IMAGE DENOISING USING WAVELET BAYESIAN NETWORK MODELS

Jinn Ho , Wen-Liang Hwang

Institute of Information Science, Academia Sinica, Taiwan

ABSTRACT

A number of techniques have been developed to deal with image denoising, which is regarded as the simplest inverse problem. In this paper, we propose an approach that constructs a Bayesian network from the wavelet coefficients of a single image such that different Bayesian networks can be obtained from different input images. Then, we utilize the maximum-a-posterior (MAP) estimator to derive the wavelet coefficients. Constructing a graphical model usually requires a large number of training images. However, we demonstrate that by using certain wavelet properties, namely, inter-scale data dependency, decorrelation between wavelet coefficients, and sparsity of the wavelet representation, a robust Bayesian network can be constructed from one image to resolve the denoising problem. Our experiment results show that, in terms of the peak-signal-to-noise-ratio (PSNR) performance, the proposed approach outperforms state-of-art algorithms on several images with various amounts of white Gaussian noise.

Index Terms— Image Denoising, Bayesian Network, Wavelet Transform

1. INTRODUCTION

Complex phenomena usually involve a large number of hidden variables and data sources. Graphical models provide a unifying framework for modeling the probability distributions of complex phenomena by decomposing joint probability distributions into a set of local constraints and dependencies [1]. The models are particularly useful in signal and image processing applications, computer vision, machine learning, and time series analysis. After formulating a problem as a graphical model, a wide range of statistical learning and inference algorithms can be applied directly to derive a solution.

Bayesian networks are probably the most popular type of graphical model. In this paper, our objective is to construct a Bayesian network from a single image for denoising purposes. The construction of a Bayesian network involves prior knowledge of the probability relationships between the variables of interest. Learning approaches are widely used to construct Bayesian networks that best represent the joint probabilities of training data [2]. In practice, an optimization process based on a heuristic search technique is used to find the best DAG structure over the space of all possible networks. However, the approach is computationally intractable because it must explore several combinations of variable dependencies to derive the optimal Bayesian network. Two wavelet properties can be exploited to reduce the computational complexity of learning a Bayesian network. First, the wavelet transform of a natural image tends to be sparse with large coefficients at the edges and in smooth regions. The sparsity reduces the number of variables required to construct a graph. Second, the magnitudes of the wavelet coefficients tend to propagate through the scales of the quad-trees; thus, the dependencies of variables in adjacent scales can be derived from the multi-scale quad-trees of the wavelet coefficients. The second property motivated the authors of [3] to use the hidden Markov tree (HMT) model to capture the joint statistics of wavelet coefficients across scales.

Our approach shares a common framework with the Bayesian approach in that we first construct a Bayesian network from the undecimated discrete wavelet coefficients (DWT) of an image. Then, we convert the network into a factor graph and use the sum-product algorithm to derive the MAP solution.

2. BAYESIAN NETWORKS AND FACTOR GRAPHS

Bayesian networks and factor graphs are graph models that express how the joint probability of several variables is factored into the product of the local functions (factors) of smaller sets of variables. The models are closely connected because one representation can always be converted into the equivalent form of the other representation [4]. Aji and McEliece [5] demonstrated that many well-known probabilistic inference algorithms, such as the belief propagation algorithm in Bayesian networks and the sum-product algorithm in factor graphs, can solve the "marginalize product-of-functions" problem [6].

In a Bayesian network, the probability inference problem involves assigning the most probable values to unobserved variables given the values of the observed variables. Although the problem is generally NP-hard, when the Bayesian network forms a DAG, the efficient message passing scheme, called the belief propagation scheme, can be used to solve the problem. Specifically, in each iteration of belief propagation, every node sends a message to each of its neighboring nodes

