entropy

MDPI

Article

A Stochastic Model for Block Segmentation of Images Based on the Quadtree and the Bayes Code for It †

Yuta Nakahara [1,*] and Toshiyasu Matsushima [2]

[1] Center for Data Science, Waseda University, 1–6–1 Nisniwaseda, Shinjuku-ku, Tokyo 169-8050, Japan
[2] Department of Pure and Applied Mathematics, Waseda University, 3–4–1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan; toshimat@waseda.jp
* Correspondence: yuta.nakahara@aoni.waseda.jp
† This paper is an extended version of our paper published in 2020 Data Compression Conference (DCC), Snowbird, UT, USA, 24–27 March 2020; pp. 293–302.

Abstract: In information theory, lossless compression of general data is based on an explicit assumption of a stochastic generative model on target data. However, in lossless image compression, researchers have mainly focused on the coding procedure that outputs the coded sequence from the input image, and the assumption of the stochastic generative model is implicit. In these studies, there is a difficulty in discussing the difference between the expected code length and the entropy of the stochastic generative model. We solve this difficulty for a class of images, in which they have non-stationarity among segments. In this paper, we propose a novel stochastic generative model of images by redefining the implicit stochastic generative model in a previous coding procedure. Our model is based on the quadtree so that it effectively represents the variable block size segmentation of images. Then, we construct the Bayes code optimal for the proposed stochastic generative model. It requires the summation of all possible quadtrees weighted by their posterior. In general, its computational cost increases exponentially for the image size. However, we introduce an efficient algorithm to calculate it in the polynomial order of the image size without loss of optimality. As a result, the derived algorithm has a better average coding rate than that of JBIG.

Keywords: stochastic generative model; quadtree; bayes code; lossless image compression

Citation: Nakahara, Y.; Matsushima, T. A Stochastic Model for Block Segmentation of Images Based on the Quadtree and the Bayes Code for It. *Entropy* **2021**, *23*, 991. https://doi.org/10.3390/e23080991

Academic Editors: Jerry D. Gibson, Nithin Nagaraj and Kushal Shah

Received: 25 June 2021
Accepted: 28 July 2021
Published: 30 July 2021

1. Introduction

1.1. Lossless Data Compression in Information Theory

In information theory, lossless compression for general data (not only images) is based on an explicit assumption of a *stochastic generative model* $p(\boldsymbol{x})$ on target data $\boldsymbol{x}$ [1]. This assumption determines the theoretical limit, which is called entropy, of the expected code length for $p(\boldsymbol{x})$. When $p(\boldsymbol{x})$ is known, entropy codes such as Huffman code [2] and arithmetic code (see, e.g., [3]) achieve the theoretical limit. Then, researchers have considered a setup in which $p(\boldsymbol{x})$ is unknown. One method to describe the uncertainty of $p(\boldsymbol{x})$ is removing any specific assumption from the stochastic model $p(\boldsymbol{x})$ (e.g., [4,5]). Another is considering a class of parameterized stochastic generative models $p(\boldsymbol{x}|\boldsymbol{\theta})$ and assuming the class is known but the parameter $\boldsymbol{\theta}$ is unknown. We focus on the latter method in this paper. Even for this setup, researchers have proposed a variety of stochastic generative model classes and coding algorithms achieving those theoretical limits, e.g., i.i.d. model class, Markov model class, context tree model class, and so on (see, e.g., [6–10]).

In this setup, the variety of the stochastic generative model is described as that of unknown parameters or model variables. For example, the i.i.d. model can be determined by a vector $\boldsymbol{\theta}$ whose elements are occurrence probabilities of each symbol and described as $p(\boldsymbol{x}|\boldsymbol{\theta})$. Markov model contains another variable c that represents the state or context, which is a string of the most recent symbols at each time point, and the occurrence probability vector $\boldsymbol{\theta}_c$ is multiplied for each c. Then, the Markov model can be described as $p(\boldsymbol{x}|\boldsymbol{\theta}_c, c)$.

Further, when the order of the Markov model is unknown, that contains another variable k that represents the order and the occurrence probability $\boldsymbol{\theta}_c^k$, and the state variables c^k are multiplied for each k. Then, the Markov model with unknown order can be described as $p(\boldsymbol{x}|\boldsymbol{\theta}_c^k, c^k, k)$. Moreover, in the context tree model, the order depends on the context, and k is replaced by an unknown model variable m that represents a set of contexts. Finally, the context tree model can be described as $p(\boldsymbol{x}|\boldsymbol{\theta}_c^m, c^m, m)$.

It should be noted that these parameters and model variables $\boldsymbol{\theta}$, k, and m are the *statistical parameters* that govern the generation of the data $\boldsymbol{x}$. Therefore, the coding algorithm achieving the theoretical limit of these stochastic generative models inevitably contains some kind of statistically optimal action, e.g. their statistical estimation $\hat{\boldsymbol{\theta}}(\boldsymbol{x})$, $\hat{k}(\boldsymbol{x})$, $\hat{m}(\boldsymbol{x})$ as values, their estimation $p(\boldsymbol{\theta}|\boldsymbol{x})$, $p(k|\boldsymbol{x})$, $p(m|\boldsymbol{x})$ as posteriors in a Bayesian setting, or model weighting with their posteriors. The explicit assumption of the stochastic generative model and the construction of the coding algorithm with the statistically optimal action have been successful in text compression. In fact, various text coding algorithms have been derived (e.g., [8–10]).

1.2. Lossless Image Compression as a Image Processing

However, in most cases of lossless "image" compression, the main focus is on the construction of the *coding procedure* $f(\boldsymbol{x})$ that just outputs the coded sequence from the input pixel values $\boldsymbol{x}$ without the explicit assumption of a stochastic generative model. In the usual case, the coding algorithm has a *tuning parameter* a and is represented as $f(\boldsymbol{x}; a)$. This tuning parameter a is adaptively tuned to pixel values $\boldsymbol{x}$, and we express this tuning method as $\tilde{a}(\boldsymbol{x})$. Then, the coded sequence $f(\boldsymbol{x}; \tilde{a}(\boldsymbol{x}))$ from $\boldsymbol{x}$ is uniquely determined.

Therefore, the variety of the coding procedures is described as that of the tuning parameters and the tuning methods. More specifically, we give a brief review of a type of lossless image coding called predictive coding. Most of the predictive coding procedures have a form $f(x^{t-1}; a, b)$ with two parameters a and b. a is a parameter of the predictor, which predicts the next pixel value x_t from the already compressed pixels x^{t-1} at time t. b is a parameter that determines an assignment of the code length to the predictive error sequence. Note that the assignment of the code length can be represented by a vector whose sum of the elements equals 1, and it is sometimes called "probability". However, it does not represent the occurrence probability of pixel value x_t in an explicitly assumed stochastic generative model. Therefore, in this paper, we call it *code length assign vector* to distinguish them. Then, the predictive error sequence and the code length assign vector are input to the entropy codes such as the arithmetic code [3]. For example, in JPEG-LS [11], they use three predictors that are switched according to the neighboring pixels. This can be regarded as $a \in \{1, 2, 3\}$ corresponding to the index of the three predictors, and the rule to switch them is represented by $\tilde{a}(x^{t-1})$. The code length assign vector of JPEG-LS [11] is represented by a two-sided geometric distribution, which is tuned by the past sequence x^{t-1}. This can be regarded as b being a parameter of the two-sided geometric distribution and $\tilde{b}(x^{t-1})$ being its tuning method. In other studies [12–17], the authors proposed coding procedures $f(x^{t-1}; a, b, \boldsymbol{c}_a)$ in which coefficients $\boldsymbol{c}_a$ of each linear predictor are tuned by a certain method $\tilde{\boldsymbol{c}}_a(x^{t-1})$, e.g., the least squares method or weighted least squares method. In [18,19], the authors proposed coding procedures $f(x^{t-1}; a, b, \boldsymbol{c}_a, \boldsymbol{w})$ in which multiple predictors are combined according to another tuning parameter $\boldsymbol{w}$ that represents the weights of each predictor. Regarding the code length assign vector, Matsuda et al. [20] dealt with a procedure $f(x^{t-1}; a, b, \boldsymbol{c}_a, d)$ in which the code length assign vector is represented by the generalized Gauss distribution that has another tuning parameter d. (This notation is just for the explanation of the idea of the previous studies; it does not completely match the notation of each paper, and it does not contain all of the tuning parameters of each procedure.) One of the latest studies constructing a complicated coding procedure was reported by Ulacha et al. [21], in which numerous tuning parameters are tuned through careful experiments. Lossless image compression using deep learning (see, e.g., [22]) can

be regarded as one of the coding procedures with a huge number of tuning parameters that are pre-trained.

These studies have been practically successful. However, it should be noted that the tuning parameters a and b are not the statistical parameters that govern the generation of pixel values $\boldsymbol{x}$ since they are introduced just to add a degree of freedom to the coding procedure. Even the parameter b, which superficially appears to be a parameter of a probability distribution, does not directly govern the generation of pixel values $\boldsymbol{x}$ unless the coding procedure is extremely simple; it is just used to represent the code length assign vector with fewer variables. Therefore, the tuning of these parameters adaptive to $\boldsymbol{x}$ is not theoretically grounded by the statistics nor information theory. If our task were not lossless compression, e.g., lossy compression, image super-resolution, and so on, this parameter tuning would be evaluated from various points of view, e.g., subjective evaluation. It is because such tasks have difficulty in the performance measure itself. Besides, in lossless image compression, it should be evaluated from an information-theoretical perspective. These parameters should be tuned to decrease the difference between the expected code length and the entropy of the assumed stochastic generative model, and we have to say any other tuning methods are heuristic unless they pursue the added value except for the coding rate. However, such an information-theoretical evaluation is impossible because there is no explicit assumption of the stochastic generative model $p(\boldsymbol{x})$, and the entropy—the theoretical limit of the expected code length—itself is not defined. This is a critical problem of the previous studies above. In addition, the more tuning parameters are introduced, the more difficult the construction of the tuning method becomes since there is no confirmation of the optimality of each tuning method.

