



Bayesian Image Denoising with Multiple Noisy Images

Shun Kataoka¹ · Muneki Yasuda²

Received: 3 February 2019 / Accepted: 20 June 2019 / Published online: 16 July 2019
© The Author(s) 2019

Abstract

In this paper, we propose a fast image denoising method based on discrete Markov random fields and the fast Fourier transform. The purpose of the image denoising is to infer the original noiseless image from a noise corrupted image. We consider the case where several noisy images are available for inferring the original image and the Bayesian approach is adopted to create the posterior probability distribution of the denoised image. In the proposed method, the estimation of the denoised image is achieved using belief propagation and an expectation–maximization algorithm. We numerically verified the performance of the proposed method using several standard images.

Keywords Image denoising · Discrete Markov random field · Belief propagation · EM algorithm · FFT

1 Introduction

Bayesian image processing based on Markov random fields (MRFs) is an important framework in the field of image processing [1, 2]. An MRF is a undirected graph representation of probability distribution, and many applications of MRFs exist in the image processing and computer vision fields [3–5]. MRFs have also been applied to other research fields, including traffic engineering [6, 7] and earth science [8, 9]. In Bayesian image processing, the objective image can be inferred based on the posterior probability distribution.

Recently, we proposed a fast image denoising method for the case where multiple noisy images are available for inferring the original noiseless image that is based on

✉ Shun Kataoka
xskataoka@res.otaru-uc.ac.jp
Muneki Yasuda
muneki@yz.yamagata-u.ac.jp

¹ Faculty of Commerce, Otaru University of Commerce, Otaru-shi 047-8571, Japan

² Graduate School of Science and Engineering, Yamagata University, Yonezawa-shi 992-8510, Japan

Gaussian MRFs [10]. However, in the study in [10], for ease of mathematical treatment, we made an unnatural assumption that pixel values are continuous. In general, a pixel takes a discrete value from 0 to 255, and an additional framework is required to treat pixel values as discrete instead of continuous values. Therefore, in this paper, we focus on the Bayesian image denoising problem of inferring the original noiseless image from multiple noisy images when the pixel values are treated as discrete values. We created a probability model for image denoising based on the discrete MRF and Bayesian perspective. A major disadvantage of an image processing model based on discrete MRFs is the computational complexity. In fact, the inference problem from discrete MRFs belongs to the NP-hard class. Therefore, an approximate inference technique is required to infer the objective image from a discrete MRF. Belief propagation [11] is known as one such effective technique. In this paper, we propose an effective image denoising algorithm for multiple noisy images that applies belief propagation. The main contributions of this paper are that an MRF model for image denoising with multiple noisy images is defined and a fast effective denoising algorithm based on our discrete MRF model and the fast Fourier Transform (FFT) is proposed.

The remainder of this paper is organized as follows. In Sect. 2, we define a probability model for image denoising with multiple noisy images based on the discrete MRF and Bayesian perspective. In Sect. 3, we derive an image denoising algorithm based on the posterior probability distribution defined in Sect. 2. In Sect. 4, we describe the framework for estimating the parameters in the posterior probability distribution. We explain the implementation of our denoising method using the FFT in Sect. 5. In Sect. 6, we describe the numerical verification of the performance of the proposed method. Finally, in Sect. 7, we present our concluding remarks.

2 Framework of Bayesian Image Denoising Method

In this section, we briefly explain the framework of the Bayesian image denoising method for the case where multiple noisy images are available. Suppose that we have K degraded images that are independently obtained by adding additive white Gaussian noise (AWGN) to the original image. We assume that the images are composed of $N = h \times w$ pixels. Let $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_N]^T$ and $\mathbf{y}^{(k)} = [y_1^{(k)} \ y_2^{(k)} \ \dots \ y_N^{(k)}]^T$ be N dimensional vectors corresponding to the original image and the k -th noisy image, respectively. Vectors \mathbf{x} and $\mathbf{y}^{(k)}$ can be easily obtained by raster scanning the images. We assume that $x_i (i = 1, 2, \dots, N)$ takes L discrete values from 0 to $L - 1$.

The purpose of the image denoising is to infer the original noiseless image \mathbf{x} from K noisy images $Y = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(K)}\}$. In the Bayesian framework, the original image \mathbf{x} can be inferred using the posterior probability distribution $P(\mathbf{x}|Y)$ that is expressed as

$$P(\mathbf{x}|Y) = \frac{P(Y|\mathbf{x})P(\mathbf{x})}{\sum_{\mathbf{x}} P(Y|\mathbf{x})P(\mathbf{x})}, \quad (1)$$

where $\sum_{\mathbf{x}}$ denotes the multiple summations over all the possible L^N states of \mathbf{x} . The framework of the proposed Bayesian image denoising method is illustrated in Fig. 1. From the definition of $\mathbf{y}^{(k)}$, the probability density function $P(Y|\mathbf{x})$ is expressed as

