Estimation of Multiple Illuminant Colors Using Color Line Features

Quan XIU HO\(^{[a]}\), Nonmember, Takao JINNO\(^{[b]}\), Member, Yusuke UCHIMI\(^{[b]}\), Nonmember, and Shigeru KURIYAMA\(^{[c]}\), Senior Member

SUMMARY  The colors of objects in natural images are affected by the color of lighting, and accurately estimating an illuminant’s color is indispensable in analyzing scenes lit by colored lightings. Recent lighting environments enhance colorfulness due to the spread of light-emitting diode (LED) lightings whose colors are flexibly controlled in a full visible spectrum. However, existing color estimations mainly focus on the single illuminant of normal color ranges. The estimation of multiple illuminants of unusual color settings, such as blue or red of high chroma, has not been studied yet. Therefore, new color estimations should be developed for multiple illuminants of various colors. In this article, we propose a color estimation for LED lightings using Color Line features, which regards the color distribution as a straight line in a local area. This local estimate is suitable for estimating various colors of multiple illuminants. The features are sampled at many small regions in an image and aggregated to estimate a few global colors using supervised learning with a convolutional neural network. We demonstrate the higher accuracy of our method over existing ones for such colorful lighting environments by producing the image dataset lit by multiple LED lightings in a full-color range.

key words: illuminant color, Color Lines, multiple color estimation, CNN regression

1. Introduction

Various computer vision tasks, such as object recognition, image retrieval, and image classification, can have imprecise results for images processed with incorrect computational color constancy, as demonstrated in [1]. The performance of these tasks can be improved by estimating illuminant colors, especially for the scenes illuminated by lightings whose colors have high chroma. In addition, some image correction methods, such as simple normalization or chromatic adaptation transformation [2], can be applied by giving the illuminants’ colors of scenes. However, these correction methods were developed for the natural colors emitted from solar or traditional light sources, such as filament bulbs or fluorescent lamps.

The diffusion of color-controllable lightings with light-emitting diodes (LEDs) creates colorful lighting environments with the expressive power of colors whose range spans most of the visible spectrum. The color estimation on objects becomes difficult for images captured under such colorful scenes lit by multiple colored lightings.

In this article, we focuses on the color estimation of multiple illuminants from images. We propose color estimation models using the method called Color Lines that regards the color distribution as a straight line in a local area on an object. Such features are fed to train a convolutional neural network (CNN) for estimating the multiple colors of LED lightings in a scene.

2. Related Works

2.1 Single Illuminant Color

Various approaches have been proposed for single illuminant estimation using physical assumption and statistical values; i.e., the color constancy and color variance can be roughly categorized into statistics- and learning-based. The most common one introduces the grey world assumption [3], which assumes that the average of objects’ colors converges to grey. The Grey Edge [4] introduces this assumption for the achromatic average edge difference in a scene.

Recent studies have attempted to apply machine learnings for estimating an illuminant color because learning-based methods are often more powerful than statistics-based ones. One of the first learning-based algorithms [5] introduced a neural network (NN) to estimate the illumination chromaticity from images. Cardie et al. [6] proposed an approach that combines multiple algorithms consisting of grey world, white patch, and NNs, where they are selectively applied depending on the scene. Gijsenij et al. [7] introduced the parameters of the Weibull distribution to identify the characteristics of color images. A maximum-likelihood classifier was trained based on a mixture of Gaussians for selecting the best estimation algorithm against a particular image. Bianco et al. [8] introduced a method based on supervised learning with CNNs designed to take image patches as input to estimate local illuminant colors. The local estimates were aggregated for a single illuminant to train a local-to-global regressor for predicting a global color. Shi et al. [9] proposed a selection network that combines two networks to choose an estimation by generating multiple illuminant hypotheses for each local patch. However, this hypothesis
cannot estimate multiple different colors.

2.2 Multiple Illuminant Colors

Grey Pixels [10] detects achromatic regions for estimating multiple illuminant colors from a single image. In this method, the lighting component is regarded as constant by assuming their invariance among nearby pixels. The standard deviation within a local region is considered a reference value independent of the lighting conditions. This method estimates the illuminant color by assuming that most natural images contain achromatic regions. It can also estimate multiple illuminant colors if the scene contains corresponding achromatic regions. Shi et al. [11] proposed an estimation method for an illuminant color using patch-wise bright pixels. This method divides a down-sampled image into patches, and brighter patches are selected to estimate illuminant color using the gray world-based method [3]. It is also applicable to the estimation of multiple illuminant colors. However, this method is sometimes ineffective when the scenes have color saturation, as we experimentally evaluate in Sect. 4.

