to 1.7%). However, office light modeling usually emphasis on area for working, which usually does not include the area near the walls. So, L-RBFNN modeling can be concluded to be successful, and the number of training sensors can be reduced to be the same with lamp number.

5. Conclusions

This paper presents an illumination modeling method that uses a novel L-RBFNN. To model both lighting coefficients and an integrated lighting environment, single L-RBFNNs

Fig. 16 Difference of L-RBFNN results trained by 13 sensors.

are trained for all lamps to represent their illuminance distribution. Then, an integrated neural network is built to calculate the luminosity inside the offices according to the dimming ratios of lamps.

Different from the conventional RBFNN structure, L-RBFNN uses a Lambertian neuron to represent the rough theoretical illuminance distribution of the lamp. Besides, the RBF neurons are modified to fit the nature of lamp illuminance. Thus compared with conventional RBFNN, the proposed L-RBFNN:

- 1) Provides modeling results with lower error and better distribution shape;
- 2) Solves the instability problem of RBFNN and achieves stable modeling performance that makes it better for real-world applications;
- 3) Can reflect the prior knowledge of lights such as the locations and some physical parameters.

DIALux simulations of both single lamp modeling and office light modeling were made to check the performance of L-RBFNN. Methods of LF, NLF, CSI and conventional RBFNN are also simulated to be compared with the proposed L-RBFNN. The results prove that:

- 1) L-RBFNN is a successful modeling method for artificial

light with very small modeling error which is unperceivable for human;

- 2) Compared with other existing methods, L-RBFNN can provide better performance, which has lower difference with the distribution of verifying data;
- 3) The number of training sensors can be reduced to be the same with the number of lamps, thus makes the modeling method easier to be applied in real-world lighting systems.

References

- [1] I. Moreno and C.-C. Sun, "Modeling the radiation pattern of LEDs," *Optics Express*, vol.16, no.3, pp.1808–1819, 2008.
- [2] D. Caicedo, A. Pandharipande, and G. Leus, "Occupancy-based illumination control of LED lighting systems," *Lighting Research and Technology*, vol.43, no.2, pp.217–234, 2011.
- [3] A. Pandharipande and D. Caicedo, "Daylight integrated illumination control of LED systems based on enhanced presence sensing," *Energy and Buildings*, vol.43, no.4, pp.944–950, 2011.
- [4] N. Ekren, B. Dursun, and E. Aykut, "Lighting Computer Programs in Lighting Technology," *G.U. Journal of Science*, vol.21, no.1, pp.15–20, 2007.
- [5] M. Miki, Y. Kasahara, T. Hiroyasu, and M. Yoshimi, "Construction of illuminance distribution measurement system and evaluation of illuminance convergence in intelligent lighting system," *Proceedings of IEEE Sensors*, pp.2431–2434, 2010.
- [6] Z. Wang and Y.K. Tan, "Illumination control of LED systems based on neural network model and energy optimization algorithm," *Energy and Buildings*, vol.62, pp.514–521, 2013.
- [7] F. Dong, V. Hombal, and A.C. Sanderson, "Adaptive light field sampling and sensor fusion for smart lighting control," *15th International Conference on Information Fusion*, pp.40–47, 2012.
- [8] S. Afshari, S. Mishra, A. Julius, F. Lizarralde, and J.T. Wen, "Modeling and feedback control of color-tunable LED lighting systems," *American Control Conference (ACC)*, pp.3663–3668, 2012.
- [9] W. Si, X. Pan, H. Ogai, K. Hirai, N. Yamauchi, and T. Li, "Illumination Modeling Method for Office Lighting Control by Using RBFNN," *IEICE Trans. Inf. & Syst.*, vol.E97-D, no.12, pp.3192–3200, 2014.
- [10] E.J. Hartman, J.D. Keeler, and J.M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Computation*, vol.2, no.2, pp.210–215, 1990.
- [11] J. Park and I.W. Sandberg, "Universal approximation using radial-basis function networks," *Neural Computation*, vol.3, no.2, pp.246–257, 1991.
- [12] S.A. Billings, H.-L. Wei, and M.A. Balikhin, "Generalized multiscale radial basis function network," *Department of Automatic Control and Systems Engineering, Neural network*, vol.20, no.10, pp.1081–1094, 2007.
- [13] Y.-J. Qu and B.-G. Hu, "RBF networks for nonlinear models subject to linear constraints," *2009 IEEE International Conference on Granular Computing*, pp.482–487, 2009.
- [14] S. Haykin, "Neural Networks: A Comprehensive Foundation," *Neural Networks*, 2, 2004.
- [15] S. Milanič, S. Strmčnik, D. Šel, N. Hvala, and R. Karba, "Incorporating prior knowledge into artificial neural networks an industrial case study," *Neurocomputing*, vol.62, pp.131–151, 2004.
- [16] Y.-J. Wen and A.M. Agogino, "Wireless Networked Lighting Systems for Optimizing Energy Savings and User Satisfaction," *Proceedings of Wireless Hive Networks Conference*, pp.1–7, Austin, Texas, USA, 2008.
- [17] M. Miki, T. Yoshii, H. Ikegami, Y. Azuma, A. Emi, and K. Yoshida,

"A lighting control system to optimize brightness distribution on working places," 2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI), pp.23–28, 2013.

Wa Si was born in Hefei, China, on May 6th, 1987. She received a Bachelor degree of Science from Fudan University (China) and a Ph.D degree of Engineering from Waseda University. She is presently a research associate in the Graduate School of Information, Production and Systems, Waseda University. She has worked on optimization algorithms and neural networks.

Xun Pan was born in Zhenjiang, China, on February 23th, 1989. He received a Bachelor degree of Engineering from Nanjing University (China) and a Master degree of Engineering from Waseda University. He is presently a Ph.D student in the Graduate School of Information, Production and Systems, Waseda University. He has worked on optimization algorithms and computer vision.

Harutoshi Ogai was born in Kitakyushu, Japan, on March 1st, 1952. He received a Ph.D degree of Engineering from Tokyo Institute of Technology in 1997 and is presently a professor in the Graduate School of Information, Production and Systems, Waseda University. His research area includes process control, process modeling and simulation, process analysis and data mining, automotive control and bridge diagnosis. He is also member of SICE, ISCIE, ISIJ, JSAI and IPSJ.

Katsumi Hirai was born in Tokyo, Japan, on February 10th, 1944. He received a Master degree of Engineering from University of Tokyo in 1968 and is presently the president of Hakutsu Technology Corporation.