$$\begin{aligned}
 P(Y|\mathbf{x};\sigma^2) &= \prod_{k=1}^K P(y^{(k)}|\mathbf{x};\sigma^2) \\
 &= \prod_{k=1}^K \prod_{i \in V} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (y_i^{(k)} - x_i)^2\right),
 \end{aligned}
 \tag{2}$$

where $V = \{1, 2, \dots, N\}$ and σ^2 is the variance of the AWGN. We express the parameters of the probability model by its arguments after the semicolon as Eq. (2). We define the prior probability distribution as

$$P(\mathbf{x};\alpha) = \frac{1}{Z_{\text{prior}}(\alpha)} \exp\left(-\alpha \sum_{ij \in E} \phi(x_i - x_j)\right),
 \tag{3}$$

where E is a set of edges of the $h \times w$ lattice graph and $\phi(x)$ is a downward convex even function taking its minimum at $x = 0$. In this study, we assumed the periodic boundary condition on the graph structure E , as demonstrated in Fig. 2. $\alpha > 0$ is the parameter of the prior probability distribution; if α is set to a large value, neighboring x_i and x_j tend to take close values. $Z_{\text{prior}}(\alpha)$ is a normalization constant defined as

$$Z_{\text{prior}}(\alpha) = \sum_{\mathbf{x}} \exp\left(-\alpha \sum_{ij \in E} \phi(x_i - x_j)\right).
 \tag{4}$$

By substituting Eqs. (2) and (3) into Eq. (1), the posterior probability distribution $P(\mathbf{x}|Y)$ is expressed as

$$\begin{aligned}
 P(\mathbf{x}|Y;\alpha, \sigma^2) &= \frac{1}{Z_{\text{post}}(\alpha, \sigma^2)} \exp\left(-\frac{1}{2\sigma^2} \sum_{i \in V} \psi_i(x_i) - \alpha \sum_{ij \in E} \phi(x_i - x_j)\right),
 \end{aligned}
 \tag{5}$$

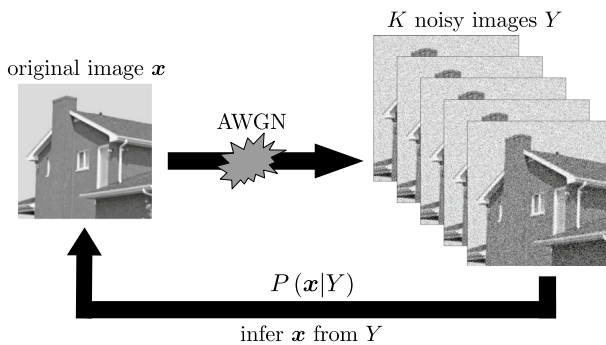
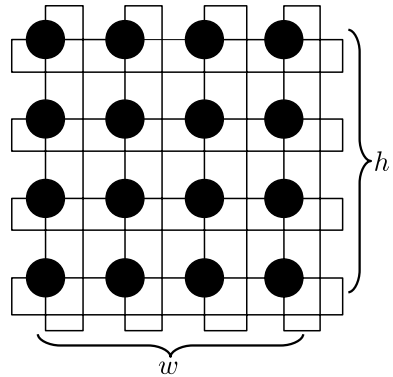


Fig. 1 Illustration of the proposed Bayesian image denoising method

Fig. 2 Periodic boundary condition for $h \times w$ lattice graph E , when $h = 4$ and $w = 4$



where

$$\psi_i(x_i) = \sum_{k=1}^K \left(x_i - y_i^{(k)}\right)^2, \tag{6}$$

and

$$Z_{\text{post}}(\alpha, \sigma^2) = \sum_x \exp\left(-\frac{1}{2\sigma^2} \sum_{i \in V} \psi_i(x_i) - \alpha \sum_{ij \in E} \phi(x_i - x_j)\right), \tag{7}$$

respectively.

3 Inference Algorithm Based on Belief Propagation

The image denoising is achieved by finding the image \hat{x} that maximizes the posterior probability distribution in Eq. (5):

$$\hat{x} = \underset{x}{\operatorname{argmax}} P(x|Y; \alpha, \sigma^2). \tag{8}$$

However, the problem of determining such an image is intractable, because this maximization problem belongs to the NP-hard class. Therefore, we need an approximate inference method to find \hat{x} . In this section, we explain an effective approximate inference method called belief propagation for inferring the denoised image \hat{x} .

