Abstract: From the perspective of the Bayesian approach, the denoising problem is essentially a prior probability modeling and estimation task. In this paper, we propose an approach that exploits a hidden Bayesian network, constructed from wavelet coefficients, to model the prior probability of the original image. Then, we use the belief propagation (BP) algorithm, which estimates a coefficient based on all the coefficients of an image, as the maximum-a-posteriori (MAP) estimator to derive the denoised wavelet coefficients. We show that if the network is a spanning tree, the standard BP algorithm can perform MAP estimation efficiently. Our experiment results demonstrate that, in terms of the peak-signal-to-noise-ratio and perceptual quality, the proposed approach outperforms state-of-the-art algorithms on several images, particularly in the textured regions, with various amounts of white Gaussian noise.

Keywords: Bayesian network, image denoising, 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 such phenomena by decomposing joint probability distributions into a set of local constraints and dependencies [1]. 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 (directed) graphical model. In this paper, our objective is to construct a Bayesian network from a single image for denoising purposes. To do this, we need to overcome two difficulties: 1) constructing a Bayesian network is computationally in efficient and 2) the data over-fitting problem, which exaggerates minor fluctuations in the input data.

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] - [5]. In practice, an optimization process based on a heuristic search technique is used to find the best structure over the space of all possible networks.

However, the approach is computationally intractable because it must explore several combinations of dependent variables to derive an optimal Bayesian network. The difficulty is resolved in this paper by representing the data in wavelet domains and restricting the space of possible networks by using certain techniques, such as the “maximal weighted spanning tree” (MWST). Three wavelet properties-sparse, clustering, and persistence-can be exploited to reduce the computational complexity of learning Bayesian network. First, the wavelet transform of a natural image tends to be sparse with large coefficients at the edges. The sparsity reduces the number of variables required to construct a graph. Second, the adjacent wavelet coefficients tend to have similar values as a cluster. Third, wavelet coefficients at the same location and orientation tend to be positively correlated in adjacent scales.

The over-fitting problem occurs because the underlying network is too complex; for example, there may be too many parameters for the number of observations [6]. In the effort of using hidden Markov tree (HMT) model [7]-[9] to capture the joint statistics of wavelet coefficients, the marginal probability of each wavelet coefficient is modeled as a mixed density function with a hidden state variable; for example, an M-state Gaussian mixture model for a wavelet coefficient consists of M states and M Gaussian conditional probability density functions (pdf), one for each state.

To generate wavelet coefficients, the HMT first draws a state value based on the pdf $p$, and then draws an observation according to the conditional probability $f$ of the state. The pdf of the wavelet coefficient is given by $s=1 p(s|j)$, where the conditional pdf $f(s|j)$ is written as a parametric formula that depends on certain parameters. Because several parameters are used to estimate a wavelet coefficient, the HMT approach can only be used to model the marginal probability when the number of training images is large; then the underlying parameters can be estimated accurately, if there is only one image, the over-fitting problem under the HMT approach would be serious. Thus, instead of associating each wavelet coefficient with a random variable, we split all the wavelet coefficients into equal-size blocks and assume that the blocks are independently sampled from a matrix of random variables (called a wavelet patch). The approach allows us to estimate non-parametric statistics, which do not require a pdf assumption, from the samples in the wavelet patch.

The image denoising problem is particularly serious in modern image capturing devices because the increase in the sensor’s density per unit area of a chip reduces the signal-to-noise (SNR) and increases the capturing device’s sensitivity to noise [10]. The state-of-the-art denoising algorithms are based on the non-local means approach [11]-[13], which exploits the self-similarity and redundancy in an image. The most representative approach is the block matching and 3-D filtering (BM3D) algorithm [13]. It combines similar 2-D patches that can be over lapped to form a 3-D group, and the nuses collaborative 3-D filtering techniques to perform non-local filtering. The filtered blocks are returned to their original positions, and the final estimate of a pixel is...
computed as the weighted average of the estimates of the pixel in several different blocks. This simple approach is very efficient and it yields better results than regularized local mean approaches. In [14], denoising Wiener filter, motivated by the statistical analysis of the performance bounds of patch-based methods, is proposed. The filter’s parameters are estimated from geometrically as well as photometrically similar patches. Recent developments in sparse representations have been used together with thenon-local mean approach for image denoising purposes [15],[16]. The sparse model approach assumes that image patches can be represented sparsely by an over-complete redundant dictionary, which can be learned from a family of training data. In [17], Milanfar demonstrates that the non-local structure, the modeling of which depends on known variance, can be extracted adaptively to the local structure of an image. He also presents a general framework for understanding the basic principles behind the approaches.