1.3. Lossless Image Compression on an Explicitly Redefined the Stochastic Generative Model

However, there are some coding procedures $f(\boldsymbol{x}; a)$ [11–14,16–20,23] whose tuning parameter a can be regarded as a statistical parameter of an implicitly assumed statistical generative model $p(\boldsymbol{x}|a)$ by changing the viewpoint. (In some of these studies, the assumption of the stochastic generative model is claimed, but the distinction between the stochastic generative model and the code length assign vector is ambiguous, and the discussion about the difference between the expected code length and the entropy of the stochastic generative model is insufficient.) Further, its parameter tuning method $\tilde{a}(\boldsymbol{x})$ could be regarded as a heuristic approximation of a statistically optimal estimation $\hat{a}(\boldsymbol{x}) \approx \tilde{a}(\boldsymbol{x})$. Then, explicitly redefining the implicit stochastic generative model behind the previous coding procedures, we can construct a statistical generative model supported by their practical achievements. Moreover, if we derive the coding algorithm that minimizes the difference between the expected code length and the entropy of the constructed stochastic generative model under some kind of criterion, this algorithm inevitably contains a statistically optimal action that is an improved version of $\tilde{a}(\boldsymbol{x})$. Further, such an action is not necessarily the estimation $\hat{a}(\boldsymbol{x})$ as a value. We can also estimate its posterior $p(a|\boldsymbol{x})$ or mix the coding procedures weighted by the posterior $p(a|\boldsymbol{x})$.

To derive such a coding algorithm, we can utilize the coding algorithms in text coding. Although image data are different from the text data, their stochastic generative models may contain a similar structure, and we may utilize the estimating algorithm in the text coding. In fact, we utilize the efficient algorithm for the context tree model class [8–10] for our stochastic generative model in this paper.

It is true that the coding algorithm constructed in this approach does not necessarily work for real images, since the optimality is guaranteed only for the stochastic generative model, and it is difficult to prove that the real images generated from the assumed stochastic generative model. Therefore, the constructed coding algorithm might be inferior to the existing one in the initial stage of this approach. However, we claim that this problem should not be solved by a heuristic tuning of the parameter in the coding procedure but an explicit extension of the stochastic generative model, as much as possible. Such parameter tuning should be done in the final stage before implementation or standardization.

We already adopted this approach in the previous studies [24,25]. In [24], we proposed a two-dimensional autoregressive model and the optimal coding algorithm by interpreting the basic procedure [11–13,16,23] of the predictive coding as a stochastic generative model. In [25], we proposed a two-dimensional autoregressive hidden Markov model by interpreting the predictor weighting procedure around a diagonal edge [18] as a stochastic generative model. However, these stochastic generative models do not have enough flexibility to represent the non-stationarity among segments of an image. Therefore, we proposed a stochastic generative model for the non-stationarity in [26]. This paper is an extended version of it.

1.4. The Contribution of This Paper

Then, our target data are the images in which the properties of pixel values are different depending on the segments. In this paper, we achieve the following purposes.

1. We propose a stochastic generative model that effectively represents the non-stationarity among the segments in an image.
2. We derive the optimal code that minimizes the difference between the expected code length and the entropy of the proposed stochastic model under the Bayes criterion.
3. We derive an efficient algorithm for the implementation of the code without loss of the optimality.

A trivial way to represent the non-stationarity as a stochastic generative model is to divide the image into fixed-size blocks and assume different probability distributions for each block. However, such a stochastic generative model is not flexible enough to represent the smaller segments and inefficient to represent the larger segments than the block size.

On the other hand, one of the most efficient lossless image coding procedures [20] contains preprocessing to determine a quadtree that represents a variable block size segmentation. Then, different predictors are assigned to each block to mitigate the non-stationarity. The quadtree is also used in various fields of image and video processing to represent the variable block size segmentation, and its flexibility and computational efficiency are reported by a number of studies, e.g., in H.265 [27]. However, the quadtree in these studies is a tuning parameter of a procedure. There are no studies that regard the quadtree as a statistical model variable m of a stochastic generative model $p(\boldsymbol{x}|m)$ governing the generation of pixel values $\boldsymbol{x}$ and construct the optimal code that minimizes the difference between the expected code length and the entropy of it in the Bayes criterion, to the best of our knowledge.

In this paper, we propose a novel stochastic generative model based on the quadtree, so that our model effectively represents the non-stationarity among segments by the variable block size segmentation. Then, we construct the optimal code that minimizes the difference between the expected code length and the entropy of the proposed stochastic generative model under the Bayes criterion. The optimal code is given by a weighted sum of all the possible model quadtrees m, and the optimal weight is given by its posterior $p(m|\boldsymbol{x})$. In general, its computational cost increases exponentially for the image size. However, we introduce a computationally efficient algorithm to implement our code without loss of optimality, taking in the knowledge of the text coding [8–10]. A similar algorithm is also used for decision tree weighting in machine learning [28]. It is in contrast to the previous lossless image coding procedure [20] that fixes a single quadtree in the preprocessing, which statistically corresponds to some kind of model selection.

Although the main theme of this paper is lossless image compression, the substantial contribution of our results is the construction of the stochastic model. Therefore, the proposed stochastic model contributes to not only lossless image compression but also any other stochastic image processing such as recognition, generation, feature extraction, and so on.

The organization of this paper is as follows. In Section 2, we describe the proposed stochastic generative model. In Section 3, we derive the optimal code for the proposed model. In Section 4, we derive an efficient algorithm to implement the derived code.

In Section 5, we perform some experiments to confirm the flexibility of our stochastic generative model and the efficiency of our algorithm. In Section 6, we describe future works. Section 7 is the conclusion of this paper.

2. The Proposed Stochastic Model

At first, we define some notations. Note that the following notations are independent of those in Section 1. Let $\mathcal{V}$ denote a set of possible values of a pixel. For example, $\mathcal{V} = \{0, 1\}$ for binary images, $\mathcal{V} = \{0, 1, \ldots, 255\}$ for gray scale images, and $\mathcal{V} = \{0, 1, \ldots, 255\}^3$ for color images. Let $\mathbb{N}$ denote the set of natural numbers. Let $h \in \mathbb{N}$ and $w \in \mathbb{N}$ denote the height and width of an image, respectively. Although our model is able to represent any rectangular images and its block segmentation, we assume that $h = w = 2^{d_{\max}}$ for $d_{\max} \in \mathbb{N}$ in the following for the simplicity of the notation. Then, let V_t denote the random variable of the tth pixel value in order of the raster scan and $v_t \in \mathcal{V}$ denote its realized value. Note that V_t is at $x(t)$th row and $y(t)$th column, where t divided by w is $x(t)$ with a reminder of $y(t)$. In addition, let V^t denote the sequence of pixel values $V_0, V_1, \ldots, V_t$. Note that all the indices start from zero in this paper.

We consider the pixel value V_t is generated from various probability distributions depending on a model $m \in \mathcal{M}$ and parameters $\boldsymbol{\theta}^m \in \boldsymbol{\Theta}^m$. Therefore, they are represented by $p(v_t | v^{t-1}, \boldsymbol{\theta}^m, m)$ in general. Note that the model m and the parameters $\boldsymbol{\theta}^m$ are unobservable and should be estimated in actual situations. The definitions of m and $\boldsymbol{\theta}^m$ are as follows.

Definition 1. *Let $s_{(x_1y_1)(x_2y_2)\cdots(x_dy_d)}$ denote the following index set called "block"*

$$s_{(x_1y_1)(x_2y_2)\cdots(x_dy_d)} := \left\{ (i,j) \in \mathbb{Z}^2 \;\middle|\; \sum_{d'=1}^{d} \frac{x_{d'}}{2^{d'}} \le \frac{i}{2^{d_{\max}}} < \left(\sum_{d'=1}^{d} \frac{x_{d'}}{2^{d'}} + \frac{1}{2^d} \right), \sum_{d'=1}^{d} \frac{y_{d'}}{2^{d'}} \le \frac{j}{2^{d_{\max}}} < \left(\sum_{d'=1}^{d} \frac{y_{d'}}{2^{d'}} + \frac{1}{2^d} \right) \right\}, \tag{1}$$

where $x_{d'}, y_{d'} \in \{0, 1\}$, $d \le d_{\max}$, and $\mathbb{Z}$ denotes the set of integers. In addition, let s_λ be the set of whole indices $s_\lambda := \{0, 1, \ldots h-1\} \times \{0, 1, \ldots, w-1\}$. Then, let $\mathcal{S}$ denote the set which consists of all the above index sets, namely

$$\mathcal{S} := \{s_\lambda, s_{(00)}, \ldots, s_{(11)}, s_{(00)(00)}, \ldots, s_{(11)(11)}, \ldots, s_{(11)(11)\cdots(11)}\}. \tag{2}$$

Example 1. *For $d_{\max} = 2$,*

$$s_{(01)} = \{(i,j) \in \mathbb{Z}^2 \mid 0 \le i < 2, 2 \le j < 4\} = \{(0,2), (0,3), (1,2), (1,3)\}. \tag{3}$$

Therefore, it represents the indices of the upper right region. In a similar manner, $s_{(01)(11)} = \{(i,j) \in \mathbb{Z}^2 \mid 1 \le i < 2, 3 \le j < 4\} = \{(1,3)\}$. It should be noted that the cardinality $|s|$ for each $s \in \mathcal{S}$ represents the number of pixels in the block.

