博士論文

# Study of Noise Robust Bit-Depth Expansion for High Dynamic Range Imaging

# 高ダイナミックレンジ画像処理のための ノイズにロバストなビット長拡張に関する研究

Akira MIZUNO

Graduate School of Information Science and Technology,

Hokkaido University

February, 2016

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

Bit-depth expansion (BDE) is desired in high dynamic range (HDR) imaging. Prior research on an HDR display system [16] has shown that more than 10 bits are necessary for the number of quantization levels to cover the desired range without quantization artifacts.

When displaying a relatively low bit-depth (LBD) image on a monitor, the brightness gets discontinuous values. Histogram equalization or other image enhancement methods may generate a sparse histogram. These discontinuities of brightness create an artifact called a false contour, appearing in low spatial frequency or low contrast regions such as the sky, sea, or skin. The false contour, which does not exist in the original scene, seriously degrades the image quality. Therefore, a method for reconstructing the original smooth gradation by interpolating the false contour is required. Coming up with an optimized reconstruction method is currently the most important issue in BDE.

## 1.2 Goal of This Thesis

In the analog part of an imaging system, signal intensity fluctuations occur due to noise (e.g. thermal noise in the image sensor). After that, in the digital part, the intensities are rounded off to limited levels. The latter process, which is quantization, increases the intensity of fluctuation errors caused by stochastic resonance. These errors are viewed as false contour artifacts in the gradation region. Our goal was to obtain the original signal from the quantized noisy signal. We formulated a probabilistic model based on this quantization process, and successfully reconstructed smooth gradations from noisy contours. Subjective evaluation by voting clarified that the output image has higher quality.

## 1.3 Outline

First, Chapter 2 and Chapter 3 are written as introductory topics in digital image processing and probabilistic inference respectively. The Chapter 2 includes explanations about terms commonly needed in image processing. The Chapter 3 includes basic definitions about probability theory, and introduces how to guess unknown events using Bayesian probabilities.

After that, our researches about the BDE is written. In Chapter 4, we discuss about situations when the BDE is needed, and prepare definitions generally used in BDE problem. The important term “false contour artifact” is expressed in this chapter.

Chapter 5 gives one of the methods in our research—smooth gradation reconstruction from the contour artifacts using Poisson equation.

In Chapter 6, the problem in BDE for natural images are discussed. Difficulty of BDE for noisy false contour artifacts is explained.

In Chapter 7, our BDE method targeting natural images is explained. We formulated a probabilistic model based on this quantization process, and evaluate image quality of outputs.

# Chapter 2

# Digital Image Processing

The digital image processing is required in various scenes because of the spread of multimedia products. This chapter explains fundamental information for digital image processing.

## 2.1 Analog to Digital Conversion

Analog signals must be converted to digital signals for processing on our computers. The way to convert continuous signal to discrete signal is explained in this section.

### 2.1.1 Sampling

The sampling means discretization of time. The timings of the sampling is indicated as vertical lines in Fig. 2.1. The analog signals are recorded in fixed interval. When we discuss about digital images, the time points are synonyms of spatial positions of pixels.

### 2.1.2 Quantization

The quantization discretizes the intensities of the signal after sampling. Horizontal lines in Fig. 2.1 indicates levels of the quantization. The analog intensities are trimmed to the integer values by rounding down their fractions. Normally, the number of the quantization levels is a binary number (i.e. $2^8 = 256$ levels) to be used in computers. In this paper, "bit-depth" means the data size of its quantization level. Fig. 2.2 shows same pictures in multiple bit-depth. We see that the higher bit-depth images have higher qualities.

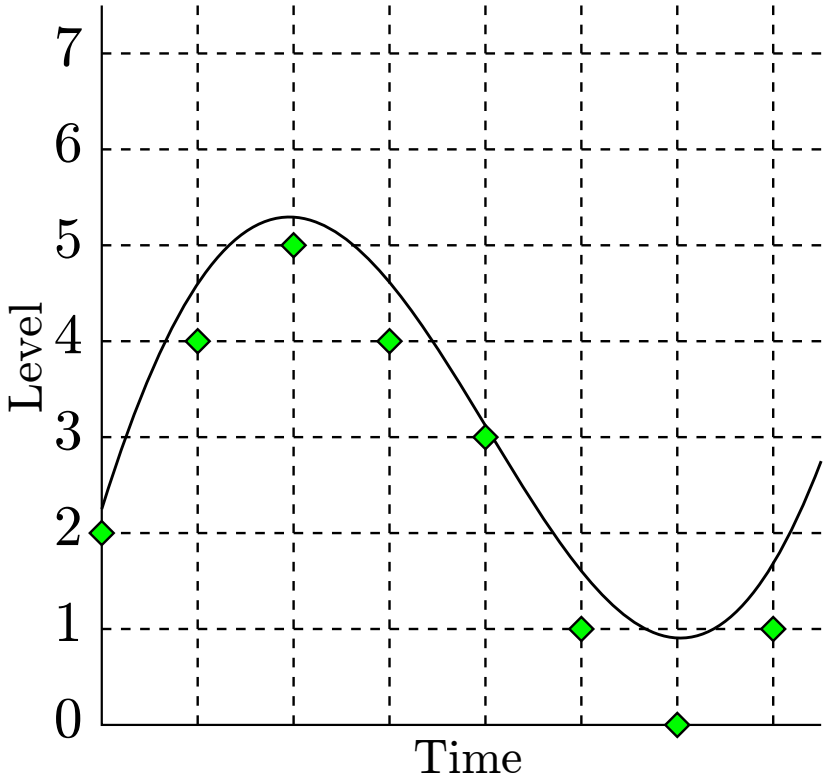



Fig. 2.1: Analog-to-digital conversion: sampling and quantization

## 2.2 Color Mixing

In photophysics, "color" is explained as frequency and intensity of light. Thus, there are infinite colors in real world.

In graphics systems, we want to manage the many colors as much as possible. This section shows you how to create a lot of colors using finite data.

### 2.2.1 Additive Color Mixing

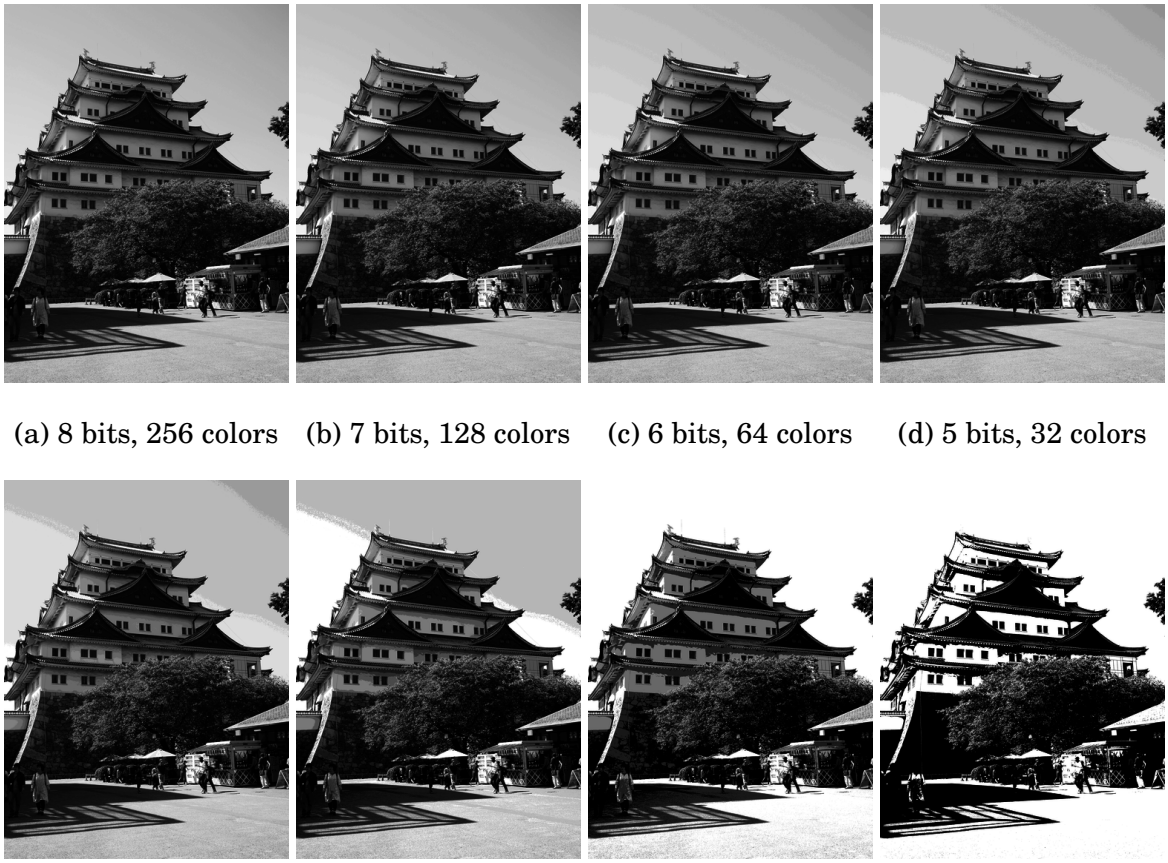
The additive color is used for display monitors. A color in the additive color system has 3 channel values: red, green, and blue channels. Colors are created by mixing the primary colored lights as shown in Fig. 2.3. The background color of the monitors is black (dark), and the other colors are gotten by emitting the 3 types of light.

### 2.2.2 Subtractive Color Mixing

The subtractive color mixing is used for printing systems. Its primary colors are cyan, magenta, and yellow. The colors are created overlaying white papers by inks (see Fig. 2.4). Generally, for better quality of black color, almost all of printing systems uses black ink in addition to the cyan, magenta, and yellow inks; this is called as "CMYK color".

### 2.2.3 Number of Colors

If each channel of the mixing systems having 3 primary colors (i.e. RGB) is stored as 8-bits data, the number of the expressible colors is $(2^8)^3 = 16,777,216$ (16.7 million colors). This is known as "True color (24-bits color)", and used as



(a) 8 bits, 256 colors (b) 7 bits, 128 colors (c) 6 bits, 64 colors (d) 5 bits, 32 colors

(e) 4 bits, 16 colors (f) 3 bits, 8 colors (g) 2 bits, 4 colors (h) 1 bits, 2 colors

Fig. 2.2: Gray scale images with multiple bit-depth

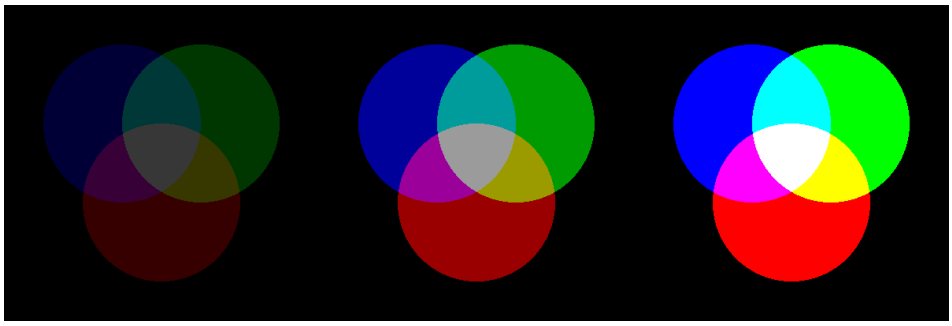


Fig. 2.3: Additive color mixing

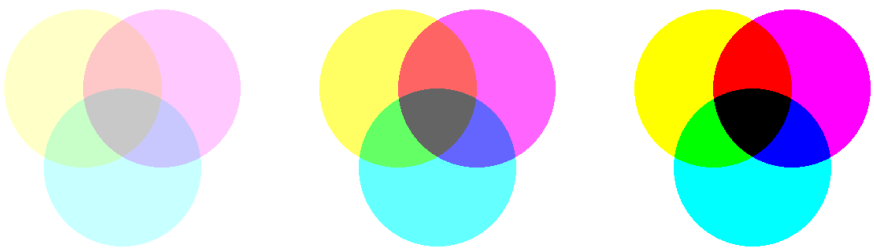


Fig. 2.4: Subtractive color mixing

standard data size of digital images. In gray scale images, however, the number of expressible colors is only $(2^8)^1 = 255$ because the gray colors are created by mixing same values of RGB.

## 2.3 Data Structure of Digital Image

The typical data structures of digital images are shown in this section. Raster images (in 2.3.1) are used for photographs, and vector images (in 2.3.2) are used for illustrations. Since our research is mainly performed for the photographs, the term "image" means the raster image in other sections of this paper.

### 2.3.1 Raster Image

The structure of raster images is shown in Fig. 2.5. A pixel of the color image has 3 channel values: red, green, and blue. Then, the pixels are set in a 2-dimensional arrangement.

The raster images consist of many pixels to draw meaningful pictures. Fig. 2.6 shows a same picture at two different ranges: distant view (Fig. 2.6(a)) and close view (Fig. 2.6(b)). If we look the picture at close range, the local areas of the image are only seen as ambiguous patterns.

### 2.3.2 Vector Image

The vector images can draw illustrations by using path and figures. If we want to draw a line segment in vector image, parameters of the line (e.g. anchor positions, line width, and color of the line) are stored in the file. This type of image formats has infinite resolution. Fig. 2.7 zooms Fig. 2.7(a) in raster and vector image. It is clear that the vector image (Fig. 2.7(c)) can draw beautiful lines.

## 2.4 Image Filtering

This section explain image filtering. The image filtering is primary tools of image processing. There are many types of filtering:

- Blur
- Edge detection
- Sharpening
- Contrast enhancement

Following sections explain about some of the blur filtering.

### 2.4.1 Averaging Filter

Averaging filter, also called moving average filter, local mean filter, or box filter, is one of basic blur method. The filtered pixel value is given as mean of neighbors of target pixel. Let $\mu_i$ is output pixel at position $i$, $\Omega_i$ denotes a local region centering pixel $i$, and $I$ is image, the averaging filter is defined as following equation:

$$\mu_i = \frac{1}{|\Omega_i|} \sum_{j \in \Omega_i} I_j, \tag{2.1}$$

where $|\Omega_i|$ denotes number of pixels in the local region $\Omega_i$.

Fig. 2.8 shows outputs of averaging filter with different sizes of local region. $r$ denotes filter radius. The image is strongly blurred when $r$ is large.

### 2.4.2 Gaussian Filter

Gaussian filter is improved version of the averaging filter by weighting. The weights of this filter is given by using the Gaussian function.

$$\mathrm{G}(x) = \exp(-\frac{x^2}{2\sigma^2}), \tag{2.2}$$

$$g_i = \frac{\sum_{j\in\Omega_i} \mathrm{G}(I_i - I_j) I_j}{\sum_{j\in\Omega_i} \mathrm{G}(I_i - I_j)}, \tag{2.3}$$

where $\sigma^2$ is a variance of the Gaussian function.

In the local region of the Gaussian filter, pixels around a center are weighted larger than outer pixels. This filter can remove high frequency signal efficiently (See high frequency part in Fig. 2.9). Therefore, the Gaussian filter is commonly used as low-pass filter.

### 2.4.3 Median Filter

Median filter takes median value of local region for each pixel. Fig. 2.10 shows the filtered images by the median filter.

The median filter rejects exceptional value in local region. Therefore, the median filter can be used to remove impulse noise. Experimental result of denoise by median filter are shown in Fig. 2.11.

## 2.5 Quality Evaluation

This section explains image quality evaluation. When image quality is degraded (e.g. image compression), we want to know how much is the quality degraded? In such case, the image quality can be defined as distance between original image

and degraded image. There are many different definitions of the distance between two images. We discuss about the most often-used quality measure, Peak Signal-to-Noise Ratio.

### 2.5.1 Mean Squared Error (MSE)

Let $\Phi$ is image domain, and $I$ and $J$ denote compared images. MSE is defined as follows:

$$\mathrm{MSE} = \frac{1}{|\Phi|} \sum_{i \in \Phi} (I_i - J_i)^2, \tag{2.4}$$

where $I_i$ is represented as a pixel at position $i$ in image $I$, and $|\Phi|$ denotes number of pixels.

### 2.5.2 Peak Signal-to-Noise Ratio (PSNR)

When each image is $8$-bits gray scale image, PSNR is calculated as

$$\mathrm{PSNR} = 20 \log_{10} \left( \frac{255}{\sqrt{\mathrm{MSE}}} \right). \tag{2.5}$$

Fig. 2.12 shows three levels of degraded images by noise and their PSNR. The stronger noise affects, the lower PSNR value becomes. If the PSNR value is higher than $40$ dB, it is hard to tell the two images apart. Therefore, PSNR$\geq 40$ dB is typically regarded as high quality.

The PSNR is very useful for objective quality evaluation. However, we must note that it is not almighty tool.

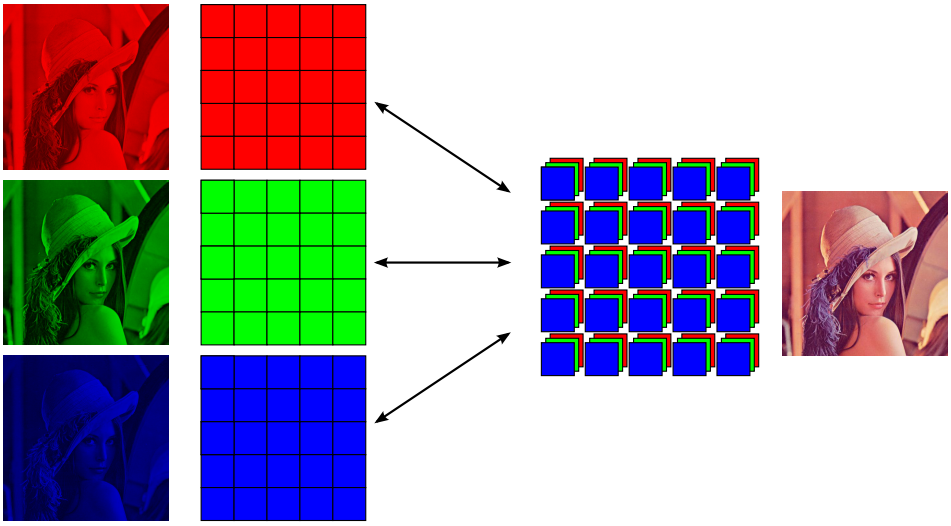


Fig. 2.5: Color image and its channels

(a) Distant view (whole image, $350 \times 300$ pixels)

(b) Close view (parts of (a), $11 \times 11$ pixels)

Fig. 2.6: Distant and close views of raster image

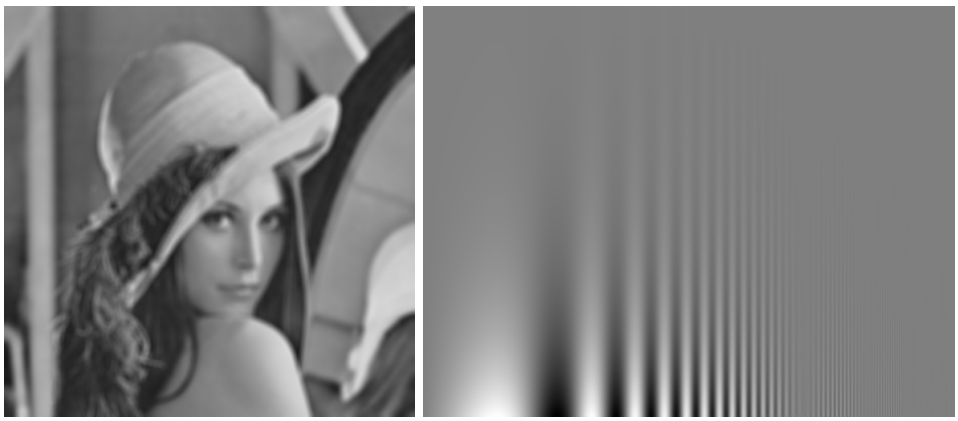

(a) Original image ($\times 1$)