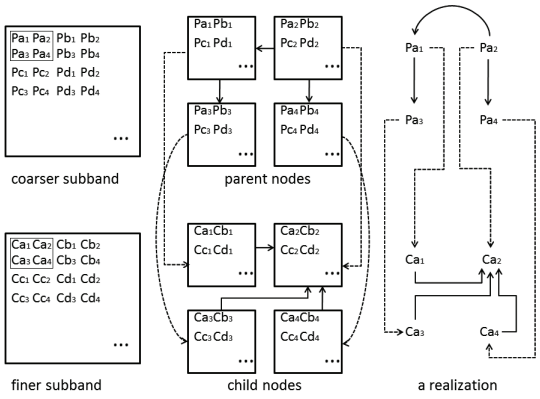


Fig. 1. Constructing inter-scale edges E_o^u and intra-scale edge E_i^u for two subbands: left: two corresponding subbands at adjacent scales; middle: the nodes and edges in the Bayesian network (the dashed line and solid lines denote inter-scale edges and intra-scale edges respectively); right: a realization of the Bayesian network in the middle figure.

and receives messages from those neighbors.

A factor graph is a bipartite graph whose vertices are divided into two disjoint sets, U and V, where U is a set of variables and V is a set of functions (factors) such that every (undirected) edge connects a vertex in U to a vertex in V. Each edge expresses the variable that is the argument of a particular local function. Let $\mu_{x\to f}$ denote the message sent from variable node x to function node f, and let $\mu_{f\to x}$ denote the message sent from function node f to variable node x. In addition, let $\mathcal{N}(x)$ and $\mathcal{N}(f)$ denote the neighboring nodes of x and f respectively. The message is computed by the max-product algorithm for a factor graph based on the following update rules:

variable to local function:

$$\mu_{x\to f}(x) = \prod_{h\in\mathcal{N}(x)\setminus f} \mu_{h\to x}(x); \tag{1}$$

local function to variable:

$$\mu_{f\to x}(x) = \max_{\sim\{x\}} f(X) \prod_{y\in\mathcal{N}(f)\setminus x} \mu_{y\to f}(y), \tag{2}$$

where $\max_{\sim\{x\}}$ indicates that all variables except x are maximized and X is the set of arguments of the function f.

The MAP solution of the max-product algorithm is defined as

$$\hat{x} = \arg\max_x \prod_{f\in\mathcal{N}(x)} \mu_{f\to x}(x). \tag{3}$$

3. CONSTRUCTING WAVELET BAYESIAN NETWORKS

A Bayesian network, denoted as $B = (V, E, P)$, comprises a set of random variables and their conditional dependencies represented by a directed acyclic graph in which the nodes represent the elements in V. Each edge element in E takes the form of a directed arc $x \to y$, where x and $y \in V$. The likelihood $p(y \mid x) \in P$ of an edge $x \to y \in E$ is the conditional probability of observing y given that x exists.

We call the Bayesian networks constructed in wavelet domains wavelet Bayesian networks (WBNs). To construct a WBN, we first group subbands with the same orientation together to obtain a horizontal-group(h), a vertical-group(v), and a diagonal-group(d) of wavelet coefficients. Next, we explain how to construct the Bayesian network $B^u(V^u, E^u, P^u)$ that corresponds to the u-orientation with $u \in \{h, v, d\}$.

A. Vertex Set V^u:

Let the size of the input image F be $N \times N$. If J wavelet decompositions are applied to F, there will be J subbands of size $N \times N$ in each orientation. Given a parameter m, without loss of generality, we assume that m divides N. For each subband, m^2 variable nodes are formed and $(\frac{N}{m})^2$ wavelet coefficients sampled from the subband are assigned to each variable node. Let $x_j^u(i,k)$, with $j = 1, \cdots, J$ and $i, k = 0, \cdots, m-1$, denote the (i,k) variable node in the j-th subband. We denote the vertex set of Bayesian network B^u to be

$$V^u = \{x_j^u(i,k) | i,k = 0, \cdots, m-1; j = 1, \cdots, J\}. \tag{4}$$

Because images are usually modeled as Markov random fields and the wavelet transforms of real-world images tend to be approximately decorrelated, it can be assumed that the wavelet coefficients sampled with large pixel distances are independent of each other. Thus, the $(\frac{N}{m})^2$ wavelet coefficients are independently sampled from some (unknown) distribution of a random variable.

