Denoising via MCMC-based Lossy Compression

Shirin Jalali* and Tsachy Weissman†,

*Center for Mathematics of Information, CalTech, Pasadena, CA 91125

†Department of Electrical Engineering, Stanford University, Stanford, CA 94305

Abstract

It has been established in the literature, in various theoretical and asymptotic senses, that universal lossy compression followed by some simple post-processing results in universal denoising, for the setting of a stationary ergodic source corrupted by additive white noise. However, this interesting theoretical result has not yet been tested in practice in denoising simulated or real data. In this paper, we employ a recently developed MCMC-based universal lossy compressor to build a universal compression-based denoising algorithm. We show that applying this iterative lossy compression algorithm with appropriately chosen distortion measure and distortion level, followed by a simple de-randomization operation, results in a family of denoisers that compares favorably (both theoretically and in practice) with other MCMC-based schemes, and with the Discrete Universal Denoiser (DUDE).

Index Terms

Denoising, Compression-based denoising, Universal lossy compression, Markov chain Monte Carlo, Simulated annealing.

I. INTRODUCTION

Consider a finite-alphabet random process $\mathbf{X} = \{X_i\}_{i=1}^{\infty}$ corrupted by an additive white (i.e., i.i.d.) noise process $\mathbf{Z} = \{Z_i\}_{i=1}^{\infty}$. The receiver observes the noisy process $\mathbf{Y} = \{Y_i\}_{i=1}^{\infty}$, where

$$Y_i = X_i + Z_i,$$

and desires to recover the noise-free signal $\mathbf{X}$. For simplicity and concreteness, we assume that starting here and throughout the paper, the source, noise, and reconstruction alphabets are the M-ary alphabet, i.e., $\mathcal{X} = \hat{\mathcal{X}} = \mathcal{Z} = \{0, 1, \ldots, M-1\}$, and the noise is additive modulo-M. The idea and the approach can be extended to more general settings as well.

Let π denote the probability mass function (pmf) of the noise process $\mathbf{Z}$, i.e., $\mathrm{P}(Z_i = z) = \pi(z)$, for $z \in \mathcal{Z}$ and $i \in \mathbb{N}$, and assume that $\pi(z) > 0$, for each $z \in \mathcal{Z}$.

If $\mathcal{Y}$ and $\hat{\mathcal{X}}$ denote the alphabets of the received and the reconstruction signals, respectively, then, an n-block denoiser can be described by its block length n and denoising mapping

$$\theta_n : \mathcal{Y}^n \to \hat{\mathcal{X}}^n,$$

such that $\hat{X}^n = \theta_n(Y^n)$. Define the average distortion between source and reconstruction blocks x^n and $\hat{x}^n$ as

$$d_n(x^n, \hat{x}^n) \triangleq \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i), \tag{1}$$

where $d : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}^+$ is a per-letter distortion measure. The performance of an n-block denoiser θ_n is measured in terms of its expected average loss defined as

$$L_{\theta_n}(\mathbf{X}, \pi) \triangleq \mathrm{E}[d_n(X^n, \theta_n(Y^n))], \tag{2}$$

where the expectation is with respect to the randomness both in signal and in noise. Hence, the performance of a given denoiser depends on both the distribution of the signal and the distribution of the noise.

In the case where the denoiser has knowledge of the probability distributions of the source and noise processes $\mathbf{X}$ and $\mathbf{Z}$, an optimal denoiser is a Bayesian denoiser whose i^{th} reconstruction is given by

$$\hat{X}_i^{Bayes} = \hat{X}_i^{Bayes}(Y^n) \triangleq \arg\min_{\hat{x} \in \hat{\mathcal{X}}} \mathrm{E}[d(X_i, \hat{x}) | Y^n], \tag{3}$$

where $Y^n = X^n + Z^n$. Let $L^{opt,B}(\mathbf{X}, \pi)$ denote the asymptotic performance of the Bayesian denoiser defined by the set of denoiserz $\{\hat{X}_i^{Bayes}\}_{i=1}^n$ for stationary ergodic source $\mathbf{X}$ corrupted by an additive

white noise process $\mathbf{Z}$ distributed according to π. In other words,

$$L^{opt,B}(\mathbf{X},\pi) = \lim_{n\to\infty} \mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n} d(X_i, \hat{X}_i^{Bayes})\right], \tag{4}$$

where the limit exists by sub-additivity. In the case where $\mathbf{X}$ is Markov, the solution of (3) can be obtained efficiently via the backward-forward recursions [1], [2].

In many practical situations, assuming that the denoiser knows the source distribution is not realistic. However, it can be shown that the knowledge of the noise distribution π is enough for achieving $L^{opt,B}(\mathbf{X},\pi)$. In other words, there exists a family of n-block denoisers that achieves the optimal performance $L^{opt,B}(\mathbf{X},\pi)$, for any stationary ergodic $\mathbf{X}$, as long as the noise distribution π is known by the denoiser [3]. In fact, not only is this performance achievable for any stationary ergodic process $\mathbf{X}$, but it can be achieved practically with linear-complexity via the discrete universal denoising (DUDE) algorithm proposed in [3].

In DUDE, the denoiser first estimates the probability distribution of the source using its observed noisy version, and then it performs a Bayesian-type denoising operation using the estimated statistics. A different approach to denoising is based on lossy compression of the noisy signal. This method provides an alternative and more implicit method for learning the required statistics. After obtaining such statistics, the Bayesian estimation can be done similar to the DUDE. The intuition behind this approach is as follows. In universal lossy compression, to encode a sequence y^n, the encoder looks for a more compressible sequence $\hat{y}^n$, which is also 'close' to y^n. At a high level, lossy compression of y^n at distortion level D can be done by searching among all sequences $\hat{y}^n$ that are within radius D of y^n, i.e., $d_n(y^n,\hat{y}^n) \le D$, and choosing the one that has the lowest 'complexity' or 'description length'. On the other hand, adding noise to a signal, always increases its entropy, i.e., since $I(X^n+Z^n;Z^n) \ge 0$,

$$H(X^n+Z^n) - H(X^n+Z^n|Z^n) = H(X^n+Z^n) - H(X^n) \ge 0, \tag{5}$$

and therefore $H(Y^n) \ge H(X^n)$. Hence, in universal lossy compression of noisy sequence $Y^n = X^n + Z^n$, if the distortion level is set appropriately, a reasonable reconstruction sequence can be the original sequence X^n.