(b) $\times 10$ if (a) is raster image

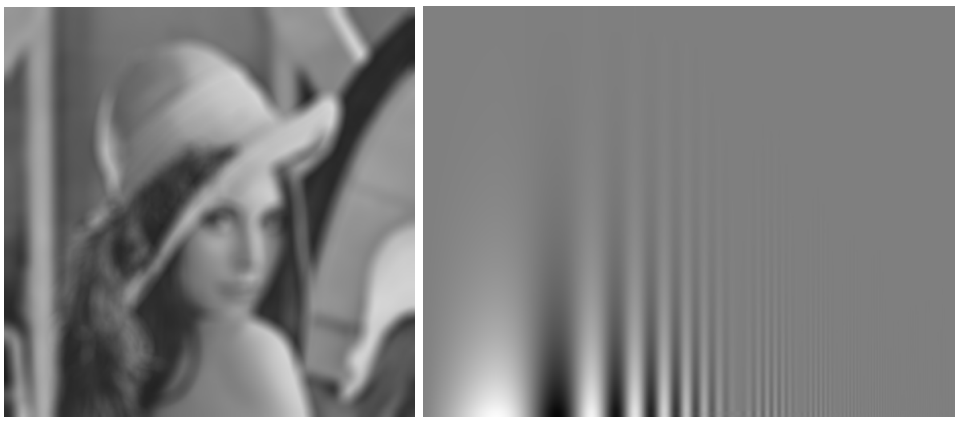
(c) $\times 10$ if (a) is vector image

Fig. 2.7: Raster and vector images

(a) Original images



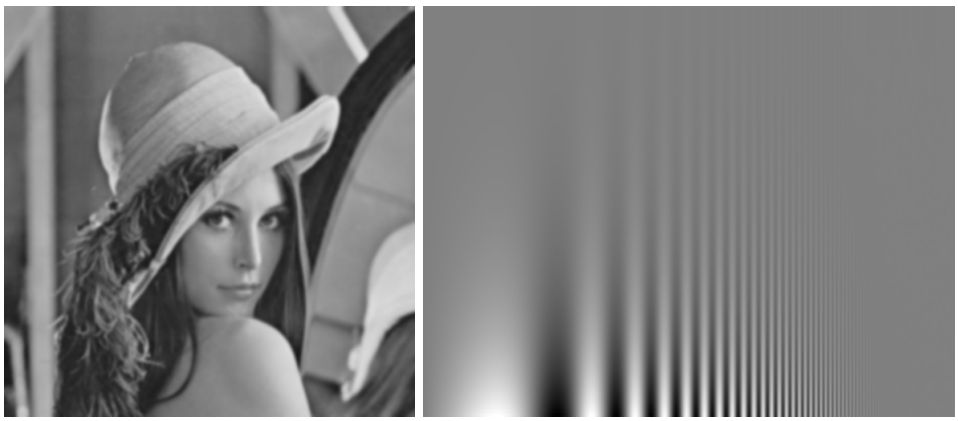
(b) Output of averaging filter ($r = 5$)



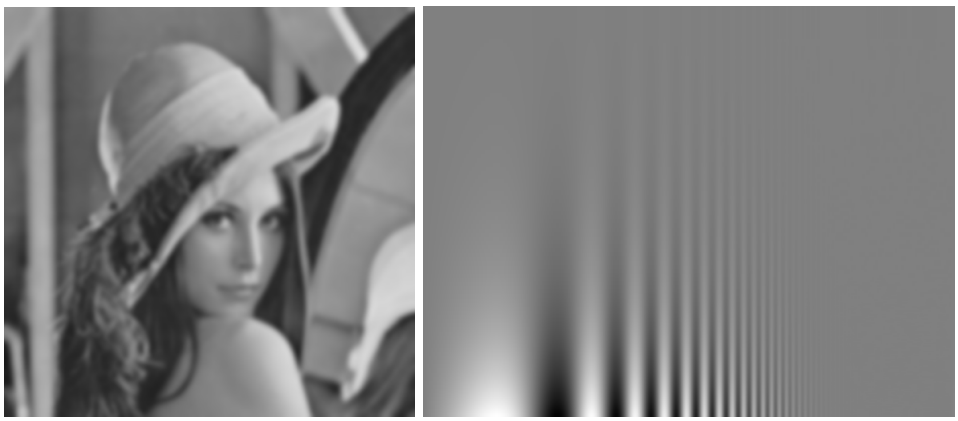
(c) Output of averaging filter ($r = 10$)

Fig. 2.8: Averaging filter

(a) Original images



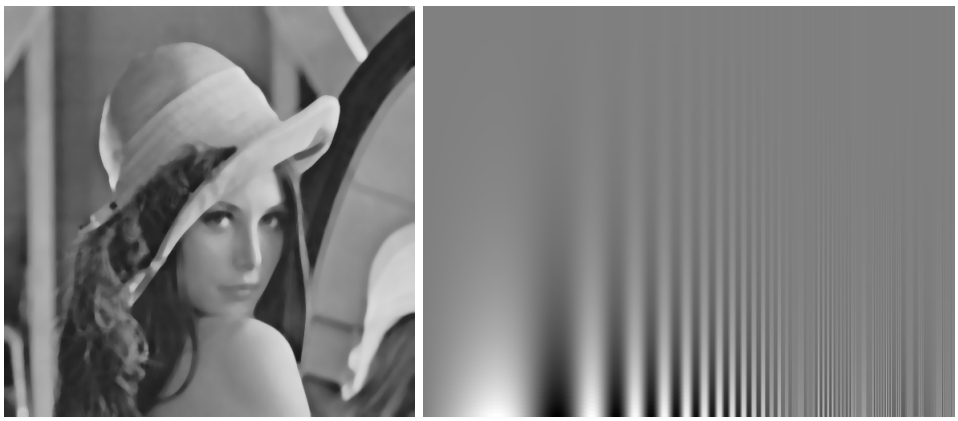
(b) Output of Gaussian filter ($r = 5$)



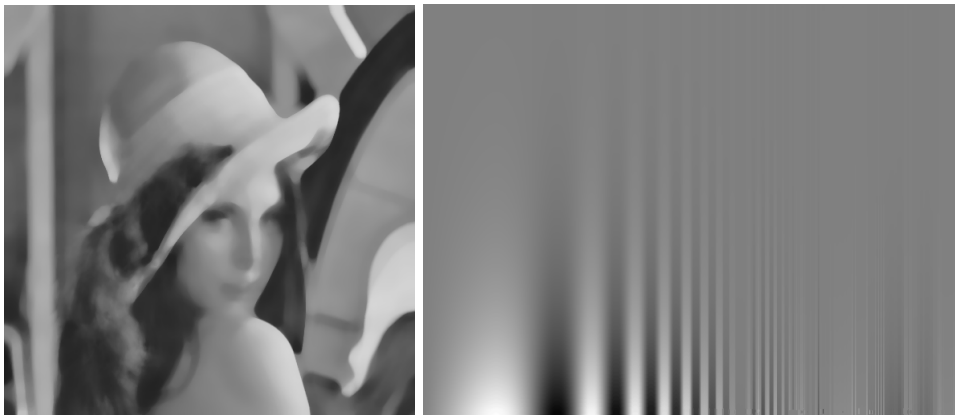
(c) Output of Gaussian filter ($r = 10$

Fig. 2.9: Gaussian filter

(a) Original images



(b) Output of median filter ($r = 5$)



(c) Output of median filter ($r = 10$)

Fig. 2.10: Median filter


(a) Original images


(b) Original image with impulse noise


(c) Denoised image by median filter ($r = 1$)

Fig. 2.11: Denoising by median filter

(a) Original image

(b) PSNR = 45.8 dB

(c) PSNR = 32.1 dB

(d) PSNR = 26.2 dB

Fig. 2.12: Quality degradation and PSNR values

# Chapter 3

# Probability and Inference

The word 'probability' has two different meanings depending on philosophical viewpoints. One side of the viewpoints is called frequentism. It describes the probabilities as *average frequencies* of outcomes in repeatable random experiments. The other side of the viewpoints, which is Bayesian, says the probabilities are understood as *degrees of belief* [12]. This represents that the Bayesian probabilities can quantify credence or assurance of estimations in one's mind. Thus the Bayesian probabilities are an useful tool to express one's assumptions and inferences given those assumptions.

In this paper, 'probability' is used as a synonym for 'Bayesian probability'. The 'Bayesian probability' and some basic terms are expounded at first of this chapter. After that, inferences based on probabilistic models are discussed.

## 3.1 Probability

This section is written about keywords in probability theory. They are common knowledge to both of the two viewpoints (frequentism and Bayesian).

### 3.1.1 Probability Distribution

The probability of $x$ is written as $P(x)$ in this paper. The sum of the probabilities is $\mathrm{PMF}(x)$ is called a probability mass function, if

$$P(x) = \mathrm{PMF}(x), \tag{3.1}$$

$$\sum_{u_i \in \mathcal{U}} \mathrm{PMF}(x = u_i) = 1 \quad (0 \leq \mathrm{PMF}(u_i) \leq 1)\,, \tag{3.2}$$

where $x$ is the value of a discrete random variable, which takes one of a set of possible values $\mathcal{U} = \{u_1, u_2, \cdots, u_m\}$.

When the variable $x$ is a continuous random variable, $\mathrm{PDF}(x)$ is called a probability density function. The probabilities is defined as integration of the function, i.e.

$$P(x_1 \leq x \leq x_2) = \int_{x_1}^{x_2} \mathrm{PDF}(x)\mathrm{d}x, \tag{3.3}$$

$$\int_{-\infty}^{\infty} \mathrm{PDF}(x)\mathrm{d}x = 1 \quad (0 \leq \mathrm{PDF}(x))\,. \tag{3.4}$$

The probability that the continuous variable $x$ lies between values $x_1$ and $x_2$ can be obtained as $\int_{x_1}^{x_2} \mathrm{PDF}(x)\mathrm{d}x$.

**Range of probability distribution function**

The probability mass functions $\mathrm{PMF}(x)$ must be less than or equal to $1$ because the sum of the function values is $1$. In the continuous case, however, the probability density function $\mathrm{PDF}(x)$ may take values greater than $1$. For example, the normal distribution function Eq.(3.6) returns large values with small $\sigma^2$ (e.g. $\mathrm{Normal}(x = \mu; \mu, \sigma^2 = 0.01) = 3.99$).

#### Examples of Probability Distribution

The probability mass (density) function is also called the probability distribution function. Many discrete and continuous distributions are known in probability

theory. Here are some examples of the distributions.

■Poisson distribution The Poisson distribution is one of the discrete probability distributions.

$$\mathrm{Poisson}(r;\lambda) = \exp(\lambda)\frac{\lambda^r}{r!}, \tag{3.5}$$

where $r$ is a number of non–negative integers $r \in \{0, 1, 2, \cdots\}$, and the parameter $\lambda$ is a positive real number. The Poisson distribution arises by counting the occurrence number of random events during a fixed time interval (e.g. photon number arriving in a pixel). The probabilities of arriving photon number $r$ are obtained by $\mathrm{Poisson}(r|\lambda)$; $\lambda$ is an average number. Fig. 3.1 shows function shapes with several $\lambda$.

■Normal distribution The normal (or Gaussian) distribution function is the best–known of continuous distributions.

$$\mathrm{Normal}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{3.6}$$

where $\mu$ and $\sigma^2$ are mean and variance of the distribution respectively. Fig. 3.2 shows function shapes with different parameters.

■Uniform distribution If the density (mass) function has same values for its support, it is the uniform distribution. The uniform distribution can be defined as both the discrete and continuous functions.

The discrete uniform distribution function is defined as follows:

$$\mathrm{Uniform_d}(x;a,b) = \frac{1}{b-a+1} \quad (a \le x \le b), \tag{3.7}$$

where $x$, $a$, and $b$ are integers.

The continuous uniform distribution function is:

$$\mathrm{Uniform_c}(x;a,b) = \begin{cases} \frac{1}{b-a} & (a \le x \le b), \\ 0 & \text{otherwise}, \end{cases} \tag{3.8}$$

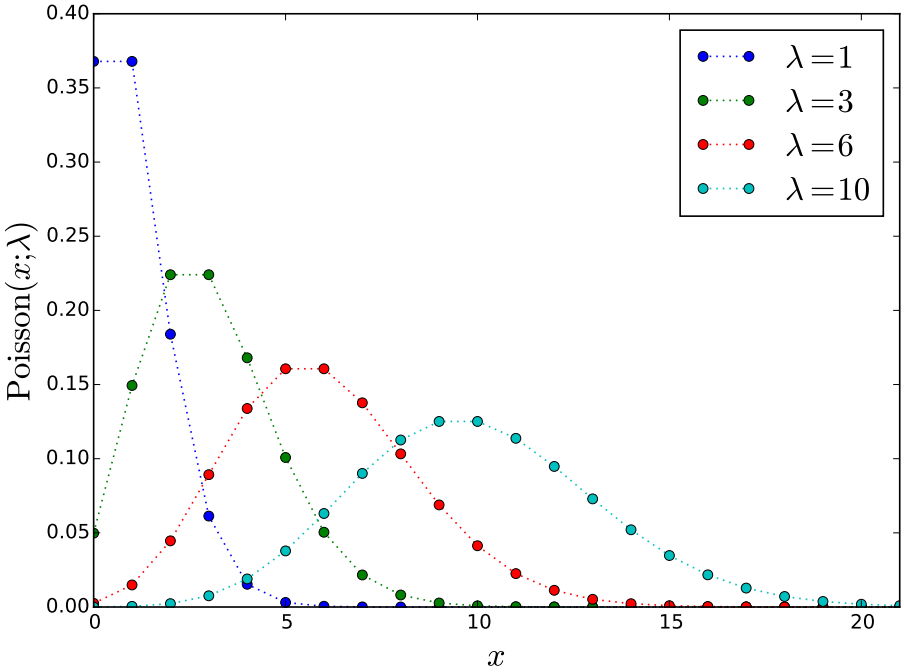



Fig. 3.1: Poisson distribution

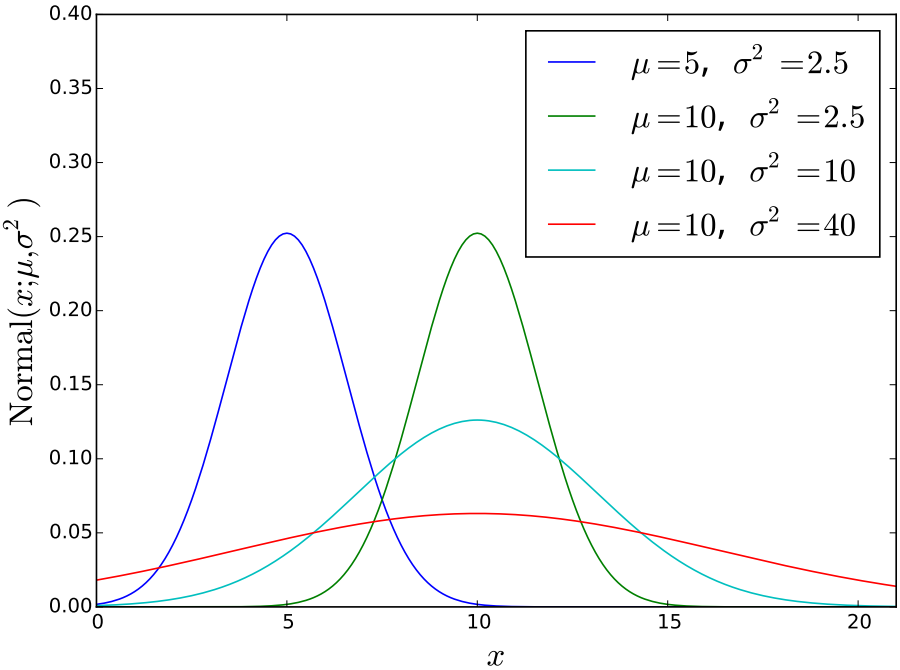



Fig. 3.2: Normal distribution

where $x$, $a$, and $b$ are real numbers.

The discrete and continuous uniform distribution functions are shown in Fig. 3.3(a) and Fig. 3.3(b) respectively.

**Relationship between Poisson and Normal distribution**

When the parameter $\lambda$ of the Poisson distribution is a sufficiently large value, the Normal distribution with parameters $\mu = \lambda$ and $\sigma^2 = \lambda$ can be used as an approximation of the Poisson distribution (See $\lambda = 10$ on Fig. 3.1 and $\mu = \sigma^2 = 10$ on Fig. 3.2).

### 3.1.2 Joint Probability

Let two or more random variables $x \in \mathcal{U} = \{u_1, u_2, \cdots, u_m\}$, $y \in \mathcal{V} = \{v_1, v_2, \cdots, v_n\}$, ..., the combinations of their outcomes are obtained like $(x = u_i, y = v_j, \cdots)$. The probabilities of the combinations are called joint probabilities. The joint probabilities of $x$ and $y$ are written as $P(x, y)$ (or $P(y, x)$).

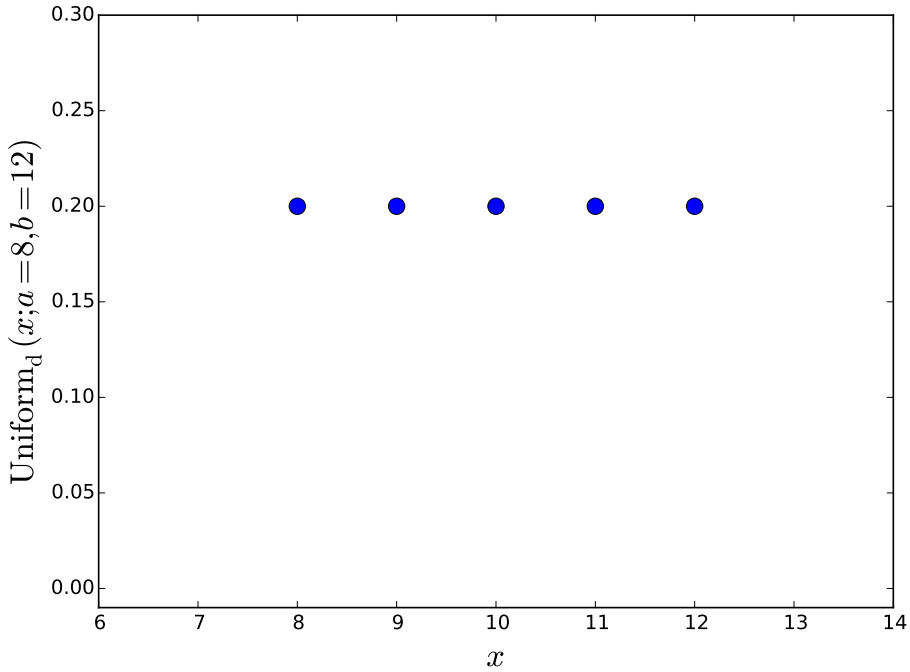
### 3.1.3 Marginal Probability

When the joint probabilities distribution $P(x, y)$ are known, we can obtain the marginal probability $P(x = u_i)$ by summing $P(x = u_i, y)$ up for overall patterns of $y$,

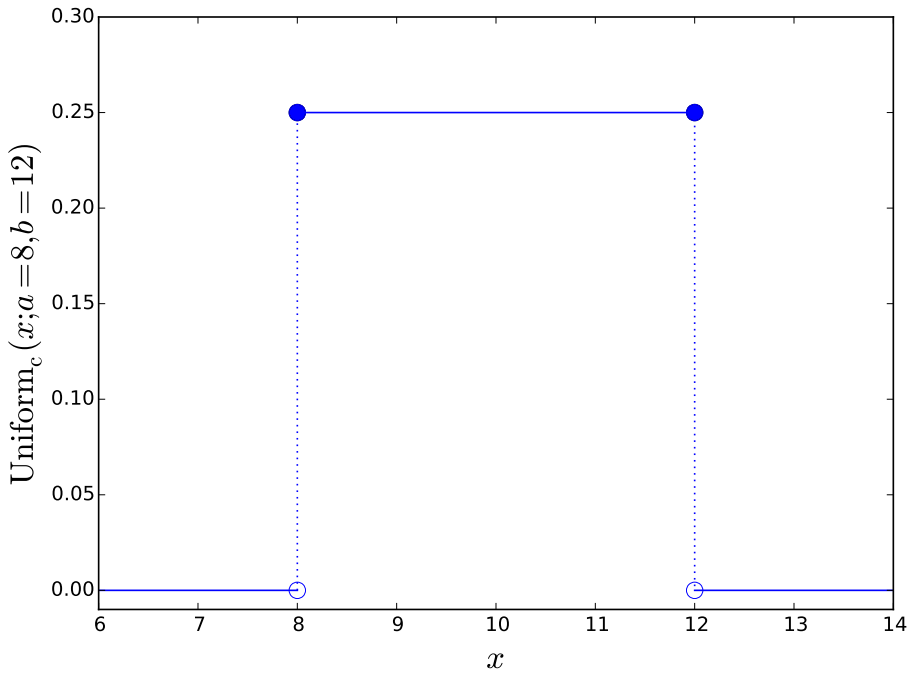
$$P(x = u_i) = \sum_{y \in \mathcal{V}} P(x = u_i, y). \tag{3.9}$$

Similarly, marginal probabilities of $y$ are,

$$P(y) = \sum_{x \in \mathcal{U}} P(x, y). \tag{3.10}$$




(a) Discrete uniform distribution




(b) Continuous uniform distribution

Fig. 3.3: Uniform distribution

### 3.1.4 Conditional Probability

The conditional probability, the probability of $x = u_i$ given $y = v_j$, is defined as:

$$P(x = u_i | y = v_j) = \frac{P(x = u_i, y = v_j)}{P(y = v_j)}, \tag{3.11}$$

where $P(y = v_j) \neq 0$. If $P(y = v_j) = 0$, the conditional probability $P(x = u_i | y = v_j)$ is undefined.