The Bayesian approach is also widely used to solve the image demarking problem. The Bayesian formula indicates that the denoising problem is essentially a prior probability modeling and estimation task. If $y=x+n$, where $n$ is white Gaussian noise with known variance, then the Bayesian formula is $P(y|x)=P(y|x)P(x)/P(y)$. The joint probability of the pixels in its neighborhood is modeled by Markov Random Fields (MRFs), where the probability of a pixel depends solely on the joint probability of the pixels in its neighborhood. According to the Hammersley-Clifford theorem, the probability distribution of an MRF is the Gibbs distribution whose energy function is the sum of the potential functions defined on the cliques (i.e., maximal complete subgraphs) in image neighborhoods.

Many wavelet-based denoising algorithms integrate the wavelet properties in MRF stores in the structure of a denoised image. The main differences between the algorithms are the methods used for neighborhood selection, the modeling of the original image over the neighborhood, and the techniques employed to derive solutions. In [18], the estimated wavelet coefficient at index $l$ (position, scale), $w_l$, is obtained by

$$w_l = P(x_l|M) = w_l P(M|x_l) P(x_l),$$

where $w_l$ is the observed wavelet coefficient, $x_l$ is the hidden label, $P(x_l|M)$ is the probability of the label $l$ as a nudge, and $M$ is the measurement derived by inter-scale Lipschitz exponent estimation [19],[20].

The prior probability $P(x_l|l)$ at the scale of $l$ is modeled by a $2	imes 2$ MRF, where the potential functions between two neighboring wavelet coefficients on hand-chosen cliques are defined as $x_{l,j}$, with each variable taking the value 0 or 1. The approach is extended in [21] by incorporating robust inter-scale estimators $M$ and $P(x_l|M)$ and generalized an isotropic MRF prior $P(x_l)$ on each scale.

Simon Celli examined the empirical statistical properties of images in a adjacent scales and presented an inter-scale probability model for the wavelet coefficients in two adjacent scales [22]. The joint statistical model assumes that the density of an estimated wavelet coefficient is conditionally Gaussian, where the variance is a linear combination of the squared coefficients in a local neighborhood.

The neighborhood consists of coefficients at other orientations and adjacent scales, as well as adjacent spatial locations. For example, Simon Celli’s model uses a neighborhood comprised of the 12 nearest spatial neighbors in the same sub band, the 5 nearest coefficients in sub bands at other orientations on the same scale, the 9 nearest coefficients in the adjacent subband of the coarser scale, and some coefficients in other subbands.

The BLS-GSM algorithm [23] models the distribution of a vector of wavelet coefficients in a $3 \times 3$ region, together with the coefficient at the center location and the same orientation at the next coarser scale, as a Gaussian Scale Mixture (GSM). Then, the Bayesian least square method is used to estimate the wavelet coefficient at the center of the neighborhood system. In [24], a mixture of Gaussian Scale Mixtures (MGSM)is proposed to make the GSM model adaptive to the image content. Its denoising performance is almost as good as that of the BM 3D algorithm. In [25], a dimension reduction technique is applied to the MGSM to reduce the computational cost and avoid the curse of dimensionality problem in learning the high number of free parameters of the model.

The MRF is modeled on a run directed graph. In this paper, we propose an approach that uses a hidden directed graph to model the prior probability of an image. Specifically, the graph is a Bayesian network with a multi-layer network structure constructed from the wavelet coefficients of an image. Our approach has two advantages over existing approaches.

First, the MAP solution can be derived by using the standard BP algorithm [26], [27]. BP inference passes messages forward from coefficients at coarser scales to finer scales and backward from finer scales to coarser scales. Forward message passing tends to smooth fluctuations in fine scales, while backward message passing tends to retain the finer details of an image. Thus, our method is more capable of retaining the fine structure of an image than existing approaches. The second advantage is that the hidden structure is derived by a data-adaptive process.

In addition, the prior probabilities over inter-scale edges and intra-scale edges are modeled in a similar way to those in [22] and [28] respectively.

We also analyze the complexity of estimating the solution of a Bayesian network and show that BP inference can derive
the MAP solution efficiently provided that the Bayesian network is a spanning tree.