Minimum Kolmogorov Complexity Estimator (MKCE) proposed in [4] is constructed based on the idea described above. If X^n is a binary sequence, and $Y^n = X^n + Z^n$, where $\{Z_i\}$ is an i.i.d. sequence

and $Z_i \sim \mathrm{Bern}(\delta)$, the MKCE denoiser looks for $\hat{X}^n$ solving the problem

$$\begin{aligned} \min \quad & K(\hat{x}^n) \\ \text{subject to} \quad & \hat{x}^n \in \{0,1\}^n, \\ & d_n(Y^n, \hat{x}^n) \leq \delta. \end{aligned} \tag{6}$$

In (6), $K(\hat{x}^n)$ represents the Kolmogorov complexity of $\hat{x}^n$, which is the length of the shortest program that generates $\hat{x}^n$ and halts [5]. Basically, $K(\hat{x}^n)$ measures the complexity or compressibility of $\hat{x}^n$. It is shown in [4] that the performance of MKCE is strictly worse than an optimal denoiser, but by a factor no larger than 2. Later this result was refined in [6]. It was shown in [6] that replacing (6) with a universal lossy compressor, and then performing some post-processing operation results in a universal denoising algorithm with optimal performance. As explained before, the role of universal lossy compressor is helping the denoiser to estimate the distribution of the source. Using the estimated source statistics, the post-processing operation performs Bayesian denoising.

Compression-based denoising algorithms have been proposed before by a number of researchers. It has been studied both from a theoretical standpoint [4], [6] and from a practical point of view. (See [7]–[11] and references therein.) While the theoretical results, specially the work of [6], suggest that compression-based denoising is able to achieve the optimal performance, there is yet a gap between the theory and practice in this area. The implementable algorithms, while achieving promising results, are suboptimal, and the theoretical results have not yet led to practical compression-based denoising algorithms. In this paper, we show how combining the lossy compression algorithm proposed in [12] and the denoising approach of [6] leads to an implementable universal denoiser. Our simulation results show that the performance of the resulting scheme is comparable with the performance of the discrete universal denoising (DUDE) algorithm, when applied to one-dimensional or two-dimensional binary data, as proposed in [13].

The lossy compression algorithm proposed in [12] is based on Gibbs sampling, and simulated annealing [14]–[16]. Consider the probability distribution p over $\mathcal{X}^n$, such that for $x^n \in \mathcal{X}^n$, $p(x^n) \propto f(x^n)$, where $f(x^n) \geq 0$, for all x^n. In many applications, it is desirable to sample from such a distribution, which is only specified through another function f. Clearly, $p(x^n) = f(x^n)/Z$, where $Z = \sum_{x^n \in \mathcal{X}^n} f(x^n)$. However, since the size of the sample space grows exponentially with n, computing Z in general requires

exponential computational complexity. Therefore, to sample from p, one usually looks for sampling methods that do not directly require the computation of Z. Markov chain Monte Carlo (MCMC) methods are a class of algorithms that address this issue. They consist of a class of sampling algorithms that sample from distribution p by generating a Markov chain whose stationary distribution is p. Hence, running such a Markov chain, after it reaches the steady state, its state is a sample drawn from the distribution p. The Gibbs sampler, also known as the *heat bath* algorithm, is an instance of the MCMC methods. It is applied in the cases where the computational complexity of finding the conditional distributions of each variable given the rest, i.e., $p(x_i|x^{n\setminus i})$, is manageable.

Simulated annealing is a well-known method in discrete optimization problems. Its goal is to find the minimizer of a given function $h(x^n)$ over $\mathcal{X}^n$, i.e., $x^n_{\min} = \arg\min_{x^n\in\mathcal{X}^n} h(x^n)$. In order to perform simulated annealing, a sequence of probability distributions $\{p_i(x^n)\}_{i=1}^{\infty}$ corresponding to the temperatures $\{T_i\}_{i=1}^{\infty}$, such that $T_i \to 0$ as $i \to \infty$, is considered. At each time i, the algorithm runs one of the relevant MCMC methods in an attempt to sample from distribution $p_i \propto e^{-\beta_i f(\mathbf{s})}$, where $\beta_i = 1/T_i$. Note that as $T_i \to 0$, or $\beta_i \to \infty$, p_i converges to a probability distribution which is uniform over the set of minimizers of h, and zero otherwise. Hence, clearly a sample from this distribution gives a minimizer of h. On the other hand, starting from an extremely low temperature results in a Markov chain that requires exponential time to reach the steady state. Therefore, in simulated annealing, the algorithm starts from a moderate temperature, and decreases the temperature gradually. It can be proved that, if the temperature drops slowly enough, the probability of getting the minimizing state as the output of the algorithm approaches one [14].

The organization of this paper is as follows. Section II introduces the notation and definitions used in this paper. Section III reviews the universal lossy compression algorithm proposed in [12]. In Section IV, we show how the universal lossy compression algorithm of [12] can be employed to construct a universal denoising algorithm. Section V presents some experimental results. The paper is concluded in Section VI.

II. Notation and definitions

Calligraphic letters such as $\mathcal{X}$ and $\mathcal{Y}$ denote sets. The size of a set $\mathcal{X}$ is denoted by $|\mathcal{X}|$. Bold letters such as $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ and $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ represent n-tuples, where n is implied by the context . An n-tuple $\mathbf{X}$ or $\mathbf{x}$ of length n is also represented as X^n or x^n when we want to be

explicit about n. For $1 \leq i \leq j \leq n$, $x_i^j = (x_i, x_{i+1}, \ldots, x_j)$. For two vectors x^i and y^j, $x^i y^j$ denotes a vector of length $i+j$ formed by concatenating the two vector as $(x_1, \ldots, x_i, y_1, \ldots, y_j)$. Capital letters represent random variables and capital bold letters represent random vectors. For a random variable X, let $\mathcal{X}$ denote its alphabet set. For vectors $\mathbf{u}$ and $\mathbf{v}$ both of length n, let $\|\mathbf{u}-\mathbf{v}\|_1$ denote the ℓ_1 distance between $\mathbf{u}$ and $\mathbf{v}$ defined as $\|\mathbf{u}-\mathbf{v}\|_1 \triangleq \sum_{i=1}^n |u_i - v_i|$.