### 3.1.5 Bayes' Theorem

From the definition of conditional probabilities Eq.(3.11), following relationships are obtained,

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x). \tag{3.12}$$

This is called the chain rule.

The Bayes' theorem is derived from Eq.(3.12),

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}. \tag{3.13}$$

**Independent random variables and their probabilities**

If two random variables $x$ and $y$ are independent, the conditional probabilities $P(x|y)$ are not varied depending on outcomes of $y$. So the conditional probabilities are:

$$P(x|y) = P(x). \tag{3.14}$$

Then, from Eq.(3.12) and Eq.(3.14), the joint probabilities of independent random variables are obtained as:

$$P(x, y) = P(x)P(y). \tag{3.15}$$

## 3.2 Bayesian Inference

In this section, we discuss about the inference using the Bayesian probabilities. Now we have two random variables $x$ and $y$, where true distribution of $x$ is unknown, and we can observe outcomes of $y$. This situation is often seen in inverse problems of stochastic processes. Fig. 3.4 shows a simple scheme of stochastic processes. The outcomes generated from $x$ are stochastically changed (e.g. occurrence of errors), and the results $y$ of the flow are observed by us.

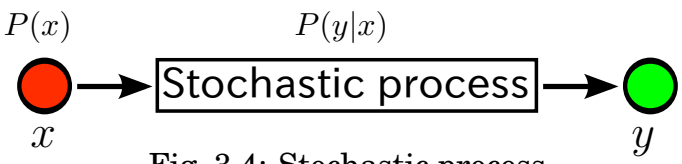



Fig. 3.4: Stochastic process

In the probability theory, the inference—estimating hidden state, forecasting future, or restoring original—is accomplished by deriving a probability $P(x|y = v_j)$ of unknown $x$ given outcome $y = v_j$. Then, a most probable value of $x$ is assumed as $x = u_i$ when $P(x = u_i|y = v_j)$ takes a maximum probability with fixed $y = v_j$.

To calculate $P(x|y = v_j)$, we rewrite it as multiplications of other probabilities from the Bayes' theorem (Eq.(3.13)):

$$P(x|y = v_j) = \frac{P(y = v_j|x)P(x)}{P(y = v_j)}, \tag{3.16}$$

where the each term of the equation is called as follow in the Bayesian inference:

$P(x)$ Prior probability,

$P(x|y = v_j)$ Posterior probability,

$P(y = v_j|x)$ Likelihood,

$P(y = v_j)$ Evidence (or Marginal likelihood).

Meanings of the terms are explained in 3.2.1–3.2.4. Please see that the right hand side of Eq.(3.16) is constructed from the probabilities of the forward direction (generation process in Fig. 3.4). When we formulate a model for the forward stochastic process, the required equation of the inverse problem $P(x|y = v_j)$ can be obtained.

### 3.2.1 Prior Probability

The prior probabilities $P(x)$ mean our assumptions or known information about $x$. The prior probability distribution does not have to be truth distribution of $x$. Therefore, it can be decided subjectively. If we think that the random variable $x$ probably follows the normal distribution, then its distribution is set like $P(x) = \mathrm{Normal}(x; \mu, \sigma^2)$.

In another case, if we do not have anything information or hints about its distribution, the uniform distribution is used normally.

The prior probability distribution is frequently written as 'prior'.

### 3.2.2 Posterior Probability

The posterior probabilities $P(x|y = v_j)$ are probabilities of $x$ given evidence $y = v_j$. That is defined when we obtain the outcome of $y$, where $y$ is related to $x$. In other words, the posterior probabilities are modified probabilities of $x$ using evidences. We can expect that the posterior (modified) probability distribution is closer to the true distribution of $x$ than the prior distribution.

### 3.2.3 Likelihood

$P(y = v_j|x)$ is the likelihood. The term likelihood has not same meaning to probability. For the equation $P(y|x)$, it is the probability function if $x$ is fixed,

and is the likelihood function if $y$ is fixed.

### 3.2.4 Evidence

$P(y = v_j)$ is called the evidence. The evidence has not dependence on $x$. When we want to discuss about probabilities relating to $x$, the evidence is considered as only a normalizing constant because it is defined as marginal likelihood:

$$P(y = v_j) = \sum_{x \in \mathcal{U}} P(y = v_j | x). \tag{3.17}$$

Table 3.1: Joint distribution when $x$ and $y$ is independent ($P(x, y) = P(x)P(y)$).

| $P(x, y)$ | | $x$ | | | | $P(y)$ |
|---|---|---|---|---|---|---|
| | | a | b | c | d | |
| $y$ | a | 0.08 | 0.1 | 0.02 | 0 | 0.2 |
| | b | 0.16 | 0.2 | 0.04 | 0 | 0.4 |
| | c | 0.12 | 0.15 | 0.03 | 0 | 0.3 |
| | d | 0.04 | 0.05 | 0.01 | 0 | 0.1 |
| $P(x)$ | | 0.4 | 0.5 | 0.1 | 0 | |

# Chapter 4

# False Contour Artifacts and Bit-Depth Expansion

## 4.1 Overview of Bit-Depth Expansion

The image data is quantized by sensors. The quantization discretizes intensity levels of the original image signal. In this case, the image quantization process is formulated as

$$\mathbf{y} = Q(\mathbf{x}), \tag{4.1}$$

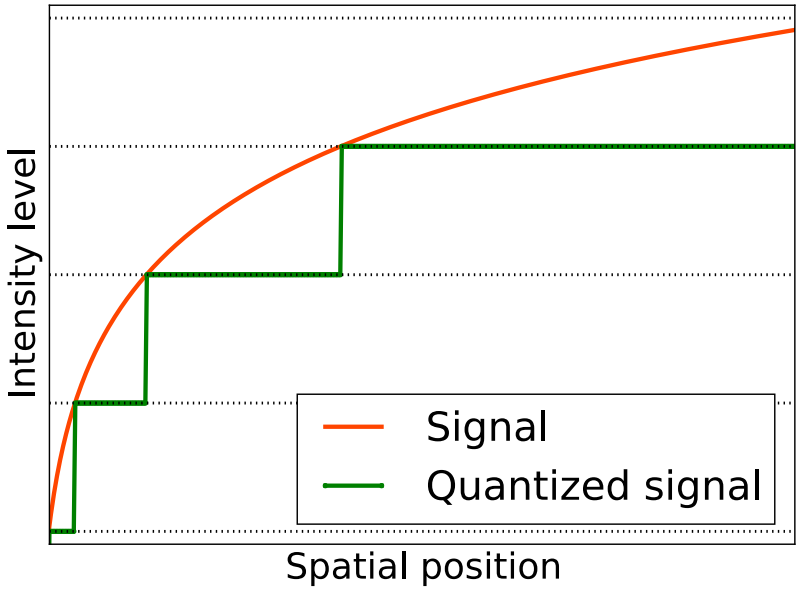
where $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T \in \mathbb{R}^n$ is an original signal, $\mathbf{y} \in \mathbb{Z}^n$ is a quantized signal, and $Q$ is a quantizer function defined as rounding off fractions in this paper. Eq.(4.1) requires that the pixel values of an estimated signal is in per-pixel quantization bins:

$$x_i^* \in [y_i, y_i + 1)\,, \tag{4.2}$$

where $x_i^*$ is the estimated value of $x_i$ $(1 \leq i \leq n)$.

The bit-depth reduction often causes the false contour artifacts. Fig. 4.1 shows appearance of the false contours in one-dimensional data. The contour artifacts

looks like a step function (See Fig. 4.1(a)). Fig. 4.1(b) shows the quantized signal and its quantized bins. We can assume that the original signal is in the gray region because of Eq.(4.2).



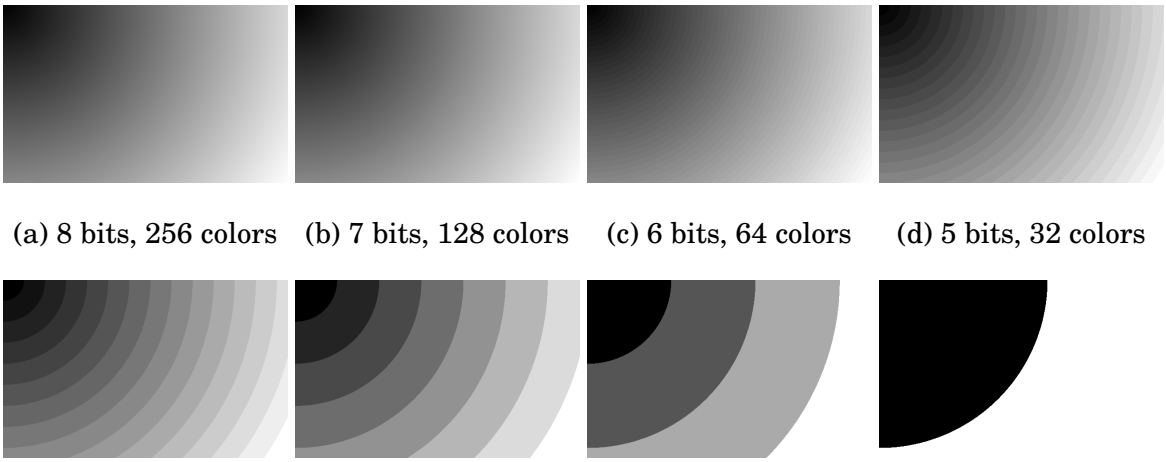

(a) Quantization


(b) Zoom–in. Quantized signal (green) and its quantization bins (gray)

Fig. 4.1: Signal quantization

## 4.2 False Contour Artifacts

The false contour is known as artifacts appeared in gradation region of low bit-depth images. Fig. 4.2 shows simple gradation images with multiple bit-depth. When the bit-depth of the image is low, the image must draw gradations by restricted number of colors. If the image has high bits (e.g. 12 bits), the gradation is drawn smooth. The low bit-depth images, however, the gradation is drawn as color bands. See also Fig. 2.2 as the examples of natural images. If the false contour artifacts appear in gradation region, the image quality is degraded seriously.



(a) 8 bits, 256 colors (b) 7 bits, 128 colors (c) 6 bits, 64 colors (d) 5 bits, 32 colors

(e) 4 bits, 16 colors (f) 3 bits, 8 colors (g) 2 bits, 4 colors (h) 1 bits, 2 colors

Fig. 4.2: False contour artifacts in gradation region

## 4.3 Bit-Depth Reduction and Expansion

In this thesis, the bit-depth reduction is done by truncating least significant bits (LSBs). The bit-depth reduction from $p$ bits to $q$ bits $(q < p)$ is defined as follows:

$$I_{\text{low}} = \lfloor \text{gain}(p, q) \cdot I_{\text{high}} \rfloor, \tag{4.3}$$

$$\text{gain}(a, b) = \frac{2^b}{2^a} = 2^{b-a}, \tag{4.4}$$

where $I_{\text{high}}$ is $p$-bit integer $(0 \leq I_{\text{high}} \leq 2^p - 1)$, $I_{\text{low}}$ is $q$-bit integer $(0 \leq I_{\text{l}} \leq 2^q - 1)$, and $\lfloor x \rfloor$ is a floor function defined as conversion from the real number $x$ to a greatest integer less than or equal to $x$,

$$\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \leq x\}, \tag{4.5}$$

where $\mathbb{Z}$ denotes the set of integers.

From Eq.(4.3), the bit-depth expansion from $q$ bits to $p$ bits is:

$$I_{\text{high}} = \text{gain}(q, p) \cdot (I_{\text{low}} + \xi), \tag{4.6}$$

where $\xi$ denotes the truncated error by the floor function. The error is called quantization error. We see that estimating the value of $\xi$ is required if $\xi$ is unknown.

### 4.3.1 Zero Padding

Zero padding is a easiest way to get higher bit-depth data. Several "zero" are simply added as LSBs. The zero padding can be written as:

$$I_{\text{high}} = \text{gain}(q, p) \cdot I_{\text{low}}. \tag{4.7}$$

This means that the truncated error $\xi$ in Eq.(4.6) is ignored. Therefore the errors are enlarged by the function $\text{gain}(\cdot)$ (Fig. 4.3).

The zero padding is used as a fastest method because it can be implemented by bit shifts only. The image quality, however, is not good because the false contour artifacts remain. The zero padding or other similar methods touch pixels independently. To reduce the false contour artifacts, we should look not just a target pixel but also its neighborhoods.

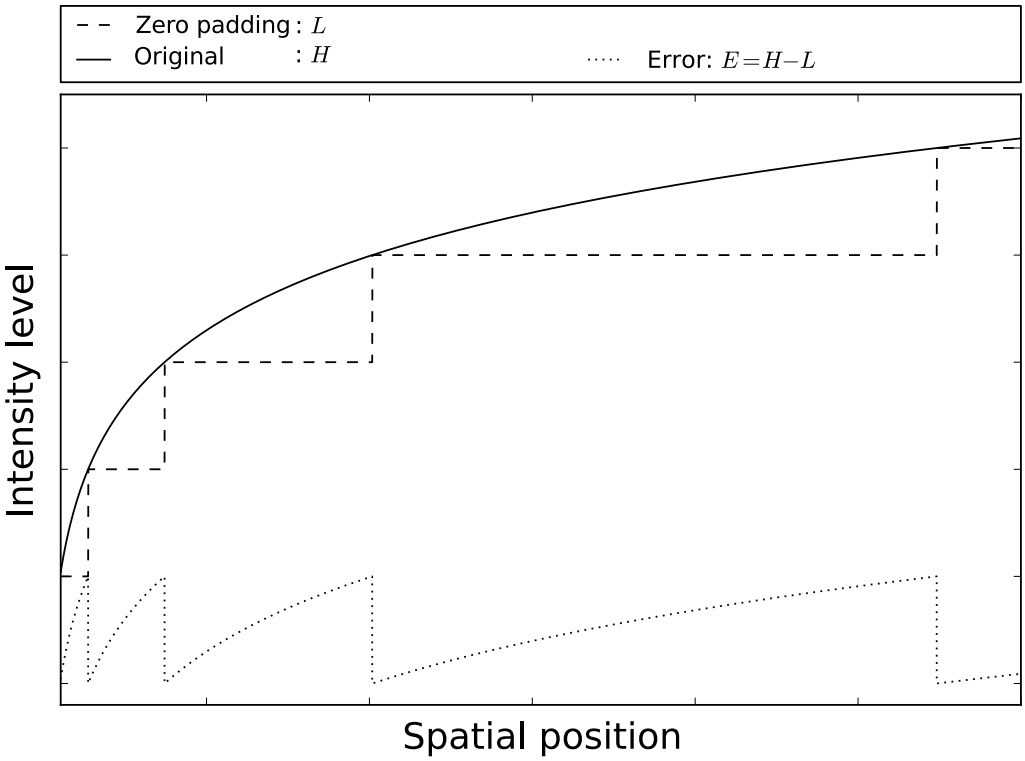



Fig. 4.3: Zero padding and its enlarged errors

## 4.4 Countermeasures for False Contour Artifacts

This section shows conventional methods to reduce the false contour artifacts.

### 4.4.1 Dithering

The dithering is a technique to generate pseudo high bit-depth images in restricted bit-depth. Fig. 4.4 shows a result of dithering. Fig. 4.4(b) and Fig. 4.4(c) are generated from Fig. 4.4(a) by reducing their bit-depth. Both of the Fig. 4.4(b) and Fig. 4.4(c) have same bit-depth (5 bits). We can see that the dithered image has better quality.

A problem of the dithering technique is that it must be applied at the time of the bit-depth reduction. If the image bit-depth is reduced already, the dithering can no longer be used to improve their image quality.

### 4.4.2 Linear Interpolation

The linear interpolation is a method to improve spatial resolution and level number of signals. The interpolated signal in Fig. 4.5 can imitate the shape of the analog signal in Fig. 2.1.

This can be used as a simple and effective method to reduce the false contour artifacts in one–dimensional domain. Continuous gradations are reconstructed by the linear interpolation as shown in Fig. 4.6(a). However, for two–dimensional domain data such as digital image, the linear interpolation is not simple. Fig. 4.6 shows directions of interpolation in two–dimensional domain like image. One or more interpolated values can be defined for same pixel. Resolving the conflict and constructing two–dimensional continuous gradations are difficult problems.

## 4.5 Conventional Methods

Many attempts have been made to address this need in previous work. Low-pass filtering methods [1, 6, 7, 10, 11, 14, 19, 23] have been proposed to smooth the


(a) Original image (8 bits)


(b) Low bit-depth image (8→5 bits)


(c) Dithered image (8→5 bits)

Fig. 4.4: Dithering

contour region. Dithering-based methods [1, 2, 5, 23] have been shown to reduce the visibility of undesirable artifacts. Flooding-based methods [4, 20, 22] can convert the 2D extrapolation problem into a 1D interpolation. Optimization-based methods [21, 22] maximize both accuracy and smoothness as much as possible.

Some of the methods are described in this section.

### 4.5.1 Flooding-based Linear Interpolation

To apply the linear interpolation to two-dimensional domain, the flooding-based methods are proposed.

Its conceptual scheme is shown in Fig. 4.7 in one-dimensional domain. Now, we want to decide the level of a interpolated pixel. The interpolated pixel exists between the edges of the step function. Then, we can see that the ratio of the spatial distances from target pixel to edges $(D_1, D_2)$ and its distances of range $(V_1, V_2)$ are same:

$$D_1 : D_2 = V_1 : V_2. \tag{4.8}$$

Therefore we can get the interpolated value $V_1$ as

$$V_1 = \frac{D_1}{D_1 + D_2} \cdot (V_1 + V_2) \tag{4.9}$$

$$= \frac{D_1}{D_1 + D_2} \cdot V_{\mathrm{const}}. \tag{4.10}$$

This law can be used in two-dimensional space. If we know the distances from target pixel to upper and lower contours, the two-dimensional linear interpolation can be applied. The conventional methods count the distances from contour edges by iteration (See Fig. 4.8). The distance counting continues until distances fulfilled for all pixels. Therefore the iteration number is depending on the contour width.

### 4.5.2 BDE using Spectral Graph Theory

Wan et al. developed one of the optimization-based methods. We describe a brief description of the graph Fourier transform and an application for BDE are provided.

#### Graph Fourier transform

The graph Fourier transform is a technique of graph signal processing. It enables analyzing signals on graphs in the graph spectral domains. We can use this to measure the smoothness of an image.

Now we define a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ that consists of a set of vertices $\mathcal{V}$ (i.e., pixels of an image), a set of edges $\mathcal{E}$, and an adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$. $\mathbf{W}$ store edge weights between one pixel and another. If there is an edge $e = (i, j)$ connecting vertices $v_i$ and $v_j$, the entry $W_{i,j}$ represents the weight of the edge; otherwise, $W_{i,j} = 0$.