Definition 2. *We define the model m as a full quadtree whose nodes are elements of $\mathcal{S}$. Let $\mathcal{L}^m \subset \mathcal{S}$ and $\mathcal{I}^m \subset \mathcal{S}$ denote the set of the leaf nodes and the inner nodes of m, respectively. Then, $\mathcal{L}^m$ corresponds to a pattern of variable block size segmentation, as shown in Figure 1. Let $\mathcal{M}$ denote the set of full (i.e., every inner node has exactly four child nodes) quadtrees whose depth is smaller than or equal to $d_{\max}$.*

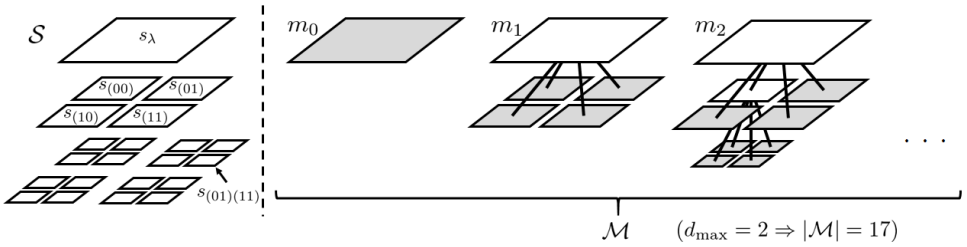

Figure 1. An example of node set $\mathcal{S}$ and models m.

Definition 3. *Each leaf node $s \in \mathcal{L}^m$ of the model m has a parameter θ_s^m whose parameter space is Θ_s^m. We define $\boldsymbol{\theta}^m$ as a tuple of parameters $\{\theta_s^m\}_{s\in\mathcal{L}^m}$, and let $\boldsymbol{\Theta}^m$ denote the total parameter space of them.*

Under the model $m \in \mathcal{M}$ and the parameters $\boldsymbol{\theta}^m \in \boldsymbol{\Theta}^m$, we assume that the tth pixel value $v_t \in \mathcal{V}$ is generated as follows.

Assumption 1. *We assume that*

$$p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m) = p(v_t|v^{t-1}, \theta_s^m), \tag{4}$$

where $s \in \mathcal{L}^m$ satisfies $(x(t), y(t)) \in s$.

Thus, the pixel value V_t depends only on the parameter of the block s which contains V_t under the past sequence V^{t-1}.

3. The Bayes Code for the Proposed Model

If we know the true model m and the parameters $\boldsymbol{\theta}^m$, we are able to compress the pixel value v_t up to the entropy of $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ by a well-known entropy code such as the arithmetic code. However, the true m and $\boldsymbol{\theta}^m$ are unobservable. One reasonable solution is to estimate them and substitute the estimated ones $\hat{m}$ and $\hat{\boldsymbol{\theta}}^m$ into $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$. Then, we can use $p(v_t|v^{t-1}, \hat{\boldsymbol{\theta}}^m, \hat{m})$ as a coding probability of the entropy code.

However, there is another powerful solution, in which we assume prior distributions $p(m)$ and $p(\boldsymbol{\theta}^m|m)$. Then, we estimate the true coding probability $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ itself instead of m and $\boldsymbol{\theta}^m$ by $q(v_t|v^{t-1})$ so that $q(v_t|v^{t-1})$ can minimize the *Bayes risk function* based on the *loss function* between the expected code length of entropy code using $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ and that using $q(v_t|v^{t-1})$. The code constructed by such a method is called the Bayes code (see, e.g., [29,30]).

It is known that the expected code length of the Bayes code converges to the entropy of the true stochastic model for sufficiently large data length t, and its convergence speed achieves the theoretical limits [30]. In fact, the Bayes code achieves remarkable performances in text compression (e.g., [8]).

Therefore, we derive the Bayes code for the proposed stochastic model. According to the general formula in [29], the optimal coding probability for v_t in the scheme of the Bayes code is derived as follows:

Proposition 1. *The optimal coding probability $q^*(v_t|v^{t-1})$ which minimizes the Bayes risk function is*

$$q^*(v_t|v^{t-1}) = p(v_t|v^{t-1}) = \sum_{m\in\mathcal{M}} p(m|v^{t-1}) \int p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m) p(\boldsymbol{\theta}^m|v^{t-1}, m) \mathrm{d}\boldsymbol{\theta}^m. \tag{5}$$

We call $q^*(v_t|v^{t-1})$ the Bayes optimal coding probability.

Proposition 1 implies that we should calculate the posterior distributions $p(m|v^{t-1})$ and $p(\boldsymbol{\theta}^m|v^{t-1}, m)$. Then, we should use the coding probability which is a weighted mixture of $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ for every block segmentation pattern m and parameters $\boldsymbol{\theta}^m$ according to the posteriors $p(m|v^{t-1})$ and $p(\boldsymbol{\theta}^m|v^{t-1}, m)$.

4. The Efficient Algorithm to Calculate the Coding Probability

Unfortunately, the Bayes optimal coding probability (5) contains computationally difficult calculations. As the depth $d_{\max}$ of full quadtree increases, the amount of calculation for the sum with respect to $m \in \mathcal{M}$ increases exponentially. Moreover, the posterior $p(m|v^{t-1})$ does not have a closed-form expression in general. (Strictly speaking, a few problems are also left. Both the integral with respect to $\boldsymbol{\theta}^m$ and the posterior $p(\boldsymbol{\theta}^m|m, v^{t-1})$ do not have closed-form expressions in general. These problems can be solved in various methods depending on the setting of $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ and $p(\boldsymbol{\theta}^m|m)$ and almost independent of our proposed model. Therefore, we describe an example of a feasible setting of $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ and $p(\boldsymbol{\theta}^m|m)$ in the next section. Other settings are described in Section 6 as future works.)

Similar problems are studied in text compression, and efficient algorithms to calculate the coding probability have been constructed (see, e.g., [8–10]). In these algorithms, the weighted sum of the context trees is calculated instead of the quadtrees. We apply it for our proposed model. In this section, we focus to describe the procedure of the constructed algorithm. Its validity is described in Appendix A.

First, we assume the following priors on m and $\boldsymbol{\theta}^m$.

Assumption 2. *We assume that each node $s \in \mathcal{S}$ has a hyperparameter $g_s \in [0, 1]$, and the model prior $p(m)$ is represented by*

$$p(m) = \prod_{s \in \mathcal{L}^m} (1 - g_s) \prod_{s' \in \mathcal{I}^m} g_{s'}, \tag{6}$$

where $g_s = 0$ for s whose cardinality $|s|$ equals 1, and the empty product equals 1.

The idea of this form is to represent $p(m)$ as a product of the probability that the block s is divided. Such a probability is denoted by g_s in (6). Note that $|s| = 1$ means that the block s consists of only 1 pixel and it cannot be divided. A proof that the above prior satisfies the condition $\sum_{m \in \mathcal{M}} p(m) = 1$ is in Appendix A. Note that the above assumption does not restrict the expressive capability of the general prior in the meaning that each model m still has possibility to be assigned a non-zero probability $p(m) > 0$.

Assumption 3. *For each model $m \in \mathcal{M}$, we assume that*

$$p(\boldsymbol{\theta}^m|m) = \prod_{s \in \mathcal{L}^m} p(\theta^m_s|m). \tag{7}$$

Moreover, for any $m, m' \in \mathcal{M}$, $s \in \mathcal{L}^m \cap \mathcal{L}^{m'}$, and $\theta_s \in \Theta_s$, we assume that

$$p(\theta_s|m) = p(\theta_s|m') =: p_s(\theta_s). \tag{8}$$

Therefore, each element θ^m_s of the parameters $\boldsymbol{\theta}^m$ depends only on s and is independent of both the other elements and the model m.

From Assumptions 1 and 3, the following lemma holds.

Lemma 1. *For any $m, m' \in \mathcal{M}$, $s \in \mathcal{L}^m \cap \mathcal{L}^{m'}$, and $v^t \in \mathcal{V}^t$, if $(x(t), y(t)) \in s$, then*

$$p(v_t|v^{t-1}, m) = p(v_t|v^{t-1}, m'). \tag{9}$$

Then, we represent it by $\tilde{q}(v_t|v^{t-1}, s)$ because it does not depend on m but s.

The proof of Lemma 1 is in Appendix A. Lemma 1 means that the optimal coding probability for v_t depends only on the leaf node block s which contains v_t, and it can be calculated as $q(v_t|v^{t-1}, s)$ if s is known.

At last, the efficient algorithm to compute the Bayes optimal coding probability $q^*(v_t|v^{t-1})$ is represented as an iteration of updating g_s and summing the functions $\tilde{q}(v_t|v^{t-1}, s)$ weighted by g_s for nodes on a path of the complete quadtree on $\mathcal{S}$.

