2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

A New Scheme for Color Night Vision by Quaternion Neural Network

Hiromi Kusamichi, Teijiro Isokawa, Nobuyuki Matsui
Division of Computer Engineering,
University of Hyogo, Japan
hiromi@comp.himeji-tech.ac.jp, {*isokawa, matsui*}*@eng.u-hyogo.ac.jp*

Yuzo Ogawa, Kazuaki Maeda
Technical Development Office, R & D Division,
TOA Corporation, Japan
{*ogawa_yuzo, maeda_kazuaki*}*@toa.co.jp*

Abstract

A neural network for extracting color information from gloomy images is presented in this paper. We adopt a feedforward quaternion neural network, where neuronal parameters are represented as 4-dimensional vectors. The network is trained by imposing a gloomy image as input and its finer one as target by quaternion version of back propagation algorithm, and then relations between them are established. We show the performance and the generalization ability of quaternion neural network through numerical results. Our proposed method will be useful in realizing a so-called color night vision system for the benefit of its simplicity and cost-effectiveness.

Keywords: Quaternion Neural Network, Color Night Vision, Gloomy image.

1 Introduction

Color information plays an important role in our visual recognition. The color discrimination in biological visual systems is ascribed to perceiving the difference between the frequencies of light waves reflected or transmitted by various objects. We need some intensity of light in perceiving colors. Therefore, in the night or under twilight environments, it is hard to capture color information. Artificial visions, such as Charge-Coupled Device (CCD), also requires a certain intensity of light in order to output signals, so that they would be faced with the same difficulty. So, it is important for surveillance, monitoring, or navigation applications to develop a method extracting color information from gloomy images in night scenes.

The simplest way to brighten up gloomy images is to amplify signals from CCDs. But the images obtained from this method, are unclear due to the so-called shot noise. Another way is to extend the exposure time or to expose repeatedly in order to collect more photons coming from objects. This is rather effective but the frame rate of images tends to become lower.

Recently, image fusion techniques are studied in order to make gloomy scene vivid [1, 2]. These are realized by fusing signals obtained from the sensors for different bandwidth, and can be used for real-time applications. However, their cost performance is not effective due to multiple sensors.

In this paper, we propose a novel scheme of quaternion neural network for color night vision extracting color information from gloomy images. It requires only a single CCD camera for visible light. Our strategy is to train this feed-forward neural network in order to transform pixel values in the gloomy image to finer ones by preparing images taken from a scene under different illuminations. It is reported that our quaternion-valued neural network has the advantage of treating color information, as compared with conventional(real-valued) one [3, 4]. We perform the color night vision experiments by using images taken from several scenes, and explore the qualities and properties of reconstructed images by our method.

2 Quaternion Neural Network

2.1 Quaternion and its ability of geometrical operators

Quaternions form a class of hypercomplex numbers that consist of a real number and three kinds of imaginary number, $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$. Formally, a quaternion is defined as a vector in a 4-dimensional vector space, i.e.,

$\boldsymbol{x} = x^{(\mathrm{e})} + x^{(\mathrm{i})}\boldsymbol{i} + x^{(\mathrm{j})}\boldsymbol{j} + x^{(\mathrm{k})}\boldsymbol{k}$

where $x^{(\mathrm{e})}$ and $x^{(\mathrm{i})}$, $x^{(\mathrm{j})}$, $x^{(\mathrm{k})}$ are real numbers. $\boldsymbol{K}^4$, the division ring of quaternion, thus constitutes the 4-dimensional vector space over the real numbers with the bases 1,$\boldsymbol{i}$,$\boldsymbol{j}$,$\boldsymbol{k}$. It is also written using 4-tuple or 2-tuple notations as