Belief propagation is a method of computing the approximate marginal distributions $b_i(x_i)$ and $b_{ij}(x_i, x_j)$ for each $i \in V$ and $ij \in E$. In the belief propagation framework, the approximate marginal distributions $b_i(x_i)$ and $b_{ij}(x_i, x_j)$ are given by

$$b_i(x_i) = \frac{1}{Z_i} \exp\left(-\frac{1}{2\sigma^2} \psi_i(x_i)\right) \prod_{k \in \partial i} M_{k \rightarrow i}^{\text{post}}(x_i), \tag{9}$$

$$Z_i = \sum_{x_i} \exp\left(-\frac{1}{2\sigma^2} \psi_i(x_i)\right) \prod_{k \in \partial i} M_{k \rightarrow i}^{\text{post}}(x_i), \tag{10}$$

and

$$b_{ij}(x_i, x_j) = \frac{1}{Z_{ij}} \exp(-\alpha\phi(x_i - x_j)) m_{\sqrt{j \rightarrow i}}^{\text{post}}(x_i) m_{\sqrt{i \rightarrow j}}^{\text{post}}(x_j), \tag{11}$$

$$Z_{ij} = \sum_{x_i} \sum_{x_j} \exp(-\alpha\phi(x_i - x_j)) m_{\sqrt{j \rightarrow i}}^{\text{post}}(x_i) m_{\sqrt{i \rightarrow j}}^{\text{post}}(x_j), \tag{12}$$

respectively, where $\partial i = \{k \in V | ik \in E\}$ is a set of all the neighboring pixels of pixel i . $M_{k \rightarrow i}^{\text{post}}(x_i)$ in Eqs. (9) and (10) is a message from pixel k to pixel i and is obtained by the convergence point of the message update rule

$$M_{j \rightarrow i}^{\text{post}}(x_i) = \frac{1}{Z_{j \rightarrow i}} \sum_{x_j} \exp(-\alpha\phi(x_i - x_j)) m_{\sqrt{i \rightarrow j}}^{\text{post}}(x_j), \tag{13}$$

$$m_{\sqrt{j \rightarrow i}}^{\text{post}}(x_i) = \exp\left(-\frac{1}{2\sigma^2} \psi_i(x_i)\right) \prod_{k \in \partial i \setminus \{j\}} M_{k \rightarrow i}^{\text{post}}(x_i), \tag{14}$$

where $Z_{j \rightarrow i}$ is a normalization constant to ensure algorithmic stability. The estimation of the denoised image $\hat{x} = \{\hat{x}_i | i \in V\}$ is achieved by finding

$$\hat{x}_i = \underset{x_i}{\operatorname{argmax}} b_i(x_i), \tag{15}$$

for $i \in V$ in the belief propagation framework.

4 Parameter Estimation Using Expectation–Maximization Algorithm

In the preceding section, we explained the method for inferring the denoised image \hat{x} based on belief propagation. In our framework, the denoised image is inferred from the posterior probability distribution in Eq. (5), which has two parameters, α and σ^2 . It is obvious that the inferred denoised image \hat{x} depends on these parameters. In this section, we explain the method for determining these parameters from degraded images Y based on the expectation–maximization (EM) algorithm [12].

The EM algorithm is a statistical inference method to infer the maximum likelihood estimates

$$\hat{\alpha}, \hat{\sigma}^2 = \underset{\alpha, \sigma^2}{\operatorname{argmax}} \sum_{\mathbf{x}} P(\mathbf{x}, Y; \alpha, \sigma^2), \tag{16}$$

by an iterated method. In the EM algorithm framework, the parameters α and σ^2 are estimated by iterative maximization of the Q function defined as

$$\begin{aligned}
 Q(\alpha, \sigma^2; \alpha_t, \sigma_t^2) &= \sum_{\mathbf{x}} P(\mathbf{x}|Y; \alpha_t, \sigma_t^2) \log P(\mathbf{x}, Y; \alpha, \sigma^2) \\
 &= -\frac{1}{2\sigma^2} \sum_{i \in V} \langle \psi_i(x_i); \alpha_t, \sigma_t^2 \rangle_{\text{post}} - \frac{NK}{2} \log \sigma^2 \\
 &\quad - \alpha \sum_{ij \in E} \langle \phi(x_i - x_j); \alpha_t, \sigma_t^2 \rangle_{\text{post}} - \log Z_{\text{prior}}(\alpha) \\
 &\quad + \text{Const.},
 \end{aligned}
 \tag{17}$$

where α_t and σ_t^2 are estimates of the parameters at the t -th iteration and

$$\langle f(\mathbf{x}); \alpha_t, \sigma_t^2 \rangle_{\text{post}} = \sum_{\mathbf{x}} f(\mathbf{x}) P(\mathbf{x}|Y; \alpha_t, \sigma_t^2).
 \tag{18}$$