For $y^n \in \mathcal{Y}^n$, let the $|\mathcal{Y}| \times |\mathcal{Y}|^k$ matrix $\mathbf{m}(y^n)$ denote the $(k+1)^{\text{th}}$ order empirical distribution of y^n. Each column of matrix $\mathbf{m}(y^n)$ is indexed by a k-tuple $b^k \in \mathcal{Y}^k$, and each row is indexed by an element $\beta \in \mathcal{Y}$. The element in row β and column b^k of $\mathbf{m}(y^n)$ is defined as

$$m_{\beta,b^k}(y^n) \triangleq \frac{1}{n-k} \left| \left\{ k+1 \leq i \leq n : y_{i-k}^{i-1} = b^k, y_i = \beta \right\} \right|, \tag{7}$$

i.e., the fraction of occurrences of the $(k+1)$-tuple $b^k\beta$ along the sequence. Let $H_k(y^n)$ denote the conditional empirical entropy of order k induced by y^n, i.e.,

$$H_k(y^n) = \sum_{b^k \in \mathcal{Y}^k} \|\mathbf{m}_{\cdot,b^k}(y^n)\|_1 \mathcal{H}\left(\mathbf{m}_{\cdot,b^k}(y^n)\right), \tag{8}$$

where $\mathbf{m}_{\cdot,b^k}(y^n)$ denotes the column in $\mathbf{m}(y^n)$ corresponding to b^k, and for a vector $\mathbf{v} = (v_1, \ldots, v_\ell)$ with non-negative components, $\mathcal{H}(\mathbf{v})$ denotes the entropy of the random variable whose pmf is proportional to $\mathbf{v}$. Formally,

$$\mathcal{H}(\mathbf{v}) = \begin{cases} \sum_{i=1}^{\ell} \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{\|\mathbf{v}\|_1}{v_i} & \text{if } \mathbf{v} \neq (0, \ldots, 0) \\ 0 & \text{if } \mathbf{v} = (0, \ldots, 0), \end{cases} \tag{9}$$

where $0 \log(0) \triangleq 0$ by convention. Alternatively, $H_k(y^n) \triangleq H(U_{k+1}|U^k)$, where U^{k+1} is distributed according to the $(k+1)^{\text{th}}$ order empirical distribution induced by y^n, i.e., $\mathrm{P}(U^{k+1} = [b^k, \beta]) = m_{\beta,b^k}(y^n)$.

Consider lossy compression of a stationary ergodic source $\mathbf{X} = \{X_i\}_{i=1}^{\infty}$ by block coding. The encoder maps each source output block of length n, X^n, to a binary sequence $f_n(X^n)$, i.e.,

$$f_n : \mathcal{X}^n \to \{0,1\}^*,$$

where $\{0,1\}^*$ denotes the set of all finite-length binary sequences. The index $f_n(X^n)$ is then losslessly transmitted to the decoder. The decoder maps $f_n(X^n)$ into a reconstruction block $\hat{X}^n = g_n(f_n(X^n))$,

where

$$g_n : \{0,1\}^* \to \hat{\mathcal{X}}^n.$$

The performance of a lossy coding algorithm $\mathcal{C}_n = (f_n, g_n)$ with block length n is measured by its induced rate R_n and distortion D_n. Let $l(f_n(X^n))$ denote the length of the binary sequence assigned to sequence X^n. The rate R_n of code $\mathcal{C}_n$ is defined as the expected average number of bits per source symbol, i.e.,

$$R_n \triangleq \mathrm{E}\left[\frac{l(f_n(X^n))}{n}\right].$$

The distortion D_n induced by code $\mathcal{C}_n$ is defined as the average expected distortion between source and reconstruction blocks, i.e.,

$$D_n \triangleq \mathrm{E}[d_n(X^n, \hat{X}^n)], \tag{10}$$

where $d_n(x^n, \hat{x}^n)$ is defined according to (1), and, as before, $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ defines a single-letter distortion measure.

For any $D \geq 0$, and stationary ergodic process $\mathbf{X}$, the minimum achievable rate (cf. [5] for exact definition of achievability) is characterized as [17]–[19]

$$R(D, \mathbf{X}) = \lim_{n\to\infty} \min_{p(\hat{X}^n|X^n):\mathrm{E}[d_n(X^n,\hat{X}^n)]\leq D} \frac{1}{n} I(X^n; \hat{X}^n). \tag{11}$$

III. Universal lossy compression via MCMC

Lossy compression algorithms can be divided into three groups as: i) fixed-rate, ii) fixed-distortion, and iii) fixed-slope. A family of universal lossy compression algorithms $\mathcal{C}_n = (f_n, g_n)$, $n \geq 1$, is called fixed-rate, if, for every stationary ergodic source $\mathbf{X}$, $R_n \leq R$, for all n, and $\limsup_n D_n \leq D(R, \mathbf{X})$. Similarly, a family of codes is called fixed distortion, if, for every stationary ergodic source, $\mathbf{X}$, $D_n \leq D$, for all n, and $\limsup_n R_n \leq R(D, \mathbf{X})$. Finally, a family of codes is called fixed distortion, if, for every stationary ergodic source $\mathbf{X}$, $\limsup_n [R_n + \alpha D_n] = \min_D [R(D, \mathbf{X}) + \alpha D]$. For a given source $\mathbf{X}$ with rate distortion function $R(D)$, a fixed-slope universal lossy compression algorithm at slope α asymptotically achieves the point (D_α, R_α), which is the point on the rate-distortion curve where the slope, $\frac{\partial R(D)}{\partial D}$, is equal to $-\alpha$.

Now for a fixed slope $\alpha > 0$, consider the quantization mapping $\hat{x}^n : \mathcal{X}^n \to \hat{\mathcal{X}}^n$ defined as

$$\hat{x}^n = \arg\min_{y^n} \left[H_k(y^n) + \alpha d_n(x^n, y^n) \right]. \tag{12}$$

Finding the sequence $\hat{x}^n$ from (12), and losslessly conveying it to the decoder using a universal lossless compression algorithm such as the Lempel-Ziv (LZ) algorithm constitutes a lossy compression algorithm. It can be proved that the described scheme attains the optimum rate-distortion performance at the slope α, universally for any stationary ergodic process [12], [20] in a strong almost sure sense. In other words, for any stationary ergodic source $\mathbf{X}$,

$$\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n) + \alpha d_n(X^n, \hat{X}^n) \stackrel{n\to\infty}{\longrightarrow} \min_{D\geq 0} \left[R(D, \mathbf{X}) + \alpha D \right], \tag{13}$$

almost surely. In (19), $\ell_{\mathsf{LZ}}(\hat{X}^n)$ denotes the description length of $\hat{X}^n$ by the LZ algorithm [21], [22].