$\boldsymbol{x} = (x^{(\mathrm{e})}, x^{(\mathrm{i})}, x^{(\mathrm{j})}, x^{(\mathrm{k})}) = (x^{(\mathrm{e})}, \boldsymbol{w})$

where $\boldsymbol{w} = \{x^{(\mathrm{i})}, x^{(\mathrm{j})}, x^{(\mathrm{k})}\}$. In this representation $x^{(\mathrm{e})}$ is the scalar part of $\boldsymbol{x}$ and $\boldsymbol{w}$ forms the vector part. In

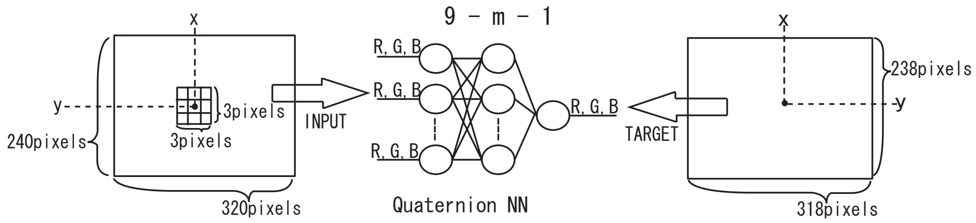


Figure 1: The schematic of the training method of quaternion neural network.

the case $x^{(e)} = 0$ in $\boldsymbol{x}$, i.e., $\boldsymbol{x} = x^{(i)}\boldsymbol{i} + x^{(j)}\boldsymbol{j} + x^{(k)}\boldsymbol{k} \in \boldsymbol{I}$ where $\boldsymbol{I}$ denotes purely imaginary quaternion space, we call this purely imaginary quaternion.

The quaternion conjugate is defined as
$\boldsymbol{x}^* = (x^{(e)}, -\boldsymbol{w}) = x^{(e)} - x^{(i)}\boldsymbol{i} - x^{(j)}\boldsymbol{j} - x^{(k)}\boldsymbol{k}$.

Quaternion bases satisfy the following identities, known as the Hamilton rules:

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1,$$
$$\boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k}, \quad \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i}, \quad \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}.$$

The outer product between $\boldsymbol{p} = (p^{(e)}, \boldsymbol{v})$ and $\boldsymbol{q} = (q^{(e)}, \boldsymbol{w})$, is defined as

$$\boldsymbol{p} \otimes \boldsymbol{q} = (p^{(e)}q^{(e)} - \boldsymbol{v}\cdot\boldsymbol{w}, p^{(e)}\boldsymbol{w} + q^{(e)}\boldsymbol{v} + \boldsymbol{v}\times\boldsymbol{w})$$

where $\boldsymbol{v}\cdot\boldsymbol{w}$ and $\boldsymbol{v}\times\boldsymbol{w}$ denote the dot and cross products respectively between three-dimensional vectors $\boldsymbol{w}$ and $\boldsymbol{w}$.

By the use of quaternion, we can define three geometrical operators; these are Translation, Scaling, and Rotation. Let $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{u}$ be purely imaginary quaternions representing coordinates in 3-dimensional space.

Translation and Scaling are defined in the same way as in 2D space. The sum of $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e., $\boldsymbol{x}+\boldsymbol{y}$ results in the point with coordinate $\boldsymbol{x}$ that is shifted by $\boldsymbol{y}$. Multiplying $\boldsymbol{x}$ by the scalar $\boldsymbol{y}$, i.e., $b\boldsymbol{x}$ represents the point with coordinate $\boldsymbol{x}$ scaled by b.

Rotation is defined by a generic quaternion $\boldsymbol{a}$ with $|\boldsymbol{a}| = 1$ and its conjugate, as

$$\boldsymbol{y} = \boldsymbol{a} \otimes \boldsymbol{x} \otimes \boldsymbol{a}^*.$$

When $\boldsymbol{a}$ is denoted by $\boldsymbol{a} = \cos\alpha + (\sin\alpha)\boldsymbol{u}$ where $|\boldsymbol{u}| = 1$ and $|\alpha| < \pi$, $\boldsymbol{y}$ is the point with coordinate $\boldsymbol{x}$ rotated by the angle 2α around $\boldsymbol{u}$.