Using belief propagation, we can approximate the expectations in Eq. (17) as

$$\langle \psi_i(x_i); \alpha_t, \sigma_t^2 \rangle_{\text{post}} = \sum_{x_i} \psi_i(x_i) b_i^{(t)}(x_i),
 \tag{19}$$

and

$$\langle \phi(x_i - x_j); \alpha_t, \sigma_t^2 \rangle_{\text{post}} = \sum_{x_i} \sum_{x_j} \phi(x_i - x_j) b_{ij}^{(t)}(x_i, x_j),
 \tag{20}$$

respectively, where $b_i^{(t)}(x_i)$ and $b_{ij}^{(t)}(x_i, x_j)$ are the approximate marginal distributions of the posterior probability distribution $P(\mathbf{x}|Y; \alpha_t, \sigma_t^2)$ computed using Eqs. (9) and (11). The parameter update at iteration t is given by

$$\alpha_{t+1}, \sigma_{t+1}^2 = \underset{\alpha, \sigma^2}{\text{argmax}} Q(\alpha, \sigma^2; \alpha_t, \sigma_t^2),
 \tag{21}$$

and the maximum likelihood estimates in Eq. (16) are given as the convergence point of the above iterative estimation. By differentiating the Q function with respect to α and σ^2 and considering the conditions for the extremal value, the updated parameter α_{t+1} in Eq. (21) is expressed as the solution of the equation

$$\sum_{ij \in E} \langle \phi(x_i - x_j); \alpha_t, \sigma_t^2 \rangle_{\text{post}} = \sum_{ij \in E} \langle \phi(x_i - x_j); \alpha \rangle_{\text{prior}},
 \tag{22}$$

and the updated parameter σ_{t+1}^2 is calculated as

$$\sigma_{t+1}^2 = \frac{1}{NK} \sum_{i \in V} \langle \psi_i(x_i); \alpha_t, \sigma_t^2 \rangle_{\text{post}},
 \tag{23}$$

where $\langle \phi(x_i - x_j); \alpha \rangle_{\text{prior}} = \sum_{\mathbf{x}} \phi(x_i - x_j) P(\mathbf{x}; \alpha)$; this expectation can also be computed approximately using belief propagation similarly to Eq. (20). Using the bisection method, we can easily find α_{t+1} that satisfies Eq. (22).

5 Proposed Algorithm: Fast Implementation Based on the Fast Fourier Transform

The image denoising algorithm based on our probabilistic model in Eq. (5) described in Sects. 2–4 is summarized in Algorithm 1, together with the differences in the computation times of the naive implementation and the proposed method, which is explained in this section. The worst computation time of the naive implementation of this algorithm is $O(T_{EM}T_{BP}NL^2)$, where T_{EM} and T_{BP} are the maximum number of updates for the parameter update in Eq. (21) and the message update in Eq. (13), respectively. In Algorithm 1, we terminate the parameter and message updates in iteration T_{EM} and T_{BP} , respectively. Because we assume the periodic boundary condition for the graph structure E , the number of edges is $|E| = 2N$. Therefore, the number of messages passing each edge is $O(N)$. The computation time of the naive message update from pixel j to pixel i is $O(L^2)$, because Eq. (13) must be computed for each $x_i = 0, 1, \dots, L - 1$ to update a message $M_{j \rightarrow i}^{post}(x_i)$.

Algorithm 1 : Image Denoising Algorithm

Require: $Y = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(K)}\}$

- 1: Initialize α_0 and σ_0^2
- 2: **for** $t = 0, 1, 2, \dots, T_{EM} - 1$ **do**
- 3: initialize all messages $M_{j \rightarrow i}(x_i)$
- 4: **for** $t' = 0, 1, 2, \dots, T_{BP} - 1$ **do**
- 5: **for all** $j \rightarrow i$ **do**
- 6: update $M_{j \rightarrow i}(x_i)$ using Eq. (13) (naive: $O(L^2) \rightarrow$ propose: $O(L \log L)$)
- 7: **if** all messages $M_{j \rightarrow i}(x_i)$ are converged **then** break
- 8: **for all** $i \in V$ **do**
- 9: compute $\langle \psi_i(x_i); \alpha_t, \sigma_t^2 \rangle_{post}$ using Eq. (19)
- 10: **for all** $ij \in E$ **do**
- 11: compute $\langle \phi(x_i - x_j); \alpha_t, \sigma_t^2 \rangle_{post}$ using Eq. (20) (naive: $O(L^2) \rightarrow$ propose: $O(L \log L)$)
- 12: compute α_{t+1} using Eq. (22) and bisection method (naive: $O(L^2) \rightarrow$ propose: $O(L \log L)$)
- 13: compute σ_{t+1}^2 using Eq. (23)
- 14: **if** α and σ^2 are converged **then** break
- 15: initialize all messages $M_{j \rightarrow i}(x_i)$
- 16: **for** $t' = 0, 1, 2, \dots, T_{BP} - 1$ **do**
- 17: **for all** $j \rightarrow i$ **do**
- 18: update $M_{j \rightarrow i}(x_i)$ using Eq. (13) (naive: $O(L^2) \rightarrow$ propose: $O(L \log L)$)
- 19: **if** all messages $M_{j \rightarrow i}(x_i)$ are converged **then** break
- 20: **for all** $i \in V$ **do**
- 21: compute $b_i(x_i)$ using Eq. (9)
- 22: compute \hat{x}_i using Eq. (15)

It should be noted that the computation time of the message update can be reduced to $O(L \log L)$ using the FFT [13]. It has been confirmed that this FFT-based method in fact accelerates the message computation for the probabilistic image denoising model in Eq. (5) for the case where $K = 1$ [14]. However, there exist

additional $O(L^2)$ computation terms in Algorithm 1: Steps 11 and 12. In this section, we show that these $O(L^2)$ computation terms can also be computed in $O(L \log L)$ using the FFT. Therefore, we can reduce the worst computation time of Algorithm 1 to $O(T_{EM}T_{BP}NL \log L)$.