B. Edge Set E^u:

The arcs (directed edges) in B^u can be divided into two disjoint sets, E_o^u and E_i^u, where E_o^u is comprised of (inter-scale) edges incident to vertices at different scales, and E_i^u are the (intra-scale) edges incident to vertices at the same scale. The persistence property of the wavelet transforms indicates that large/small values of wavelet coefficients tend to occur at the same spatial locations in subbands at adjacent scales and orientations. The property can be used to construct arcs in E_o^u by linking a vertex at the coarser scale $j+1$ to the vertex of the same index at the finer scale j; that is,

$$E_o^u = \{x_{j+1}^u(i,k) \to x_j^u(i,k) |\ i,k = 0, \cdots m-1, \text{ and } j = 1, \cdots, J-1\}. \tag{5}$$

The edges in E_i^u represent the connections between vertices at the same scale and orientation. Constructing the edges corresponds to deriving the Bayesian network on the nodes $x_j^u(i,k)$ that best represent the joint probability of the nodes at the same scale j and orientation u. However, this optimization process is computationally intractable because it searches for the best DAG structure over the space of all possible networks of nodes $x_j^u(i,k)$. We limit the solution space to spanning trees so that we can derive an efficient solution by using

the *maximal weighted spanning tree* (MWST) algorithm [7, 8]. The optimum weighted spanning tree can be derived by minimizing the relative entropy (Kullback-Leibler distance) $D(p||q)$ between the probability functions p and q.

Let $\mathbf{x}$ be the vector of variables $x_1, \cdots, x_n$; let $p(i)$ and $b(i)$ denote the indices of the parent nodes and the sibling nodes of x_i respectively; and let q be the induced probability of the spanning tree. Then, we have

$$q(\mathbf{x}) = \prod_{j=1}^{n} p(x_j \mid x_{p(j)}, x_{b(j)}). \tag{6}$$

To find the optimal spanning tree, we minimize the relative entropy between $q(\mathbf{x})$ and the joint probability $p(\mathbf{x})$ as follows:

$$D(p||q) = \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \tag{7}$$

Since minimizing $D(p||q)$ is equivalent to maximizing $\sum_{\mathbf{x}} p(\mathbf{x}) \log q(\mathbf{x})$, we can derive that minimizing $D(p||q)$ over q is equivalent to maximizing the weighted summation of conditional mutual information $\sum_{j,b(j)\neq\emptyset} \sum_{p(j)} p(x_{p(j)}) \cdot I(x_j, x_{b(j)} \mid x_{p(j)})$, where the weight of the arc $x_i \rightarrow x_j$ is defined as $p(x_{p(i)})I(x_i, x_j|x_{p(i)})$. To choose a tree with the maximum total arc weight of n nodes, we use Kruskal's algorithm [9].

C. Probability Model P^u:

There are two types of arcs in a Bayesian network B^u: (1) the inter-scale parent-child arc, which connects a node with its coarser-scale parent; and (2) the intra-scale sibling arc, which connects two nodes of the same scale. Simoncelli [10] exploited the persistence property of wavelet transforms and proposed a joint statistical model of a "child" coefficient conditioned on the coarse-scale "parent" coefficients at the same spatial locations in all orientations. In our algorithm we assume there is only one parent coefficient x_p for x and the bias of variance α is zero. Then, the joint probability of the parent-child arc in E_o^u is modeled as

$$f_o^u(x|x_p) = \mathcal{N}(0; 2wx_p^2), \tag{8}$$

where w is a chosen parameter.

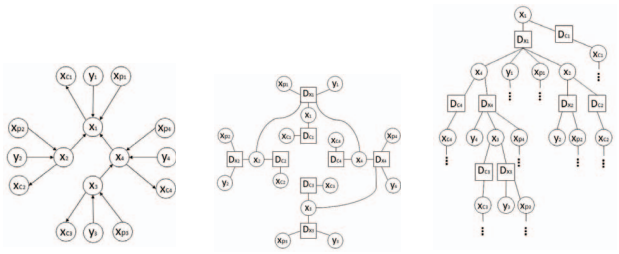
In the construction of intra-scale edges, Kruskal's algorithm selects the arc $x \rightarrow z \in E_i^u$, where the mutual information between x and z is high; that is, x and z are highly correlated. Thus, we utilize a similar concept to model the probability of z conditioned on x as the following Laplacian distribution function:

$$f_i^u(z|x) \propto \lambda \exp(-\lambda|x - z|), \tag{9}$$

where λ is the scale parameter of the Laplacian distribution.