Definition 4. *Let $\mathcal{S}_t$ denote the set of nodes which contain $(x(t), y(t))$. They construct a path from the leaf node $s_{(x_1y_1)(x_2y_2)\cdots(x_{d_{\max}}y_{d_{\max}})} = \{(x(t), y(t))\}$ to the root node s_λ on the complete quadtree whose depth is $d_{\max}$ on $\mathcal{S}$, as shown in Figure 2. In addition, let $s_{\text{child}} \in \mathcal{S}_t$ denote the child node of $s \in \mathcal{S}_t$ on that path.*

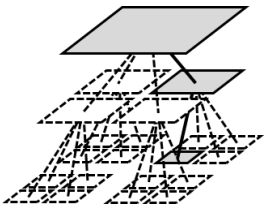

Figure 2. An example of a path constructed from $\mathcal{S}_t$.

Definition 5. *We define the following recursive function $q(v_t|v^{t-1}, s)$ for $s \in \mathcal{S}_t$.*

$$q(v_t|v^{t-1}, s) := \begin{cases} \tilde{q}(v_t|v^{t-1}, s), & |s| = 1, \\ (1 - g_{s|t-1})\tilde{q}(v_t|v^{t-1}, s) + g_{s|t-1}q(v_t|v^{t-1}, s_{\text{child}}), & \text{otherwise}, \end{cases} \tag{10}$$

where $g_{s|t}$ is also recursively updated as follows.

$$g_{s|t} := \begin{cases} g_s, & t = -1 \\ g_{s|t-1}, & t \geq 0 \wedge (s \notin \mathcal{S}_t \vee |s| = 1) \\ \frac{g_{s|t-1}q(v_t|v^{t-1}, s_{\text{child}})}{q(v_t|v^{t-1}, s)}, & t \geq 0 \wedge s \in \mathcal{S}_t \wedge |s| > 1. \end{cases} \tag{11}$$

Then, the following theorem holds.

Theorem 1. *The Bayes optimal coding probability $q^*(v_t|v^{t-1})$ for the proposed model is calculated by*

$$q^*(v_t|v^{t-1}) = q(v_t|v^{t-1}, s_\lambda). \tag{12}$$

The proof of Theorem 1 is in Appendix A. Theorem 1 means that the summation with respect to $m \in \mathcal{M}$ in (5) is able to be replaced by the summation with respect to $s \in \mathcal{S}_t$ and it costs only $O(d_{\max})$. In a sense, $(1 - g_{s|t-1})$ can be regarded as the marginal posterior probability that the true block division was stopped at s. Then, the proposed algorithm takes a mixture of the coding probability $\tilde{q}(v_t|v^{t-1}, s)$, weighting such a case with $(1 - g_{s|t-1})$ and the other cases with $g_{s|t-1}$.

5. Experiments

We performed three experiments. The purpose of the first experiment was to confirm the Bayes optimality of $q(v_t|v^{t-1}, s_\lambda)$. Therefore, we used synthetic images randomly generated from the proposed model. The purpose of the second experiment was to demonstrate the flexibility of our model. Therefore, we used a well-known benchmark image. We also used the Bayes optimal code for fixed block size segmentation for comparison in these two experiments. (Let 2^d be the fixed block size. Such a model is derived by substituting $g_s = 1$ for s whose depth is smaller than $d_{\max} - d$ and $g_s = 0$ otherwise.) The purpose of the third

experiment was to compare average coding rates of our proposed algorithm with a current image coding procedure on real images.

5.1. Experiment 1

In Experiments 1 and 2, we assumed $\mathcal{V} = \{0, 1\}$. In other words, we treated only binary images. $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ was assumed to be the Bernoulli distribution $\mathrm{Bern}(v_t|\theta_s^m)$ for s which satisfies $(x(t), y(t)) \in s$. Each element of $\boldsymbol{\theta}^m$ was i.i.d. distributed with the beta distribution $\mathrm{Beta}(\theta|\alpha, \beta)$, which is the conjugate distribution of the Bernoulli distribution. Therefore, the integral in (5) had a closed-form. The hyperparameter g_s of the model prior was $g_s = 1/2$ for every $s \in \mathcal{S} \setminus \{s_\lambda\}$ and $g_{s_\lambda} = 1$, and the hyperparameters of the Beta distribution were $\alpha = \beta = 1/2$.

The setting of Experiment 1 was as follows. The width and height of images were $w = h = 2^{d_{\max}} = 64$. Then, we generated 1000 images according to the following procedure.

1. Generate m according to (6).
2. Generate θ_s^m according to $p(\theta_s^m|m)$ for $s \in \mathcal{L}^m$.
3. Generate pixel value v_t according to $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ for $t \in \{0, 1, \ldots, hw - 1\}$.
4. Repeat Steps (1)–(3) 1000 times.

Examples of the generated images are shown in Figure 3. Then, we compressed these 1000 images. The size of the image was saved in the header of the compressed file using 4 bytes. The coding probability calculated by the proposed algorithm was quantized in 2^{16} levels and substituted into the range coder [31].

Figure 3. Examples of the generated images in Experiment 1.

The coding rates (bit/pel) averaged over all the images are shown in Table 1. Our proposed code has the minimum coding rate as expected by the Bayes optimality. Additionally, we compressed them by a standard lossless binary image coder called JBIG [32]. It did not work for the generated images. It is probably because JBIG [32] is not designed for synthetic images but mainly for real images such as faxes. A more detailed comparison was done in Experiment 3.

Table 1. The average coding rates (bit/pel). The bold number shows the minimum coding rate.

Quadtree (Proposed)	Fixed Size 4	Fixed Size 8	Fixed Size 16	JBIG [32]
0.619	0.705	0.659	0.679	1.826

5.2. Experiment 2

In Experiment 2, we compressed the binarized version of `camera.tif` from Wat [33], where the threshold of binarization was 128. The settings of the header and the range coder were the same as those of Experiment 1. Figure 4 visualizes the maximum a posteriori (MAP) estimation $m^{\mathrm{MAP}} = \arg\max_m p(m|v^{hw-1})$, which was calculated as a by-product of the compression by the algorithm detailed in Appendix B. It shows that our proposed model has the flexibility to represent the non-stationarity among the regions. The coding rate for `camera.tif` is shown in Table 2. For this image, the proposed algorithm a showed better coding rate than JBIG [32].

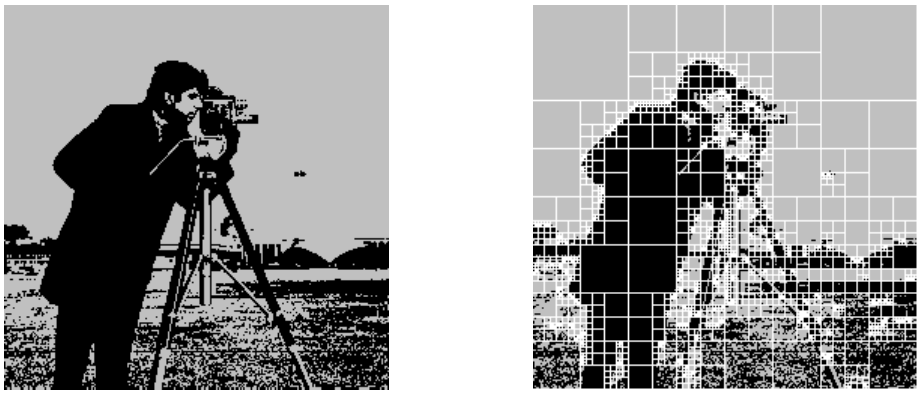

Figure 4. The original image (**left**) and the MAP estimated model m^{MAP} (**right**).

Table 2. The coding rates for the `camera.tif` from Wat [33] (bit/pel). The bold number shows the minimum coding rate.

Quadtree (Proposed)	Fixed Size 4	Fixed Size 8	Fixed Size 16	JBIG [32]
0.323	0.427	0.388	0.430	0.348

5.3. Experiment 3

In Experiment 3, we compared the proposed algorithm with JBIG [32] on real images from Wat [33]. They were binarized in a similar manner to Experiment 2. The settings of the header and the range coder were the same as those of Experiments 1 and 2. The results are shown in Table 3. The algorithm labeled as Proposed 1 in Table 3 is the same as that in Experiments 1 and 2. In the algorithm labeled as Proposed 2 in Table 3, we assumed that $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ is the Bernoulli distribution $\mathrm{Bern}(v_t|\theta^m_{s;v_{t-w-1}v_{t-w}v_{t-w+1}v_{t-1}})$, which depends on the neighboring four pixels. (If the indices go out of the image, we used the nearest past pixel in Manhattan distance.) In other words, there were 16 parameters $\theta^m_{s;0000}, \theta^m_{s;0001}, \ldots, \theta^m_{s;1111}$ for each block s of model m, and one of them was chosen by the realized values v_{t-w-1}, v_{t-w}, v_{t-w+1}, and v_{t-1} in the past. Each parameter was i.i.d. distributed with the beta distribution whose parameters were $\alpha = \beta = 1/2$.

Table 3. The coding rates for the images from Wat [33] (bit/pel). The bold number shows the minimum coding rate.