2.2 Quaternion neuron model and feedforward network

Quaternion neuron model, proposed in [3, 4], adopts purely imaginary quaternions as the inputs and the output of a neuron. The geometrical operators are used in processing neuronal inputs.

The output of quaternion j, $\boldsymbol{y}_j$, is expressed as

$$\boldsymbol{s}_j = \sum_i \frac{\boldsymbol{w}_{ji} \otimes \boldsymbol{x}_i \otimes \boldsymbol{w}_{ji}^*}{|\boldsymbol{w}_{ji}|} - \boldsymbol{\theta}_j, \quad \boldsymbol{y}_j = \boldsymbol{f}(\boldsymbol{s}_j),$$

where i denotes the indices of neurons in the previous layer, and $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{s} \in \boldsymbol{I}, \boldsymbol{w} \in \boldsymbol{K}^4$ respectively are the vector of inputs to the neurons, the vector of outputs from the neurons, the threshold, the internal potential, and the weights of the connections to the neurons in layer i. Each of the output signals from the neurons at the input side of neuron j is dilated and rotated by the weight factors. The output of the neuron is determined by the output of the activation function $\boldsymbol{f}$, which is defined by

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{s}) &= f(s^{(i)})\boldsymbol{i} + f(s^{(j)})\boldsymbol{j} + f(s^{(k)})\boldsymbol{k}, \\ f(x) &= 1/(1+e^{-x}). \end{aligned}$$

We construct a feedforward network by quaternion neurons and derive Back-Propagation (BP) algorithm for training this network. For quaternion equivalent of BP algorithm, we define the error function E between the output value of the output and the target training data as

$$E = \frac{1}{2}\sum_n \sum_{\upsilon\in\{i,j,k\}} (d_n^{(\upsilon)} - y_n^{(\upsilon)})^2$$

where $\boldsymbol{d}_n$ and $\boldsymbol{y}_n$ are the desired output and the output from the neuron n respectively. The network parameter $\boldsymbol{p}$ ($\boldsymbol{p}$ being weights between neurons and thresholds of neurons) are updated along the gradients of E with respect to $\boldsymbol{p}$:

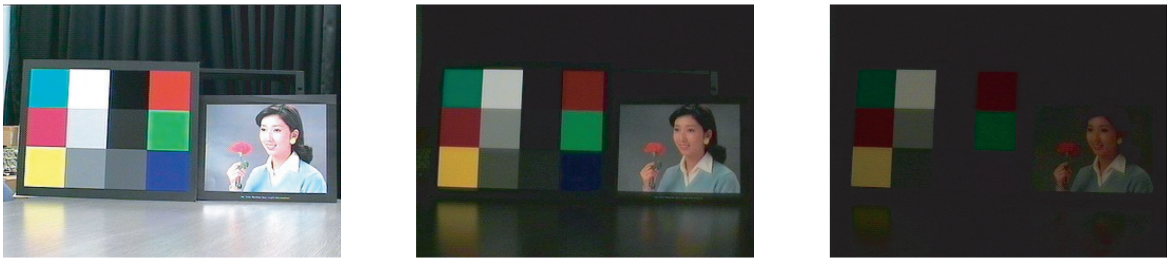
$$\boldsymbol{p}^{new} = \boldsymbol{p}^{old} - \eta\frac{\partial E}{\partial \boldsymbol{p}}$$

where η is a constant denoting the learning rate. For complete derivation of the learning algorithm, see [4].