The graph Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where the degree matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix satisfying $D_{i,i} = \sum_j W_{i,j}$. The graph Fourier transform is defined as

$$\lambda = \mathbf{v}^T \mathbf{L} \mathbf{v}, \tag{4.11}$$

where $\mathbf{v} = [v_1, v_2, \ldots, v_n]^T \in \mathbb{R}^n$ is a graph signal. When the graph signal $\mathbf{v}$ is smooth, the scalar $\lambda$ takes a small value. Thus, the graph Fourier transform can be used to measure the smoothness of a graph signal. Refer to (Shuman et al., 2013[17]) for more details.

#### BDE via the maximum a posteriori (MAP) estimation

One particular BDE method estimates the original image by using a graph Fourier transform and MAP estimation [21]. To find the most probable $\mathbf{x}$ from

an observed $\mathbf{y}$, they use a MAP solution (defined in Eq.(7.3)).

Their optimization problem is formulated as (see also their paper[21]):

$$\min_{\mathbf{x}_A} \sigma_l^2 \mathbf{x}_A^T \mathbf{M}^p \mathbf{x}_A + \sigma_q^2 ||\mathbf{x}_A - \mathbf{f}_A||_2^2, \tag{4.12}$$
$$s.t. |x_{A_i} - y_{A_i}| < 0.5, \forall i, \sum_i x_{A_i} = 0,$$

where $\sigma_l, \sigma_q$ are parameters to decide the balance of terms, $\mathbf{x}_A$ is an AC component of $\mathbf{x}$, $\mathbf{M}$ is a graph Laplacian matrix designed for their problem, $p$ is a positive integer, and $\mathbf{f}_A$ is the fidelity. The problem Eq.(4.12) is iteratively solved. The fidelity $\mathbf{f}_A$ is initialized as $\mathbf{y}_A$, which is an AC component of observed $\mathbf{y}$, at first iteration. After that, $(k+1)$–th solution $\mathbf{x}_A^{*(k+1)}$ is obtained by solving Eq.(4.12) with fidelity updated to $(k)$–th solution: $\mathbf{f}_A = \mathbf{x}_A^{*(k)}$. The graph Fourier transform is used as a smoothness term.

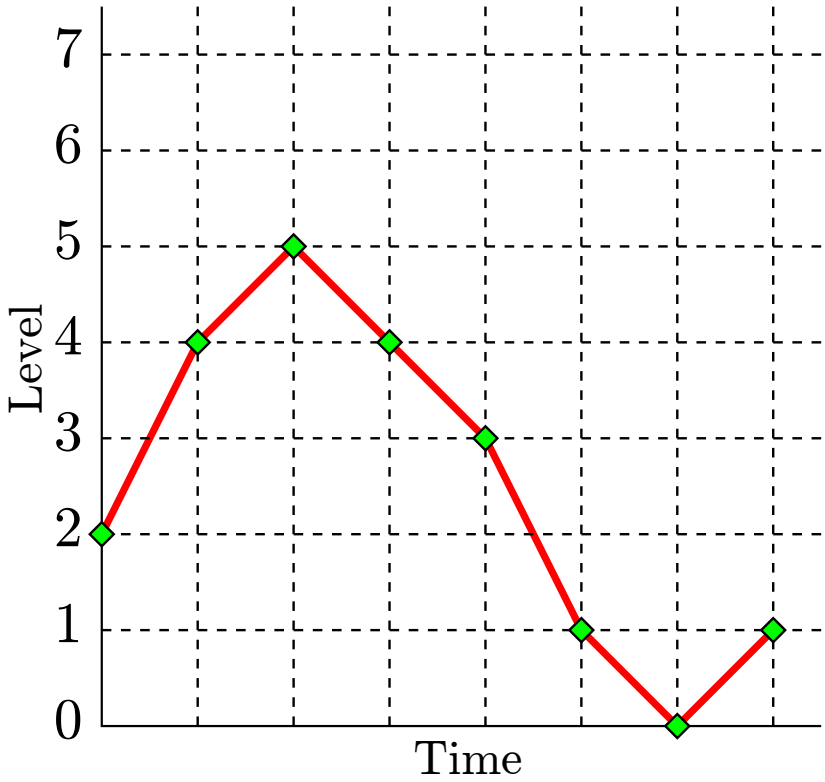
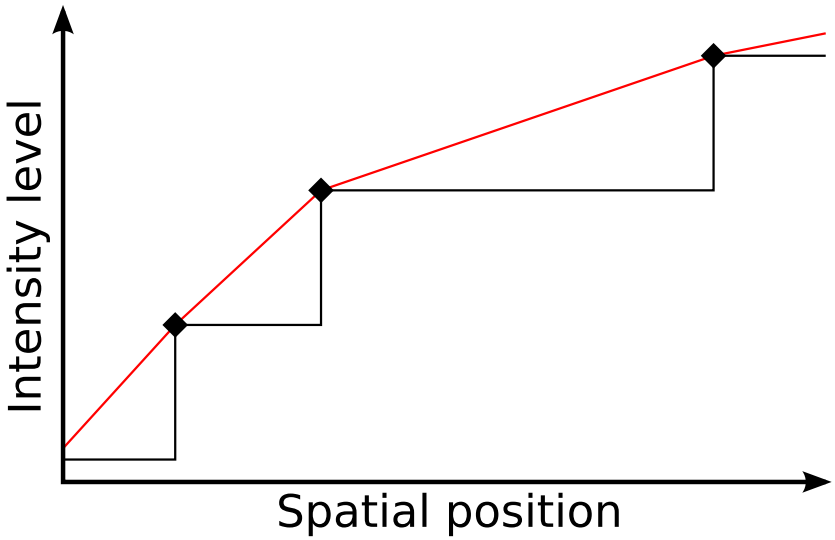
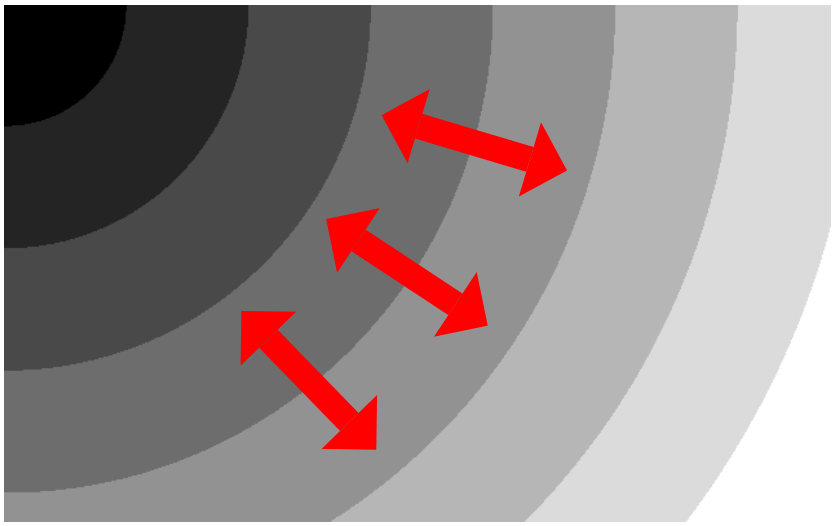



Fig. 4.5: Linear interpolation




(a) Contour interpolation in one–dimensional domain



(b) Directions of interpolation in two–dimensional domain

Fig. 4.6: Gradation reconstruction by linear interpolation

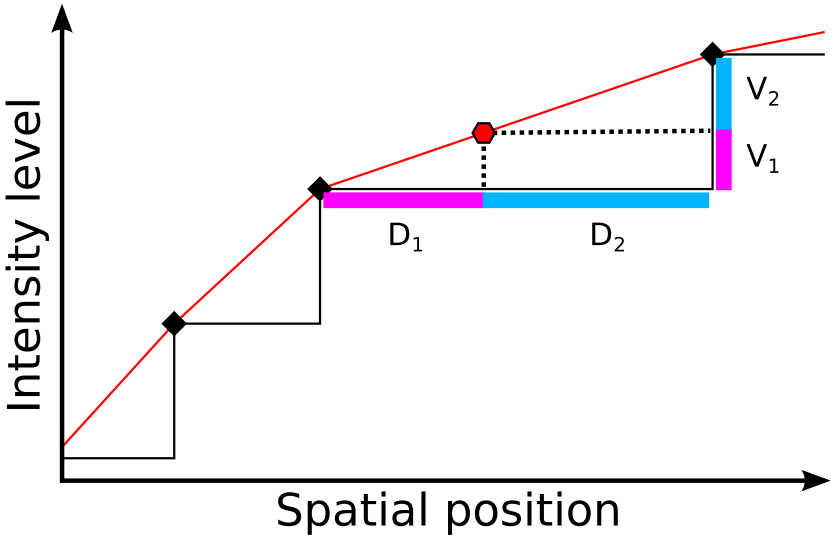



Fig. 4.7: Flooding-based linear interpolation

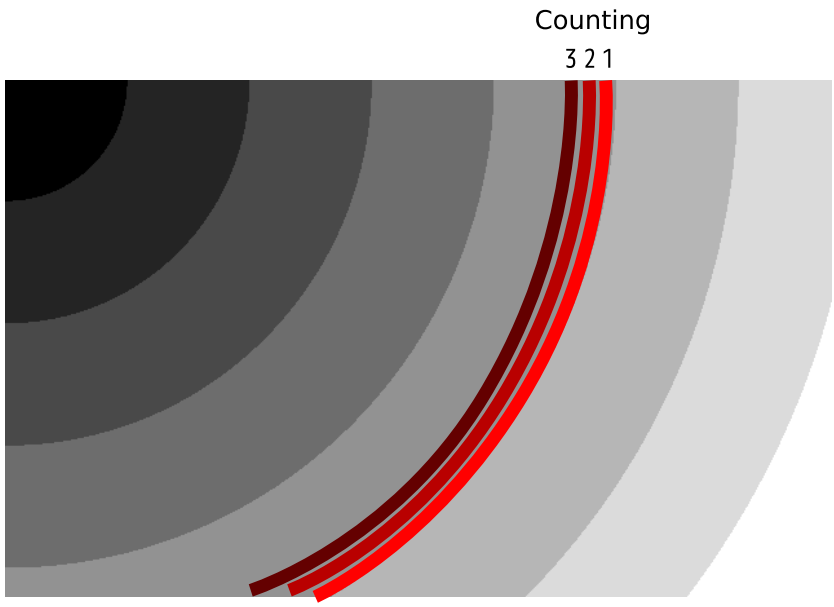



Fig. 4.8: Distance counting from contour edge in two-dimensional domain. The count is incremented $+1$ by one iteration cycle.

# Chapter 5

# Bit-depth Expansion using Poisson Equation

Interpolation in two-dimensional space (images) is more difficult problem than the interpolation in one-dimensional space because the shapes of contours in higher dimensional space has higher degrees of freedom. Features of our method in this chapter are follows:

- The two-dimensional interpolation problem is split to one-dimensional problems.
- Smooth gradations can be output.

## 5.1 Poisson Equation

The Poisson equation—one of the partial differential equations—is defined as:

$$\Delta f = g, \tag{5.1}$$

where $f$ and $g$ denote functions. When $f$ is a unknown function and $g$ is a known (fixed) function, $f$ can be obtained by solving Eq.(5.1) with boundary conditions. The definitions of the first-order differential operator $\nabla$ and the second-order differential operator $\Delta$ in two-dimensional space are as follows:

$$\nabla f = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}, \tag{5.2}$$

$$\Delta f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{5.3}$$

The $\Delta$ is also called "Laplacian operator". In the image processing, the differential operators are often used for gradient field of images[8, 9, 15, 18].

## 5.2 Formulation

If a function is differentiable, the function is looked as "smooth" function in mathematics. In the quantization, the bit-depth reduction (Eq.(4.3)) generates undifferentiable functions because of the floor function $\lfloor \cdot \rfloor$. To obtain smooth gradations from the quantized images, we focus on the differentiability of output images. Furthermore, we expect to simplify the interpolation problem by splitting an image (two-dimensional array) into one-dimensional arrays.

### 5.2.1 Multidimensional Interpolation using Poisson Equation

First, we split the image data into one-dimensional arrays. Now, we can get arrays as rows and columns depending on the splitting directions.

Let a set of the rows of image $I$ are written as $\boldsymbol{R}$, and a set of the columns are $\boldsymbol{C}$. The relationships between the split data and the image $I$ are:

$$\boldsymbol{R} = \{R_n \mid R_n(m) = I(m, n)\}, \tag{5.4}$$

$$\boldsymbol{C} = \{C_m \mid C_m(n) = I(m, n)\}, \tag{5.5}$$

where $I(x, y)$ is a pixel value of the image $I$ at position $(x, y)$, $R_y$ is a $y$-th row of the image, and $C_x$ is a $x$-th column of the image. If domain of the image is $1 \leq x \leq W$ and $1 \leq y \leq H$, the set $\boldsymbol{R}$ consists of $H$ functions, and the set $\boldsymbol{C}$ consists of $W$ function.

If the image function $I$ is a differentiable function, following equations are derived from the definitions of the $R_y$ and $C_x$;

$$\frac{\partial I(x, y)}{\partial x} = \frac{\mathrm{d}R_y(x)}{\mathrm{d}x}, \tag{5.6}$$

$$\frac{\partial I(x, y)}{\partial y} = \frac{\mathrm{d}C_x(y)}{\mathrm{d}y}. \tag{5.7}$$

The $R_y$ and $C_x$ are functions of two variables, and the differentials are defined for the variables $x$ and $y$ respectively. The total derivative $\mathrm{d}$, therefore, is used instead of the partial derivative $\partial$ in this paper.

Applying the second order differential operator Eq.(5.3) to Eq.(5.6) and Eq.(5.7), we obtain

$$\begin{aligned} \Delta I(x, y) &= \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \\ &= \frac{\mathrm{d}^2 R_y(x)}{\mathrm{d}x^2} + \frac{\mathrm{d}^2 C_x(y)}{\mathrm{d}y^2}. \end{aligned} \tag{5.8}$$

The Eq.(5.8) can be seen as the Poisson equation form shown in Eq.(5.1). Thus, if the right hand side of Eq.(5.8) is given, we can obtain the unknown image function $I$ solving the Poisson equation.

The data of right hand side of 5.8 are row-wise or column-wise independent (Fig. 5.1). Since the each element in $\boldsymbol{R}$ and $\boldsymbol{C}$ is one-dimensional array, the interpolation problem in two-dimensional space is split into multiple one-dimensional interpolation problems.

### 5.2.2 Weighted Linear Interpolation

5.2.1 shows that we can only design one-dimensional interpolation method. In this section, the one-dimensional interpolation used for our method are explained. Based on the simple piecewise linear approximation, the weighted linear interpolation is designed purposing the additional feature; texture preserving.

From Eq.(4.6), we must estimate the omitted fractions of $I$. Let $\tilde{I}(x)$ is the interpolated data of $I(x)$. The value of the $\tilde{I}(x)$ is limited in range:

$$I(x) \leq \tilde{I}(x) \leq I(x) + 1, \tag{5.9}$$

where $x$ denotes a position in the one-dimensional space.

#### Fixed Points

We define the fixed points as the invariant points before and after interpolation. A set of the fixed points $\boldsymbol{F}$ is defined as follows:

$$\boldsymbol{F} = \boldsymbol{F}_{\mathrm{up}} \cup \boldsymbol{F}_{\mathrm{down}} \cup \boldsymbol{F}_{\mathrm{edge}}, \tag{5.10}$$

$$\boldsymbol{F}_{\mathrm{up}} = \{m \mid 0 < I(m) - I(m-1) < t\}, \tag{5.11}$$

$$\boldsymbol{F}_{\mathrm{down}} = \{m \mid 0 < I(m) - I(m+1) < t\}, \tag{5.12}$$

$$\boldsymbol{F}_{\mathrm{edge}} = \{m \mid t \leq |I(m) - I(m-1)|\} \cup \{m \mid t \leq |I(m) - I(m+1)|\}, \tag{5.13}$$

where sets $\boldsymbol{F}_{\mathrm{up}}$ and $\boldsymbol{F}_{\mathrm{down}}$ are fixed points in the false contour region, and $\boldsymbol{F}_{\mathrm{edge}}$ denotes fixed points of edges. The edge is defined by given threshold parameter $t$ (normally $t = 2$).

#### Piecewise Linear Interpolation

The fixed points classify the domain of $I(x)$ into multiple intervals. Then the piecewise linear interpolation consists of the linear interpolation for each inter-

val.

The interpolated interval is defined as a closed-interval $(x_k, x_{k+1})$, where $k(1 \leq k)$ is index number of the fixed points in ascending order. Then, the relationship between $x_k$ and $x_{k+1}$ is:

$$x_{k+1} = \min\{m \in \boldsymbol{F} \mid x_k < m\}. \tag{5.14}$$

A function $J(x)$ interpolating the interval between two fixed points $I(x_k)$ and $I(x_{k+1})$ is written as:

$$J(x) = \frac{I(x_{k+1}) - I(x_k)}{L}(x - x_k) + I(x_k), \quad (x \in (x_k, x_{k+1})), \tag{5.15}$$

where $L$ is $L = |x_{k+1} - x_k|$. Eq.(5.15) interpolates the step functions in the false contour region as linear functions.

### Interpolation in Local Minima/Maxima

If Eq.(5.15) satisfies $J(x) = I(x)$ in an entire interval, the interval is defined as a local maximum region of the function. Similarly, a local minimum satisfies $J(x) = I(x) + 1$. Let the local maxima and minima are written as $\boldsymbol{\Omega}_\wedge$ and $\boldsymbol{\Omega}_\vee$ respectively. They are defined as follows:

$$\boldsymbol{\Omega}_\wedge = \{m \notin \boldsymbol{F} \mid J(m) = I(m)\}, \tag{5.16}$$

$$\boldsymbol{\Omega}_\vee = \{m \notin \boldsymbol{F} \mid J(m) = I(m) + 1\}. \tag{5.17}$$

To improve accuracy of interpolations in local maxima and minima, we define the functions interpolating that regions. The interpolating function $J_\wedge(x)$ for local maxima $\boldsymbol{\Omega}_\wedge$ is defined as a upward convex function:

$$J_\wedge(x) = \begin{cases} \frac{G(L,\sigma)}{L}(x - x_k) + I(x_k), & \left(x \in (x_k, \frac{x_{k+1}+x_k}{2}]\right) \\ -\frac{G(L,\sigma)}{L}(x - x_{k+1}) + I(x_{k+1}). & \left(x \in (\frac{x_{k+1}+x_k}{2}, x_{k+1})\right) \end{cases} \tag{5.18}$$

Similarly, the interpolating function for local minima $\boldsymbol{\Omega}_\vee$ is defined as a downward convex function:

$$J_\vee(x) = \begin{cases} -\frac{G(L,\sigma)}{L}(x - x_k) + I(x_k), & \left(x \in (x_k, \frac{x_{k+1}+x_k}{2}]\right) \\ \frac{G(L,\sigma)}{L}(x - x_{k+1}) + I(x_{k+1}). & \left(x \in (\frac{x_{k+1}+x_k}{2}, x_{k+1})\right) \end{cases} \tag{5.19}$$

The $G(d, \sigma)$ is weighing function to adjust the heights of the peaks in the interpolated data. The weights are varied depending on the length of the interval $(x_k, x_{k+1})$:

$$G(d, \sigma) = 1 - \exp\left(-\frac{d^2}{2\sigma^2}\right), \tag{5.20}$$

where value range of the function is $0 \leq G(d, \sigma) \leq 1$.

The weighted linear interpolating functions are rewritten from Eq.(5.15), Eq.(5.18), and Eq.(5.19).

$$\tilde{I}(x) = \begin{cases} I(x), & (x \in \boldsymbol{F}) \\ J_\wedge(x), & (x \in \boldsymbol{\Omega}_\wedge) \\ J_\vee(x), & (x \in \boldsymbol{\Omega}_\vee) \\ J(x). & (\text{otherwise}) \end{cases} \tag{5.21}$$

We can interpolate contour with preserving textures by setting suitable $\sigma$. The interpolations in false contour region, local maxima, and local minima are shown in Fig. 5.2.