2.3 Estimation with Color Lines

Omer et al. [12] proposed the method of Color Lines for applications such as color clustering and re-coloring. In the RGB space, the Color Line of each pixel bends towards the point of an illuminant color. Assuming that the color of an object’s surface is uniform, the color distribution in the local area spreads along the color direction of the illuminant (Fig. 1). After estimating the distribution by fitting a straight line, all lines computed for every local region in the same scene may have an intersection corresponding to the point of illuminant colors. However, we experimentally found that accurately detecting such an intersection is often difficult due to the noise and disturbance involved in natural images. Therefore, instead of relying on the intersection, we introduce a learning model to use the Color Line features.

3. Color Estimations Using Color Lines

We estimate illuminant colors in an image by effectively using the method of Color Lines. Instead of taking raw images as the input of a deep learning model [8], our approach trains a model with Color Line features obtained during preprocessing. Our method divides images into superpixels, by which colors are more effectively estimated by extracting only valid regions for computing Color Lines.

Figure 2 shows the flowchart of our method. First, we compute image features based on Color Lines [12], connecting the actual object color and illuminant color. Then, we estimate illuminant colors using these features with a convolutional NN (CNN). This CNN model takes the input of the Color Line features via adaptive sampling, whose dimensions are determined by a sampling resolution, and estimates different colors of multiple illuminants by considering positional dependency between pixels.

3.1 Feature Extraction Based on Color Lines

Our method computes the Color Line features for each local region of an input image. For extracting the local regions of images satisfying the color uniformity assumed in Sect. 2.3, we divide the image into small segments using linear spectrum clustering [13], which clusters connected pixels of similar brightness and color properties. Since the adequate number of segments (or clusters) differs depending on the texture’s density in the image, we employ the fast Fourier transform to decide the number of segments by evaluating the amplitude spectrum in each frequency domain channel. The number of segments $N_{seg}$ is therefore determined by the product of the absolute mean of FFT’s coefficients $F_{u,v}^c$ averaged over color channels $c = R, G, B$ as,

$$N_{seg} = \frac{k}{3 |U| |V|} \sum_{c=R,G,B} \sum_{u \in U, v \in V} |F_{u,v}^c|,$$

where $|U|, |V|$ denote the number of coefficients along each frequency dimension, and we experimentally set the value of constant by $k = 50$.

Figure 3 shows examples of images with different numbers of segments. For scenes with fine textures, the number of segments increases while the area of each segment decreases.

Color Lines in the RGB space are calculated using principal component analysis in our method. For every clustered segments, we compute a variance covariance matrix of all pixel values within the region to obtain its eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$, and corresponding eigenvectors $v_1^c, v_2^c, v_3^c$. Then, we set an eigenvector $v_i^c$, which corresponds to the most significant eigenvalue $\lambda_i$, as the direction vector of a Color Line. Given the mean values of color distribution $\vec{X} = \{r, g, b\}$, the equation of Color Lines is formulated as $\vec{X} + \bar{t}v_i^c$, where $\bar{t}$ is a parameter and $v_i^c = \{v_{r,i}, v_{g,i}, v_{b,i}\}$ denotes the direction vector. Since each Color Line feature is given by the $\vec{X}$ and $v_i^c$, it can be represented as a six-dimensional vector.

The Color Lines cannot be accurately estimated for the segments causing color saturation because the lines bend to-
ward a wrong point. In addition, the estimation becomes unstable when the irradiation intensity remains unchanged within the segment because the distribution of pixel values tends to be nonlinear. We detect such unstable condition by evaluating the flatness ratio of the color distribution as $\lambda_1/\sum_{i=1}^3 \lambda_i < 0.8$. We remove all segments having saturation or the above-mentioned flatness from the Color Line features (Fig. 4).