To find the minimizer of (12), one needs to search the space of all possible reconstruction sequences which is of size $|\mathcal{X}|^n$. Hence, although the described scheme is theoretically appealing, it is an impractical algorithm and its implementation requires an exhaustive search.

In [12], we showed how simulated annealing enables us to get close to the performance of the impractical exhaustive search coding algorithm. In the rest of this section, we briefly review the lossy compression algorithm proposed in [12] based on simulated annealing.

To each reconstruction sequence $y^n \in \hat{\mathcal{X}}^n$, assign the energy $\mathcal{E}(y^n) \triangleq n[H_k(y^n) + \alpha d_n(x^n, y^n)]$, and define the probability distribution p_β on the space of reconstruction sequences $\hat{\mathcal{X}}^n$ as

$$p_\beta(y^n) = \frac{1}{Z_\beta} e^{-\beta \mathcal{E}(y^n)}, \tag{14}$$

where β and Z_β denote the inverse temperature parameter and the normalization constant (partition function), respectively. Sampling from this distribution for some large β results in a sequence Y^n that with high probability its energy is very close to the minimum energy, i.e.,

$$H_k(Y^n) + \alpha d_n(x^n, Y^n) \approx \min_{y^n} \left[H_k(y^n) + \alpha d_n(x^n, y^n) \right]. \tag{15}$$

However, sampling from the distribution p_β for large values of β is a challenging task. A well-known approach to circumvent this difficulty is the idea of *simulated annealing*. The main idea in simulated annealing is to explore the search space for the state of minimum energy using a time-dependent random

walk. The random walk is designed such that the probability of moving from the current state to one of its *neighboring* states depends on the difference between their energies. To give the algorithm the ability to escape from local minima, the Markov chain allows leaving a state of lower energy to reach a state of higher energy. As time proceeds, the system freezes ($\beta \to \infty$), and the probability of having such energy-increasing jumps decreases to zero.

The lossy compression algorithm based on simulated annealing presented in [12] is described in Alg. 1. In Alg. 1, $\mathrm{P}(Y_i = \cdot | Y^{n\backslash i} = y^{n\backslash i})$ denotes the conditional probability of Y_i given $Y^{n\backslash i} \triangleq (Y_n : n \neq i)$ under p_{β_t}. For $a \in \hat{\mathcal{X}}$, $\mathrm{P}(Y_i = a | Y^{n\backslash i} = y^{n\backslash i}) = p_\beta(a|y^{n\backslash i})$ can be expressed as

$$\begin{aligned}
\mathrm{P}(Y_i = a | Y^{n\backslash i} = y^{n\backslash i}) &= \frac{p_\beta(y^{i-1}\ a\ y_{i+1}^n)}{\sum_{b\in\hat{\mathcal{X}}} p_\beta(y^{i-1}\ b\ y_{i+1}^n)} \\
&= \frac{e^{-\beta\mathcal{E}(y^{i-1}ay_{i+1}^n)}}{\sum_{b\in\hat{\mathcal{X}}} e^{-\beta\mathcal{E}(y^{i-1}by_{i+1}^n)}} \\
&= \frac{e^{-\beta(H_k(y^{i-1}ay_{i+1}^n)+\alpha d_n(x^n, y^{i-1}ay_{i+1}^n))}}{\sum_{b\in\hat{\mathcal{X}}} e^{-\beta(H_k(y^{i-1}by_{i+1}^n)+\alpha d_n(x^n, y^{i-1}by_{i+1}^n))}} \\
&= \frac{1}{\sum_{b\in\hat{\mathcal{X}}} e^{-\beta(\Delta H_k(a,b,y^{i-1},y_{i+1}^n)+\alpha\Delta d(a,b,x_i))}},
\end{aligned} \tag{16}$$

where

$$\Delta H_k(a, b, y^{i-1}, y_{i+1}^n) \triangleq H_k(y^{i-1}by_{i+1}^n) - H_k(y^{i-1}ay_{i+1}^n)$$

and

$$\Delta d(b, a, x_i) \triangleq d_n(x^n, y^{i-1}by_{i+1}^n) - d_n(x^n, y^{i-1}ay_{i+1}^n) = \frac{d(x_i, b) - d(x_i, a)}{n}.$$

Computing the conditional probability distributions described in (16) forms the main step of Alg. 1.

Algorithm 1 Generating the reconstruction sequence

Input: x^n, k, α, $\{\beta_t\}_t$, r
Output: a reconstruction sequence $\hat{x}^n$

1: $y^n \leftarrow x^n$
2: **for** $t = 1$ to r **do**
3: Draw an integer $i \in \{1, \ldots, n\}$ uniformly at random.
4: For each $b \in \hat{\mathcal{X}}$ compute $p_{\beta_t}(b|y^{n\backslash i})$ given in (16).
5: Update y^n by replacing its i^{th} component y_i by Z, where $Z \sim p_{\beta_t}(\cdot|y^{n\backslash i})$.
6: Update $\mathbf{m}(y^n)$ and $H_k(y^n)$.
7: **end for**
8: $\hat{x}^n \leftarrow y^n$

Note that

$$\Delta H_k(a,b,y^{i-1},y_{i+1}^n) = H_k(y^{i-1}by_{i+1}^n) - H_k(y^{i-1}ay_{i+1}^n)$$
$$\sum_{b^k\in\mathcal{Y}^k} \left[\|\mathbf{m}_{\cdot,b^k}(y^{i-1}by_{i+1}^n)\|_1 \mathcal{H}\left(\mathbf{m}_{\cdot,b^k}(y^{i-1}by_{i+1}^n)\right)\right.$$
$$\left. -\|\mathbf{m}_{\cdot,b^k}(y^{i-1}ay_{i+1}^n)\|_1 \mathcal{H}\left(\mathbf{m}_{\cdot,b^k}(y^{i-1}ay_{i+1}^n)\right)\right]. \tag{17}$$

On the other hand, changing the i^{th} element of $y^{i-1}by_{i+1}^n$ from b to a affects at most $2k+1$ columns of the matrix $\mathbf{m}(y^{i-1}by_{i+1}^n)$. In other words, at least $2^k - 2k - 1$ columns of $\mathbf{m}(y^{i-1}by_{i+1}^n)$ and $\mathbf{m}(y^{i-1}ay_{i+1}^n)$ are exactly the same. Hence, from (17), the number of operations required for computing $\Delta H_k(a,b,y^{i-1},y_{i+1}^n)$ is linear in k, and independent of n.