### 5.2.3 Uniting One-Dimensional Interpolated Functions

This section shows method to merge interpolations in 5.2.2 using 5.2.1

First, the two types of split data $\boldsymbol{R}$ and $\boldsymbol{C}$ are obtained from input image. And a set of the fixed points is provided:

$$\boldsymbol{\Omega}_{\text{fixed}} = \{(m, n) \mid \ m \in \boldsymbol{F}_{R_n} \vee n \in \boldsymbol{F}_{C_m}\}, \tag{5.22}$$

where $\boldsymbol{F}$ is defined in Eq.(5.10), and $\boldsymbol{F}_{R_n}$ and $\boldsymbol{F}_{C_m}$ denote the sets of fixed points of $R_n$ and $C_m$ respectively.

Second, the weighted linear interpolation shown in 5.2.2 is applied to $R_1, \cdots, R_H$ and $C_1, \cdots, C_W$.

The following simultaneous equations are obtained from Eq.(5.8) and Eq.(5.21):

$$\tilde{I}(x,y) = \begin{cases} I(x,y), & ((x,y) \in \boldsymbol{\Omega}_{\text{fixed}}) \\ \tilde{R}_1(x), & (y < 1) \\ \tilde{R}_H(x), & (H < y) \\ \tilde{C}_1(y), & (x < 1) \\ \tilde{C}_W(y), & (W < x) \end{cases} \tag{5.23}$$

$$\Delta\tilde{I}(x,y) = \frac{\mathrm{d}^2\tilde{R}_y(x)}{\mathrm{d}x^2} + \frac{\mathrm{d}^2\tilde{C}_x(y)}{\mathrm{d}y^2}, \qquad \text{(otherwise)} \tag{5.24}$$

where $\tilde{R}_n$ and $\tilde{C}_m$ are interpolations for $R_n$ and $C_m$ by Eq.(5.21), respectively. Eq.(5.23) denotes fixed points and boundary conditions. Then Eq.(5.24) is solved for other unfixed regions.

## 5.3 Experimental Results

Quality evaluations and time measurements are in this section.

The testing set are shown in Fig. 5.3. The test images include artificial images without noise (Fig. 5.3(a) and Fig. 5.3(b)) and natural images with noise (Fig. 5.3(c), Fig. 5.3(d), and Fig. 5.3(e)).

### 5.3.1 Relationship between Parameters and Image Quality

We investigate the relationship between the image quality and the parameter $\sigma$ explained in 5.2.2. Fig. 5.4 shows outputs of our method with different $\sigma$. When $\sigma$ is small, the noise in contour region remains (Fig. 5.4(c)). When $\sigma$ is suitable value, the noise is reduced (Fig. 5.4(d)). On the other hands, the larger $\sigma$, the smaller PSNR value shown in Fig. 5.4(b). Thus, note that the higher PSNR value does not mean the higher quality result.

In this section, we commonly use $\sigma = 12$.

### 5.3.2 Quality Evaluation

In this section, the proposed method is compared with the conventional methods based on the piecewise linear approximation[4, 20].

The comparison of PSNR values is shown in Table 5.1. Our method has higher PSNR values for artificial images. For the natural images, however, PSNR values of our method is lower. This tells us the existence of the noise influences PSNR values of our method.

The smoothness of gradation regions are compared in spatial-frequency domain. Fig. 5.5(a) shows the spatial-frequency spectrum for each method. Fig. 5.5(b) shows direction dependence of the power spectrums. Our method has no dependence for any directions. In the conventional methods, however, there are strong concentration to the specified directions. This results shows that the gradation smoothness of conventional methods are not complete.

Fig. 5.6 shows actual outputs. The conventional methods have two (Fig. 5.6(e)) or four (Fig. 5.6(h)) directional patterns in the output image. In contrast, the output of our method achieves the smooth gradations (Fig. 5.6(k)).

### 5.3.3 Processing Time

Table 5.2 shows the processing time of each method. In maximum case, the processing time of our method is 70 times faster than other methods. Note that the processing time of our method is depending on the solver algorithm and input image.

We solved the Eq.(5.24) using successive over relaxation method. All of the software programs are implemented in C++ language. And the programs were run on the Pentium Dual-Core $2.70$-GHz CPU without downsampling and multi-threading.

Table 5.1: PSNR comparison with conventional methods.

| Image | Type | PSNR (dB) | | |
|---|---|---|---|---|
| | | Cheng [4] | Wan [20] | Proposed |
| Fig. 5.3(a) | Artificial | 47.6 | 49.2 | **51.5** |
| Fig. 5.3(b) | Artificial | 39.7 | 44.6 | **48.5** |
| Fig. 5.3(c) | Natural | 36.2 | **36.8** | 34.2 |
| Fig. 5.3(d) | Natural | 36.9 | **37.2** | 35.2 |
| Fig. 5.3(e) | Natural | 37.1 | **38.0** | 36.2 |

Gray level: input $= 16$, output $= 256$

Parameter $\sigma$ of the proposed method $= 12$.

Table 5.2: Time comparison with conventional methods

| Image | Size | Time (sec) | | |
|---|---|---|---|---|
| | | Cheng [4] | Wan [20] | Proposed |
| Fig. 5.3(a) | $640 \times 400$ | 3.50 | 22.3 | **1.11** |
| Fig. 5.3(b) | $640 \times 400$ | 1.33 | 18.2 | **1.09** |
| Fig. 5.3(c) | $1600 \times 1200$ | 58.1 | 435 | **8.37** |
| Fig. 5.3(d) | $2304 \times 1728$ | 173 | 1316 | **17.9** |
| Fig. 5.3(e) | $1728 \times 2304$ | 127 | 1163 | **19.3** |

Gray level: input $= 16$, output $= 256$

Parameter $\sigma$ of the proposed method $= 12$.

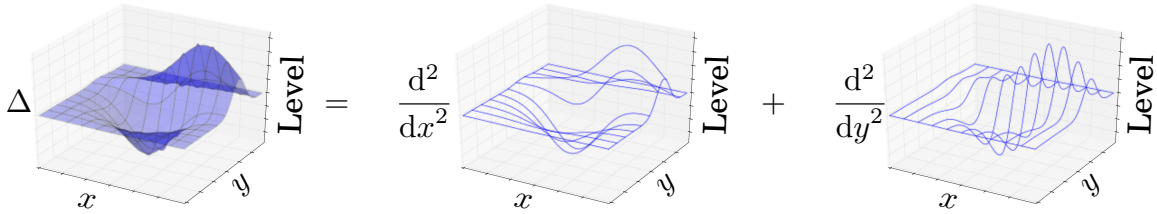



Fig. 5.1: Function splitting using the Poisson equation. From left to right: smooth function $I(x, y)$, split rows $R_y(x)$, and split columns $C_x(y)$.

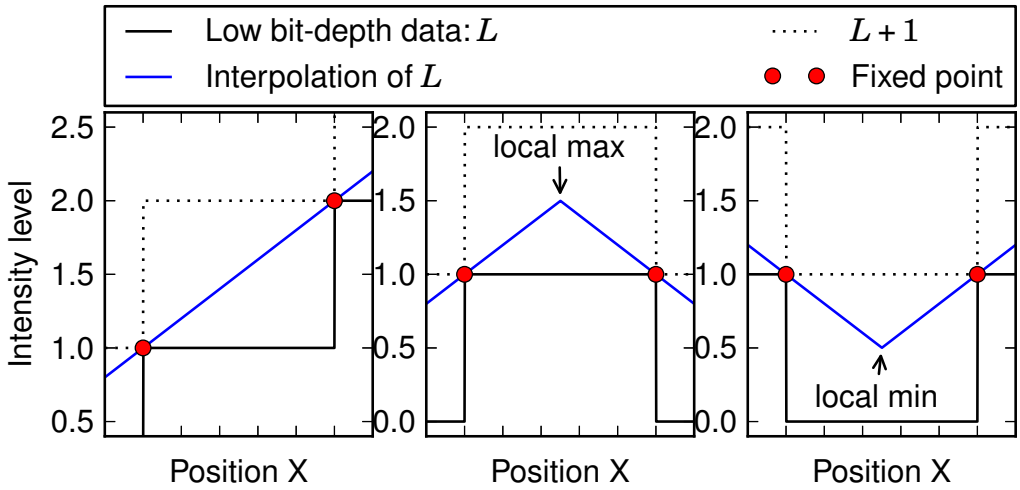
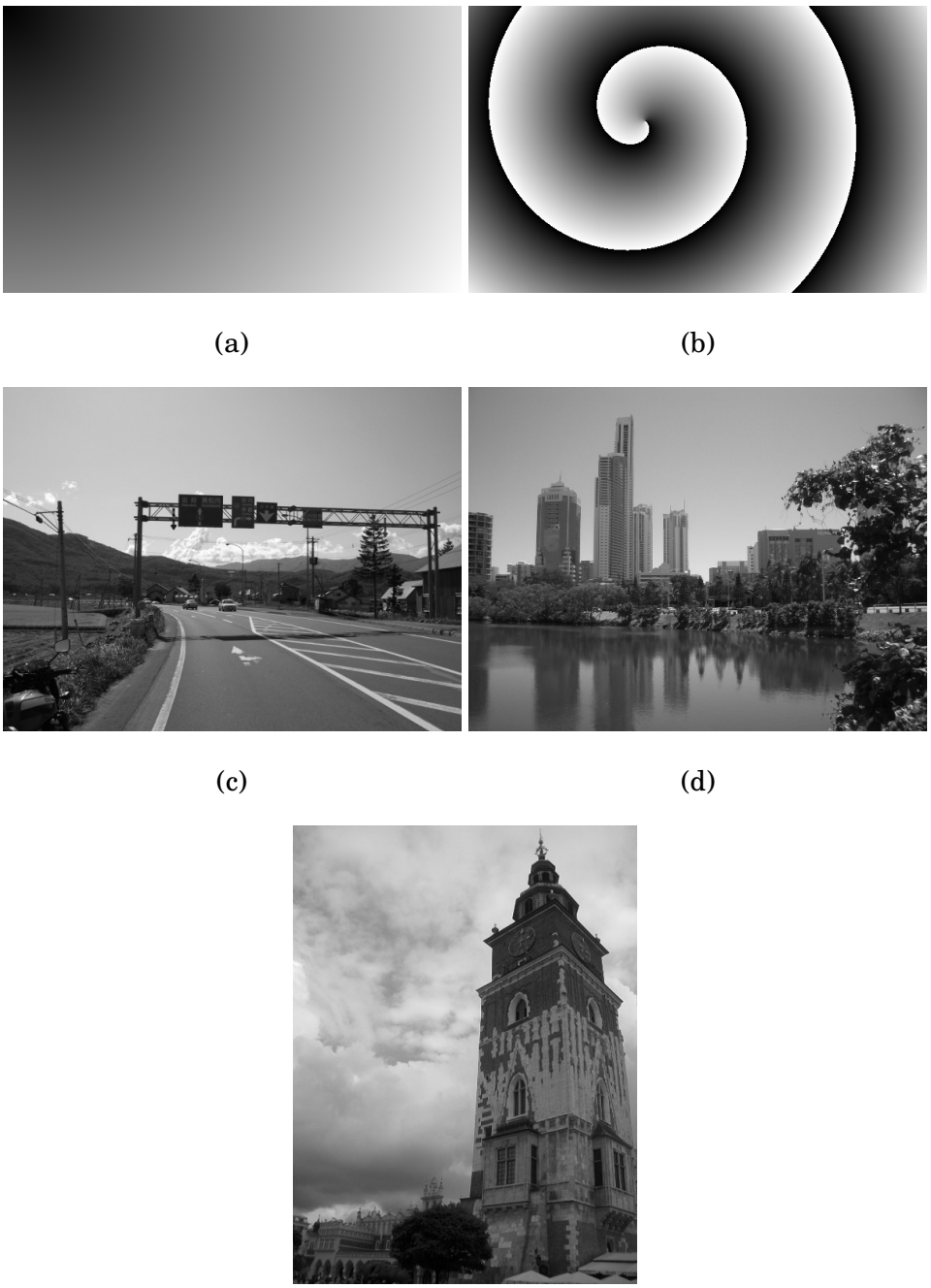



Fig. 5.2: Weighted linear interpolation. From left to right: interpolation in contour region $J(x)$, local maximum region $J_\wedge(x)$, and local minimum region $J_\vee(x)$.



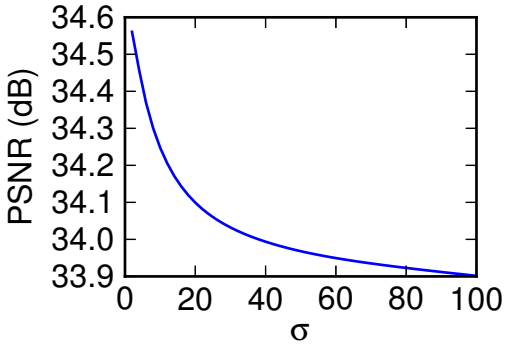
(a) (b)

(c) (d)

(e)

Fig. 5.3: Test images.

(a) Input: gray level = 16

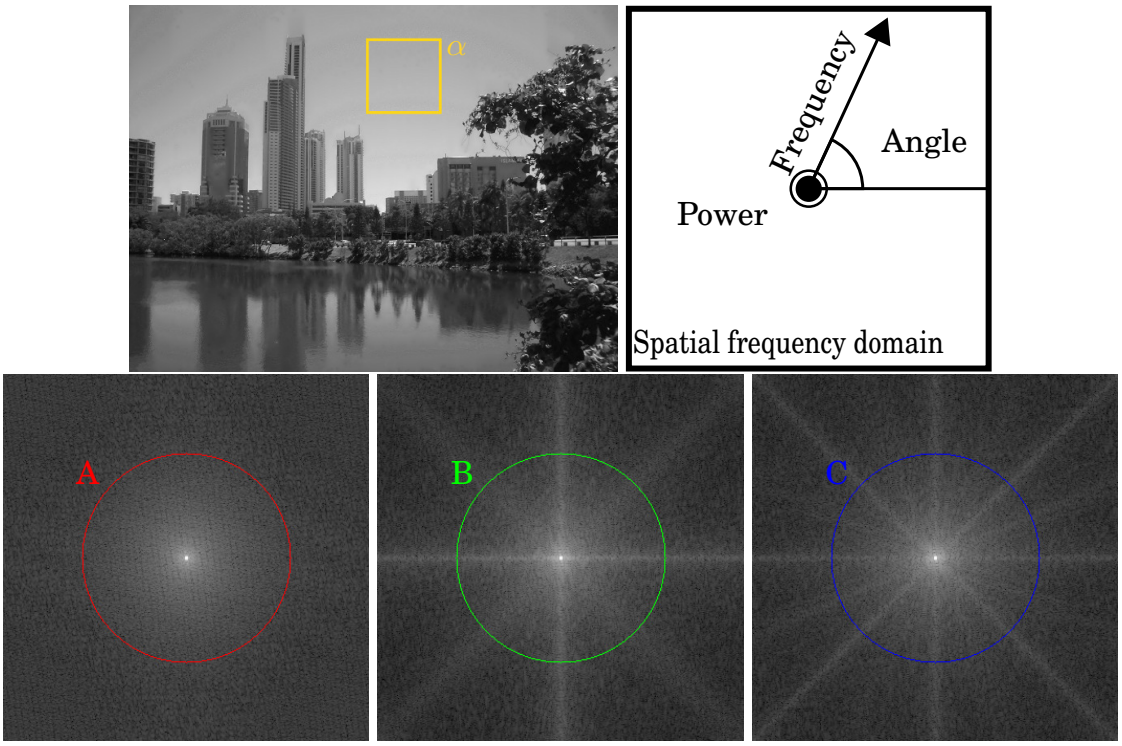



(b) PSNR vs. $\sigma$

(c) Output and edges: $\sigma = 2$

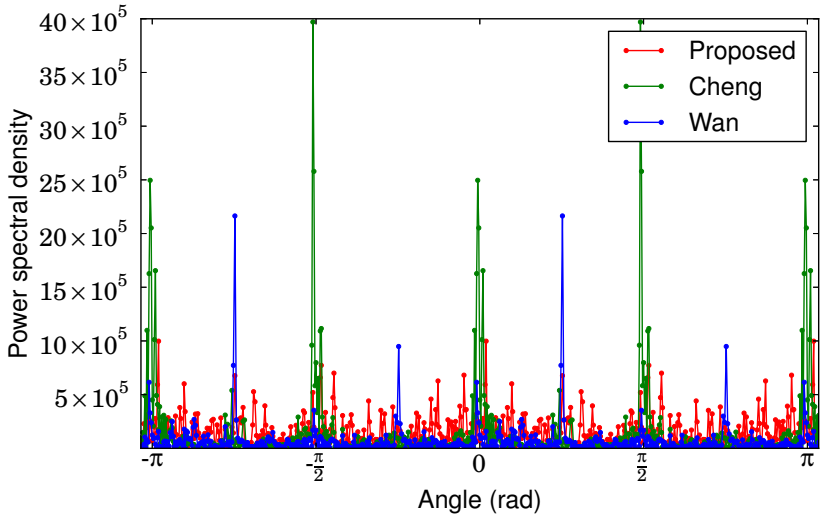
(d) Output and edges: $\sigma = 12$

Fig. 5.4: Results with different $\sigma$.



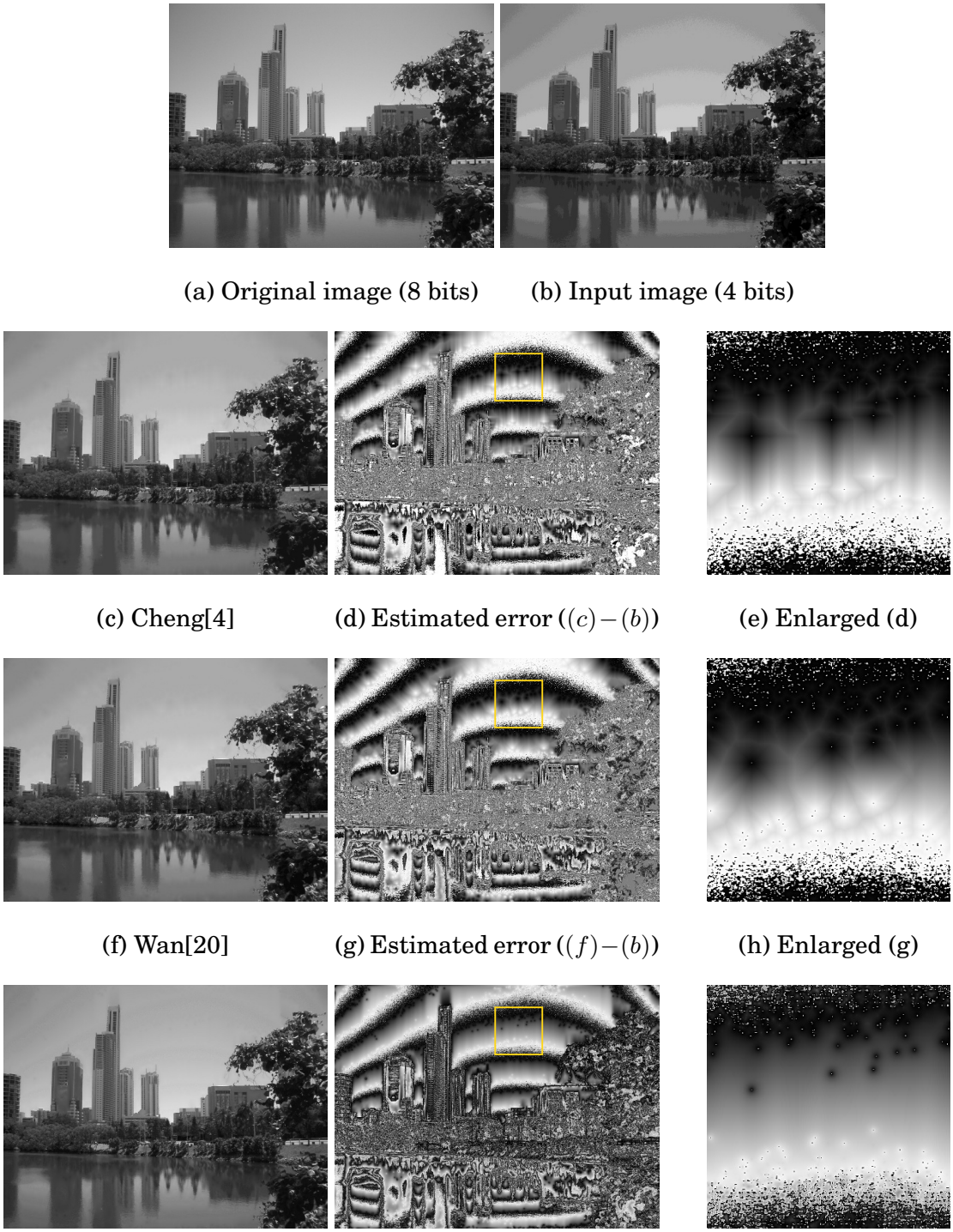

Proposed Cheng[4] Wan[20]

(a) Spatial-frequency spectrums in $\alpha$




(b) Power spectrums on the circumferences A, B, and C

Fig. 5.5: Spatial-frequency characteristics in gradation region.