To evaluate our approach, we compare its performance with that of other approaches, including BM 3D, and demonstrate that it yields a better peak-signal-to-noise ratio (PSNR) as well as better perceptual quality on the textured as of an image.

The remainder of this paper is organized as follows. In Section II, we explain the rationale behind the proposed data-adaptive graph modeling of an image. We also analyze the computational cost of using BP inference to estimate the MAP solution for the proposed approach. In Section III, we present the method used to construct and model wavelet Bayesian networks; and in Section IV, we describe how the networks are used for denoising. In Section V, we discuss the proposed denoising algorithm and compare its performance with that of other approaches. Section VI contains some concluding remarks.

Two problems may arise during the above construction procedure: 1) the coefficient and wavelet patch association problem, which involves associating subband coefficients with a wavelet patch; and 2) the graph selection problem, i.e., determining the type of graph to construct. To solve the first problem, we propose the following heuristic procedure. Assume that the wavelet patch is a matrix of \( m \times m \) random variables. Let the size of a subband be \( N \times N \) and let \( m \) divide \( N \). We partition the subband into \( m \times m \) rectangular blocks, each of which contains \( m \times m \) coefficients. Then, the coefficient at location \((i,j)\) in each block is assigned as a realization of the random variable at location \((i,j)\) in the wavelet patch. Thus, each random variable has \( m^2 \) sampled observations.

For the second problem, we analyze the computational cost of a graph structure for which the MAP solution can be derived efficiently by the standard BP algorithm. The standard implementation of the message passing algorithm in BP on \( m \times m \) cliques runs in \( O(N^2 k m^{-m} T) \), where \( N^2 \) is the number of coefficients in a subband, \( k \) is the number of labels for each coefficient, and \( T \) is the number of iterations. Basically, computing each message takes \( O(k m^{-m}) \) time and there are \( O(N^2) \) messages per iteration. The computational cost of the conventional BP algorithm can be reduced if the algorithmic techniques in [29]–[31] are used for Bayesian inference.

If parallelism is also exploited as described in [32], BP inference can achieve a near linear parallel scaling performance. We use a graph \( G(V, E) \) to represent the (one-layer) subgraph structure, where \( V \) is the node set and \( E \) is the arc set. If \( G \) is a loopy Bayesian network, the BP algorithm sometimes yields surprisingly good approximate results; however, sometimes, it fails to produce any results, even after a large number of iterations. BP can derive an exact MAP solution in two iterations if \( G \) is a directed-acyclic-graph (DAG), but the inference cost depends on the structure of the DAG. A DAG that incurs a high inference cost can be constructed as follows: let the nodes in \( G \) be indexed from 1 to \( m^2 \). For each node \( j \), passing at node \( j \). For typical values of \( k \) and \( m \), which are 512 and 4 respectively, BP inference of the DAG is too high to be of practical use.

If we assume that \( G \) is a spanning tree structure, we can show that the average computation time for BP inference is
The tree contains \( O(m^2 k) \) operations on a node. If the average number of message passing operations is at most 1, the average number of messaging passing arcs of a node is at most \( k \). Because the average in-degree and out-degree arc of a node are \( k \) and \( k \), respectively, to pass Messages to a node.

The joint probability of a spanning tree \( G = (V, E) \) can be formulated by the dependency structure in G as follows.

Let \( f(u | v_i) \) be the probability function associated with arc \( u \rightarrow v_i \), where \( u, v_i \in V \), and let \( u \) be the root of the tree with probability \( f(u) \). Then, the joint probability of G is

\[
f(G) = f(v_i | u) f(u).
\]

Note that the intra-scale clustering property of wavelet coefficients indicates that the neighboring coefficients in a subband are positively correlated. Thus, the pairwise joint probability can be modeled as a measurement of \( |u_i - v_j| \), resembling the marginal statistics of the gradient of neighboring nodes with values \( u_i \) and \( v_j \). We can stack the subgraphs of two adjacent scales to form a two-layer graph. Let \( G_{c} = (V_c, E_c) \) be the graph corresponding to a subband at the coarser scale of \( G \). Based on the inter-scale persistence property, inter-scale arcs can be constructed between \( V_c \) and \( V \), denoted as \( A \), by combining the parent node and child node (nodes at the same location) in \( G_c \) and \( G \) respectively. Let \( p(vk) \) in \( V_c \) be the parent node of \( vk \) in \( V \); and let \( uc \) and \( u \) be the root of the tree in \( G_c \) and \( G \) respectively. Because of the node dependency in the graphs, the joint probability of the resulting directed graph can be formulated as follows:

\[
f(G_c, G) = f_c(u_c) f_c(u) f(u | p(u)) f(v) f(v | \phi(v, p(v))) \]

where \( p(u) \rightarrow u \in A \), and \( p(v) \rightarrow v \in A \). Equation (3) gives the joint probability of all the nodes in the graph as shown in Fig.2(c). The prior probability of the hidden coefficients in the graph is derived as follows. We take one coefficient from each node to form a data group; for example, Fig. 2(d) has four data groups.

The joint probability of each data group can be written in the form of Equation (3) by replacing the node variables with the hidden coefficients. Finally, it is assumed that the joint probability of all the data groups is the product of that of each data group.

After constructing the hidden network, we create layers of observation nodes for each noisy wavelet subband. Next, we assign the noisy coefficients to the observation nodes in the same way as the ideal subband coefficients are assigned to the hidden nodes, as shown in Fig.3(a) and (b).

Then, arcs are created to link the observation nodes to the corresponding hidden nodes, as shown in Fig.3(c). BP
inference can now be applied to derive the MAP solution for the denoising problem. Recall that each node in a wavelet patch has \( m^2 \) realizations. If we take one wavelet coefficient from each node and use the BP algorithm to estimate the solution, then denoising process makes \( mN \) BP inferences.

Note that, as shown in Fig. 3(d), all the wavelet coefficients located in the same rectangular blocks in various subbands are estimated simultaneously by one BP inference.

We calculate the computational cost of image denoising under the proposed model as follows. Because each node has one in-degree and one out-degree inter-scale edge at most, the number of operations that BP inference performs for the multi-layer graph is \( J \) times the number performed for one wavelet patch.

For a spanning tree, each inference involves an average of \( O(m^2) \) operations and there are \((m^2)^2\) BP; thus, the proposed algorithm performs an average of \( O(N^2k) \) operations for one wavelet patch. As a result, the average BP inference cost of the \( J \) layers network is \( O(JN^2k) \).

\[
\begin{align*}
B &= (V, E, P) \\
V &= \{ h, v, u \} \\
E &= \{ (h, v), (v, u) \} \\
P &= \{ (h, v),odel relation. \}
\end{align*}
\]

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 \( x \rightarrow y \), where \( x \) and \( y \) are elements in \( V \). The likelihood \( p(y|x) \in P \) of an edge \( x \rightarrow y \) is the conditional probability of observing \( y \) given that \( x \) exists.

We call the Bayesian networks constructed in wavelet domains wavelet Bayesian networks (WBNs). Our primary objective is to construct a WBN from the undecimated discrete wavelet transform (DWT) of a single image. Initially, wavelet decomposition of an image \( F \) yields three images of wavelet coefficients with horizontal, vertical, and diagonal orientations respectively, and one approximate image of \( F \). Then, at the next scale, the approximated image is further decomposed to obtain three images of the wavelet coefficients and one coarser approximate image of \( F \).

Let \( W^h_j \) denote the horizontal-group, \( W^v_j \) the vertical-group, and \( W^d_j \) the diagonal-group of wavelet coefficients.

\[
W^h_j \mathcal{F}(u, v), W^v_j \mathcal{F}(u, v), \text{ and } W^d_j \mathcal{F}(u, v) \text{ denote, respectively, the horizontal, vertical, and diagonal images of the wavelet coefficients at scale } j \text{.}
\]

Let \( A_j \) represent the approximated image at the same scale. If the undecimated DWT is decomposed \( J \) times, we will have wavelet coefficients \( W^h_j \mathcal{F}, W^v_j \mathcal{F}, \text{ and } W^d_j \mathcal{F} \) with \( j = 1, \ldots, J \).}

To construct a WBN, we first group subbands with the same orientation together to obtain a horizontal-group, vertical-group, and a diagonal-group of wavelet coefficients. Then, we construct a Bayesian network \( B \) for each group. Let \( B^h = (V^h, E^h, P^h) \), \( B^v = (V^v, E^v, P^v) \), and \( B^d = (V^d, E^d, P^d) \) denote the Bayesian networks constructed from the horizontal-group, vertical-group, and diagonal-group of wavelet coefficients respectively.