4. WAVELET BAYESIAN NETWORKS FOR DENOISING

In this section, we consider the image denoising problem, which involves removing additive white Gaussian noise with zero mean and known variance from an image.



(a) (b) (c)

Fig. 2. Converting B_n to F_n and message passing: (a) a small B_n, where x and y represent variable nodes and observation nodes respectively; (b) the equivalent factor graph F_n, where D and x are factor nodes and variable nodes respectively; and (c) forward message passing from the leaf (level by level) to the root x_1, and backward message passing (level by level) from the root to the leaves.

To infer the probability for denoising, we associate each variable node x in Bayesian network B with an observation node y and create the arc $y \rightarrow x$. The probability function of x conditioned on the observed value of y is modeled as

$$f_n(x|y) \propto \frac{1}{\sigma_n} \exp(-\frac{(x-y)^2}{2\sigma_n^2}), \tag{10}$$

where σ_n^2 is the variance of the zero mean Gaussian white noise.

We use the message passing algorithm to obtain the estimated wavelet coefficients of each realization. In our implementation, we first convert WBN B_n to a factor graph F_n, and then use the max-product algorithm to derive the estimated wavelet coefficients. In the last step of the max product algorithm, the marginal probability of each variable node V in F_n is calculated based on Equation (3) Let $\mathcal{N}(x)$ represent the neighboring factor nodes of variable node x in F_n. In addition, let x_p and x_c denote, respectively, the parent variable node and child variable node of x in B_n; and let $\{x_j\}$ denote the sibling variable nodes of x in B_n. The value of $\hat{x}$ can be estimated based on whether x has a child node.

Case 1: x has a child node x_c.

$$\begin{aligned} \hat{x} &= \arg\max_x \prod_{D\in\mathcal{N}(x)} \mu_{D\rightarrow x}(x) \\ &= \arg\max_x \left(\frac{1}{x}\exp\left[-J_c(x)\right]\right), \end{aligned} \tag{11}$$

where

$$\begin{aligned} J_c(x) &= \frac{(x-y)^2}{2\sigma_y^2} + \frac{x^2}{2\omega {x_p}^2} + \frac{{x_c}^2}{2\omega x^2} \\ &+ \lambda \sum_j |x - x_j| + \Psi. \end{aligned} \tag{12}$$

In Equation (12), $\Psi = \Psi(y, x_p, x_c, \{x_j\})$ is independent of x, and σ_y^2 is the variance of the wavelet coefficients associated with observation node y. The variance σ_y^2 can be written as $\sigma_n^2\rho$, where ρ depends on the scale and the wavelets. We use

(a) (b) (c) (d)

Fig. 3. Comparison of the denoised images derived by $BLS-GSM$, $BM3D$, and our algorithm. The noise standard deviation is $\sigma = 25$: (a) the original Lena image; (b) the denoised result of the $BLS-GSM$ algorithm; (c) the denoised result of the $BM3D$ algorithm; (d) the denoised result of our algorithm.

an iterative quadratic approximation to estimate the root $\hat{x}$ of $\frac{1}{x} + J_c'(x) = 0$.

Case 2: x does not have a child node (x is a node at the finest wavelet scale). We can set $x_c = 0$ in Equation (12) and obtain

$$J(x) = \frac{(x-y)^2}{2\sigma_y^2} + \frac{x^2}{2\omega {x_p}^2} + \lambda \sum_j |x - x_j| + \Psi. \qquad (13)$$

We also use an iterative approximation to estimate the root $\hat{x}$ of $J'(x) = 0$.

Table 1

Image	Method	PSNR		
		$\sigma_n = 15$	$\sigma_n = 25$	$\sigma_n = 35$
Einstein	$BLS-GSM$	32.6818	31.0201	29.9372
512×512	$BM3D$	33.0331	31.4186	30.3777
	Our	**33.2768**	**31.7125**	**30.6061**
Barbara	$BLS-GSM$	30.7724	27.8214	25.9775
512×512	$BM3D$	33.0666	30.7176	28.8879
	Our	**33.2223**	**30.8669**	**28.6159**
Lena	$BLS-GSM$	34.1105	31.7891	30.278
512×512	$BM3D$	34.8782	32.5501	31.0301
	Our	**35.0229**	**32.6287**	**31.0807**
Baboon	$BLS-GSM$	27.8549	24.9483	23.2643
512×512	$BM3D$	28.139	25.3495	23.6554
	Our	**28.3655**	**25.5593**	**23.8645**