(a) Original image (8 bits) (b) Input image (4 bits)

(c) Cheng[4] (d) Estimated error ($(c)-(b)$) (e) Enlarged (d)

(f) Wan[20] (g) Estimated error ($(f)-(b)$) (h) Enlarged (g)

(i) Proposed method (j) Estimated error ($(i)-(b)$) (k) Enlarged (j)

Fig. 5.6: Reconstruction results of false contours.

# Chapter 6

# Noisy False Contour in Natural Image

Some conventional methods work well on images with no noise (e.g., computer graphics). Unfortunately, however, none of them can work on natural images.

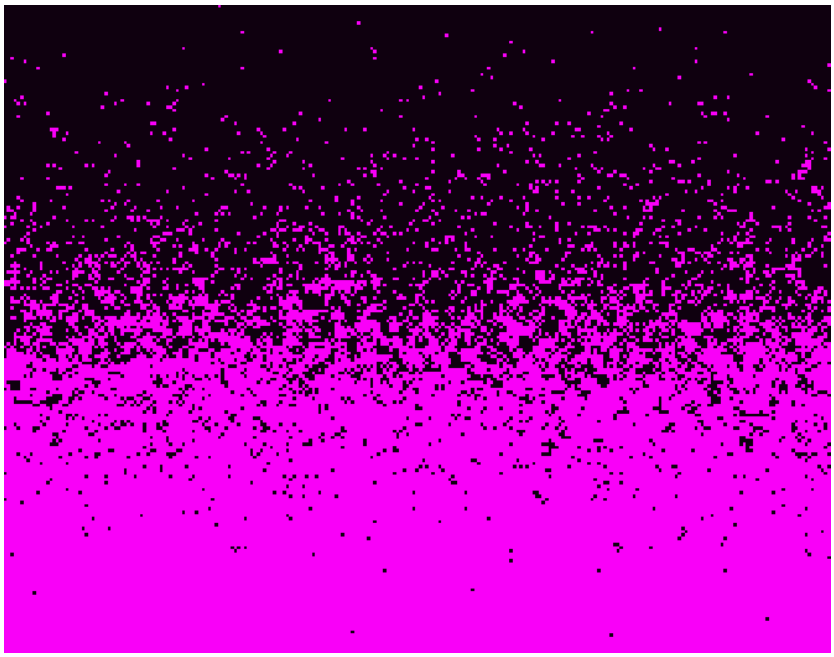
When false contours appear in natural images, the contour lines are not clear. High frequency noise scatters the boundaries of contours. The scattered contour artifacts are shown in Fig. 6.1.

In the BDE for the natural images, we must remove the noise and reconstruct smooth gradations from that artifacts.

## 6.1 Inverse Quantization Problem from Noisy Contour Artifact

Fig. 6.2 shows noisy input signal and its quantized data. The noisy contour artifacts have two different regions, flat and noisy region. Their quantization bins are hardly fluctuated (Fig. 6.2(b)). If there is no noise, the quantization

(a) Image having false contours



(b) Zoom-in. The two regions are painted in different colors for visibility.

Fig. 6.1: Noisy false contour artifacts in natural image

process generates clear step function as shown in Fig. 4.1. In noisy case, however, their contour edges are not clear because of high frequency noise.

Now, we want to estimate the original signal from quantized signal. When the conventional methods applied to the noisy contours, the original signal is estimated like Fig. 6.3(a) because the estimated signal can not take values out of the quantization bins by Eq.(4.2). This is not what we really want. If we have an ideal estimator, it should generate the estimated signal like Fig. 6.3(b). Our real goal is developing a method that can take values out of the quantization bins.

To understand noisy contour problem, four types of signal are drawn in Fig. 6.4. The upper two signals are noisy, and the lower two are not noisy. The right two are continuous signals, and the left hand side denotes discontinuous. All of the methods discussed in Chapter 4 and Chapter 5 are free of noise problems (estimating the lower right from the lower left in Fig. 6.4). Our required task is solving a problem "estimating the lower right from the upper left in Fig. 6.4".

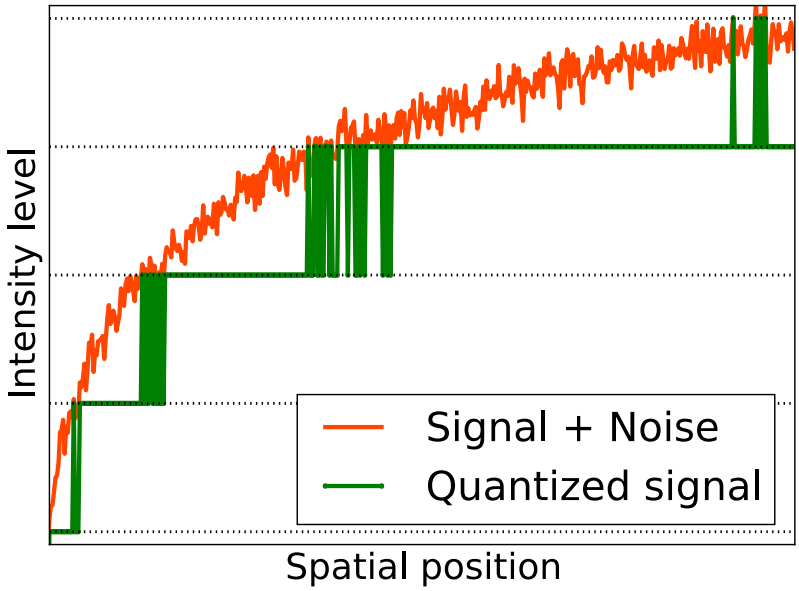
Here are our early investigations to attack the problem.

## 6.2 Noise Removal using Median Filter

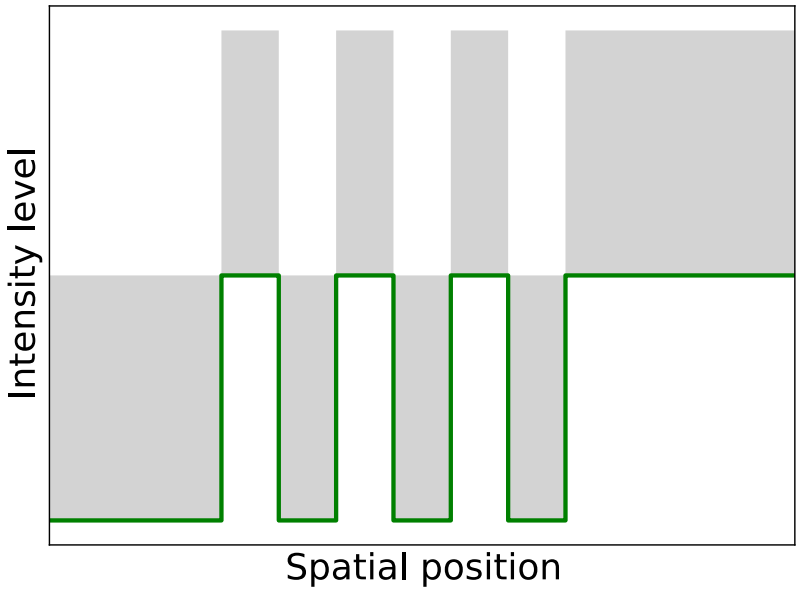
Our first idea has two steps.

- Removing noise from noisy contours (from upper left to lower left in Fig. 6.4)
- Applying classic BDE methods for the denoised signal (from lower left to lower right)

The conventional methods can be used as the second step. Therefore, we have only to attack the denoising problem. An experimental result of denoising by the median filter is shown in Fig. 6.5. We can see the noise is decreased. Unfortu-

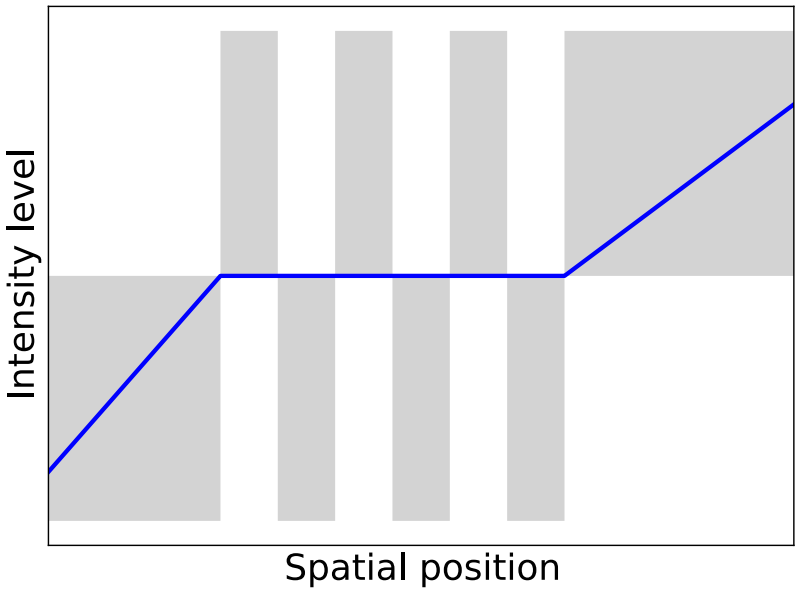



(a) Noisy signal and quantized signal

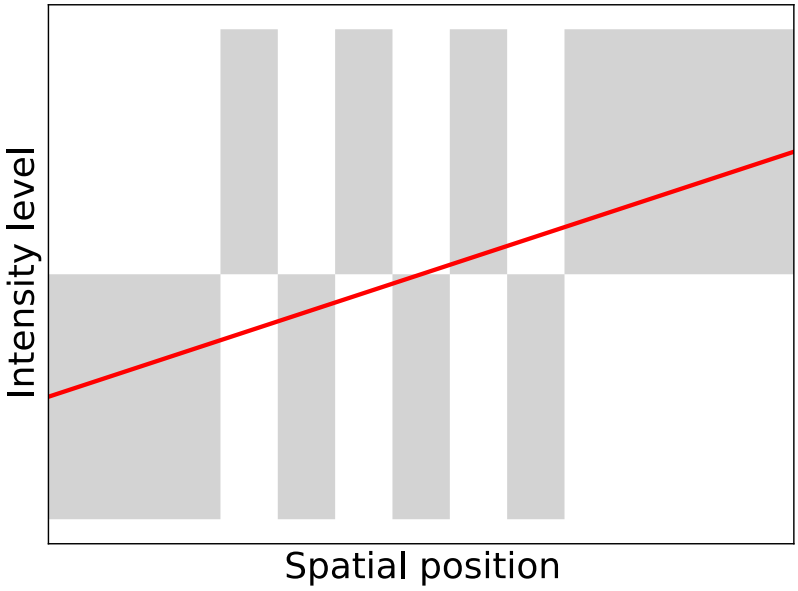



(b) Zoom-in. The quantized signal (green) and its quantization bins (gray)

Fig. 6.2: Quantization of noisy signal




(a) Estimation based on the model Eq.(4.2)




(b) Ideal estimation

Fig. 6.3: Estimation of original signal from noisy contour artifact

nately, however, some outliers remain (See Fig. 6.5(b)).

## 6.3 Low-pass Filter

Fig. 6.6 shows the simulation result of decontouring using a low-pass filter. When the flat region is in the input signal, the contour effect remains in the result. The low–pass filtering can remove high–frequency effects in quantized signal, but can not remove flat region because flat signal is completely low–frequency signal. The filter can only convert the step function (quantized signal in Fig. 6.6) to the smoothed step function (result signal in Fig. 6.6). Actually, we want to obtain a smooth slope–like function from a step function. Therefore, the low–pass based methods fails to reconstruct smooth gradation from large contours typically appearing in the sky, sea or skin.

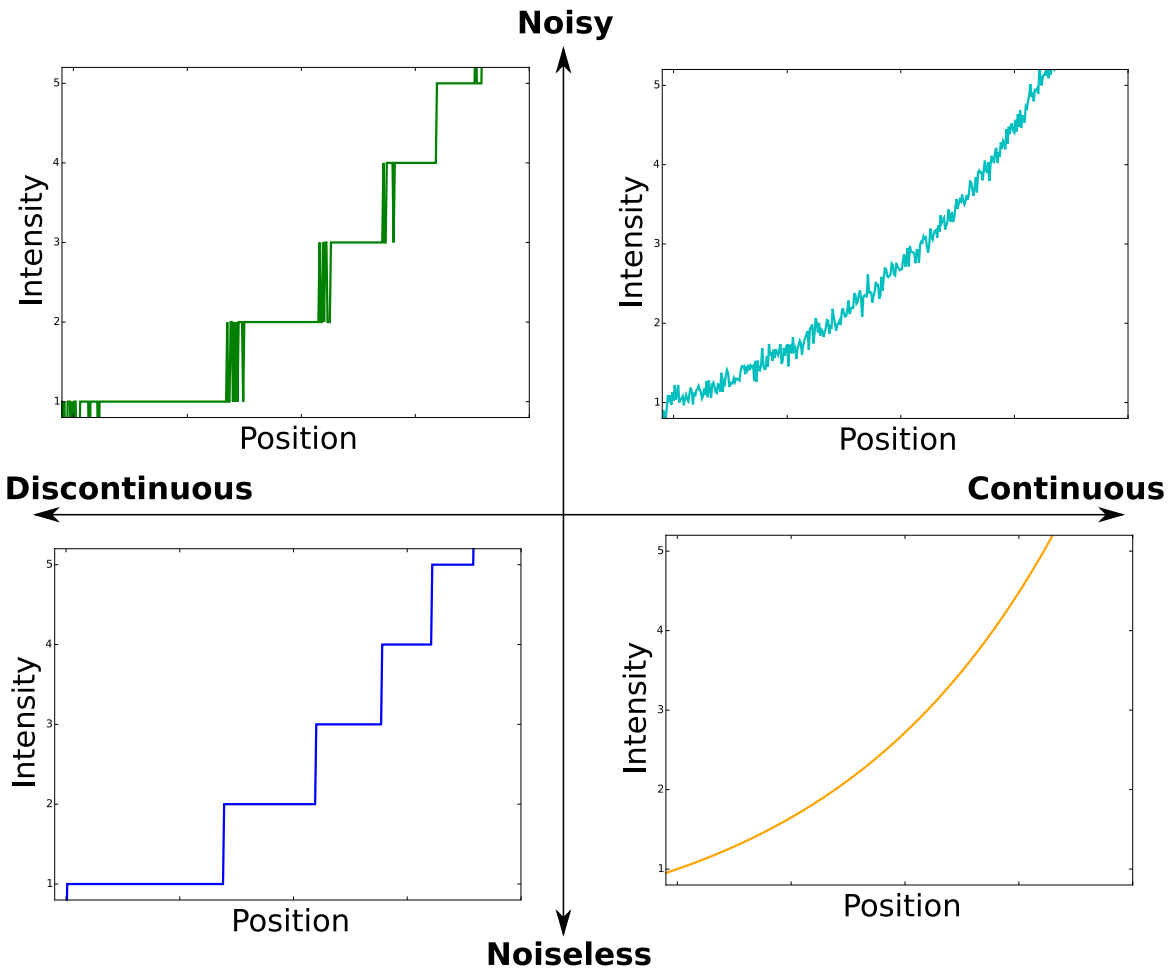



Fig. 6.4: Signal types focusing on continuity and noise

(a) Original LBD image

(b) Median filtered

Fig. 6.5: Median filter for nosy contours

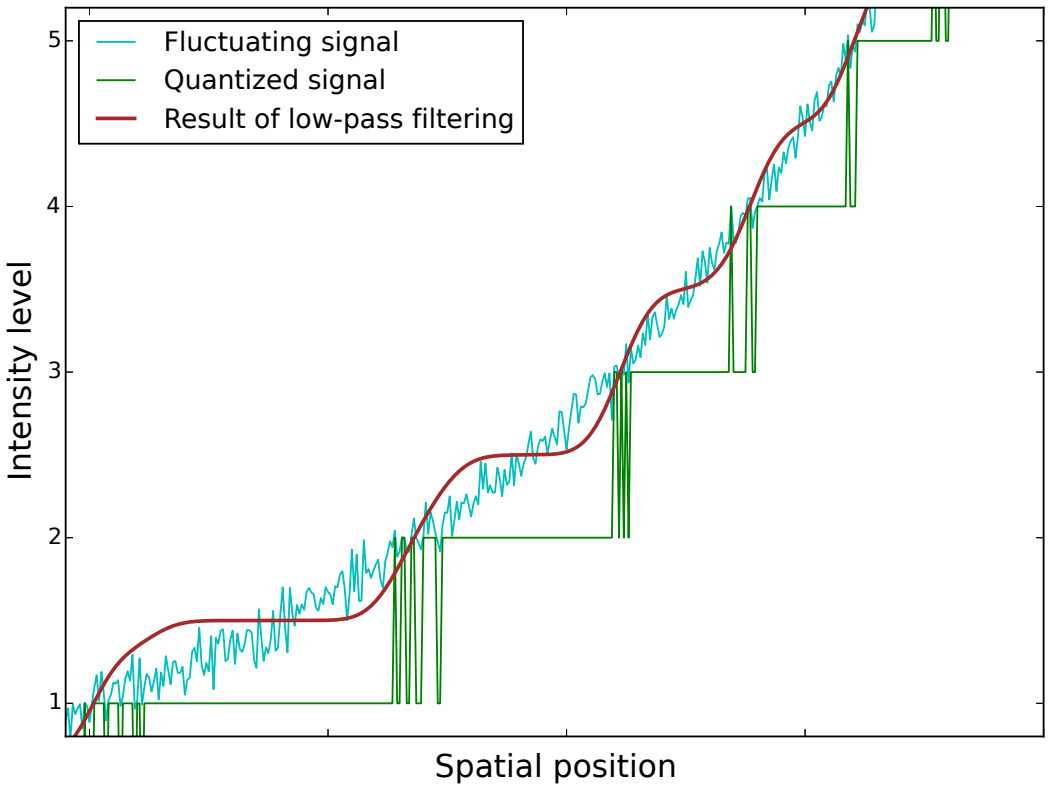



Fig. 6.6: Simulation of low-pass filtering

# Chapter 7

# Bit-Depth Expansion for Natural Images

In natural images (photographs), there are noises in the false contour artifacts. The conventional BDE methods and our method in Chapter 5 are designed only for images having no noise. Their outputs for natural images are not good. Therefore we develop a BDE method to manage the natural images in this chapter.

## 7.1 Quantization Model for Noisy Signal

The implicit premise in the conventional researches is that no noise is added to the original signal. The quantization model without noise (Eq.(4.1)) is discussed in section 4.1.

We use a more accurate formulation because natural images are inseparable from noise:

$$\mathbf{y} = Q(\mathbf{x} + \xi), \tag{7.1}$$

where $\xi \in \mathbb{R}^n$ denotes the noise. Eq.(7.1) only states that the fluctuating signal

is in the quantization bins:

$$x_i^* + \xi_i \in [y_i, y_i + 1)\,. \tag{7.2}$$

It does not decide the value range of estimated signal $\mathbf{x}^*$. The removing constraint of $\mathbf{x}^*$ is an important point of this paper.

Fig. 7.1 shows example results of BDE using two different models, Eq.(4.1) and Eq.(7.1). The quantized signal (green) given from the fluctuating signal (cyan) is input for a BDE method, and then the dequantized signal (blue or red) is obtained. We can see that the quantized signal consists of some flat regions and noisy ones. The upper of Fig. 7.1(a) is generated by one of flooding–based method[4], and the lower by our method described later in this paper. The result of Eq.(4.1) fails to reconstruct smooth gradations because the dequantized signal must be in quantization bins by reason of Eq.(4.2). The allowed range for dequantized signal, the quantization bins denoted as gray in Fig. 7.1(b), is raised and collapsed incessantly at noisy area. Therefore, to achieve a continuous function, the dequantized signal is compelled to be flat by taking lower bounds of the raised bins and upper of the collapsed bins.