3 Experimental Setup

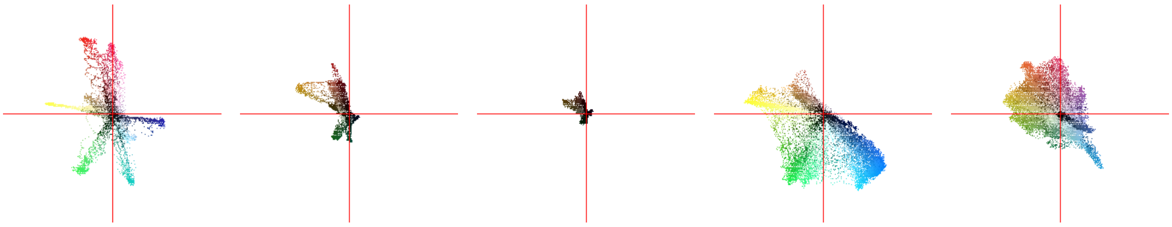
3.1 Configuration and training of the network

Figure 1 shows the schematic of our method. A quaternion feedforward neural network is used for reconstructing color information, which is trained by BP algorithm. Two images from one scene are necessary for training the network: One is taken under the condition of enough illumination, which is used for the desired output, and another for the input is taken under the condition of a little illumination. These images consist of $320 \times 240 (= 76800)$ pixels, each of which has its color values represented by 24 bits in RGB (Red, Green, and Blue) color space.



(a) Chart-Bright (b) Chart-Gloomy1 (c) Chart-Gloomy2

Figure 2: The images from the scene 'Chart' in three illuminations. The color version of the figures in this paper will be available at `http://wwwj1.comp.eng.himeji-tech.ac.jp/research/cnv/`.



(a) Chart-Bright (b) Chart-Gloomy1 (c) Chart-Gloomy2 (d) SII-Bright (e) Ham-Bright

Figure 3: Chroma distributions of images used in the experiments. The horizontal axis (or the vertical axis) in these figure displays the intensity of U (or V).

The network is trained by applying color values of each pixel in the images so that a transformation from gloomy color to finer one can be accomplished. For the input to the network, we use color values of nine pixels in the input image, i.e., a pixel located at (x, y) and its direct neighbors located at $(x+i, y+j)$ where $(i, j) \in \boldsymbol{N} = \{ (-1,-1), (0,-1), (1,-1), (-1,0), (1,0), (-1,1), (0,1), (1,1)\}$. Note that we define the location of the pixel at leftmost and uppermost as $(0,0)$ and that at rightmost and lowermost as $(319,239)$. Its corresponding desired output is the color values of a pixel located at (x, y) in the target images. The pixels on edges of the images are not use for the training of the network because they have only partial neighbors; $\{(x+i,y+j)|\{(i,j)\} \subset \boldsymbol{N}\}$. All pixels in the region of $1 \leq x \leq 318$ and $1 \leq y \leq 238$ are used for training, thus the number of training data is $318 \times 238 (= 75684)$.

Quaternion neurons can treat three components (dimensions) of data, thus the number of neurons in the input or output layer just equals to the number of pixels for the input or output. The structure of the network is three layered, there are nine neurons in the input layer and one neuron in the output layer. We denote the configuration of this network as 9-m-1, where m represents the number of neurons in the hidden layer.

We compare our quaternion neural network with a conventional (=real-valued) network. So the real-valued network is prepared and trained for the same task. In the case of real-valued neural network, the number of neurons in the input and output layer is three times as much as that for quaternion network, thus the network configuration is denoted as 27-m-3.

3.2 Images for training and evaluating

We obtain the images under a variety of illuminations for three scenes. We denote these scenes are named Chart, SII, and Ham, respectively. We introduce an averaged brightness ABr of the image for the quantitative evaluation of the performance. ABr for a image consists of $X \times Y$ pixels is defined as