The WBN \( B = (V, E, P) \) is derived from \( B^h \), \( B^v \) and \( B^d \) by:

- \( V = V^h \cup V^v \cup V^d \) \hspace{1cm} (4)
- \( E = E^h \cup E^v \cup E^d \) \hspace{1cm} (5)
- \( P = P^h \cup P^v \cup P^d \) \hspace{1cm} (6)

Next, we explain how to construct the Bayesian network \( B^h = (V^h, E^h, P^h) \) that corresponds to the \( h \)-orientation, where \( a \in \{ h, v, d \} \).

\[
\begin{align*}
\text{(a) Top layer and the bottom (dashed) layer of each group are composed of hidden subband coefficients and coefficients observed at the corresponding subband, respectively. (b) Coefficients in observed (dashed) layers are organized in the same way as the coefficients in the hidden layers. (c) Observation nodes } &\{ y_i , y_i \} \text{ are created and linked to hidden nodes } \{ x , x \}. \\
\text{(d) For denoising proposes, data in the network is organized into four groups and BP inference is applied to each group. For example, } &\{ \{ y_i, p_i \} , p_i \text{, yai } , a(i = 1, 2, 3, 4) \text{ is a group. The other three data groups can be derived in a similar manner.}
\end{align*}
\]
A. Vertex Set \( V^o \)

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. Let a wavelet patch (matrix of random variable) be of size \( m \times m \). For each subband, a graph of \( m^2 \) variable nodes are formed. We then associate each random variable in the wavelet patch to a variable node \((i,k)\) blocks, each of size \( m \times m \). Then, \( (m^2)^2 \) wavelet coefficients sampled from the subband are assigned to each variable node.

Let \( x_{ij}(i,k) \), with \( j = 1, \ldots, J \) and \( i, k = 0, \ldots, m-1 \), denote the \((i,k)\) variable node in the \( j \)-th subband. In our \( J \) construction, the \((m^2)^2\) wavelet coefficients assigned to nodex \( j \) \((i,k)\) are sampled from \( W^j \) \( F(i+mp, k+mq) \) with \( p, q = 0, \ldots, m^2 - 1 \). If we represent each node as a vertex in the Bayesian network \( B^o \) will be \( V^o = \{ x_{ij} \} \) \( j = 0, \ldots, m-1 \), \( i = 1, \ldots, J \} \) and the \((m^2)^2\) wavelet coefficients can be regarded as sampled from some unknown distribution of a random variable. Figs. 1(a), (b), and (c) show the procedure used to construct the vertex set for a subband of \( 4 \times 4 \) coefficients.

B. Edge Set \( E^o \)

The arcs (directed edges) in \( B^o \) can be divided into two disjoint sets, \( E^o_{\text{in}} \) and \( E^o_{\text{out}} \), where \( E^o_{\text{in}} \) comprises the (interscale) edges incident to vertices at different scales, and \( E^o_{\text{out}} \) comprises 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. The property can be used to construct arcs in \( E^o_{\text{in}} \) by linking a vertex at the coarser scale \( j+1 \) to the vertex of the same index at the finer scale \( j \); that is,

\[
E^o_{\text{in}} = \{ \{ x^{|j+1|(i,k)} \rightarrow x^{|j|(i,k)} \} \mid j = 0, \ldots, m-1 \}
\]

The edges in \( E^o_{\text{in}} \) represent the connections between vertices at the same scale and orientation. Constructing the edges corresponds to deriving the Bayesian network on the nodes \( x^{|j|(i,k)} \) that best represent the joint probability of the nodes at the same scale \( j \) and orientation \( n \). However, as discussed in Section II, BP inference is computationally intractable if the Bayesian network is a general graph. Thus, 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 [33]–[36]. A maximal weighted spanning tree is a spanning tree whose weight is greater than or equal to the total weight of every other spanning tree. The optimal weighted spanning tree can be derived by minimizing the relative entropy (Kullback-Leibler distance) \( D(p|q) \) between the probability functions \( p \) and \( q \).
Algorithm 1 Kruskal’s Algorithm