The key idea for accelerating the message updates in Eq. (13) is to consider the update rule a convolution calculation. If we define the function $f(x; \alpha)$ as

$$f(x; \alpha) = \exp(-\alpha\phi(x)), \tag{24}$$

we can reformulate the message update rules as

$$m_{j \rightarrow i}^{\text{post}}(x_i) = \sum_{x_j} f(x_i - x_j; \alpha) m_{i \rightarrow j}^{\text{post}}(x_j), \tag{25}$$

$$M_{j \rightarrow i}^{\text{post}}(x_i) = m_{j \rightarrow i}^{\text{post}}(x_i) / \sum_{l=0}^{L-1} m_{j \rightarrow i}^{\text{post}}(l). \tag{26}$$

The calculation of $m_{j \rightarrow i}^{\text{post}}(x_i)$ in Eq. (25) is a convolution calculation. Therefore, we can calculate $m_{j \rightarrow i}^{\text{post}}(x_i)$ for $x_i = 0, 1, \dots, L - 1$ in $O(L \log L)$ computation time using the FFT, and the computation time of $M_{j \rightarrow i}^{\text{post}}(x_i)$ in Eq. (26) is linear with respect to L . Therefore, we can update a message $M_{j \rightarrow i}^{\text{post}}(x_i)$ in $O(L \log L)$ computation time.

Now, we show that the expectation in Eq. (20) can be calculated in $O(L \log L)$ computation time by using the FFT. By substituting Eqs. (11) into (20), the expectation calculation can be expressed as

$$\langle \phi(x_i - x_j); \alpha, \sigma^2 \rangle_{\text{post}} = \frac{G}{H}, \tag{27}$$

$$G = \sum_{x_i} \sum_{x_j} \phi(x_i - x_j) f(x_i - x_j; \alpha) m_{i \rightarrow j}^{\text{post}}(x_j) m_{j \rightarrow i}^{\text{post}}(x_i), \tag{28}$$

$$H = Z_{ij} = \sum_{x_i} \sum_{x_j} f(x_i - x_j; \alpha) m_{i \rightarrow j}^{\text{post}}(x_j) m_{j \rightarrow i}^{\text{post}}(x_i). \tag{29}$$

If we define functions $g(x_i)$ and $h(x_i)$ as

$$g(x_i) = \sum_{x_j} \phi(x_i - x_j) f(x_i - x_j; \alpha) m_{i \rightarrow j}^{\text{post}}(x_j), \tag{30}$$

$$h(x_i) = \sum_{x_j} f(x_i - x_j; \alpha) m_{i \rightarrow j}^{\text{post}}(x_j), \tag{31}$$

respectively, we can reformulate the expectation calculation as

$$G = \sum_{x_i} g(x_i) m_{j \rightarrow i}^{\text{post}}(x_i), \tag{32}$$

and

$$H = \sum_{x_i} h(x_i)m_{\sqrt{j \rightarrow i}}^{\text{post}}(x_i), \tag{33}$$

respectively. Therefore, the computation time of the expectation in Eq. (27) is $O(L \log L)$, because the total computation time to calculate convolutions $g(x_i)$ and $h(x_i)$ in Eqs. (30) and (31) for all $x_i = 0, 1, \dots, L - 1$ is $O(L \log L)$, and Eqs. (32) and (33) can be computed in $O(L)$ computation time.

In Eq. (22), we need to calculate the messages and expectation of prior probability distribution $P(\mathbf{x}; \alpha_t)$ to find α_{t+1} that satisfies this equation using the bisection method. It should be noted that, because we assume the periodic boundary condition for the graph structure E , we can calculate the messages and expectation of prior probability distribution faster than those of posterior probability distribution by considering the translational symmetry assumption. If we assume both a periodic boundary condition and translational symmetry, the messages and expectation of prior probability distribution become not dependent on the position of the edges $ij \in E$. Therefore, the message update rule and expectation calculation for prior probability distribution are expressed as

$$M^{\text{prior}}(x_i) = \frac{1}{Z^{\text{prior}}} \sum_{x_j} \phi(x_i - x_j)f(x_i - x_j; \alpha)(M^{\text{prior}}(x_j))^3, \tag{34}$$

$$Z^{\text{prior}} = \sum_{x_i} \sum_{x_j} \exp(-\alpha\phi(x_i - x_j))(M^{\text{prior}}(x_j))^3, \tag{35}$$