Images	JBIG [32]	Proposed 1	Proposed 2
`bird`	0.149	0.121	**0.099**
`bridge`	0.386	0.390	**0.373**
`camera`	0.348	0.323	**0.310**
`circles`	0.102	0.100	**0.060**
`crosses`	**0.083**	0.140	0.110
`goldhill1`	0.359	0.371	**0.353**
`horiz`	0.078	0.075	**0.022**
`lena1`	0.217	0.254	**0.216**
`montage`	0.164	0.176	**0.163**
`slope`	0.096	0.091	**0.056**
`squares`	0.076	**0.005**	0.010
`text`	**0.301**	0.468	0.468
avg.	0.197	0.209	**0.187**

Proposed 2 outperforms JBIG [32] without any specialized tuning of the hyperparameters from the perspective of average code rates. On the other hand, JBIG [32] outperforms our algorithms for `crosses` and `text`. This is because JBIG [32] is designed for text images

such as faxes and our stochastic generative model is for images with non-stationarity among segments. The structure of the text images should not be represented by the proposed quadtree-based stochastic generative model but the stochastic model $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ in each block. Although refinement of $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ for target images is out of the scope of this paper, it is an important problem in the future (see the next section).

6. Future Works

In this paper, we focus only on the stochastic representation of the non-stationarity among the segments. The discussion about the stochastic model $p(v_t|v^{t-1}, \boldsymbol{\theta}^m, m)$ and the prior $p(\boldsymbol{\theta}^m|m)$ to be assumed in each block is out of the scope. This is the first future work. For example, our model also works on the pairs of categorical distribution and Dirichlet distribution, normal distribution and normal-gamma distribution, and two-dimensional autoregressive model and normal-gamma distribution [24]. Moreover, using an approximative Bayesian estimation such as the variational Bayesian method, we expect that more complicated stochastic models (e.g., [25]) can be assumed.

The second future work is to apply our model to other stochastic image processing: image recognition, image generation, image inpainting, future extraction, etc. In particular, image generation and image inpainting may be suitable because the whole structure of stochastic image generation is described in our model and the parameters of the stochastic model can be learned optimally.

7. Conclusions

We propose a novel stochastic model based on the quadtree so that our model effectively represents the variable block size segmentation of images. Then, we construct a Bayes code for the proposed stochastic model. Moreover, we introduce an efficient algorithm to implement it in polynomial order of data size without loss of optimality. As a result, the derived algorithm has a better average coding rate than that of JBIG [32].

Author Contributions: Conceptualization, Y.N. and T.M.; methodology, Y.N.; software, Y.N.; validation, Y.N. and T.M.; formal analysis, Y.N. and T.M.; investigation, Y.N. and T.M.; resources, Y.N.; data curation, Y.N.; writing—original draft preparation, Y.N.; writing—review and editing, Y.N. and T.M.; visualization, Y.N.; supervision, T.M.; project administration, T.M.; and funding acquisition, T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by JSPS KAKENHI Grant Numbers JP17K06446 and JP19K04914.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://links.uwaterloo.ca/Repository.html (accessed on 30 July 2021) .

Acknowledgments: We would like to thank the members of Matsushima laboratory for their meaningful discussions.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Validity of the Proposed Algorithm

Appendix A.1. The Property of the Model Prior $p(m)$

First, we prove the following lemma for a general case. Note that the empty product equals 1 as usual.

Lemma A1. *Consider the k-ary complete tree $\tilde{T}$ with its depth D, in which each node u has a parameter $g_u \in [0,1]$. Let $\mathcal{T}$ denote the set of full subtrees which contain the root node λ of $\tilde{T}$. Then, the following holds.*

$$\sum_{T\in\mathcal{T}}\left(\prod_{u\in\mathcal{L}^T}(1-g_u)\prod_{u'\in\mathcal{I}^T}g_{u'}\right)=1, \tag{A1}$$

where $\mathcal{L}^T$ and $\mathcal{I}^T$ denote the set of leaf nodes and inner nodes of T, respectively, and $g_u = 0$ for u whose depth is D.

Proof. Lemma A1 is proved by induction with respect to the depth D. Let $[\lambda]$ denote the tree which consists of only the root node λ of $\tilde{T}$. When $D = 0$,

$$\begin{aligned}\sum_{T\in\mathcal{T}}\left(\prod_{u\in\mathcal{L}^T}(1-g_u)\prod_{u'\in\mathcal{I}^T}g_{u'}\right)&=\prod_{u\in\mathcal{L}^{[\lambda]}}(1-g_u)\prod_{u'\in\mathcal{I}^{[\lambda]}}g_{u'} &\text{(A2)}\\ &=1-g_\lambda &\text{(A3)}\\ &=1, &\text{(A4)}\end{aligned}$$

where (A2) is because $\mathcal{T}=\{[\lambda]\}$; (A3) is because $\mathcal{L}^{[\lambda]}=\{\lambda\}$, $\mathcal{I}^{[\lambda]}=\emptyset$, and the empty product equals to 1; (A4) is because the assumption of the statement, that is $g_u = 0$ for u whose depth is D.

If we assume (A1) for $D = d \geq 0$ as the induction hypothesis, then the following holds for $D = d + 1$.

$$\begin{aligned}\sum_{T\in\mathcal{T}}\left(\prod_{u\in\mathcal{L}^T}(1-g_u)\prod_{u'\in\mathcal{I}^T}g_{u'}\right)&=(1-g_\lambda)+\sum_{T\in\mathcal{T}\setminus\{[\lambda]\}}\left(\prod_{u\in\mathcal{L}^T}(1-g_u)\prod_{u'\in\mathcal{I}^T}g_{u'}\right) &\text{(A5)}\\ &=(1-g_\lambda)+g_\lambda\sum_{T\in\mathcal{T}\setminus\{[\lambda]\}}\left(\prod_{u\in\mathcal{L}^T}(1-g_u)\prod_{u'\in\mathcal{I}^T\setminus\{\lambda\}}g_{u'}\right). &\text{(A6)}\end{aligned}$$

Since each subtree $T \in \mathcal{T}\setminus\{[\lambda]\}$ is identified by k sub-subtrees whose root nodes are the child nodes of λ, let $\lambda_{\mathrm{child},i}$ denote the ith child node of λ for $0 \leq i \leq k-1$ and $\mathcal{T}^{\lambda_{\mathrm{child},i}}$ denote the set of sub-subtrees whose root node is $\lambda_{\mathrm{child},i}$. Then, the summation in (A6) are factorized as follows.

$$\begin{aligned}&\sum_{T\in\mathcal{T}\setminus\{[\lambda]\}}\left(\prod_{u\in\mathcal{L}^T}(1-g_u)\prod_{u'\in\mathcal{I}^T\setminus\{\lambda\}}g_{u'}\right) &\text{(A7)}\\ &=\sum_{T_0\in\mathcal{T}^{\lambda_{\mathrm{child},0}}}\cdots\sum_{T_{k-1}\in\mathcal{T}^{\lambda_{\mathrm{child},k-1}}}\\ &\qquad\left\{\left(\prod_{u\in\mathcal{L}^{T_0}}(1-g_u)\prod_{u'\in\mathcal{I}^{T_0}}g_{u'}\right)\times\cdots\times\left(\prod_{u\in\mathcal{L}^{T_{k-1}}}(1-g_u)\prod_{u'\in\mathcal{I}^{T_{k-1}}}g_{u'}\right)\right\} &\text{(A8)}\\ &=\left\{\sum_{T_0\in\mathcal{T}^{\lambda_{\mathrm{child},0}}}\left(\prod_{u\in\mathcal{L}^{T_0}}(1-g_u)\prod_{u'\in\mathcal{I}^{T_0}}g_{u'}\right)\right\}\\ &\qquad\times\cdots\times\left\{\sum_{T_{k-1}\in\mathcal{T}^{\lambda_{\mathrm{child},k-1}}}\left(\prod_{u\in\mathcal{L}^{T_{k-1}}}(1-g_u)\prod_{u'\in\mathcal{I}^{T_{k-1}}}g_{u'}\right)\right\}. &\text{(A9)}\end{aligned}$$

Using (A1) for $D = d$ as the induction hypothesis,

$$\sum_{T_i \in \mathcal{T}^{\lambda}_{\mathrm{child},i}} \left(\prod_{u \in \mathcal{L}^{T_i}} (1 - g_u) \prod_{u' \in \mathcal{I}^{T_i}} g_{u'} \right) = 1 \tag{A10}$$

for $0 \leq i \leq k-1$. Then,

$$(\mathrm{A6}) = (1 - g_\lambda) + g_\lambda \cdot 1^k = 1. \tag{A11}$$

Therefore, Lemma A1 holds for any D. □

Using this lemma, the following corollaries hold for our model.

Corollary A1. *The prior assumed in Assumption 2 satisfies* $\sum_{m \in \mathcal{M}} p(m) = 1$.

Corollary A2. *Under Assumption 2 and for any* $s \in \mathcal{S}$,

$$\sum_{m \in \{m' \in \mathcal{M} | s \in \mathcal{L}^{m'}\}} p(m) = (1 - g_s) \prod_{s' \in \mathcal{A}_s} g_{s'}, \tag{A12}$$

where $\mathcal{A}_s$ *denotes the set of the ancestor nodes of* s. *(Let* $\mathcal{A}_{s_\lambda}$ *be the empty set.)*

Proof of Corollary 2: Since each $m \in \{m' \in \mathcal{M} \mid s \in \mathcal{L}^{m'}\}$ has the right-hand side of (A12) as the factor in its prior,

$$\begin{aligned} &\sum_{m \in \{m' \in \mathcal{M} | s \in \mathcal{L}^{m'}\}} p(m) \\ &= (1 - g_s) \prod_{s' \in \mathcal{A}_s} g_{s'} \sum_{m \in \{m' \in \mathcal{M} | s \in \mathcal{L}^{m'}\}} \left(\prod_{s' \in \mathcal{L}^m \setminus \{s\}} (1 - g_{s'}) \prod_{s'' \in \mathcal{I}^m \setminus \{\mathcal{A}_s\}} g_{s''} \right). \end{aligned} \tag{A13}$$

Then, factorizing the sum in a similar manner from (A7)–(A9) and using Lemma A1 for the subtrees whose root nodes are out of $\mathcal{A}_s$, Corollary A2 is proved.