3.2 Regression with CNN

Our method introduces a CNN that directly takes Color Line features as the input. After extracting Color Line features for every superpixel, we split the entire image equally into $16 \times 16$ blocks. Then, we calculate the six-dimensional weighted average of the Color Line features in each block, and Color Line features in an image are converted into a $16 \times 16 \times 6$ dimensional vector. The weighting values are computed by the area’s ratio of the corresponding superpixel overlapped with the block. This block-wise aggregation standardizes the dimension, which is suited to the CNN model of fixed input size.

Our CNN consists of three convolutional layers whose kernel size is $3 \times 3$, followed by fully connected layers (Fig. 5). We apply batch normalization over the outputs of the three convolutional layers, each followed by a leaky rectified linear unit as activation with a negative slope of 0.2. The network weight parameters are updated with the loss function of mean-square error against the ground truth colors, using Stochastic Gradient Descent optimization. For scenes with a single illuminant, the final layer has the output of three channels for the RGB space, and for scenes with $N$ illuminant colors, the final layer is modified to have the output of $3N$ channels.

4. Experimental Evaluations

4.1 Metric for Evaluation

We evaluated our method using three-fold cross-validation; Fig. 6 shows the flow of our evaluation. The Macbeth color checker (MCC) in each image was masked before training and testing for estimating illuminant color. We obtained the ground truths of the illuminant colors by calculating the average values of the achromatic patches of the MCC, and the error metric by angular error [14], which is given by $\cos^{-1}(\mathbf{e}_{est} \cdot \mathbf{e}_{gt})$, where $\mathbf{e}_{est}$ and $\mathbf{e}_{gt}$ are the normalized vectors representing the estimation and ground truth of illuminant color, respectively, in the RGB space. Then, the angular errors are evaluated using the mean, median, trim mean [7], Best25, and Worst25 indices. In the following sections, we demonstrate the comparison against the existing methods for different datasets.

4.2 Preparation of Colored Illuminant Dataset

The images whose scenes include various of colored illuminants can be synthesized with computer graphics technology. However, such artificial images are unsuited to training samples due to the lack of authenticity. To produce realistic scenes and increase the variety of illuminant colors for training and testing, we collected natural images to create datasets of two types: scenes lit by single and multiple lightings. In every scene, objects are placed in the center with a black background and lit with various colors using Philips.
hue bloom, where all images were taken in long exposures mode.

We augmented the training images data by rotating them with a random angle between $-30^\circ \sim +30^\circ$ and up-down and left-right flips. The images were also randomly cropped and resized to the same size.

4.2.1 Single-Colored Illuminant Dataset

Although our final target was to estimate multiple illuminants colors, we prepared a dataset of images lit by a single colored LED for evaluating our performance in globally estimating a color.

The single-colored dataset consists of 420 images from 10 scenes. Each scene has 42 images, and the control signals for colored lightings are set by 21 primary colors, whose details are shown in Table A-1 of appendices, with two saturation levels. All images include the MCC for obtaining the ground truth. Notably, the values of primary colors are just the control parameters specified by a manufacturer, and they do not represent the ground truth of colored illuminants; we actually estimate primary colors by the pixel values on the MCC in each image.

We employed 3-fold cross-validation and therefore split single- and two-colored illuminant datasets into the ratio of 2:1 for training and test, respectively.

4.2.2 Two-Colored Illuminant Dataset

We also prepared a two-colored illuminant dataset comprising 570 images with 20 scenes, and two types of settings are adopted for every 10 scenes. This dataset was used for evaluating the performance of our method to estimate two colors separately. We set these colors using two approaches.

In the first case, the colors of lightings were set to be the randomly selected 18 combinations of 6 colors (red, green, blue, orange, cyan, and magenta), and the brightness of each color was also randomly altered in three levels ($n/4, n = 2, 3, 4$). Since each scene had 18 images, 180 images were taken in this setting.

In the second case, the colors of lightings were set to be predefined 19 combinations of the eight colors extracted from the abovementioned primary colors (blue, aqua, crimson, orange, lime, violet, medium violet red, and saddle brown). The combination was altered by swapping the corresponding lightings to make $2 \times 19 = 38$ combinations, and we added white vs. white combinations as reference.

As 10 scenes were taken for each setting, the total of $(1 + 38) \times 10 = 390$ images were collected. Notably, every image includes two MCCs for obtaining the ground truth of each illuminant. We also employed 3-fold cross-validation in a similar way to the single- and two-colored illuminant dataset.