Let $\hat{X}_{\alpha,r}^n(X^n)$ denote the (random) outcome of Algorithm 1 when taking $k = k_n$ and $\boldsymbol{\beta} = \{\beta_t\}_t$ to be deterministic sequences satisfying $k_n = o(\log n)$ and $\beta_t = \frac{1}{T_0^{(n)}}\log(\lfloor\frac{t}{n}\rfloor + 1)$, for some $T_0^{(n)} > n\Delta$, where

$$\Delta = \max_i \max_{\left\{\begin{array}{c} u^{i-1}\in\hat{\mathcal{X}}^{i-1},\\ u_{i+1}^n\in\hat{\mathcal{X}}^{n-i},\\ a,b\in\hat{\mathcal{X}}\end{array}\right.} \left|\mathcal{E}(u^{i-1}au_{i+1}^n) - \mathcal{E}(u^{i-1}bu_{i+1}^n)\right|, \tag{18}$$

applied to the source sequence X^n as input.[1] By the previous discussion, for given n and k, the computational complexity of the algorithm at each iteration is independent of n and linear in k. It can be proved that this choice of parameters yields a universal lossy compression algorithm, i.e., for any stationary ergodic process $\mathbf{X}$,

$$\lim_{n\to\infty}\lim_{r\to\infty}\left[\frac{1}{n}\ell_{\mathsf{LZ}}\left(\hat{X}_{\alpha,r}^n(X^n)\right) + \alpha d_n(X^n,\hat{X}^n)\right]$$
$$= \min_{D\geq 0}\left[R(D,\mathbf{X}) + \alpha D\right], \tag{19}$$

almost surely.

[1]Here and throughout it is implicit that the randomness used in the algorithms is independent of the source, and the randomization variables used at each drawing are independent of each other.

IV. Denoising via MCMC-based lossy compression

In [6], it is shown how a universally optimal lossy coder tuned to the right distortion measure and distortion level combined with some simple "post-processing" results in a universally optimal denoiser. In what follows we first briefly review the compression-based denoiser described in [6], and then show how the lossy coder proposed in [12] can be used for performing the lossy compression part.

Define the *difference* distortion measure $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ as

$$\rho(x, \hat{x}) \triangleq \log \frac{1}{\pi(x - \hat{x})}, \tag{20}$$

for any $(x, \hat{x}) \in \mathcal{X} \times \mathcal{X}$. As a reminder, in (20), $\pi(z)$, $z \in \mathcal{Z}$, denotes the pmf of the noise. Also as before, $\rho_n(x^n, \hat{x}^n) = n^{-1} \sum_{i=1}^n \rho(x_i, \hat{x}_i)$.

Now consider a universal sequence of lossy compression codes $\mathcal{C}_n = (f_n, g_n)$ with distortion measure ρ and fixed distortion $H(Z)$, i.e., a sequence under which

$$\mathrm{E}[\rho_n(Y^n, g_n(f_n(Y^n)))] \leq H(Z),$$

and

$$\limsup_{n\to\infty} \mathrm{E}\left[\frac{l(f_n(Y^n))}{n}\right] \to R(H(Z), \mathbf{Y}),$$

for each stationary ergodic process $\mathbf{Y} = \{Y_i\}_{i=1}^{\infty}$.

As mentioned earlier, in the denoising scheme outlined in [6], first the denoiser compresses the noisy signal appropriately, and partially removes the additive noise through lossy compression. To achieve this goal we apply the described universal lossy compression code to the noisy signal Y^n to get $\hat{Y}^n = g_n(f_n(Y^n))$.

The next step is a simple "post-processing", which involves computing the joint empirical distribution between the noisy signal and its compressed version, and then constructing the final reconstruction based on this empirical distribution. For a given integer $m = 2m_o + 1 > 0$, the empirical joint distribution $\hat{p}^{(m)}_{[Y^n, \hat{Y}^n]}(y^m, \hat{y})$ of the noisy signal Y^n and its quantized version $\hat{Y}^n$ is defined as as follows. For $y^m \in \mathcal{Y}^m$ and $\hat{y} \in \hat{\mathcal{Y}}$, let

$$\hat{p}^{(m)}_{[Y^n, \hat{Y}^n]}(y^m, \hat{y}) \triangleq \frac{1}{n - m + 1} \left| \left\{ m_o + 1 \leq i \leq n - m_o : (Y_{i-m_o}^{i+m_o}, \hat{Y}_i) = (y^m, \hat{y}) \right\} \right|. \tag{21}$$

In other words, $\hat{p}^{(m)}_{[Y^n,\hat{Y}^n]}(y^m,\hat{y})$ counts the fraction of times we observe the block y^m in Y^n ($Y_{i-m_o}^{i+m_o}=y^m$), while the position in $\hat{Y}^n$ corresponding to the middle symbol y_{m_o+1} is equal to $\hat{y}$ ($\hat{Y}_i=\hat{y}$). After constructing these count vectors, the output of the denoiser is generated through the "post-processing" or "de-randomization" process as follows

$$\hat{X}_i=\arg\min_{\hat{x}\in\mathcal{X}}\sum_{x\in\mathcal{X}}\hat{p}^{(m)}_{[Y^n,\hat{Y}^n]}(Y_{i-m_o}^{i+m_o},x)d(\hat{x},x), \tag{22}$$

where $d(\cdot,\cdot)$ is the original loss function under which the performance of the denoiser is to be measured. The described denoiser is shown to be universally optimal [6], the argument being roughly as follows: The rate-distortion function of the noisy signal $\mathbf{Y}$ under the defined difference distortion measure satisfies the Shannon lower bound with equality. It is proved in [6] that for such sources, for a fixed $\ell>0$, the ℓ^{th} order empirical joint distribution between the source block and its quantization by a sequence of universal codes, i.e.,

$$\hat{p}^{(\ell)}_{[Y^n,\hat{Y}^n]}(y^\ell,\hat{y}^\ell)\triangleq\frac{1}{n-\ell+1}\left|\left\{1\le i\le n-\ell+1:(Y_i^{i+\ell-1},\hat{Y}_i^{i+\ell-1})=(y^\ell,\hat{y}^\ell)\right\}\right|, \tag{23}$$

converges to the unique joint distribution that achieves the minimum mutual information in the k^{th} order (informational) rate-distortion function of the source. In other words, $\hat{p}^{(\ell)}_{[Y^n,\hat{Y}^n]}\xrightarrow{d}q^{(\ell)}$, where

$$q^{(\ell)}=\arg\min_{q(y^\ell,\hat{y}^\ell):\mathrm{E}_q[d_\ell(Y^\ell,\hat{Y}^\ell)]\le D} I(Y^\ell;\hat{Y}^\ell)$$