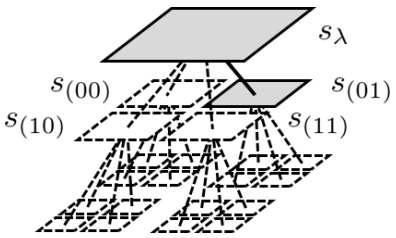

Figure A1. The example for the proof of Corollary A2.

As an example, Figure A1 shows the case where $d_{\max} = 2$, $s = s_{(01)}$, $\mathcal{A}_{s_{(01)}} = \{s_\lambda\}$. Let $\mathcal{M}^s$ denote a set of full quadtrees whose root node is s. In this case, we can factorize the sum in (A13) as follows.

$$\begin{aligned}
&\sum_{m \in \{m' \in \mathcal{M} | s_{(01)} \in \mathcal{L}^{m'}\}} \left(\prod_{s \in \mathcal{L}^m \setminus \{s_{(01)}\}} (1 - g_s) \prod_{s' \in \mathcal{I}^m \setminus \{\mathcal{A}_{s_{(01)}}\}} g_{s'} \right) \\
&= \sum_{m_{00} \in \mathcal{M}^{s_{(00)}}} \sum_{m_{10} \in \mathcal{M}^{s_{(10)}}} \sum_{m_{11} \in \mathcal{M}^{s_{(11)}}} \left\{ \left(\prod_{s \in \mathcal{L}^{m_{00}}} (1 - g_s) \prod_{s' \in \mathcal{I}^{m_{00}}} g_{s'} \right) \right. \\
&\qquad \left. \times \left(\prod_{s \in \mathcal{L}^{m_{10}}} (1 - g_s) \prod_{s' \in \mathcal{I}^{m_{10}}} g_{s'} \right) \left(\prod_{s \in \mathcal{L}^{m_{11}}} (1 - g_s) \prod_{s' \in \mathcal{I}^{m_{11}}} g_{s'} \right) \right\} \qquad \text{(A14)} \\
&= \left\{ \sum_{m_{00} \in \mathcal{M}^{s_{(00)}}} \left(\prod_{s \in \mathcal{L}^{m_{00}}} (1 - g_s) \prod_{s' \in \mathcal{I}^{m_{00}}} g_{s'} \right) \right\} \\
&\quad \times \left\{ \sum_{m_{10} \in \mathcal{M}^{s_{(10)}}} \left(\prod_{s \in \mathcal{L}^{m_{10}}} (1 - g_s) \prod_{s' \in \mathcal{I}^{m_{10}}} g_{s'} \right) \right\} \\
&\quad \times \left\{ \sum_{m_{11} \in \mathcal{M}^{s_{(11)}}} \left(\prod_{s \in \mathcal{L}^{m_{11}}} (1 - g_s) \prod_{s' \in \mathcal{I}^{m_{11}}} g_{s'} \right) \right\} \qquad \text{(A15)} \\
&= \left\{ (1 - g_{s_{(00)}}) + g_{s_{(00)}} (1 - g_{s_{(00)(00)}})(1 - g_{s_{(00)(01)}})(1 - g_{s_{(00)(10)}})(1 - g_{s_{(00)(11)}}) \right\} \\
&\quad \times \left\{ (1 - g_{s_{(10)}}) + g_{s_{(10)}} (1 - g_{s_{(10)(00)}})(1 - g_{s_{(10)(01)}})(1 - g_{s_{(10)(10)}})(1 - g_{s_{(10)(11)}}) \right\} \\
&\quad \times \left\{ (1 - g_{s_{(11)}}) + g_{s_{(11)}} (1 - g_{s_{(11)(00)}})(1 - g_{s_{(11)(01)}})(1 - g_{s_{(11)(10)}})(1 - g_{s_{(11)(11)}}) \right\} \qquad \text{(A16)} \\
&= 1 \cdot 1 \cdot 1 = 1. \qquad \text{(A17)}
\end{aligned}$$

The last equation is because $g_s = 0$ for s whose depth is $d_{\max}$. □

Appendix A.2. Proof of Lemma 1

Proof of Lemma 1.

$$\begin{aligned}
p(v_t | v^{t-1}, m) &= \int p(v_t | v^{t-1}, \boldsymbol{\theta}^m, m) p(\boldsymbol{\theta}^m | v^{t-1}, m) \mathrm{d}\boldsymbol{\theta}^m \qquad \text{(A18)} \\
&\propto \int p(v_t | v^{t-1}, \boldsymbol{\theta}^m, m) p(v^{t-1} | \boldsymbol{\theta}^m, m) p(\boldsymbol{\theta}^m | m) \mathrm{d}\boldsymbol{\theta}^m \qquad \text{(A19)} \\
&= \int p(v_t | v^{t-1}, \theta_s^m) \int p(v^{t-1} | \boldsymbol{\theta}^m, m) p(\boldsymbol{\theta}^m | m) \mathrm{d}\boldsymbol{\theta}^m_{\setminus s} \mathrm{d}\theta_s^m \qquad \text{(A20)} \\
&\propto \int p(v_t | v^{t-1}, \theta_s^m) p_s(\theta_s^m) \prod_{i \in \{i' \le t | (x(i'), y(i')) \in s\}} p(v_i | v^{i-1}, \theta_s^m) \mathrm{d}\theta_s^m, \qquad \text{(A21)}
\end{aligned}$$

where $\propto$ means that the left-hand side is proportional to the right-hand side, regarding the variables except v_t as constant, and $\boldsymbol{\theta}^m_{\setminus s}$ denotes the parameters $\boldsymbol{\theta}^m$ except θ_s^m. Here, we use Assumptions 1 and 3. As a result, Formula (A21) is independent of m. □

Appendix A.3. Proof of Theorem 1

Proof of Theorem 1. We prove the following two equations simultaneously.

$$\begin{aligned}
p(m | v^{t-1}) &= \prod_{s \in \mathcal{L}^m} (1 - g_{s|t-1}) \prod_{s' \in \mathcal{I}^m} g_{s'|t-1}, \qquad \text{(A22)} \\
q^*(v_t | v^{t-1}) &= q(v_t | v^{t-1}, s_\lambda). \qquad \text{(A23)}
\end{aligned}$$

(A22) means that the posterior distribution of the model m has the same form as the prior. (A23) is equivalent to Theorem 1.

They are proved by induction with respect to t. Therefore, the proof consists of the following four steps.

Step 1 We prove (A22) for $t = 0$.

Step 2 We prove (A23) for $t = 0$.

Step 3 We prove (A22) for $t = k + 1$ under the assumptions of (A22) and (A23) for $t = k$.

Step 4 We prove (A23) for $t = k + 1$ under the assumptions of (A22) for $t = k + 1$ and (A23) for $t = k$.

Step 1: (A22) holds for $t = 0$ because it is Assumption 2 itself.
Step 2: For $t = 0$, (A23) can be proved as follows:

$$q^*(v_0) = \sum_{m\in\mathcal{M}} p(m) \int p(v_0|\boldsymbol{\theta}^m, m) p(\boldsymbol{\theta}^m|m) \mathrm{d}\boldsymbol{\theta}^m \tag{A24}$$

$$= \sum_{s\in\mathcal{S}_0} \sum_{m\in\{m'\in\mathcal{M}|s\in\mathcal{L}^{m'}\}} p(m) \int p(v_0|\boldsymbol{\theta}^m, m) p(\boldsymbol{\theta}^m|m) \mathrm{d}\boldsymbol{\theta}^m \tag{A25}$$

$$= \sum_{s\in\mathcal{S}_0} \sum_{m\in\{m'\in\mathcal{M}|s\in\mathcal{L}^{m'}\}} p(m) \tilde{q}(v_0|s) \tag{A26}$$

$$= \sum_{s\in\mathcal{S}_0} \tilde{q}(v_0|s) \sum_{m\in\{m'\in\mathcal{M}|s\in\mathcal{L}^{m'}\}} p(m) \tag{A27}$$

$$= \sum_{s\in\mathcal{S}_0} \tilde{q}(v_0|s)(1 - g_s) \prod_{s'\in\mathcal{A}_s} g_{s'} \tag{A28}$$

$$= (1 - g_{s_\lambda}) \tilde{q}(v_0|s_\lambda) + \sum_{s\in\mathcal{S}_0\setminus\{s_\lambda\}} \tilde{q}(v_0|s)(1 - g_s) \prod_{s'\in\mathcal{A}_s} g_{s'} \tag{A29}$$

$$= (1 - g_{s_\lambda}) \tilde{q}(v_0|s_\lambda) + g_{s_\lambda} \sum_{s\in\mathcal{S}_0\setminus\{s_\lambda\}} \tilde{q}(v_0|s)(1 - g_s) \prod_{s'\in\mathcal{A}_s\setminus\{s_\lambda\}} g_{s'}. \tag{A30}$$