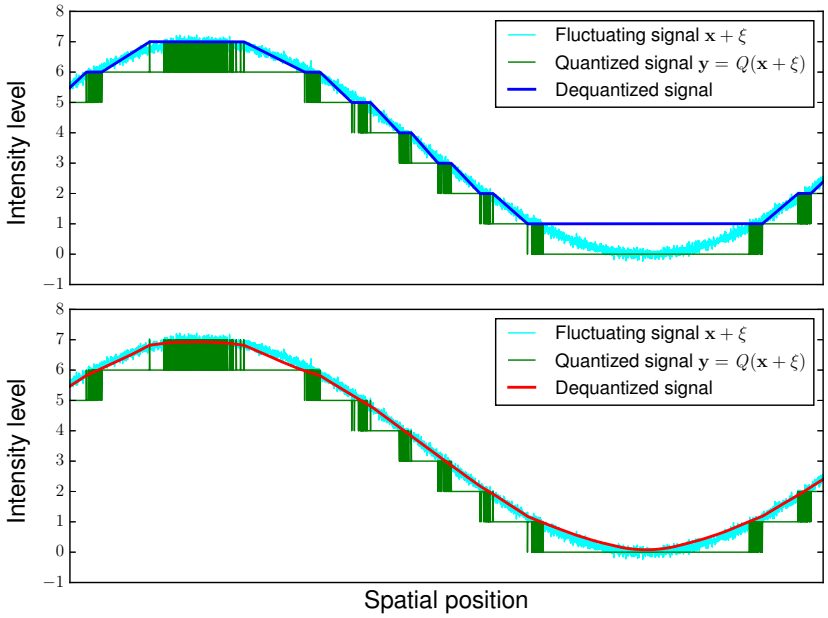
Because the result of Eq.(7.1) has smooth gradations, it is better. We explain the BDE method using Eq.(7.1) in this paper.

## 7.2 Formulation

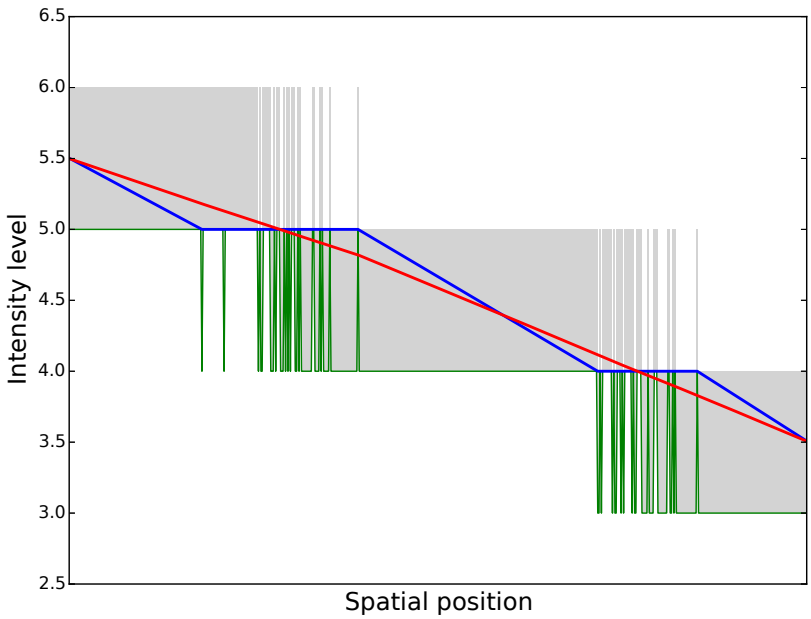
Finding the most probable $\mathbf{x}$ from an observed $\mathbf{y}$ is an inverse problem of Eq.(7.1). We use the maximum a posteriori (MAP) estimation:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} L(\mathbf{y}|\mathbf{x})P(\mathbf{x}), \tag{7.3}$$

where $L(\mathbf{y}|\mathbf{x})$ is a likelihood function of $\mathbf{x}$ given $\mathbf{y}$ and $P(\mathbf{x})$ is a prior probability of $\mathbf{x}$. In this section, we formulate the MAP estimation based on the model Eq.(7.1).




(a) Dequantization results




(b) Zoom-in. The blue result can not stray from quantization bins (gray).

Fig. 7.1: Example results of BDE methods based on different models Eq.(4.1) (blue signal) and Eq.(7.1) (red signal)

### 7.2.1 Prior Probability

A prior probability $P(\mathbf{x})$ is formulated as a function to measure smoothness of $\mathbf{x}$.

First, we define an weight parameter $w_{i,j}$ between pixels $y_i$ and $y_j$ in the quantized image $\mathbf{y}$ as follows:

$$w_{i,j} = \begin{cases} 1 & (j \in \mathcal{N}_i \wedge |y_i - y_j| \leq \kappa)\,, \\ 0 & \text{otherwise}, \end{cases} \tag{7.4}$$

where $\mathcal{N}_i$ is a set of $4$ nearest neighbors of the $i$–th pixel, and $\kappa$ is a threshold parameter. In the image domain, $w_{i,j} = 1$ denotes that the pixels $y_i$ and $y_j$ are in the same region.

Using Eq.(7.4), we define a prior probability $P(\mathbf{x})$ as

$$W(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}(x_j - x_i)^2, \tag{7.5}$$

$$P(\mathbf{x}) = \exp\left(-\frac{W(\mathbf{x})}{2\sigma_s^2}\right), \tag{7.6}$$

where the function $W(\mathbf{x})$ means a total squared variation of the image $\mathbf{x}$ segmented by $w_{i,j}$, and $\sigma_s$ is a parameter to adjust the smoothness.

### 7.2.2 Likelihood

We guess the distribution of noise $\xi$ as a normal distribution in our method. Thus, the probability distribution of $\tilde{x}_i = x_i + \xi_i$ is generated from $x_i$ with added a stochastic variable $\xi_i$, $P(\tilde{x}_i|x_i)$, can be formulated as a common normal distribution function:

$$P(\tilde{x}_i|x_i) = \frac{1}{\sqrt{2\pi\sigma_g^2}}\exp\left(-\frac{(\tilde{x}_i - x_i)^2}{2\sigma_g^2}\right), \tag{7.7}$$

where $\sigma_g$ is the variance of the distribution of $\xi_i$. From the definition of the quantization bin (Eq.(7.2)), $y_i$ is generated when $\tilde{x}_i$ is in range $[y_i, y_i+1)$ (See Fig.

7.2). Therefore the probability of $y_i$ with given $x_i$, $P_l(y_i|x_i)$, can be calculated by accumulating $P(\tilde{x}_i|x_i)$ for all possible patterns of $\tilde{x}_i$ as follows:

$$\begin{aligned} P_l(y_i|x_i) &= \int_{y_i}^{y_i+1} P(\tilde{x}_i|x_i)d\tilde{x}_i \\ &= \frac{1}{2}\left(\mathrm{erf}\left(\frac{y_i - x_i + 1}{\sqrt{2\sigma_g^2}}\right) - \mathrm{erf}\left(\frac{y_i - x_i}{\sqrt{2\sigma_g^2}}\right)\right), \end{aligned} \tag{7.8}$$

where $\mathrm{erf}(\cdot)$ is the error function. If the value of $y_i$ is known (fixed), the function $P_l(y_i|x_i)$ is considered as the likelihood function of $x_i$. Fig. 7.3 shows the function shapes of Eq.(7.8) with fixed $y_i$.

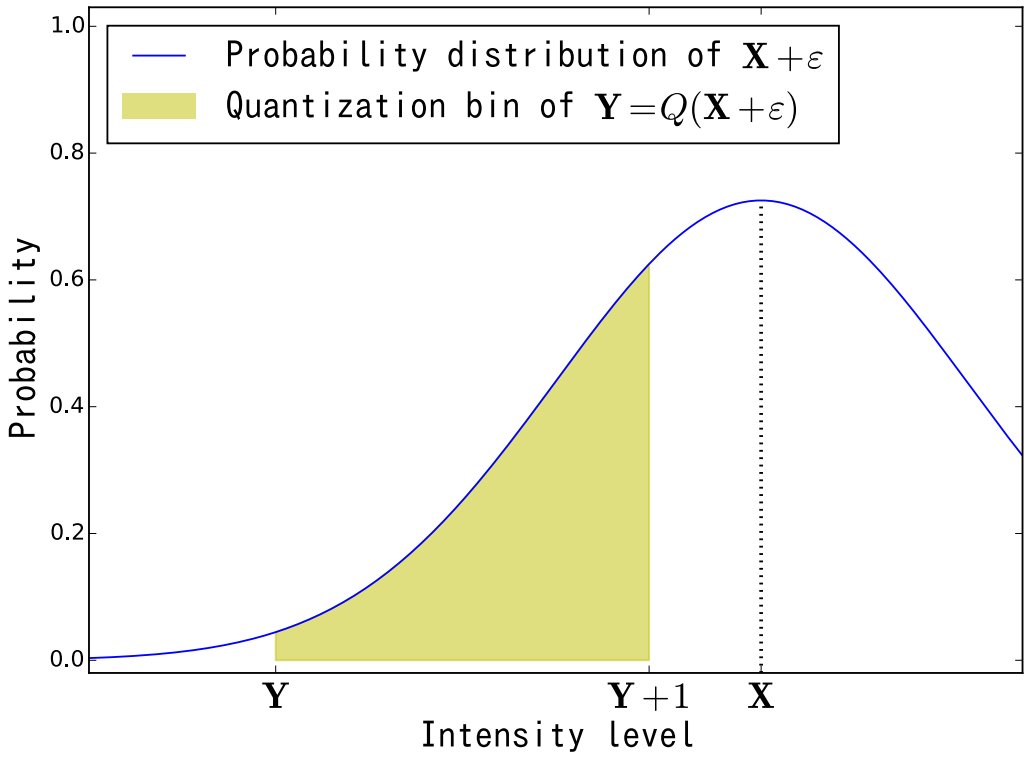



Fig. 7.2: Signal distribution and quantization bin

Here is another formulation for a special case. When $y_i$ is a maximum ($y_{\max}$) or minimum ($y_{\min}$) value of its possible range, we can empirically estimate that the signal may be saturated. At the boundary of the signal range, we define the

probability of $y_i$ as

$$P_b(y_i|x_i) = \exp\left(-\frac{b_i(x_i)}{2\sigma_b^2}\right), \tag{7.9}$$

$$b_i(x_i) = \begin{cases} (y_{\min} - x_i)^2 & (y_i = y_{\min}), \\ (y_{\max} - x_i + 1)^2 & (y_i = y_{\max}), \\ 0 & \text{otherwise.} \end{cases} \tag{7.10}$$

The likelihood that $L(\mathbf{y}|\mathbf{x})$ is calculated by using Eq.(7.8) and Eq.(7.9) is as follows:

$$L(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} P_b(y_i|x_i) P_l(y_i|x_i). \tag{7.11}$$

The probability $P_b(y_i|x_i)$ works as a bias to shift the peak of probability $P_l(y_i|x_i)$.

### 7.2.3 Optimization

We define the following optimization problem by combining Eq.(7.3), Eq.(7.6), and Eq.(7.11):

$$\mathcal{F} = \frac{1}{2\sigma_s^2} W(\mathbf{x}) + \frac{1}{2\sigma_b^2} \sum_{i=1}^{n} b_i(x_i) - \sum_{i=1}^{n} \log P_l(y_i|x_i), \tag{7.12}$$

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \mathcal{F}. \tag{7.13}$$

The partial derivatives of objective function $\mathcal{F}$ are as follows:

$$\frac{\partial \mathcal{F}}{\partial x_i} = \frac{1}{2\sigma_s^2} W'_{x_i}(\mathbf{x}) + \frac{1}{2\sigma_b^2} b'_i(x_i) - \frac{P'_l(y_i|x_i)}{P_l(y_i|x_i)}, \tag{7.14}$$

$$W'_{x_i}(\mathbf{x}) = 2\sum_{j=1}^{n} w_{i,j}(x_i - x_j), \tag{7.15}$$

$$b'_i(x_i) = \begin{cases} -2(y_{\min} - x_i) & (y_i = y_{\min}), \\ -2(y_{\max} - x_i + 1) & (y_i = y_{\max}), \\ 0 & \text{otherwise}, \end{cases} \tag{7.16}$$

$$P'_l(y_i|x_i) = \frac{1}{\sqrt{2\pi\sigma_g^2}} \left( -\exp\left( -\frac{(y_i - x_i + 1)^2}{2\sigma_g^2} \right) + \exp\left( -\frac{(y_i - x_i)^2}{2\sigma_g^2} \right) \right), \tag{7.17}$$

where $W'_{x_i}(\mathbf{x})$ is a partial derivative of the function $W(\mathbf{x})$ with respect to the variable $x_i$. The problem can be solved with the L–BFGS–B quasi-Newton method [3, 13, 24].

## 7.3 Experimentation

We compared the results of the conventional methods and our method in Fig. 7.9, Fig. 7.10, Fig. 7.11, Fig. 7.12 and Table 7.1. The outputs were estimated from a quantized image. We evaluated their image qualities by calculating the peak signal to noise ratio (PSNR) and voting by persons. Note that the PSNR values for natural images can not be calculated because a ground-truth signal, namely a photograph without noise, can not be extracted from a natural image.

Two types of the conventional methods are compared with our methods: the flooding–based one [20] and the optimization–based one [21]. The reason why the low–pass based methods are excluded is that they are not suitable to reconstruct smooth gradation from false contours (See Fig. 6.6 in Chapter 6).

### 7.3.1 Experimental Results

Fig. 7.9 shows the results for artificial data. The original image was generated with additional noise in our computer. Then, the image was quantized and inputted into each method. The conventional methods can not reconstruct smooth gradation because of noise, but our method can.

Fig. 7.10, Fig. 7.11, and Fig. 7.12 show the results for a natural image. In the sky of the input image, false contour artifacts are evident. Our method sufficiently removed the artifacts (Fig. 7.12(b)). This demonstrates that it can reconstruct smooth gradations efficiently.

Table 7.1 shows the visual quality evaluations by persons. The voters are presented three images which are outputs of Wan et al. (2012) [20], Wan et al. (2014) [21] and our method. And then, they choose one's best image.

The voting was held on a specially prepared web site. Voters access to the site and look the images with their digital monitor. For artificial images, the ground-truth image is opened to voters, and they choose an image most similar to it. On the other hand, voting for natural images, original image is not opened. The voters can only see 3 output images, and they have to decide which one is a most quality photograph.The order of the output images are randomly shuffled for each voter. It is blind test because the voters are not able to know any identities about the images. Images used for the voting are shown in Fig. 7.4, Fig. 7.5, Fig. 7.6, Fig. 7.7, and Fig. 7.8.

As a result, most of the voters chose our method in every case.

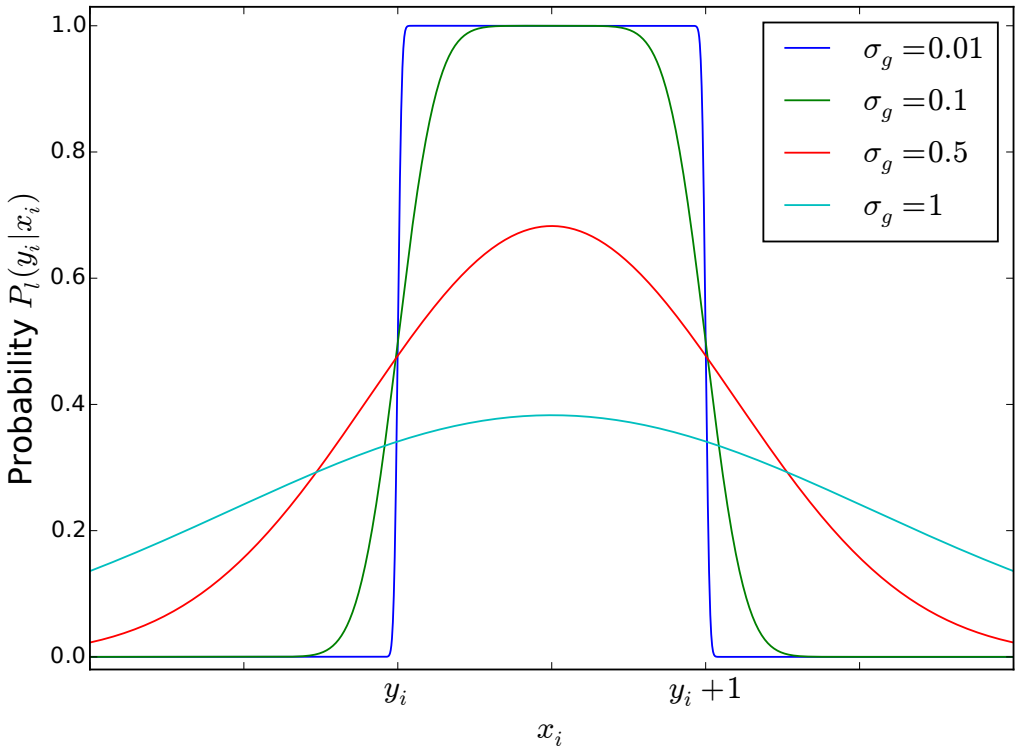



Fig. 7.3: Likelihood function

Table 7.1: Visual quality evaluation by voting

| Image | Percentage of votes (%) | | |
|---|---|---|---|
| | Wan 2012 [20] | Wan 2014 [21] | Proposed |
| Artificial I | 12.9 | 6.5 | **80.6** |
| Artificial II | 22.6 | 16.1 | **61.3** |
| Natural I | 12.9 | 6.5 | **80.6** |
| Natural II | 6.5 | 3.2 | **90.3** |
| Natural III | 38.7 | 9.7 | **51.6** |

(Gray scale $4 \rightarrow 8$ bits, Number of samples $= 31$)

**Voting Environment**

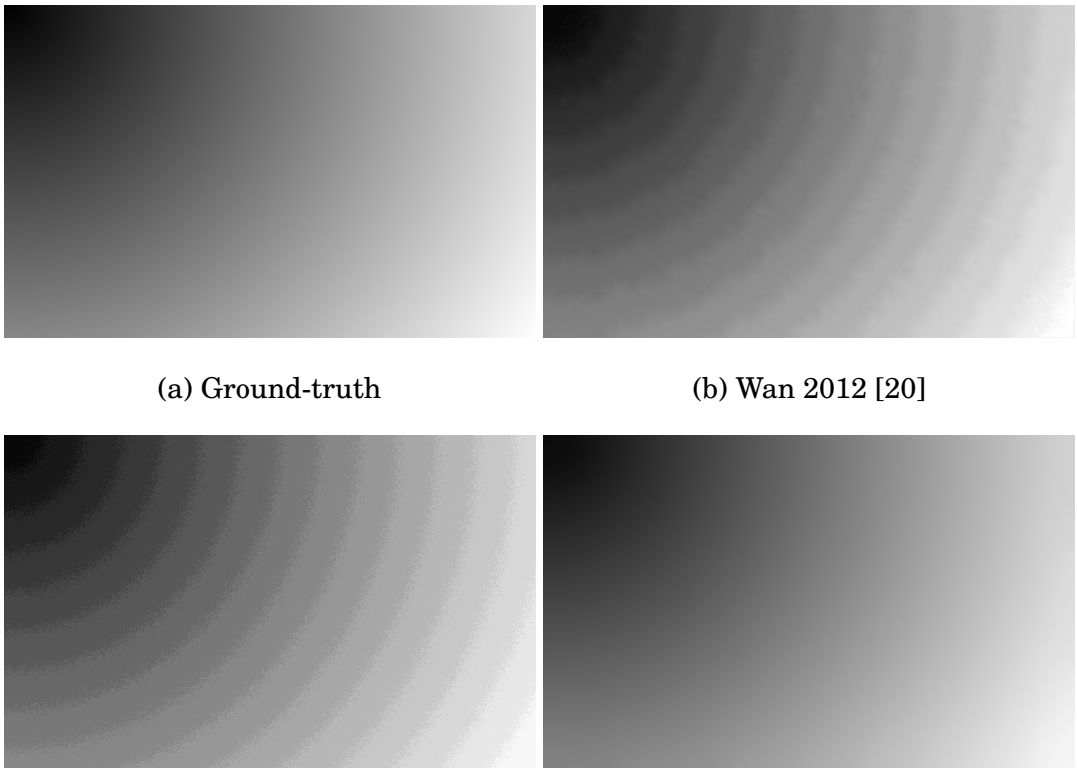
Voters accessed to the voting site from various environments through the internet. We did not fix machine platforms, web browsers and display settings for simplicity. We only limited the number of accessible time to one per voter. Here shows statistics of the voter's platforms (app.Table 7.2), web browsers (app.Table 7.3) and monitor sizes (app.Table 7.4).