4.3 Comparison of Estimation Accuracy

4.3.1 Single Color Estimation

We comparatively evaluated the estimation accuracy against eight existing methods; six methods were implemented for estimating a single illuminant color, including the supervised learning method, denoted as CNN, and Gray Pixels and Bright Pixels for comparing the accuracy of multiple color estimations.

Table 1 summarizes the results of color estimation accuracy for a single illuminant, where Best-25 % (or Worst-25 %) represents the average of the top 25% with the lowest (or highest) angular errors. As this table shows, our method had the lowest errors (decorated by bold typeface) in all metrics. Figure 7 shows the example of the resulting estimation on this single-colored dataset.

We also investigated the effectiveness of our method
Table 2  Comparative results on color-checker dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Median</th>
<th>Trimean</th>
<th>Best25</th>
<th>Worst25</th>
</tr>
</thead>
<tbody>
<tr>
<td>White Patch [15]</td>
<td>4.78</td>
<td>3.19</td>
<td>3.53</td>
<td>0.91</td>
<td>11.5</td>
</tr>
<tr>
<td>1st Gray edge [4]</td>
<td>4.39</td>
<td>3.00</td>
<td>3.28</td>
<td>1.29</td>
<td>9.92</td>
</tr>
<tr>
<td>Gray World [3]</td>
<td>6.80</td>
<td>5.45</td>
<td>5.77</td>
<td>1.94</td>
<td>14.03</td>
</tr>
<tr>
<td>Shades of Gray [15]</td>
<td>4.03</td>
<td>2.88</td>
<td>3.19</td>
<td>0.70</td>
<td>9.27</td>
</tr>
<tr>
<td>CNN [8]</td>
<td><strong>2.83</strong></td>
<td><strong>2.02</strong></td>
<td><strong>2.26</strong></td>
<td><strong>0.67</strong></td>
<td><strong>6.33</strong></td>
</tr>
<tr>
<td>Gray Pixels [10]</td>
<td>4.11</td>
<td>2.73</td>
<td>2.94</td>
<td><strong>0.64</strong></td>
<td>10.02</td>
</tr>
<tr>
<td>Our method</td>
<td>3.33</td>
<td>2.31</td>
<td>2.57</td>
<td>0.85</td>
<td>7.55</td>
</tr>
</tbody>
</table>

Fig. 8  The images of Color-checker dataset.

for the colorful lightings by comparing the accuracy of images lit by ordinary lightings of natural bulb or lamp colors.

For this experiment, we used the color-checker dataset [16] for training our NN model, which consists of 568 natural images of indoor and outdoor scenes taken by two DSLR cameras, where the dataset was split in the same way as the single- and two-colored illuminant datasets. Each image includes the MCC and is taken in auto white balance mode. Notably, the colors of illuminants in this dataset are natural (Fig. 8).

We summarize the comparison results in Table 2 following the reported results in the existing methods [9], [17], [18], where the underlined texts indicate the accuracies whose values are greater than those of Table 1. This result showed that only our method and Gray World obtained more accurate results in the colorful lighting conditions of our dataset rather than the natural lighting condition of the color-checker dataset.

CNN [8] achieved the minimum accuracies, denoted by bold typeface, except Best25, showing the advantageous property of supervised learning methodologies. Although our method was less accurate than CNN for natural lighting environment, it achieved the second superior performance, except for Best25. The difference between the two supervised methods (Our method—CNN) is greater in Table 1 than in Table 2. These observations imply that our method is superior in estimating the illuminant color of high chroma.

Here, we further investigate the feature of our method by showing the resulting estimations in Fig. 9. We observed that the images taken in outdoor scenes have lower angular errors than those taken in indoor scenes. One reason could be the imbalanced number of outdoor (350) and indoor (200) in the dataset. Although the outdoor illuminants were often achromatic, the indoor illuminants had more varieties. Therefore, the model could not have enough training to estimate the indoor scenes’ illuminant colors. In addition, the color-checker dataset includes the scenes lit by the multiple lightings of different colors. Therefore, our estimation caused large errors in estimating a single color against such mixed environments for the wrong selection among multiple candidates.