and

$$\sum_{ij \in E} \langle \phi(x_i - x_j); \alpha \rangle_{\text{prior}} = 2N \frac{G^{\text{prior}}}{H^{\text{prior}}}, \tag{36}$$

$$G^{\text{prior}} = \sum_{x_i} \sum_{x_j} \phi(x_i - x_j)f(x_i - x_j; \alpha)(M^{\text{prior}}(x_i))^3(M^{\text{prior}}(x_j))^3, \tag{37}$$

$$H^{\text{prior}} = \sum_{x_i} \sum_{x_j} f(x_i - x_j; \alpha)(M^{\text{prior}}(x_i))^3(M^{\text{prior}}(x_j))^3, \tag{38}$$

respectively, where $M^{\text{prior}}(x_i)$ is a message of the prior probability distribution. The calculation of the message and the expectation of the prior probability distribution in Eqs. (34) and (36) can be computed in $O(L \log L)$ computation time by the same calculation method as Eqs. (25, 26) and Eqs. (27)–(33), respectively.



Fig. 3 Gray scale standard images used in experiments ($L = 256$)

6 Numerical Experiments

In this section, we describe the numerical verification of the proposed method. We used the standard images in Fig. 3, which are widely used in the image processing research field. The pixel values of these images take $L = 256$ different values. All the experiments were implemented using C++ and were run single-threaded on an Ubuntu 18.04.1 LTS (64 bit) machine with an Intel Core i7-6850K CPU running at 3.60 GHz and 128 GB RAM. In this experiment, we defined the function $\phi(x)$ as

$$\phi(x) = x^2, \quad (39)$$

and set the parameters of the proposed method as follows. The initial parameter α_0 was set at 0.005 and σ_0^2 was set at the sample variance calculated from Y . The maximum numbers of iterations T_{EM} and T_{BP} were set at 100 and 1000, respectively. We considered that the messages converged if the absolute value of the average change in the messages was smaller than 10^{-4} ; the same applied to the parameters α and σ^2 . We set the search interval of the bisection method for computing α_{t+1} as $[0, 2\alpha_t]$.

First, we compared the computation time of the proposed method with that of the previous methods (belief propagation FFT (BP-FFT) and Naive). The Naive method is the naive implementation version of Algorithm 1; the worst computation time is $O(T_{EM}T_{BP}NL^2)$. BP-FFT is the method used in the study presented in [14], where only the message updates in Eqs. (13) and (14) were speeded-up by using the FFT. Tables 1 and 2 show the average computation times over 10 trials for each method where K noisy images Y were generated by adding an AWGN of $\sigma = 15$ to the original noiseless images. According to the results, the proposed method was faster than the other methods. It should be noted that the difference between the three methods is in whether Algorithm 1 is implemented using the FFT. Therefore, the image denoising results of these method are all the same.

Table 1 Average computation time over 10 trials with $\sigma = 15$, $K = 1$, and image size 128×128

	Fig. 3a	Fig. 3b	Fig. 3c	Fig. 3d
Proposed (min)	15.24	15.24	11.56	15.74
BP-FFT (min)	45.14	45.10	49.79	47.55
Naive (min)	275.17	249.64	230.18	263.57

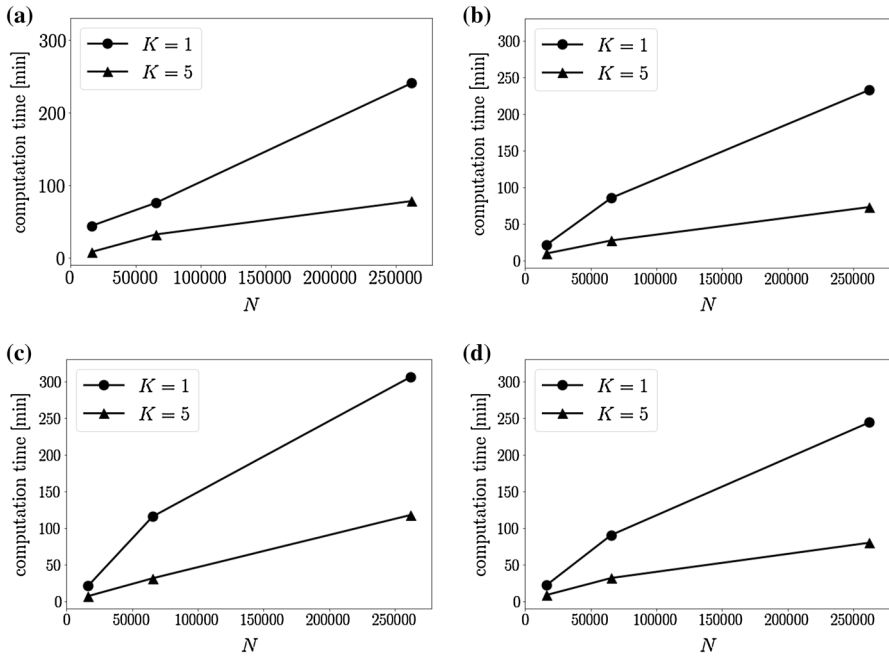


Fig. 4 Average computation time and peak signal-to-noise ratio over 10 trials versus various image sizes (128×128 , 256×256 , and 512×512) for $K = 1$ and $K = 5$. The noise level applied in these experiments was $\sigma = 15$. **a** Fig. 3a. **b** Fig. 3b. **c** Fig. 3c. **d** Fig. 3d