Note that $\mathcal{S}_0$ is defined in Definition 4. Here, we use Lemma 1 and Corollary A2 in (A26) and (A28), respectively. The recursive structure in (A28) and (A30) coincides with $q(v_0|s_\lambda)$.
Step 3: In the following, we assume (A22) and (A23) for $t = k$ as the induction hypotheses. Let $r \in \mathcal{L}^m$ satisfy $(x(k), y(k)) \in r$ and $\mathcal{S}_k$ be the same one defined in Definition 4. Then, for $t = k + 1$,

$$\prod_{s\in\mathcal{L}^m} (1 - g_{s|k}) \prod_{s'\in\mathcal{I}^m} g_{s'|k}$$

$$= \prod_{s\in\mathcal{L}^m\cap\mathcal{S}_k} (1 - g_{s|k}) \prod_{s'\in\mathcal{I}^m\cap\mathcal{S}_k} g_{s'|k} \prod_{s''\in\mathcal{L}^m\setminus\mathcal{S}_k} (1 - g_{s''|k}) \prod_{s'''\in\mathcal{I}^m\setminus\mathcal{S}_k} g_{s'''|k} \tag{A31}$$

$$= (1 - g_{r|k}) \prod_{s\in\mathcal{A}_r} g_{s|k} \prod_{s'\in\mathcal{L}^m\setminus\mathcal{S}_k} (1 - g_{s'|k}) \prod_{s''\in\mathcal{I}^m\setminus\mathcal{S}_k} g_{s''|k}. \tag{A32}$$

When $|r| = 1$, substituting (11) and (10) in this order,

$$(1 - g_{r|k}) \prod_{s\in\mathcal{A}_r} g_{s|k} = (1 - g_{r|k-1}) \prod_{s\in\mathcal{A}_r} \frac{q(v_k|v^{k-1}, s_{\mathrm{child}})}{q(v_k|v^{k-1}, s)} g_{s|k-1} \tag{A33}$$

$$= \frac{\tilde{q}(v_k|v^{k-1}, r)}{q(v_k|v^{k-1}, s_\lambda)} (1 - g_{r|k-1}) \prod_{s\in\mathcal{A}_r} g_{s|k-1}. \tag{A34}$$

Here, (A34) is given by the cancellation of the telescoping product.

When $|r| > 1$, substituting (11) and (10) in this order,

$$
\begin{aligned}
&(1-g_{r|k})\prod_{s\in\mathcal{A}_r} g_{s|k} \\
&= \left(1-\frac{q(v_k|v^{k-1},r_{\mathrm{child}})}{q(v_k|v^{k-1},r)}g_{r|k-1}\right)\prod_{s\in\mathcal{A}_r}\frac{q(v_k|v^{k-1},s_{\mathrm{child}})}{q(v_k|v^{k-1},s)}g_{s|k-1} && \text{(A35)}\\
&= \left(\frac{q(v_k|v^{k-1},r)-q(v_k|v^{k-1},r_{\mathrm{child}})g_{r|k-1}}{q(v_k|v^{k-1},r)}\right)\prod_{s\in\mathcal{A}_r}\frac{q(v_k|v^{k-1},s_{\mathrm{child}})}{q(v_k|v^{k-1},s)}g_{s|k-1} && \text{(A36)}\\
&= \left(\frac{(1-g_{r|k-1})\tilde{q}(v_k|v^{k-1},r)+q(v_k|v^{k-1},r_{\mathrm{child}})g_{r|k-1}-q(v_k|v^{k-1},r_{\mathrm{child}})g_{r|k-1}}{q(v_k|v^{k-1},r)}\right) \\
&\quad\times\prod_{s\in\mathcal{A}_r}\frac{q(v_k|v^{k-1},s_{\mathrm{child}})}{q(v_k|v^{k-1},s)}g_{s|k-1} && \text{(A37)}\\
&= \left(\frac{(1-g_{r|k-1})\tilde{q}(v_k|v^{k-1},r)}{q(v_k|v^{k-1},r)}\right)\prod_{s\in\mathcal{A}_r}\frac{q(v_k|v^{k-1},s_{\mathrm{child}})}{q(v_k|v^{k-1},s)}g_{s|k-1} && \text{(A38)}\\
&= \frac{\tilde{q}(v_k|v^{k-1},r)}{q(v_k|v^{k-1},s_\lambda)}(1-g_{r|k-1})\prod_{s\in\mathcal{A}_r}g_{s|k-1}. && \text{(A39)}
\end{aligned}
$$

Here, (A39) is again given by the cancellation of the telescoping product. As a result, (A34) and (A39) have the same form.

On the other hand, applying the updating rule (11),

$$\prod_{s'\in\mathcal{L}^m\setminus\mathcal{S}_k}(1-g_{s'|k})\prod_{s''\in\mathcal{I}^m\setminus\mathcal{S}_k}g_{s''|k}=\prod_{s'\in\mathcal{L}^m\setminus\mathcal{S}_k}(1-g_{s'|k-1})\prod_{s''\in\mathcal{I}^m\setminus\mathcal{S}_k}g_{s''|k-1}. \tag{A40}$$

Therefore, the right-hand side of (A32) is transformed as follows.

$$
\begin{aligned}
&\frac{\tilde{q}(v_k|v^{k-1},r)}{q(v_k|v^{k-1},s_\lambda)}(1-g_{r|k-1})\prod_{s\in\mathcal{A}_r}g_{s|k-1}\prod_{s'\in\mathcal{L}^m\setminus\mathcal{S}_k}(1-g_{s'|k-1})\prod_{s''\in\mathcal{I}^m\setminus\mathcal{S}_k}g_{s''|k-1} && \text{(A41)}\\
&= \frac{\tilde{q}(v_k|v^{k-1},r)}{q(v_k|v^{k-1},s_\lambda)}\prod_{s\in\mathcal{L}^m}(1-g_{s|k-1})\prod_{s'\in\mathcal{I}^m}g_{s'|k-1} && \text{(A42)}\\
&= \frac{\tilde{q}(v_k|v^{k-1},r)}{q^*(v_k|v^{k-1})}p(m|v^{k-1}) && \text{(A43)}\\
&= \frac{p(v_k|v^{k-1},m)}{p(v_k|v^{k-1})}p(m|v^{k-1}) && \text{(A44)}\\
&= p(m|v^k). && \text{(A45)}
\end{aligned}
$$

In (A43), we use (A22) and (A23) as the induction hypothesis. In (A44), we use Lemma 1 and Proposition 1. Thus, (A22) holds for $t = k+1$.

In addition, it holds that

$$\sum_{m\in\{m'\in\mathcal{M}|s\in\mathcal{L}^{m'}\}}p(m|v^k)=(1-g_{s|k})\prod_{s'\in\mathcal{A}_s}g_{s'|k}, \tag{A46}$$

since the posterior $p(m|v^k)$ has the same form as the prior $p(m)$ and can be applied Corollary A2.

Step 4: (A23) can be proved for $t = k+1$ in a similar manner to the case where $t = 0$.

$$q^*(v_{k+1}|v^k) = \sum_{s\in\mathcal{S}_{k+1}} \tilde{q}(v_{k+1}|v^k, s) \sum_{m\in\{m'\in\mathcal{M}|s\in\mathcal{L}^{m'}\}} p(m|v^k) \tag{A47}$$

$$= \sum_{s\in\mathcal{S}_{k+1}} \tilde{q}(v_t|v^k, s)(1-g_{s|k}) \prod_{s'\in\mathcal{A}_s} g_{s'|k} \tag{A48}$$

$$= (1-g_{s_\lambda|k})\tilde{q}(v_{k+1}|v^k, s_\lambda) + g_{s_\lambda|k} \sum_{s\in\mathcal{S}_{k+1}\setminus\{s_\lambda\}} \tilde{q}(v_{k+1}|v^k, s)(1-g_{s|k}) \prod_{s'\in\mathcal{A}_s\setminus\{s_\lambda\}} g_{s'|k}. \tag{A49}$$

In (A48), we use (A46). The recursive structure in (A48) and (A49) coincides with $q(v_{k+1}|v^k, s_\lambda)$. □

Appendix B. The Algorithm to Calculate $m^{\mathbf{MAP}}$

In this appendix, we derive the algorithm to calculate $\arg\max_m p(m|v^t)$. At first, $\max_m p(m|v^t)$ can be decomposed in a similar manner to the proof of Lemma A1 by replacing the sum for the max.

$$\begin{aligned}\max_{m\in\mathcal{M}} p(m|v^t) = \max\Bigg\{ & 1-g_{s_\lambda|t}, \quad g_{s_\lambda|t} \max_{m_{00}\in\mathcal{M}^{s(00)}} \left\{ \prod_{s\in\mathcal{L}^{m_{00}}} (1-g_{s|t}) \prod_{s'\in\mathcal{I}^{m_{00}}\setminus\{s_\lambda\}} g_{s'|t} \right\} \\ & \times \max_{m_{01}\in\mathcal{M}^{s(01)}} \left\{ \prod_{s\in\mathcal{L}^{m_{01}}} (1-g_{s|t}) \prod_{s'\in\mathcal{I}^{m_{01}}\setminus\{s_\lambda\}} g_{s'|t} \right\} \\ & \times \max_{m_{10}\in\mathcal{M}^{s(10)}} \left\{ \prod_{s\in\mathcal{L}^{m_{10}}} (1-g_{s|t}) \prod_{s'\in\mathcal{I}^{m_{10}}\setminus\{s_\lambda\}} g_{s'|t} \right\} \\ & \times \max_{m_{11}\in\mathcal{M}^{s(11)}} \left\{ \prod_{s\in\mathcal{L}^{m_{11}}} (1-g_{s|t}) \prod_{s'\in\mathcal{I}^{m_{11}}\setminus\{s_\lambda\}} g_{s'|t} \right\} \Bigg\}. \end{aligned} \tag{A50}$$

We define a recursive function $\phi_t : \mathcal{S} \to \mathbb{R}$ as follows.