We also observed that our method was sometimes ineffective in estimating scene including only a few small object colors, e.g., the upper right image whose border is drawn by red thick lines in Fig. 9. If the estimated colors on such objects largely differs, our estimation tends to cause large errors due to its feature-averaging property.

Table 3  Comparison on the two-colored illuminant dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Median</th>
<th>Trimean</th>
<th>Best25</th>
<th>Worst25</th>
<th>95th pct.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td><strong>6.16</strong></td>
<td><strong>5.56</strong></td>
<td><strong>5.76</strong></td>
<td><strong>2.89</strong></td>
<td><strong>10.31</strong></td>
<td><strong>11.94</strong></td>
</tr>
</tbody>
</table>

Fig. 9  Example of estimations for the color-checker dataset [16], where the left and right patch indicates the ground truth and estimated color, respectively, and the horizontal line indicates angular error in degrees.

4.3.2 Two Colors Estimation

We summarize the results of color estimation accuracy for two-colored illuminant in Table 3, where the 95th percentile denotes the angular error with the highest value remaining after the Worst 5% of the data set. For the two-colored illuminant dataset, all error metrics reduced by approximately 50% over existing two methods. Figure 10 shows an example of estimation on the two-colored dataset.

The comparative results on the three datasets suggest...
that our method more effectively estimates illuminants’ colors, even from a small size of features.

4.3.3 Effect of the Number of Color Lines

Table 4 shows the average, minimum, and maximum number of Color Lines extracted from all image samples. This comparison suggested that accuracy depended on the number of extracted Color Lines; our method had superior performance in the two-colored dataset of minimum Color Lines and inferior performance in the color-checker dataset of maximum Color Lines. This correlation implies that increasing the number of Color Lines also increases the noisy or outlier samples, which damages the estimation accuracy. This drawback is derived from the averaging strategy of Color Lines for all superpixels inside each block. Smartly screening the reliable superpixels could improve this defect. Although our method automatically removes superpixels whose colors are saturated or flattened, adaptive parameter tunings of the linear spectrum clustering [13] is required for optimally controlling the number of Color Lines, depending on the scene. Besides, we should investigate the effectiveness of our averaging strategy in computing a representative feature in each block; the method for selecting a single Color Line feature is worth considering.

### Table 4 Number of Color Lines in each dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Mean</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color-checker</td>
<td>394.0</td>
<td>74</td>
<td>1102</td>
</tr>
<tr>
<td>Single-colored</td>
<td>154.3</td>
<td>29</td>
<td>405</td>
</tr>
<tr>
<td>Two-colored</td>
<td>119.0</td>
<td>10</td>
<td>588</td>
</tr>
</tbody>
</table>

4.3.4 Evaluation with Real-World Images

Figure 11 shows the estimation results using our method for real-world images obtained from Flickr. Although there is no information on the ground truths, the appropriate colors seemed successfully estimated. These results demonstrated the versatility of our method for various outdoor scenes such as squares, gardens, monuments, buildings, and wide-open spaces such as halls.

5. Conclusion

We proposed a learning-based method to estimate illuminant colors. This method combines the efficient color features derived from Color Lines and the simple regression mechanism with a CNN. Our method outperformed the existing ones, especially in the colored illuminant datasets. Our method achieved higher accuracy than an existing CNN-based method [8], both for single and multiple illuminant environments. This performance implies the general effectiveness of the Color Line features for colored illuminants of high chroma. We also implemented support vector regression as the other regression model for a single-color dataset and confirmed an accuracy decrease (see [19] for details).

Our CNN model is relatively a simple architecture.
compared with the state-of-the-art models and still has room for improvements. Our method must change the number of channels for the output layer, according to the number of estimated color illuminants. This constraint requires individual training for every network of different output channels. Therefore, we should develop a more general network architecture that is scalable to the number of estimated illuminants’ colors. Also, detecting the number of target colors is essential for fully automating the estimation for multiple illuminants.

Our current model cannot cope with the variable number of illuminants, and the network should have more flexibility in its output dimensions. Moreover, the NN for estimating the number of illuminants (or colors) and their colors could be integrated into a single architecture. One possible solution is to set a sufficient, redundant number of output channels, e.g., for ten illuminants’ colors. We can train such a network by padding the empty output channels with the same illuminants’ colors in computing a loss function. Then, in estimation, the similar output colors could be merged as one illuminant to estimate the number of multiple lightings. The feasibility study of this approach is our future work.