5. THE PROPOSED DENOISING ALGORITHM AND EXPERIMENTAL RESULTS

For the experiments, we downloaded several gray scale images of size 512×512 from USC-SIPI image database [13], and added different amounts of white Gaussian noise to them. The parameter settings of the WBN denoising algorithm evaluated in the experiments are: $J = 4$ (the number of wavelet decompositions) , $\omega = 0.64$ (Equation (8)), $\lambda = 0.45$ (Equation (9)). Each subband is of a 512×512 image and contains 4×4 nodes ($m = 4$ in Section 3A).

Table 1 lists the PSNR results of three compared methods for four images in different noisy environments with $\sigma_n = 15, 25$ and 35. The proposed method outperforms the $BM3D$[11] and $BLS-GSM$[12] methods on all images in each noisy environment. The improvement derived by our method is not significant; even so, the results are encouraging because this is the first time that an algorithm has outperformed $BM3D$ without adopting non-local means denoising techniques. In Fig.3, we compare some images that were denoised by the three methods.

6. CONCLUSION

For image denoising, we use a Bayesian network constructed from the estimated wavelet coefficients of the input image. Different images can yield different wavelet Bayesian networks. To derive the wavelet coefficients, we use the standard probability inference algorithms for graph models. In general, a large number of training images are required to robustly estimate the parameters used to construct a Bayesian network. In our construction, it is possible to use non-parametric statistical techniques to derive a Bayesian network from a single image. We compare the denoised results of several images containing various levels of noise and demonstrate that the PSNR performance of our method is uniformly better than that of two state-of-the-art algorithms.

7. REFERENCES

[1] Edited by M. J. Jordan, "Learning in Graphical Models", *The MIT Press*, 1998.

[2] D. Heckerman, D. Geiger and D. M. Chickering, "Learning Bayesian networks: The combination of knowlege and statistical data", *Machine Learning*, Vol.20, 1995, pp.197-243.

[3] M. S. Crouse, R. D. Nowak and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models", *IEEE Transactions on Signal Processing*, Vol.46, Issue.4, Apr. 1998, pp.886-902.

[4] B. J. Frey, "Graphical Models for Machine Learning and Digital Communication", *The MIT Press*, 1998

[5] S. M. Aji and R. J. McEliece, "The Generalized Distributive Law", *IEEE Transactions on Information Theory*, Vol.46, Issue.2 , Mar 2000, pp.325-343.

[6] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference", *Margan Kaufmann Publishers, INC*, 1988.

[7] C. K.Chow and C. N. Lui, "Approximating Discrete Probability Distributions with Dependence Trees", *IEEE Transactions on Information Theory*, Vol.14, 1968, pp.462-467.

[8] D. Geiger, "An Entropy-based Learning Algorithm of Bayesian Conditional Trees", *UAI'92*, 1992, 92-97.

[9] J. B. Kruskal, "On the Shortest Spanning Subtrees of a Graph and the Traveling Salesman Problem", *Proceedings of the American Mathematical Society)*, Vol. 7, No. 1, 1956, pp. 48-50.

[10] E. P. Simoncelli, "Bayesian Denoising of Visual Images in the Wavelet Domain", *Bayesian Inference in Wavelet Based Models*, Chapter 18, pp 291–308, Lecture Notes in Statistics, Vol.141 Springer-Verlag, New York, 1999.

[11] K. Dabov, A. Foi, V. Katkovnik, and Karen Egiazarian, "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering", *IEEE Transactions on Image Processing*, Vol.16, No:8, Aug. 2007, pp.2080-2095.

[12] J. Portilla, V. Strela, M. J. Wainwright and E. P. Simoncelli, Image Denoising using Gaussian Scale Mixtures in the Wavelet Domain", *IEEE Transactions on Image Processing*, Vol.12, Issue.11, Nov. 2003, pp.1338-1351.

[13] "The USC-SIPI Image Database", http://sipi.usc.edu/database/index.html