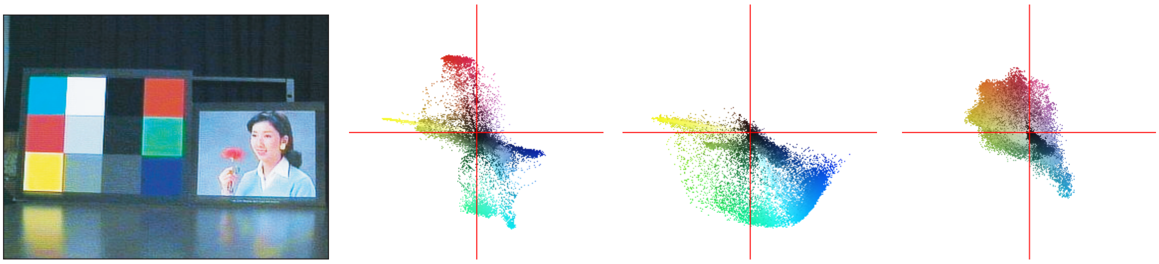
$$ABr = \frac{1}{X \times Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (0.299 R_\alpha(x,y) + 0.587 G_\alpha(x,y) + 0.114 B_\alpha(x,y))$$

where $R_\alpha(x,y), G_\alpha(x,y)$, and $B_\alpha(x,y)$ represent the intensity of color components, Red, Green, and Blue, respectively, for the pixel at (x, y) of the image . Table 1 shows the ABr of the images used for our experiments. The images attached 'Bright' label is taken under moderate illumination (that is almost equal to the condition in the daylight). The images labeled 'Gloomy1' are taken under the condition of darkish illumination. The images with the 'Gloomy2' label have smaller valued of those with the 'Gloomy1' label.

Table 1: Averaged brightness for images used for our experiments.

	Bright	Gloomy1	Gloomy2
Chart	116.32	28.36	9.32
SII	124.53	23.75	7.52
Ham	113.84	30.83	14.01

Besides averaged brightness, we also define the chroma, i.e., the purity of a color in an image as



(a) Output image for applying Chart-Gloomy1 as input

(b) Chart-Gloomy1 $ABr = 95.58$ $PSNR = 13.74(dB)$

(c) SII-Gloomy1 $ABr = 91.07$ $PSNR = 11.22(dB)$

(d) Ham-Gloomy1 $ABr = 113.75$ $PSNR = 19.54(dB)$

Figure 4: Output image and chroma distributions by the Gloomy1-network from quaternion neural network.

follows; An image represented in RGB color space is transformed into that in YUV(or YCbCr) color space. It is then projected onto UV plane (this is realized by discarding Y(brightness) information from it). Figures 2(a), (b), and (c) show three images from the scene 'Chart' and Figs.3(a), (b), and (c) show their chroma distributions. The distributions of chroma for the images under low illumination (Figs.3(b) and (c)) are contracted from that under appropriate illumination (Fig.3(a)). The task of reconstructing color information from a gloomy image corresponds to making a correct chroma distribution from contracted one. Figures 3(a), (d) and (e) are the correct(target) chroma distribution for the scenes 'Chart', 'SII' and 'Ham', respectively. This distribution can be used for the evaluation for output images from neural networks.

We also introduce the PSNR (peak-signal to noise ratio) for images with respect to the target images (images with 'Bright' label). The PSNR between two color images, named ϕ and ϕ', consist of $X \times Y$ pixels is defined as

$$PSNR = 10\log_{10}\left(\frac{255^2+255^2+255^2}{MSE}\right),$$

$$MSE = \frac{1}{X \times Y}\sum_{x=1}^{X}\sum_{y=1}^{Y}\{(R(x,y)-R'(x,y))^2 + (G(x,y)-G'(x,y))^2+(B(x,y)-B'(x,y))^2\},$$

where $R(x,y), G(x,y)$, and $B(x,y)$ in the Mean Square Error (MSE) term represent the intensities of the components for the image ϕ and $R'(x,y), G'(x,y)$, and $B'(x,y)$ are those for the image ϕ'. Table 2 shows the PSNRs for the gloomy images that labeled 'Gloomy1' or 'Gloomy2', with respect to their bright images.

Table 2: The PSNRs between the gloomy images and their bright images.