Table 7.2: Voter's platform

| Platform name | number of voters |
|---|---|
| Windows | 26 |
| iPhone | 2 |
| Macintosh | 1 |
| Linux | 1 |
| X11 | 1 |

Table 7.3: Voter's web browser

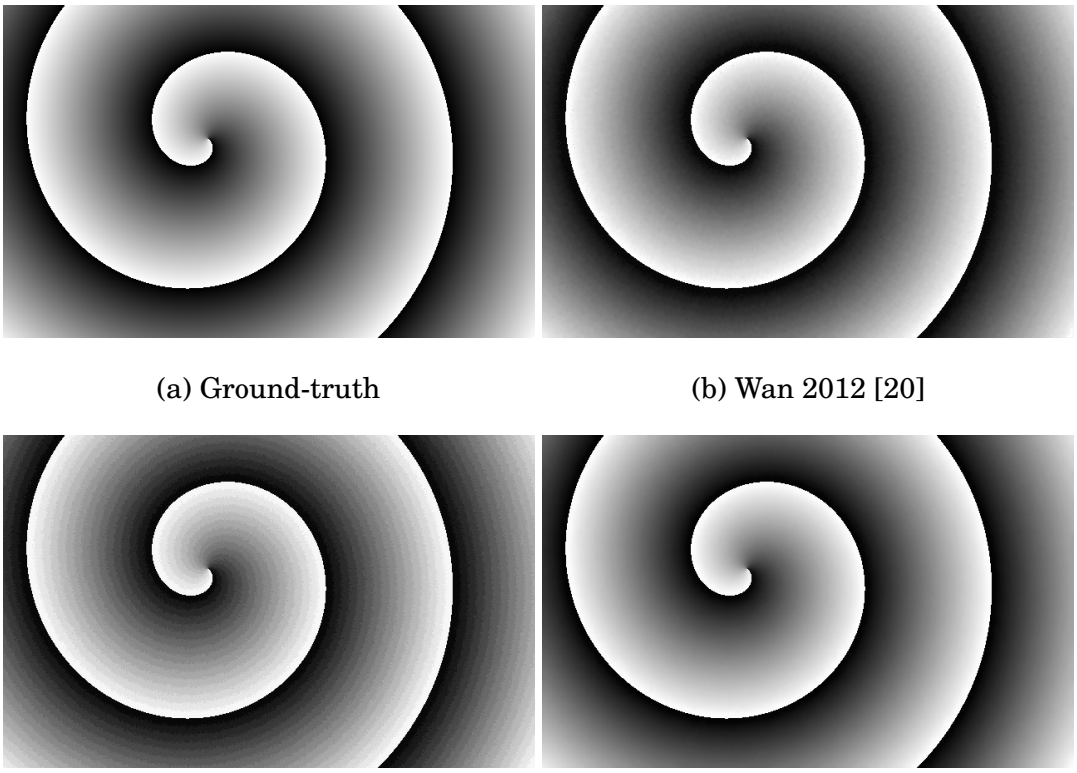
| Browser name | number of voters |
|---|---|
| Chrome | 17 |
| Firefox | 5 |
| Internet Explorer | 5 |
| Safari | 3 |
| Android | 1 |



(a) Ground-truth (b) Wan 2012 [20]

(c) Wan 2014 [21] (d) Proposed

Fig. 7.4: “Artificial I” in Table 7.1



(a) Ground-truth (b) Wan 2012 [20]

(c) Wan 2014 [21] (d) Proposed

Fig. 7.5: “Artificial II” in Table 7.1

(a) Wan 2012 [20]

(b) Wan 2014 [21]

(c) Proposed

Fig. 7.6: “Natural I” in Table 7.1

(a) Wan 2012 [20]

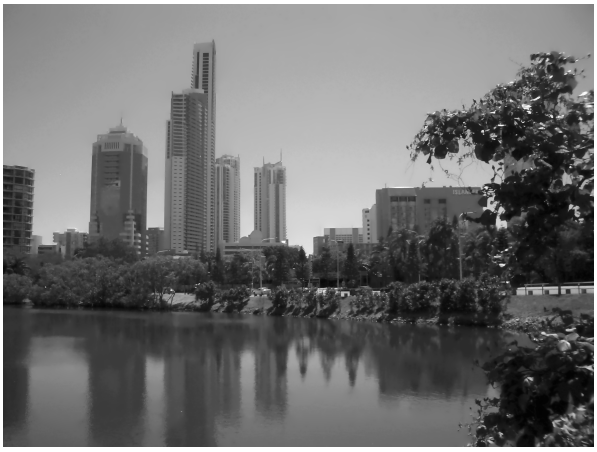
(b) Wan 2014 [21]

(c) Proposed

Fig. 7.7: “Natural II” in Table 7.1

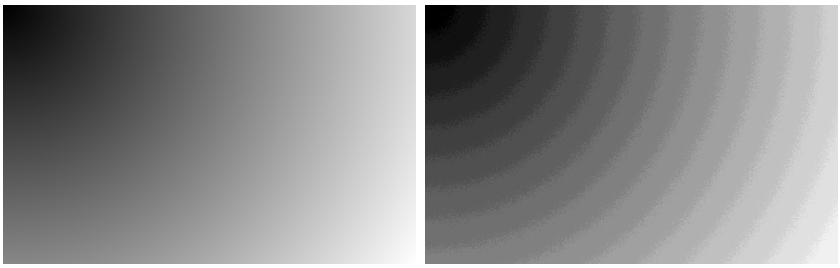
(a) Wan 2012 [20]

(b) Wan 2014 [21]

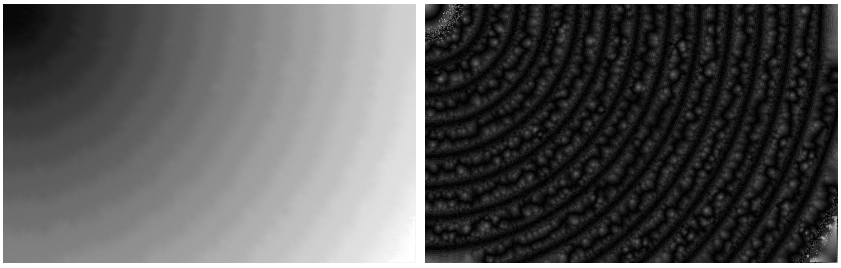


(c) Proposed

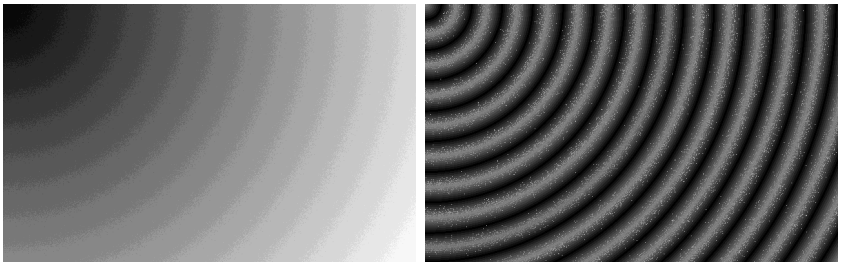
Fig. 7.8: “Natural III” in Table 7.1



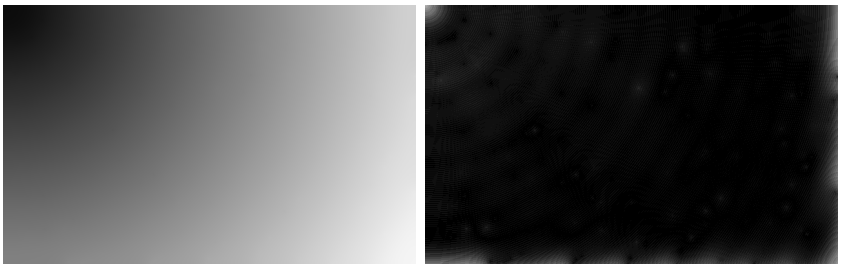
(a) Ground-truth (left) and quantized image after adding noises (right) of "Artificial I"



(b) Wan et al. (2012) [20] (PSNR:41.9 dB)



(c) Wan et al. (2014) [21] (PSNR:34.1 dB)



(d) Our method (PSNR:46.1 dB): $\kappa = 1$, $\sigma_s = 0.01, \sigma_g = 0.1$ and $\sigma_b = 0.5$

Fig. 7.9: Results and absolute error maps of BDE (gray scale $4 \rightarrow 8$ bits) for an artificially generated data. For (b)–(d), the left column shows results and the right error maps against the ground–truth. The ranges of all error maps are amplified ($\times 16$) for visibility.

(a) Original image

(b) Quantized image

Fig. 7.10: Original and quantized image of “Natural I”

(a) Wan et al. (2012) [20]

(b) Wan et al. (2014) [21]

Fig. 7.11: Results of conventional methods for “Natural I”

(a) Proposed method in Chapter 5 (2013)

(b) Proposed method in Chapter 7 (2015)

Fig. 7.12: Results of our methods in this paper for “Natural I”

Table 7.4: Voter's monitor size

| Monitor size (width×height) | number of voters |
|---|---|
| 1920×1080 | 15 |
| 1680×1050 | 5 |
| 1536×864 | 3 |
| 1280×1024 | 2 |
| 1080×1920 | 1 |
| 1366×768 | 1 |
| 2560×1440 | 1 |
| 320×568 | 1 |
| 375×667 | 1 |
| 720×1280 | 1 |

# Chapter 8

# Summary and Conclusion

This research is performed aiming at the bit-depth control of digital images whenever necessary. We, designers or researchers on image processing algorithm, expect that the input image data have sufficient bit-depth for the process. Actually, however, the bit-depth of image data is limited. The limited bit-depth causes quality degradation of image processing and displaying. Therefore, the image bit-depth expansion is required.

The most important issue for image quality is that the image has the smooth gradations. In chapter 5, we proposed smooth gradation reconstruction method. We extended the one-dimensional interpolation problem to two-dimensional BDE problem using the Poisson equation. The advantages of the approach are easiness of improvement and mathematically-guaranteed smoothness. If we improve the one-dimensional interpolation method, we can get smooth output by merging all of the interpolated rows and columns.

In chapter 7, we proposed a method for improving BDE quality for natural images. This method is designed as an inverse problem of the quantization. Our quantization model expresses the quantization process in natural image. The model is designed on the premise that noise are always with natural images.

Then we formulated the inverse quantization problem via MAP estimation focusing on noise distribution. Our method can recover smooth gradations from the noisy contour artifacts.

The BDE methods in this thesis have better quality by quality evaluations. The outputs of our method were rated as high quality by persons. From the results, our research are expected to contribute to the quality of various applications such as contrast correction, HDR imaging system, or wireless image transmission.

# Acknowledgments

This work carried out at Graduate School of Information Science and Technology, Hokkaido University, under the direction of Associate Professor Masayuki Ikebe.

I would like to express my deepest gratitude to Associate Professor Masayuki Ikebe for many helpful discussions. His comments and suggestions were of inestimable value for my study.

I would like to express my appreciation to Professor Junichi Motohisa, Professor Takashi Fukui, Professor Masato Motomura, and Assistant Professor Katsuhiro Tomioka for their insightful advises and discussions.

I am also indebt to Dr. Sousuke Shimoyama, Dr. Masaki Igarashi, Mr. Kenta Yamano, Mr. Naoto Kato, Mr. Satoshi Chikuda, Mr. Yuta Kimura, Ms. Fu Yuhan, Mr. Hotaka Kusano, Mr. Takuto Tsuji, Mr. Takayuki Yoshida, and members of our research group, for their technical supports, valuable discussions and encouragement.

I would like to thank my grandparents and my brothers for their support and encouragement. Thank you.

Finally, I would like to thank my parents, Koichi Mizuno and Makiko Mizuno, most sincerely for encouraging me to do what I want to do. Thank you so much.

# Bibliography

[1] W. Ahn and J.-S. Kim. “Flat-Region Detection and False Contour Removal in the Digital TV Display”. In: *IEEE International Conference on Multimedia and Expo, 2005. ICME 2005*. July 2005, pp. 1338–1341.

[2] S. Bhagavathy, J. Llach, and J. Zhai. “Multiscale Probabilistic Dithering for Suppressing Contour Artifacts in Digital Images”. In: *IEEE Transactions on Image Processing* 18.9 (Sept. 2009), pp. 1936–1945.

[3] R. Byrd et al. “A Limited Memory Algorithm for Bound Constrained Optimization”. In: *SIAM Journal on Scientific Computing* 16.5 (Sept. 1995), pp. 1190–1208.

[4] C.-H. Cheng et al. “Bit-depth expansion by contour region reconstruction”. In: *IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009*. May 2009, pp. 944–947.

[5] C.-H. Cheng et al. “Low color bit-depth image enhancement by contour-region dithering”. In: *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2009. PacRim 2009*. Aug. 2009, pp. 666–670.

[6] H.-R. Choi et al. “False contour reduction using directional dilation and edge-preserving filtering”. In: *IEEE Transactions on Consumer Electronics* 52.3 (Aug. 2006), pp. 1099–1106.

[7] S. J. Daly and X. Feng. “Decontouring: prevention and removal of false contour artifacts”. In: *Proc. SPIE, Human Vision and Electronic Imaging IX*. Vol. 5292. San Jose, CA: SPIE, June 2004, pp. 130–149.

[8] R. Fattal, D. Lischinski, and M. Werman. “Gradient domain high dynamic range compression”. In: *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. SIGGRAPH ’02. New York, NY, USA: ACM, 2002, pp. 249–256.

[9] G. D. Finlayson, S. D. Hordley, and M. S. Drew. “Removing Shadows from Images”. en. In: *Computer Vision—ECCV 2002*. Ed. by A. Heyden et al. Lecture Notes in Computer Science 2353. Springer Berlin Heidelberg, Jan. 2006, pp. 823–836.

[10] J. W. Lee et al. “Two-stage false contour detection using directional contrast and its application to adaptive false contour reduction”. In: *IEEE Transactions on Consumer Electronics* 52.1 (Feb. 2006), pp. 179–188.

[11] C. H. Liu et al. “Bit-depth expansion by adaptive filter”. In: *IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008*. May 2008, pp. 496–499.

[12] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. en. Cambridge University Press, Sept. 2003.

[13] J. L. Morales and J. Nocedal. “Remark on “Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-scale Bound Constrained Optimization””. In: *ACM Trans. Math. Softw.* 38.1 (Dec. 2011), 7:1–7:4.

[14] M.-H. Park et al. “False contour reduction using neural networks and adaptive bi-directional smoothing”. In: *IEEE Transactions on Consumer Electronics* 56.2 (May 2010), pp. 870–878.

[15] P. Perez, M. Gangnet, and A. Blake. "Poisson image editing". In: *ACM SIGGRAPH 2003 Papers*. SIGGRAPH '03. New York, NY, USA: ACM, 2003, pp. 313–318.

[16] H. Seetzen et al. "High dynamic range display systems". In: *ACM SIGGRAPH 2004 Papers*. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 760–768.

[17] D. Shuman et al. "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains". In: *IEEE Signal Processing Magazine* 30.3 (May 2013), pp. 83–98.

[18] J. Sun et al. "Poisson matting". In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 315–321.

[19] A. Taguchi and J. Nishiyama. "Bit-length expansion by inverse quantization process". In: *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. Aug. 2012, pp. 1543–1547.

[20] P. Wan et al. "From 2D Extrapolation to 1D Interpolation: Content Adaptive Image Bit-Depth Expansion". In: *2012 IEEE International Conference on Multimedia and Expo (ICME)*. July 2012, pp. 170–175.

[21] P. Wan et al. "Image bit depth enhancement via maximum-a-posteriori estimation of graph AC component". In: IEEE – Institute of Electrical and Electronics Engineers, Oct. 2014.

[22] P. Wan et al. "Image de-quantization via spatially varying sparsity prior". In: *2012 19th IEEE International Conference on Image Processing (ICIP)*. 2012, pp. 953–956.

[23] N. Xu and Y.-T. Kim. "A simple and effective algorithm for false contour reduction in digital television". In: *2010 Digest of Technical Papers Inter-*

*national Conference on Consumer Electronics (ICCE)*. Jan. 2010, pp. 231–232.

[24] C. Zhu et al. “Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-scale Bound-constrained Optimization”. In: *ACM Trans. Math. Softw.* 23.4 (Dec. 1997), pp. 550–560.

# List of Publications

1. 査読付学会誌

（a）水野暁, 五十嵐正樹, 池辺将之, “ポアソン方程式を用いた画像の滑らかな階調復元,” 映像情報メディア学会誌, 67 巻, 8 号, pp.J326–J333 (2013)

（b）Akira Mizuno and Masayuki Ikebe, “Smooth gradation reconstruction from false contours in quantized natural image,” ITE Transactions on Media Technology and Applications, Vol.3, No.4, pp.251–257 (2015)

2. 査読付国際会議

（a）Akira Mizuno, Masayuki Ikebe, Masaki Igarashi, Sousuke Shimoyama, and Junichi Motohisa “Image restoration and edge detection of error diffused images using local centroid of brightness,” 2011 International Symposium on Multimedia and Communication Technology, Sapporo, Japan, pp.48–51, Sep, 2011

（b）Akira Mizuno, Masayuki Ikebe, Masaki Igarashi, Sousuke Shimoyama, and Junichi Motohisa, “Fast single image dehazing using local statistics of dark channel,” 2012 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing, Honolulu, USA, pp.186–189, Mar, 2012

（c）Masaki Igarashi, Akira Mizuno, and Masayuki Ikebe, “Accuracy improvement of histogram-based image filtering,” 2013 20th IEEE In-

ternational Conference on Image Processing (ICIP), Melbourne, Australia, pp.1217–1221, Sep, 2013 (採択率 44.5%)

（d）Akira Mizuno and Masayuki Ikebe, “Bit-depth expansion for noisy contour reduction in natural images,” 2016 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, March, 2016 (採択決定済み)

3. その他講演

（a）水野暁, 池辺将之, 本久順一, “低ビット誤差拡散画像の高ビット画像復元,” 電子情報通信学会総合大会, 2011 年 2 月

（b）下山荘介, 池辺将之, 五十嵐正樹, 水野暁, 本久順一, “局所的画像変換によるマッハバンドに関する一考察,” 電子情報通信学会ソサイエティ大会, 2011 年 8 月

（c）下山荘介, 池辺将之, 五十嵐正樹, 水野暁, 本久順一, “高速局所的輝度補正技術の高画質化の検討,” 映像情報メディア学会情報センシング研究会, 2011 年 5 月

（d）下山荘介, 池辺将之, 五十嵐正樹, 水野暁, 本久順一, “局所的輝度補正手法におけるハロー効果制御の検討,” 映像情報メディア学会情報センシング研究会, 2011 年 11 月

（e）水野暁, 池辺将之, 五十嵐正樹, 本久順一, “単一画像を用いた局所適応型高速霧補正技術,” 映像情報メディア学会情報センシング研究会, 2012 年 5 月

（f）水野暁, 池辺将之, “ポアソン方程式を用いた画像のビット深度拡張,” 映像情報メディア学会情報センシング研究会, 2013 年 9 月

（g）水野暁, 池辺将之, “量子化過程の確率的モデルを用いた画像のビット深度拡張技術 Image bit-depth expansion using probabilistic model of quantization,” Optics & Photonics Japan 2015 新画像システム・情報フォトニクス研究討論会, 2015 年 10 月