1) Calculate the $m^2 \times m^2$ weights of the arcs between any two nodes. Note that the weight of arc $x \rightarrow y$ may be different from that of arc $y \rightarrow x$.
2) Sort the weights of the $m^2 \times m^2$ arcs and compile a list in non increasing order so that the weight $w_i$ is not less than the weight $w_j$ with $i < j$.
3) Assume that the initial spanning tree is empty and add the arc of weight $w_1$ to the empty tree.
4) Do until the tree is a spanning tree:
   a) Let $w_i$ be the next weight on the list.
   b) Check if adding the arc of weight $w_i$ to the tree creates a cycle.
5) End

and the sample probabilities are calculated as follows:

$$
\begin{align*}
\hat{f}(x^m_u, a) &= s(\hat{f}(x^m_u, x^m_v)) \\
\hat{f}(x^m_v) &= a\hat{f}(x^m_u, x^m_v)
\end{align*}
$$

(15)

It is well-known that the joint frequencies $f(x^m_u, x^m_v)$ and $f(x^m_v)$ are maximum-likelihood estimators for the probabilities $p(x^m_u, x^m_v)$ and $p(x^m_v)$ respectively. Therefore, the sample mutual information can be computed as

$$
\begin{align*}
I(x^m_u, x^m_v) &= \sum_{x^m_u, x^m_v} f(x^m_u, x^m_v) \log \frac{f(x^m_u, x^m_v)}{\hat{f}(x^m_u) \hat{f}(x^m_v)} \\
&= \sum_{x^m_u, x^m_v} f(x^m_u, x^m_v) \log \frac{f(x^m_u, x^m_v)}{\hat{f}(x^m_u) \hat{f}(x^m_v)}.
\end{align*}
$$

(16)

Then, we use $\hat{I}(x^m_u, x^m_v)$ instead of $I(x^m_u, x^m_v)$ when calculating the weights in the above algorithm. For each scale $\ell \in \{1, \ldots, J\}$, we execute Kruskal’s algorithm to obtain a contains all the edges in all spanning tree. The edge set $E^\ell$ the spanning trees.

Figure (d) shows an example of a network with intra-scale edges derived by the MWST algorithm from the fournodes in Fig. 1(c); and Fig. 2(c) shows an example of a multilayer network where the inter-scale and intra-scale edges are reconstructed from the nodes in Fig. 2(b). The WBN $B$ in Fig. 4 is comprised of three oriented Bayesian networks, $B^0, B^1$ and $B^2$.

C. Probability Model $P^o$

There are two types of arcs in a Bayesian network $B^o$: 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. To obtain the probability inference, we need to model the probability function on each arc.

Simoncelli [22] 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.
4. Wavelet Bayesian Networks For Denoising

In this section, we consider using the wavelet Bayesian network to model the prior probability of the original image for the image denoising problem, which involves removing white and homogeneous Gaussian noise with zero mean and known variance from an image. 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

\[
 f(x|y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-x)^2}{2\sigma^2}\right)
\]

(20)

where \( \sigma^2 \) is the variance of the zero mean Gaussian white noise and \( \rho \) depends on the scale and wavelets.

Recall that the variable nodes in the Bayesian network \( B \) are represented by \( \{x_j^i, y_j^i, \cdots, y_j^d, \cdots, y_j^k\} \) \( j = 1, \ldots, J, i; k = 0, \ldots, m - 1 \) (Equation (7)). Let \( y_j^i \) denote the observation node corresponding to \( x_j^i \), and \( P_{in} \) denote the collections of \( y_j^i \) and \( Y_j \). Let \( Y_j \rightarrow E_j^i(i, k) \), the arcs \( \{y_j^i, (i, k) \rightarrow x_j^i \} \), and the probability functions \( \{f(y_j^i|x_j^i(i, k))\} \) respectively. The WBN \( B_{(i, k)} \) for image denoising is constructed and represented as \( B_{n} = (VUY, EU_{E_{n}}, P_{in}) \).

Let a noisy image \( Z = F + N \), where \( F \) is the original image and \( N \) is zero-mean white Gaussian noise. As shown by the simple example in Fig. 3(d), each wavelet coefficient of \( Z \) is assigned to one observation node in \( B_n \). That is, the coefficient \( W_j^i(Z(i+mp, k + mq)) \) is assigned to observation node \( y_j^i \), where \( p, q = 0, \cdots, m^2 - 1 \), if \( y_j^i \) has observation value \( \{f(y_j^i|x_j^i(i, k))\} \), and \( \{f(y_j^i|x_j^i(i, k))\} \) respectively. The WBN \( B_{(i, k)} \) for image denoising is constructed and represented as \( B_{n} = (VUY, EU_{E_{n}}, P_{in}) \).