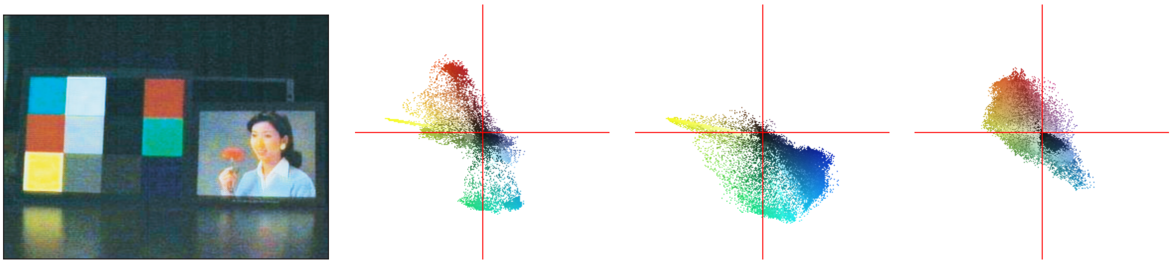
	Gloomy1	Gloomy2
Chart	7.70	5.87
SII	6.78	5.45
Ham	8.19	6.39

4 Experimental Results

This section describes the experimental results for reconstructing color information from gloomy images by our method. A Quaternion neural network with the 9-6-1 configuration (the number of trainable parameter is 261) is introduced for reconstructing, and for comparing the performance, a conventional, real-valued neural network with the 27-8-3 configuration (the number of trainable parameters is 251) is also used. The number of training iterations is 10000, where the decrease of MSE has already halted.

The images from the scene Ham are used for the training of the networks. The image with 'Ham-Bright' label (that is obtained under moderate illumination) is used to the desired output of the network, and the images with 'Ham-Gloomy1' or 'Ham-Gloomy2' label are applied as the input. When the network is trained by applying the image labeled 'Ham-Gloomy1', we name this network Gloomy1-network. Similarly, we name the network Gloomy2-network in the case of applying the image labeled 'Ham-Gloomy2' to this network in training. The images from the scenes Chart and SII are used for the evaluating of the networks.

We construct the Gloomy1-network from the quaternion neural network. After training, the test images, Chart-Gloomy1 and SII-Gloomy1, are input to the network. Figure 4 shows the outputs from the network. The output image with respect to the input Chart-Gloomy1 is shown in Fig.4(a), we see that a similar image to the original (Chart-Bright, see Fig.2(a)) can be produced by quaternion neural network. The values of ABr are 95.58, 91.07, and 113.75, for the output images from Chart-Gloomy1, SII-Gloomy1, and Ham-Gloomy1, respectively. These values increase near those of the original images. The PSNRs for the output images are calculated with respect to their target images. We obtain $13.74(dB)$, $11.22(dB)$, and $19.5\ 4(dB)$ for the output images in the case of applying Chart-Gloomy1, SII-Gloomy1, and Ham-Gloomy1, respectively. The values of PSNR are improved from those of input images (see Table 2). The chroma distributions of output images are also



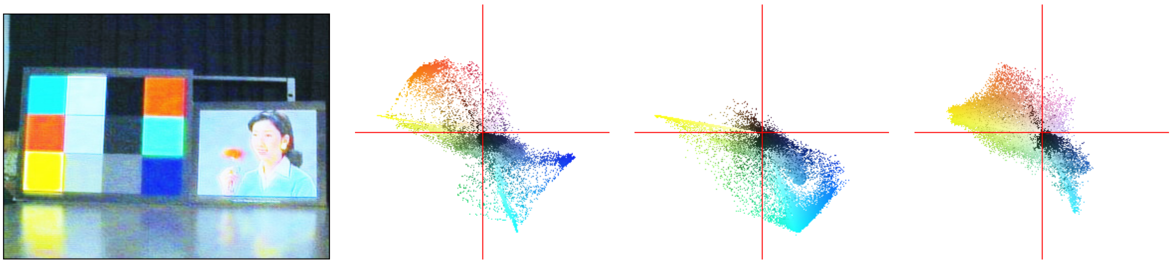
(a) Output image for applying Chart-Gloomy2 as input

(b) Chart-Gloomy2 $ABr = 72.16$ $PSNR = 11.07(dB)$

(c) SII-Gloomy2 $ABr = 72.15$ $PSNR = 9.55(dB)$

(d) Ham-Gloomy2 $ABr = 113.95$ $PSNR = 18.07(dB)$

Figure 5: Output image and chroma distributions by the Gloomy2-network from quaternion neural network.