It turns out that in quantizing the noisy signal at distortion level $H(Z)$ under the distortion measure defined in (20), $q^{(\ell)}$ is equal to the ℓ^{th} order joint distribution between the source and noisy signal [6]. Hence, the count vector $\hat{p}^{(m)}_{[Y^n,\hat{Y}^n]}(y^m,\hat{y})$ defined in (21) asymptotically converges to $p_{X_i|Y^n}$, which is what the optimal denoiser would base its decision on. After estimating $p_{X_i|Y^n}$, the post-processing step is just making the optimal Bayesian decision at each position.

The main ingredient of the described denoiser is a universal lossy compressor. Note that the MCMC-based lossy compressor described in Section III is applicable to any distortion measure. Hence, combining the MCMC-based lossy compressor and the described post-processing operation yields a universal denoiser for our additive white noise setting.

Let $-\alpha(\mathbf{Y},H(Z))<0$ denote the slope of the unique point on the rate distortion curve of the process $\mathbf{Y}=\{X_i+Z_i\}_{i=1}^{\infty}$ corresponding to the distortion level $D=H(Z)$, under the distortion measure defined

in (20). Furthermore, let $k = k_n$ and $\boldsymbol{\beta} = \{\beta_t\}_t$ be deterministic sequences satisfying $k_n = o(\log n)$ and $\beta_t = \frac{1}{T_0^{(n)}} \log(\lfloor \frac{t}{n} \rfloor + 1)$, for some $T_0^{(n)} > n\Delta$, where Δ is defined in (18). Combining the results from [6] and [12] yields the following theorem, which proves the asymptotic optimality of the proposed denoising scheme.

Theorem 1: For any stationary ergodic process $\mathbf{X}$,

$$\lim_{m\to\infty} \lim_{n\to\infty} \lim_{r\to\infty} \mathrm{E}[d_n(X^n, \hat{X}^n)] = L^{opt,B}(\mathbf{X}, \pi),$$

where $\hat{X}^n = \hat{X}^n(Y^n, \hat{Y}^n_{\alpha(\mathbf{Y},H(Z)),r}(Y^n))$ is generated by (21) and (22), and $\hat{Y}^n_{\alpha(\mathbf{Y},H(Z)),r}(Y^n)$ is the output of Alg. 1. Moreover, for each source destitution, there exists a deterministic sequence $\{m_n\}_n$, $m_n \to \infty$, such that

$$\lim_{n\to\infty} \lim_{r\to\infty} \mathrm{E}[d_n(X^n, \hat{X}^n)] = L^{opt,B}(\mathbf{X}, \pi).$$

Remark 1: The main problem is choosing the parameter α corresponding to the distortion level of interest, i.e., $\alpha(\mathbf{Y}, H(Z))$. To find the right slope, we run the quantization MCMC-based part of the algorithm independently from two different initial points α_1 and α_2. After convergence of the two runs we compute the average distortion between the noisy signal and its quantized versions. Then assuming a linear approximation, we find the value of α that would have resulted in the desired distortion, and then run the algorithm again from this starting point, and again computed the average distortion, and then find a better estimate of α from the observations so far. After a few repetitions of this process, we have a reasonable estimate of the desired α. Note that, for finding α, it is not necessary to work with the whole noisy signal, and one can consider only a long enough section of data first, estimate α from it, and then run the MCMC-based denoising algorithm on the whole noisy signal with the estimated parameter α. The outlined method for finding α is similar to what is done in [23] for finding an appropriate Lagrange multiplier.

Remark 2: Note that the deterministic sequence mentioned in Theorem 1 may depend on the source as well.

Discussion: Our proposed approach, MCMC coding and de-randomization, is an alternative not only to the DUDE, but also to MCMC-based denoising schemes that have been based on or inspired by the Geman brothers' work [14]. While algorithmically our approach has much of the flavor of previous MCMC-based

denoising approaches, ours has the merit of leading to a universal scheme, whereas the previous MCMC-based schemes guarantee, at best, convergence to a performance which is good according to the posterior distribution of the noise-free given the noisy data, but as would be induced by the rather arbitrary prior model placed on the data. In our case no assumptions, beyond ergodicity, about the distribution/model of the noise-free data are made, and optimum performance is guaranteed (in the appropriate limits).

V. Simulation results

In this section, we compare the performance of the proposed denoising algorithm to that of the discrete universal denoiser, DUDE, introduced in [3]. DUDE is a practical universal algorithm that asymptotically achieves the performance attainable by the best n-block denoiser for any stationary ergodic source. The setting of operation of DUDE is more general than what is described in the previous section, and in fact in DUDE the additive white noise can be replaced by any known discrete memoryless channel.

As the first example, consider a binary symmetric Markov source (BSMS) with transition probability $p = 0.2$. The source sequence X^n is corrupted by a binary discrete memoryless channel (DMC) with error probability δ. Figures 1 and 2 compare the performances of the DUDE with the new MCMC-based deposing algorithm, for the cases of $\delta = 0.1$ and $\delta = 0.05$, respectively. In each figure, we have plotted the average bit error rate (BER) of each algorithm over $N = 50$ simulations versus the transition probability p. Also, for the sake of comparison, we have added to each figure the performance of the forward-backward dynamic programming denoiser which achieves the optimum performance in recovering the described source from its noisy version, in the non-universal setup. In both figures the blocklength is $n = 10^4$, and the parameters of the MCMC compressors are chosen as follows: $\gamma = 0.75$, $\beta_t = (\frac{1}{\gamma})^{\lceil t/n \rceil}$, $r = 10n$, and $k = 7$.[2] It can be observed that in both figures, the performance of the proposed compression-based denoiser is very close to the performance of the DUDE algorithm.