After estimating the illuminant colors, we plan to simulate a colored relighting in the same scene against different color lightings. To perform a realistic simulation, we should develop the estimation of geometric conditions of the illumination, such as the position and beam direction of the light source.

References


Appendix: Settings of Colored Lightings

A.1 Sample Colors Used in Controlling Colored Lightings

Table A-1 shows the 21 basic colors for controlling our colored LED bulbs.

A.2 Predefined 19 Color Combinations

We predefined the 19 combinations of two colors whose hue values largely differ from each other.

- blue vs. [aqua, crimson, orange, lime, violet, medium violet red, saddle brown]

Table A-1 21 basic colors used in controlling colored lightings.

<table>
<thead>
<tr>
<th>Color name</th>
<th>R, G, B</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>255,255,255</td>
</tr>
<tr>
<td>aqua</td>
<td>0,255,255</td>
</tr>
<tr>
<td>deep sky blue</td>
<td>0,191,255</td>
</tr>
<tr>
<td>lemon chiffon</td>
<td>255,250,205</td>
</tr>
<tr>
<td>spring green</td>
<td>0,255,127</td>
</tr>
<tr>
<td>crimson</td>
<td>220,20,60</td>
</tr>
<tr>
<td>red</td>
<td>255,0,0</td>
</tr>
<tr>
<td>coral</td>
<td>255,127,80</td>
</tr>
<tr>
<td>salmon</td>
<td>250,126,144</td>
</tr>
<tr>
<td>orange</td>
<td>255,165,0</td>
</tr>
<tr>
<td>gold</td>
<td>255,215,0</td>
</tr>
<tr>
<td>golden rod</td>
<td>218,165,32</td>
</tr>
<tr>
<td>yellow</td>
<td>255,255,0</td>
</tr>
<tr>
<td>lime</td>
<td>0,255,0</td>
</tr>
<tr>
<td>blue</td>
<td>0,0,255</td>
</tr>
<tr>
<td>indigo</td>
<td>75,0,130</td>
</tr>
<tr>
<td>violet</td>
<td>238,130,238</td>
</tr>
<tr>
<td>magenta</td>
<td>255,0,255</td>
</tr>
<tr>
<td>medium violet red</td>
<td>199,21,133</td>
</tr>
<tr>
<td>pink</td>
<td>255,192,203</td>
</tr>
<tr>
<td>saddle brown</td>
<td>139,69,19</td>
</tr>
</tbody>
</table>
• crimson vs. [aqua, orange, lime, saddle brown]
• violet vs. [aqua, orange, lime, saddle brown]
• medium violet red vs. [aqua, orange, lime, saddle brown]

Quan Xiu Ho has completed her undergraduate and graduate programs in Computer Science and Engineering at Toyohashi university of technology, in 2019 and 2021, respectively. Her research is focused on estimating illuminant color from images using machine learning or deep learning. Before entering the university, she received an associate degree in Electronics and Computer Engineering at National Institute of Technology, Toyama College in 2017.

Takao Jinno received B.S., M.E., and Dr. Eng. from the University of Kitakyushu in 2007, 2009, and 2012, respectively. He had been on Toyohashi University of Technology as an assistant professor from 2012 to 2017. Now, He has been on Osaka Institute of Technology as an assistant professor since 2017. He is engaged in illuminant estimation, visualization for X-ray image and diagnosis of physiological disorders of plants.

Yusuke Uchimi received B.E. and M.E. degrees in Computer Science and Engineering at Toyohashi University in 2016 and 2018. He is currently working at SoftBank corp.

Shigeru Kuriyama received B.E. and M.E. degrees in electronics from Osaka University in 1985 and 1987, and Dr.Eng. in information systems engineering from Osaka University. He was a researcher at IBM Tokyo Research Laboratory from 1988 to 1993, and was an associate professor of Information Sciences at Hiroshima City University from 1994 to 1997. He is currently a professor of the Department of Computer Science and Engineering at Toyohashi University of Technology, since 2005. His research interests include image-based lighting controls and AI-based virtual human visualizations.