(a) Output image for applying Chart-Gloomy1 as input

(b) Chart-Gloomy1 $ABr = 135.23$ $PSNR = 14.23(dB)$

(c) SII-Gloomy1 $ABr = 120.17$ $PSNR = 13.28(dB)$

(d) Ham-Gloomy1 $ABr = 153.84$ $PSNR = 12.53(dB)$

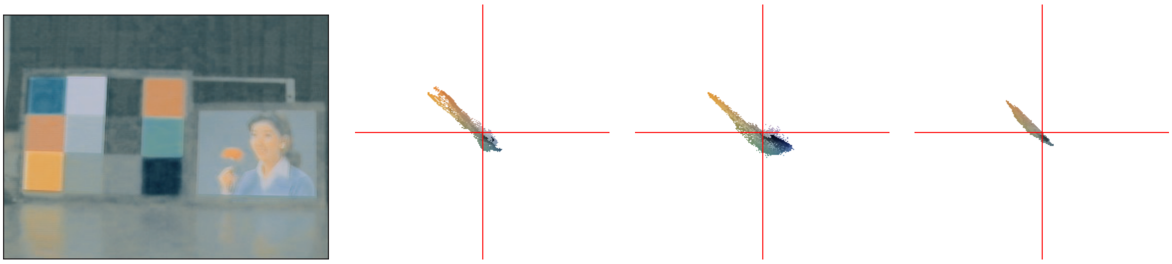
Figure 6: Output image and chroma distributions by the Gloomy2-network from quaternion neural network.

shown in Figs.4(b), (c), and (d), which correspond to Chart-Gloomy1, SII-Gloomy1, and Ham-Gloomy1 respectively. These distributions resemble those of original images, thus correct color information can be retrieved.

In Fig.5, we also show the output image and chroma distributions in the case that Gloomy2-network from the quaternion neural network is used. Because the images labeled ‘Gloomy2’ have less color intensities than those of ‘Gloomy1’, the produced images are unclear, but the chroma distributions of them show that a reconstruction ability with respect to color still remains. The values of *ABr* are 72.16, 72.15, and 113.95 for the output images from Chart-Gloomy2, SII-Gloomy2, and Ham-Gloomy2, respectively. These values increased near those of the original images. The PSNRs with respect to the target images are $11.07(dB)$, $9.55(dB)$, and $18.07(dB)$ for the input of Chart-Gloomy2, SII-Gloomy2, and Ham-Gloomy2, respectively. The values are lower than those in the case of Gloomy1-network, but are improved from those of input images (see Table 2).

We evaluate the sensitivity of the trained network. This is realized by using images with Gloomy1 label as inputs to Gloomy2-network. Figure 6 show the output image and chroma distribution with respect to the input images labeled ’Gloomy1’ to the Gloomy2-network. The output image in the case of applying Chart-Gloomy1 is shown in Fig.6(a). This image is rather whitish than Chart-Bright, color information is disappeared in some region of the output image. The values of *ABr* are 135.23, 120.17, and 153.84 for the output images from Chart-Gloomy1, SII-Gloomy1, and Ham-Gloomy1, respectively. These values are almost same or over those of the original images. The values of PSNRs are $14.23(dB)$, $13.28(dB)$, and $12.53(dB)$ with respect to the input image of Chart-Gloomy1, SII-Gloomy1, and Ham-Gloomy1, respectively. Though the PSNRs are higher than those from the Gloomy1-network, some colors in images are lost. From Figs.6(b), (c), and (d), we see that the pixel values are distributed broader than those of original images (Figs3(a), (d), and (e)). Gloomy2-network is trained to perform amplification of the input image more strongly than Gloomy1-network, thus the values of pixels in output images may sometimes be saturated.