In each case, the slope α is chosen such that the expected distortion between the noisy image and its quantized version using Alg. 1 with distortion measure

$$\rho(x, \hat{x}) = \begin{cases} -\log \delta, & \text{if } \ x \neq \hat{x} \\ -\log(1-\delta), & \text{if } \ x = \hat{x}, \end{cases} \tag{24}$$

[2]A discussion on the selection of these parameters is presented in [24].

is close to $H(Z)$. Note that

$$\mathrm{E}[\rho(X,\hat{X})] = -\mathrm{P}(X \neq \hat{X})\log\delta - \mathrm{P}(X = \hat{X})\log(1-\delta), \tag{25}$$

which is equal to $H(Z) = -\delta\log\delta - (1-\delta)\log(1-\delta)$ when $\mathrm{P}(X \neq \hat{X}) = \delta$. Hence, we require our MCMC lossy encoder to compress the noisy signal under the Hamming distortion at $D = \delta$. Fig. 3 shows the average Hamming distortion, i.e., BER, of the MCMC-based lossy compressor versus p, for the cases of $\delta = 0.05$ and $\delta = 0.1$. In both cases, the average distortion incurred by the lossy compressor is close to its desired value which is δ.

In another example, we consider denoising the 256×256 binary image shown in Fig. 5. Fig. 6 shows its noisy version which is generated by passing the original image through a binary DMC with error probability of 0.05, i.e., $\pi(1) = 1 - \pi(0) = 0.05$. Fig. VI shows the reconstructed image generated by DUDE and 8(b) depicts the reconstructed image using the described algorithm. Here, the number of pixels is $n = 256^2$. In this experiment the DUDE context structure is set as Fig. 4(c). The 2-D MCMC coder employs the context shown in Fig. 4(a), and the de-randomization block is chosen as Fig. 4(b). Note that while we require the context used in computing the conditional empirical entropy of the image to have a causal structure, i.e., only contain pixels located prior to the current pixel, for the de-randomization block we have no such constraint. While the former is used to measure the complexity of the signal, the latter is used for learning the joint distribution between the noisy signal and its quantized version.

The BERs of the DUDE and compression-based denoising algorithms are 8.17×10^{-3} and 7.59×10^{-3}, respectively. Though the performance of DUDE here is slightly better in terms of BER, the visual quality of the reconstruction is arguably better with the new denoiser, and the 'texture' of the original image seems to be better preserved with our reconstruction. This may be a result of the fact that the compression-based approach is guaranteed of recovering not only the marginal distributions of one noise-free symbol given the noisy data, as in the DUDE, but in fact that k-dimensional distributions, for every k.

VI. Conclusions

The idea of deriving a universal denoising algorithm based on a universal lossy compression scheme with some post-processing was proposed in [6]. However, this result has not yet been tested in practice. In this paper we employed the MCMC-based universal lossy compression algorithm proposed in [12] to derive a universal denoising algorithm. The algorithm first applies the MCMC-based lossy compression

algorithm to the noisy signal, using a distortion measure and level which are both functions of the channel probability distribution. Then it performs some simple post-processing operations on the compressed noisy signal. Our simulation results show that the performance of the resulting denoising algorithm is promising and comparable to the performance of the DUDE. This proves that in practical situations compression-based denoising algorithms can be quite effective.

REFERENCES

[1] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[2] R. Chang and J. Hancock. On receiver structures for channels having memory. *Information Theory, IEEE Transactions on*, 12(4):463 – 468, October 1966.

[3] T. Weissman, Erik Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger. Universal discrete denoising: Known channel. *IEEE Trans. Inform. Theory*, 51(1):5–28, 2005.

[4] D. Donoho. The kolmogorov sampler, Jan 2002.

[5] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, 2nd edition, 2006.

[6] T. Weissman and E. Ordentlich. The empirical distribution of rate-constrained source codes. *Information Theory, IEEE Transactions on*, 51(11):3718–3733, Nov 2005.

[7] B.K. Natarajan. Filtering random noise via data compression. In *Data Compression Conference, 1993. DCC '93.*, pages 60–69, 1993.

[8] B. Natarajan, K. Konstantinides, and C. Herley. Occam filters for stochastic sources with application to digital images. *Signal Processing, IEEE Transactions on*, 46(5):1434–1438, May 1998.

[9] J. Rissanen. Mdl denoising. *IEEE Trans. Inform. Theory*, 46(7):2537–2543, Nov. 2000.

[10] S. P. Awate and R. T. Whitaker. Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):364–376, 2006.

[11] S. de Rooij and P. Vitanyi. Approximating rate-distortion graphs of individual data: Experiments in lossy compression and denoising. *Computers, IEEE Transactions on*, (99), 2011.

[12] S. Jalali and T. Weissman. Rate-distortion via Markov chain Monte Carlo. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 852 –856, July 2008.

[13] E. Ordentlich, G. Seroussi, S. Verdú, M. Weinberger, and T. Weissman. A discrete universal denoiser and its application to binary images. In *Proc. IEEE Int. Conf. on Image Processing*, pages 117–120, Sep. 2003.

[14] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, Nov 1984.

[15] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[16] V. Cerny. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, Jan 1985.

[17] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423 and 623–656, 1948.

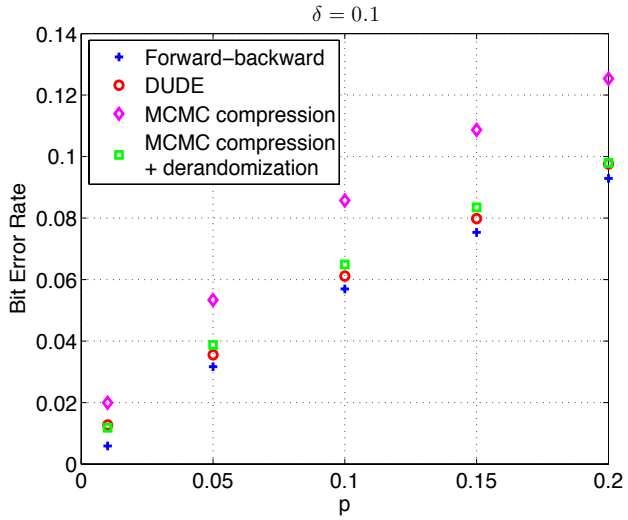


Fig. 1. Comparing the bit error rates of the denoiser based on MCMC coding plus de-randomization with DUDE and optimal non-universal Bayesian denoiser which is implemented via forward-backward dynamic programming. The source is a BSMS(p), and the channel is a DMC with error probability $\delta = 0.1$. The DUDE parameters are: $k_{\text{letf}} = k_{\text{right}} = 4$, and the MCMC compressor uses $\alpha = 0.95 : 0.3 : 2.15$, $\gamma = 0.75$, $\beta_t = (\frac{1}{\gamma})^{\lceil t/n \rceil}$, $r = 10n$, $n = 10^4$, and $k = 7$. The de-randomization window length is $2 \times 4 + 1 = 9$.