We use the message passing algorithm to obtain the estimated wavelet coefficients of each realization. First, we convert \( B_n \) to a factor graph \( F_n \) and then use the max product algorithm to derive the estimated wavelet coefficients. The conversion of \( B_n \) to \( F_n \) and the max-product message passing schemes are standard techniques. For completeness, we provide them in Appendix A. The last step of the maxproduct algorithm calculates the marginal probability of each \( x \) in \( F_n \). Let\( N(x) \) represent the neighboring factor nodes of variable node \( x \) in \( F_n \). Variable nodes \( x \) and \( x_j^i \) denote, respectively, the parent variable node and childvariable node of \( x \) in \( B_n \) and let \( \{x\} \) denote the siblingvariable nodes of \( x \) in \( B_n \). The value of \( x \) can be estimated based on whether \( x \) has a child node.

Case 1: \( x \) has a child node \( x_c \).

\[
\max_{x=x_c} P_{in}(N(x)) \exp(-J_c(x))
\]

(21)

where

\[
 J_c(x) = \frac{x^2}{2\sigma^2} + \frac{x_2^2}{2a^2\sigma^2} + \frac{x_2^2}{2a^2\sigma^2}
\]

(22)

In Equation (22), \( \_ = (x, x_j^i, \{x\}) \) is independent of \( x \), is the variance of the wavelet coefficients associated with observation node \( y \). The variance \( \sigma^2 \) can be written as \( \sigma^2 \rho \), where \( \rho \) depends on the scale and the wavelets. In Appendix B, we show that \( \rho = 1 \) if the wavelets are orthogonal. Let

\[
 K(\rho) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{\rho^2}{2\sigma^2})
\]

(23)

According to Equation (22)

\[
 J_c(x) = \frac{x^2}{\sigma^2} + \frac{x_2^2}{2a^2\sigma^2} + \frac{x_2^2}{2a^2\sigma^2}
\]

(24)

Let \( \rho = 1 \) in Equation (22)

\[
 x_j^i = \frac{x^2}{\sigma^2} + \frac{x_2^2}{2a^2\sigma^2} + \frac{x_2^2}{2a^2\sigma^2}
\]

(25)

If \( \rho = 0 \), it returns 0. Hence

\[
 J_c(x) = \frac{x^2}{\sigma^2} + \frac{x_2^2}{2a^2\sigma^2} + \frac{x_2^2}{2a^2\sigma^2}
\]

(26)

Algorithm 2 WBN Denoising Algorithm

1. Wavelet representation: calculate the ruled decimated DWT of an \( N \times N \) noisy image \( Z \) to obtain the horizontal, vertical, and diagonal subbands \( \{W_j^i(Z)|i=1,\ldots,J\} \), \( \{W_j^v(Z)|j=1,\ldots,J\} \), and \( \{W_j^d(Z)|j=1,\ldots,J\} \), respectively, as well as the coarsest approximate image \( A_Z \).

2. Create a vertex set \( V^n \): let the parameter \( m \) divide Foreach subband \( W_j^i \) \( \_ \)create Bernoulli

3. Create sibling edges \( (E_i^n) \) for nodes at the same scale and orientation: derive the empirical probability of each variable from the frequency counts of \( mN2 \) wavelet coefficients assigned to the node; then, use Kruskal’s algorithm to derive the maximal weighted spanning tree from the \( m^2 \) nodes in each subband.

4. The root of \( J \) (\( x = 0 \)) can be derived by the following fixed point algorithm. Let \( x \) be the estimated value after the \( t \)-th iteration. Then, \( x \) it can be derived from \( x_t \) as follows:
The iterative estimation stops when the difference between the values \( x_{t-1} \) and \( x_t \) is smaller than some given threshold.

\[
x_{t} = \frac{1}{1 + \sigma^2_{x_{t-1}}} 2 y - \sigma x_{t-1} (x_{t-1} - x_{t})
\]