We have investigated the same task for real-valued neural network. Figure 7 shows the results by Gloomy1-network constructed from the real-valued neural network. From Fig.7(a), we see that the quality of the image is less than that from quaternion neural network (Fig.4(a)). This is supported by the chroma distributions shown in Figs.7(b), (c), and (d). The values of *ABr* are 106.74, 82.38, and 113.03, for the output images from Chart-Gloomy1, SII-Gloomy1, and Ham-Gloomy1, respectively. These values increased near those of the original images. In this case, the PSNRs of the output from the network are $12.83(dB)$, $9.52(dB)$, and $13.81(dB)$ in the case



(a) Output image for applying Chart-Gloomy1 as input

(b) Chart-Gloomy1 $ABr = 106.74$ $PSNR = 12.83(dB)$

(c) SII-Gloomy1 $ABr = 82.38$ $PSNR = 9.52(dB)$

(d) Ham-Gloomy1 $ABr = 113.03$ $PSNR = 13.81(dB)$

Figure 7: Output image and chroma distributions by the Gloomy1-network from real-valued neural network.

of using Chart-Gloomy1, SII-Gloomy1, and Ham-Gloomy1, respectively, as input. Though the values of *ABr* and PSNR are improved from those of input images, the chroma distribution is not close to the original images. From these results, we see that the real-valued neural network does simply amplify the brightness of the input images and ignore the chroma distributions of them.

This is due to the difference in treatment of data for quaternion neural network and real-valued one. In quaternion neural network, neurons can accept three-dimensional data at a time and it is transformed geometrically by operators in neurons, hence they are expected to grasp relations in the data. On the other hand, real-valued neurons can only deal with real-valued data, i.e., one-dimensional data. For processing multi-dimensional data, the neurons of the same number as the dimension in data are necessary. The real-valued neural network has to obtain the relations between input and output data as well as the relations among component of a data. This additional task may be overhead for real-valued neural networks.

5 Conclusion

We have proposed the method for extracting color information from gloomy images. Our quaternion neural network is used for the transformations of pixel values, and the training is accomplished by applying pixels in a gloomy image as input to the network and those in its finer image as output. From the experimental results, our proposed method can extract color information from gloomy images that are not used in training. It is also confirmed by introducing the criteria, i.e., the averaged brightness, the chroma distribution, and the PSNR in the reconstructed images. We obtain the better images by using quaternion neural network for transformations, rather than real-valued one. It is due to the geometrical operators in quaternion neurons.

Our proposed networks have sensitivity with respect to input images. In other words, we obtain fine images when the brightness of input images are similar to the brightness of images used in training. In the applications of our method to the environments where illumination changes in time, we should prepare several networks trained by images with different brightness and choose one according to the brightness of the input image. The values of *ABr* for the input image will be useful for choosing the suitable network.

Our proposed method can be applied for real-time applications, such as navigation in autonomous mobile systems. This is a future work. We will proceed to hardware implementation of our method for these applications.

References

[1] A. M. Waxman, A. N. Gove, D. A. Fay, J. P. Racamato, J. E. Carrick, M. C. Seibert, and E. D. Savoye. Color Night Vision: Opponent Processing in the Fusion of Visible and IR Imagery. *Neural Networks*, 10(1):1–6, 1997.

[2] S. Das and W. K. Krebs. Sensor fusion of multispectral imagery. *Institution of Electrical Engineers: Electronics Letters*, 36:1115–1116, 2000.

[3] T. Isokawa, T. Kusakabe, N. Matsui, and F. Peper. Quaternion Neural Networks and Its Application. In *LNAI-2774 (KES2003)*, pages 318–324, 2003.

[4] N. Matsui, T. Isokawa, and H. Kusamichi. Quaternion Neural Network with Geometrical Operators. *Journal of Intelligent & Fuzzy Systems*, 2004. (in press).