[18] R.G. Gallager. *Information Theory and Reliable Communication*. NY: John Wiley, 1968.

[19] T. Berger. *Rate-distortion theory: A mathematical basis for data compression*. NJ: Prentice-Hall, 1971.

[20] En hui Yang, Z. Zhang, and T. Berger. Fixed-slope universal lossy data compression. *Information Theory, IEEE Transactions on*, 43(5):1465–1476, Sep 1997.

[21] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *Information Theory, IEEE Transactions on*, 23(3):337–343, May 1977.

[22] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *Information Theory, IEEE Transactions on*, 24(5):530–536, Sep 1978.

[23] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *Image Processing, IEEE Transactions on*, 2(2):160–175, Apr 1993.

[24] S. Jalali and T. Weissman. Rate-distortion via Markov chain Monte Carlo. *arXiv:0808.4156v2*.

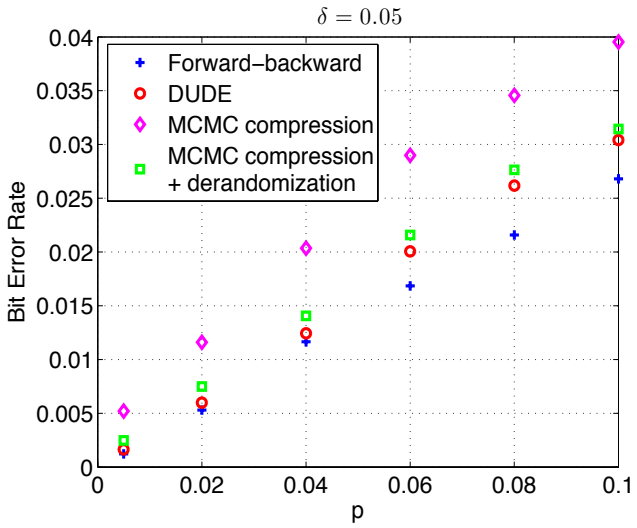
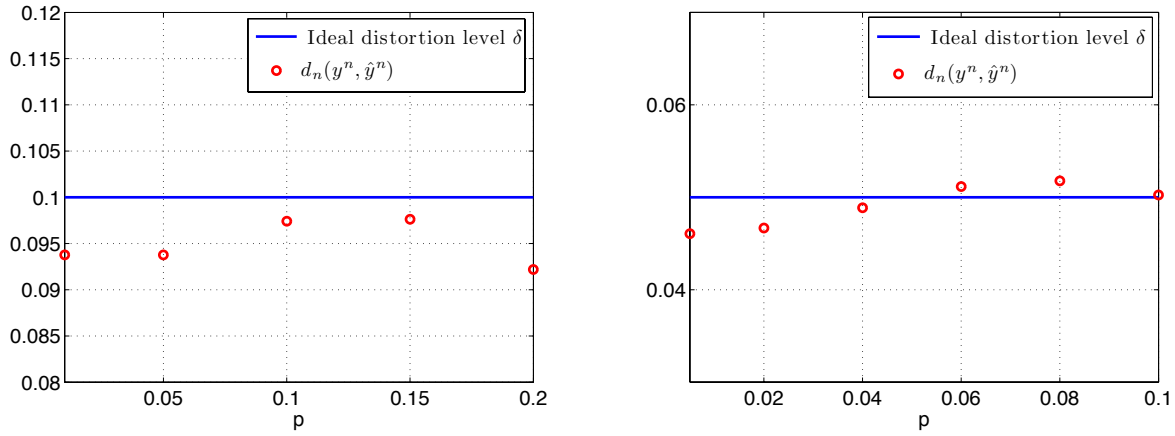
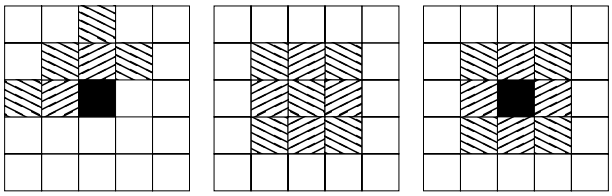


Fig. 2. Comparing the bit error rates of the denoiser based on MCMC coding plus de-randomization with the DUDE and the optimal non-universal Bayesian denoiser which is implemented via forward-backward dynamic programming. Here $\delta = 0.05$, and $\alpha = 0.9 : 0.3 : 2.4$. The rest of the parameters are identical to the setup of Fig. 1.



(a) $\delta = 0.1$ (b) $\delta = 0.05$

Fig. 3. Comparing the average BER performance of the MCMC-based lossy compressor applied to the noisy signal versus p with the optimal BER which is δ. The simulations setups here are those of Fig. 1 and Fig. 2, respectively.



(a) (b) (c)

Fig. 4. Contexts used by the MCMC compressor, DUDE and the derandomizer. (a) The 6^{th} order context used by the 2-D MCMC-based lossy compressor. (b) The de-randomization block used in MCMC-based denoising. (c) The 8^{th} order context used by DUDE.

Fig. 5. Panda image

Fig. 6. Panda image corrupted by a DMC with error probability $\delta = 0.05$

Fig. 7. The denoised image generated by the DUDE: $d_n(x^n, \hat{x}^n) = 7.97 \times 10^{-3}$.

(a) The denoised image generated by the MCMC compressor: $d_n(x^n, \hat{x}^n) = 1.01 \times 10^{-2}$. ($\alpha = 3.5$, $\beta_t = (\frac{1}{\gamma})^{\lceil t/n \rceil}$, $\gamma = 0.99$ and $r = 8n$.)

(b) The denoised image generated by the MCMC compressor plus de-randomization: $d_n(x^n, \hat{x}^n) = 8.11 \times 10^{-3}$.

Fig. 8. The MCMC-based denoiser applied to a binary image.