Table 2 Average computation time over 10 trials with $\sigma = 15$, $K = 5$, and image size 128×128

	Fig. 3a	Fig. 3b	Fig. 3c	Fig. 3d
Proposed (min)	4.86	6.46	4.22	6.05
BP-FFT (min)	14.02	18.92	16.24	16.18
Naive (min)	100.92	119.43	97.54	103.63

Figure 4 shows the average computation time versus image size for $K = 1$ and $K = 5$ over 10 trials, where the noise level of AWGN was $\sigma = 15$. The computation time of our denoising methods grows approximately linearly with the increase in the image size (it does not grow strictly linearly, because we break off the message and parameter updates according to the convergence condition).

Figure 5 shows the denoising performance of the proposed method versus various values of K for two levels of AWGN ($\sigma = 15$ and $\sigma = 30$). We evaluated the performance of the method according to the average peak signal-to-noise ratio (PSNR) over 10 trials. The PSNR is defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}, \tag{40}$$

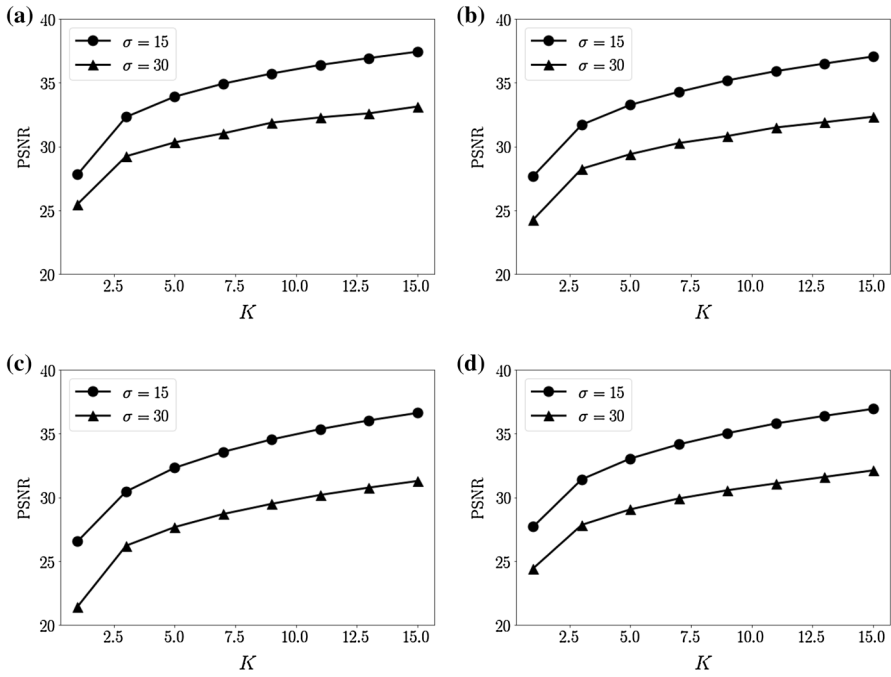


Fig. 5 Average peak signal-to-noise ratio over 10 trials against K for $\sigma = 15$ and $\sigma = 30$. **a** Fig. 3a. **b** Fig. 3b. **c** Fig. 3c. **d** Fig. 3d

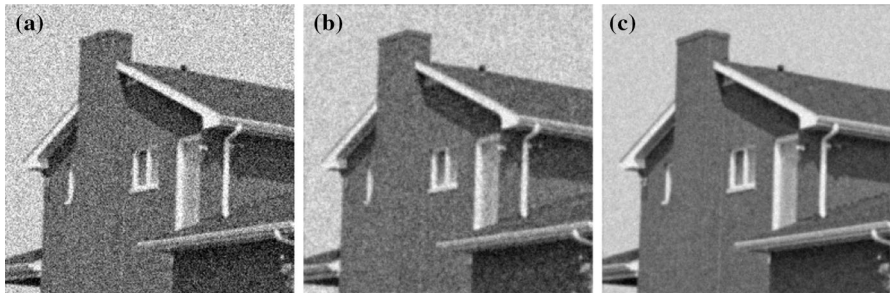


Fig. 6 Examples of the image denoising results for Fig. 3a. **a** Example of a noisy image when $\sigma = 30$ (peak signal-to-noise ratio (PSNR):18.61). **b** Denoised image obtained by the proposed method for $K = 1$ (PSNR:25.49). **c** Denoised image obtained by the proposed method for $K = 5$ (PSNR:30.33)

where MSE is the mean squared error between the original noiseless image and the inferred denoised image \hat{x} . Figure 5 confirms that the image denoising results improve as the value of K is increased. Example of image denoising results for Fig. 3a are shown in Fig. 6 for $K = 1$ and $K = 5$, respectively.