Definition A1.

$$\phi_t(s) := \begin{cases} 1, & |s| = 1 \\ \max\left\{1-g_{s|t},\ g_{s|t}\phi_t(s_{\mathrm{child}_{00}})\phi_t(s_{\mathrm{child}_{01}})\phi_t(s_{\mathrm{child}_{10}})\phi_t(s_{\mathrm{child}_{11}})\right\}, & \text{otherwise.} \end{cases} \tag{A51}$$

Here, $s_{\mathrm{child}_{00}}$, $s_{\mathrm{child}_{01}}$, $s_{\mathrm{child}_{10}}$, and $s_{\mathrm{child}_{11}}$ are child nodes of s of the complete quadtree on $\mathcal{S}$

Then, $\max_m p(m|v^t)$ can be calculated by $\phi_t(s_\lambda)$.
Next, we define the following flag variable $h_{s|t} \in \{0, 1\}$.

Definition A2.

$$h_{s|t} := \begin{cases} 0, & 1-g_{s|t} \geq g_{s|t}\phi_t(s_{\mathrm{child}_{00}})\phi_t(s_{\mathrm{child}_{01}})\phi_t(s_{\mathrm{child}_{10}})\phi_t(s_{\mathrm{child}_{11}}) \\ 1, & \text{otherwise.} \end{cases} \tag{A52}$$

We can calculate $h_{s|t}$ and $\phi_t(s)$ simultaneously. Then, $\arg\max_m p(m|v^t)$ is identified as the model which satisfies

$$s \in \mathcal{I}^m \Rightarrow h_{s|t} = 1, \tag{A53}$$

$$s \in \mathcal{L}^m \Rightarrow h_{s|t} = 0. \tag{A54}$$

Such a model can be searched by backtracking from s_λ after the calculation of $\phi_t(s_\lambda)$ and $h_{s_\lambda|t}$.

References

1. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
2. Huffman, D.A. A Method for the Construction of Minimum-Redundancy Codes. *Proc. IRE* **1952**, *40*, 1098–1101. [CrossRef]
3. Rissanen, J.; Langdon, G. Universal modeling and coding. *IEEE Trans. Inf. Theory* **1981**, *27*, 12–23. [CrossRef]
4. Ziv, J.; Lempel, A. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory* **1977**, *23*, 337–343. [CrossRef]
5. Ziv, J.; Lempel, A. Compression of individual sequences via variable-rate coding. *IEEE Trans. Inf. Theory* **1978**, *24*, 530–536. [CrossRef]
6. Davisson, L. Universal noiseless coding. *IEEE Trans. Inf. Theory* **1973**, *19*, 783–795. [CrossRef]
7. Cover, T. Enumerative source encoding. *IEEE Trans. Inf. Theory* **1973**, *19*, 73–77. [CrossRef]
8. Matsushima, T.; Hirasawa, S. Reducing the space complexity of a Bayes coding algorithm using an expanded context tree. In Proceedings of the 2009 IEEE International Symposium on Information Theory, Seoul, Korea, 28 June–3 July 2009; pp. 719–723. [CrossRef]
9. Willems, F.M.J.; Shtarkov, Y.M.; Tjalkens, T.J. The context-tree weighting method: Basic properties. *IEEE Trans. Inf. Theory* **1995**, *41*, 653–664. [CrossRef]
10. Kontoyiannis, I.; Mertzanis, L.; Panotopoulou, A.; Papageorgiou, I.; Skoularidou, M. Bayesian Context Trees: Modelling and exact inference for discrete time series. *arXiv* **2020**, arXiv:2007.14900.
11. Weinberger, M.J.; Seroussi, G.; Sapiro, G. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Trans. Image Process.* **2000**, *9*, 1309–1324. [CrossRef] [PubMed]
12. Kuroki, N.; Nomura, T.; Tomita, M.; Hirano, K. Lossless image compression by two-dimensional linear prediction with variable coefficients. *IEICE Trans. Fund. Electron. Commun. Comput. Sci.* **1992**, *75*, 882–889.
13. Wu, X.; Barthel, E.; Zhang, W. Piecewise 2D autoregression for predictive image coding. In Proceedings of the 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269), Chicago, IL, USA, 7 October 1998; pp. 901–904. [CrossRef]
14. Meyer, B.; Tischer, P. Glicbawls—Grey Level Image Compression by Adaptive Weighted Least Squares. In *Data Compression Conference;* IEEE Computer Society: Los Alamitos, CA, USA, 2001; p. 0503. [CrossRef]
15. Ye, H.; Deng, G.; Devlin, J.C. A weighted least squares method for adaptive prediction in lossless image compression. In Proceedings of the Picture Coding Symposium, Saint-Malo, France, 23–25 April 2003.
16. Liu, J.; Zhai, G.; Yang, X.; Chen, L. Lossless Predictive Coding for Images With Bayesian Treatment. *IEEE Trans. Image Process.* **2014**, *23*, 5519–5530. [CrossRef] [PubMed]
17. Weinlich, A.; Amon, P.; Hutter, A.; Kaup, A. Probability Distribution Estimation for Autoregressive Pixel-Predictive Image Coding. *IEEE Trans. Image Process.* **2016**, *25*, 1382–1395. [CrossRef] [PubMed]
18. Meyer, B.; Tischer, P. TMW—A new method for lossless image compression. In Proceedings of the 1997 Picture Coding Symposium (PCS'97), Berlin, Germany, 10–12 September 1997; pp. 533–538.
19. Martchenko, A.; Deng, G. Bayesian Predictor Combination for Lossless Image Compression. *IEEE Trans. Image Process.* **2013**, *22*, 5263–5270. [CrossRef] [PubMed]
20. Matsuda, I.; Ozaki, N.; Umezu, Y.; Itoh, S. Lossless coding using variable block-size adaptive prediction optimized for each image. In Proceedings of the 2005 13th European Signal Processing Conference, Antalya, Turkey, 4–8 September 2005; pp. 1–4.
21. Ulacha, G.; Stasiński, R.; Wernik, C. Extended Multi WLS Method for Lossless Image Coding. *Entropy* **2020**, *22*, 919. [CrossRef] [PubMed]
22. Mentzer, F.; Agustsson, E.; Tschannen, M.; Timofte, R.; Van Gool, L. Practical Full Resolution Learned Lossless Image Compression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–21 June 2019; pp. 10621–10630. [CrossRef]
23. Wu, X.; Memon, N. Context-based, adaptive, lossless image coding. *IEEE Trans. Commun.* **1997**, *45*, 437–444. [CrossRef]
24. Nakahara, Y.; Matsushima, T. Autoregressive Image Generative Models with Normal and t-distributed Noise and the Bayes Codes for Them. In Proceedings of the 2020 International Symposium on Information Theory and Its Applications (ISITA), Kapolei, HI, USA, 24–27 October 2020; pp. 81–85.
25. Nakahara, Y.; Matsushima, T. Bayes code for two-dimensional auto-regressive hidden Markov model and its application to lossless image compression. In Proceedings of the International Workshop on Advanced Imaging Technology (IWAIT) 2020, Yogyakarta, Indonesia, 5–7 January 2020; International Society for Optics and Photonics: Bellingham, WA, USA, 2020; Volume 11515, pp. 330–335. [CrossRef]
26. Nakahara, Y.; Matsushima, T. A Stochastic Model of Block Segmentation Based on the Quadtree and the Bayes Code for It. In Proceedings of the 2020 Data Compression Conference (DCC), Snowbird, UT, USA, 24–27 March 2020; pp. 293–302.
27. Sullivan, G.J.; Ohm, J.; Han, W.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [CrossRef]
28. Dobashi, N.; Saito, S.; Nakahara, Y.; Matsushima, T. Meta-Tree Random Forest: Probabilistic Data-Generative Model and Bayes Optimal Prediction. *Entropy* **2021**, *23*, 768. [CrossRef] [PubMed]
29. Matsushima, T.; Inazumi, H.; Hirasawa, S. A class of distortionless codes designed by Bayes decision theory. *IEEE Trans. Inf. Theory* **1991**, *37*, 1288–1293. [CrossRef]
30. Clarke, B.S.; Barron, A.R. Information-theoretic asymptotics of Bayes methods. *IEEE Trans. Inf. Theory* **1990**, *36*, 453–471. [CrossRef]

31. Martín, G. Range encoding: An algorithm for removing redundancy from a digitised message. In Proceedings of the Video and Data Recording Conference, Southampton, UK, 24–27 July 1979; pp. 24–27.
32. Kuhn, M. JBIG-KIT. Available online: https://www.cl.cam.ac.uk/~mgk25/jbigkit/ (accessed on 30 July 2021).
33. Image Repository of the University of Waterloo. Available online: http://links.uwaterloo.ca/Repository.html (accessed on 30 July 2021).