7 Concluding Remarks

In this paper, we defined a discrete MRF model for the Bayesian image denoising problem with multiple noisy images. We proposed a fast denoising algorithm for inferring a denoised image in $O(T_{EM}T_{BP}NL \log L)$ -time by using belief propagation and an EM algorithm based on our MRF model and FFT. We numerically verified the proposed denoising method using standard images. The results show that the proposed algorithm inferred the denoised image faster than previous implementation methods that use belief propagation.

We believe that the proposed method is the most fastest implementation of an image denoising algorithm based on a discrete MRF model that uses belief propagation and an EM algorithm. However, the method cannot yet be used for real-time processing. Therefore, we need to seek a further effective fast approximate method that preserves the restoration quality for the discrete MRF model. In our experiment, we adopted the quadratic function as the form of function $\phi(x)$. However, the proposed method is not restricted to the quadratic function: we can apply it to other types of the function $\phi(x)$, such as the Huber prior [15] and generalized sparse prior [16]. Moreover, because it can be used in any discrete MRF with $\phi(x_i - x_j)$ potential for $ij \in E$ interaction, it is expected that the proposed method is applicable to not only image denoising but also other inference problems such as sparse modeling [17, 18]. We intend to develop the method in these directions.

Acknowledgements This work was partially supported by JST CREST Grant no. JPMJCR1402 and JSPS KAKENHI Grant nos. 18K18120, 18H03303, 18K11459, and 15H03699.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741.
2. Tanaka, K. (2002). Statistical-mechanical approach to image processing. *Journal of Physics A: Mathematical and General*, 35(37), R81–R150.
3. Li, S. Z. (2009). *Markov random field modeling in image analysis*. Berlin: Springer.
4. Blake, A., Kohli, P., & Rother, C. (2011). *Markov random fields for vision and image processing*. Cambridge: MIT Press.
5. Lezoray, O., & Grady, L. (Eds.). (2012). *Image processing and analysis with graphs*. Boca Raton: CRC Press.
6. Kataoka, S., Yasuda, M., Furtlehner, C., & Tanaka, K. (2014). Traffic data reconstruction based on Markov random field modeling. *Inverse Problem*, 30(2), 025003.
7. Hara, Y., Suzuki, J., & Kuwahara, M. (2018). Network-wide traffic state estimation using a mixture Gaussian graphical model and graphical lasso. *Transportation Research Part C*, 86, 622–638.
8. Kuwatani, T., Nagata, K., Okada, M., & Toriumi, M. (2014). Markov random field modeling for mapping geofluid distributions from seismic velocity structures. *Earth, Planets and Space*, 66(5), 1–9.

9. Kuwatani, T., Nagata, K., Okada, M., & Toriumi, M. (2014). Markov-random-field modeling for linear seismic tomography. *Physical Review E*, *90*, 042137.
10. Yasuda, M., Watanabe, J., Kataoka, S., & Tanaka, K. (2018). Linear-time algorithm in Bayesian image denoising based on Gaussian Markov random field. In: *IEICE Transactions on Information and Systems*, vol. E101.D, no. 6, pp. 1629-1639.
11. Pearl, J. (1988). *Probabilistic reasoning in intelligent system: networks of plausible inference*. Burlington: Morgan Kaufmann.
12. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1–38.
13. Felzenszwalb, P. F., & Huttenlocher, D. P. (2006). Efficient belief propagation for early vision. *International Journal of Computer Vision*, *70*(1), 41–54.
14. Inoue, K. (2009). Kakuritsu denpan hou ni yoru gazou shori algorithm no kairyuu ni kansuru kenkyuu (Study on an improvement of the image processing using belief propagation), Master Thesis in Tohoku University (unpublished) (in Japanese).
15. Schulta, R. R., & Stevenson, R. L. (1994). A Bayesian approach to image expansion for improved definition. *IEEE Transactions on Image Processing*, *3*(3), 233–242.
16. Tanaka, K., Yasuda, M., & Titterington, D. M. (2012). Bayesian image modeling by means of generalized sparse prior and loopy belief propagation. *Journal of the Physical Society of Japan*, *81*(11), 114802.
17. Krzakala, F., Mézard, M., Sausset, F., Sun, Y., & Zdeborová, L. (2012). Probabilistic reconstruction in compressed sensing: Algorithms, phase diagrams, and threshold achieving matrices. *Journal of Statistical Mechanics: Theory and Experiment*, *2012*, P08009.
18. Elad, M. (2010). *Sparse and redundant representations: From theory to applications in signal and image processing*. Berlin: Springer.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.