5. Denoising Algorithm and Experimental Results

In this section, we present our denoising algorithm, explain its implementation, and compare its performance with that of other methods. The proposed algorithm is summarized in Algorithm 1. The steps are as follows: Step (1) calculates the undecimated DWT of the input image; Steps (2) to (5) construct the WBN \( B_{n} \); and Steps (6) to (8) create the WBN \( B_{n} \) used for denoising purposes; In Step (9), the wavelet coefficients are estimated from \( B_{n} \) by applying the max-product algorithm to the factor graph \( F \) for each realization of \( B_{n} \). We use CDF 9/7 filters to process the undecimated DWT. Because CDF 9/7 filters are close to orthonormal wavelet filters, the noise variance of subbands at all scales can be set at \( \sigma^2_{x} \), which is the image noise variance (see Appendix B). The variance \( \sigma^2_{x} \) is used in the Wiener filtering in Step (2) as well as in deriving the MAP estimation of the wavelet coefficients in Step (9). The frequency count in Step (4) indicates the number of wavelet coefficients in a quantization bin. The size of a subband's quantization bin is set at 14 of the standard deviation, measured from the wavelet coefficients in the subband.

We conduct experiments on two sets of images. Set I contains nine gray scale images (size 512 × 512 or 256 × 256) downloaded from the USC-SIPI image database [43]; and Set II (shown in Fig. 6) contains nine gray scale textures, some of which are from the Brodatz texture set. The parameter settings of the WBN denoising algorithm evaluated in the experiments are: \( J = 4 \) (the number of wavelet decompositions), \( \omega = 0.64 \) (Equation (18)), \( \lambda = 0.45 \) (Equation (19)), and the parameter \( m = 4 \) in Step (3). Each subband represents a 512 × 512 image and contains 4 × 4 nodes. The WBN \( B \) has 16 × 4 × 3 variable nodes because there are four subbands in each of the three (note 4 × 4 × 3 nodes, half of which are variable nodes and the rest are observation orientations. Hence, WBN \( B \) assigns 128 × 128 wavelet coefficients. Note that this number is large enough to nodes. For each observation node in \( B \), derive the empirical probability in Step (4) of the WBN algorithm. There are also 128 × 128 realizations of \( B_{n} \).

Next, we present the experimental results derived by our algorithm and compare its performance with that of two state-of-the-art algorithms: the BM3D algorithm [13] and the BLS-GSM algorithm [23]. The source codes for the BLS-GSM and BM3D methods are available from the websites of the respective authors. Our algorithm is executed on an Intel Core2Quad Q9300 CPU, with Windows XP and Matlab2007a. Tables I and II list the averages of five denoised PSNR results of the three compared methods for the images in Sets I and II respectively; white noise was added with \( \sigma^2_{x} = 10, 15, \ldots, 35 \). In the tables, we divide themages into two groups and compare the average PSNR gain of our method over the other methods on the images in each group.

\begin{table}[h]
\centering
\caption{Average PSNR Results for Image Set 1}
\begin{tabular}{|c|c|c|c|}
\hline
Image & Noise Level & \textbf{PSNR}\textsuperscript{a} \textsuperscript{b} & \textbf{PSNR}\textsuperscript{c} \textsuperscript{d} \\
\hline
\textbf{Einstein} & 10 & 34.1406 & 34.3923 \\
& 15 & 32.6818 & 33.0331 \\
& 20 & 31.7412 & 32.1694 \\
& 25 & 30.4306 & 30.8709 \\
& 30 & 28.9372 & 29.2777 \\
\textbf{Boat} & 10 & 33.5080 & 33.8853 \\
& 15 & 31.6452 & 32.1067 \\
& 20 & 30.3194 & 30.8584 \\
& 25 & 29.2446 & 29.5156 \\
& 30 & 28.4395 & 28.9054 \\
\textbf{Barbara} & 10 & 33.1518 & 34.0557 \\
& 15 & 30.7724 & 31.0666 \\
& 20 & 29.0984 & 30.7376 \\
& 25 & 27.8214 & 28.7176 \\
& 30 & 26.7998 & 27.7049 \\
\textbf{Lena} & 10 & 29.9226 & 30.6367 \\
& 15 & 28.4257 & 29.1632 \\
& 20 & 27.2311 & 27.8677 \\
& 25 & 26.0989 & 26.7085 \\
& 30 & 24.9785 & 25.6017 \\
\textbf{Cameraman} & 10 & 33.8779 & 34.1555 \\
& 15 & 31.0334 & 31.8429 \\
& 20 & 29.5762 & 30.3797 \\
& 25 & 28.5242 & 29.4118 \\
& 30 & 27.6872 & 28.5516 \\
\textbf{House} & 10 & 33.2223 & 34.0668 \\
& 15 & 32.7527 & 33.6028 \\
& 20 & 31.2357 & 32.1349 \\
& 25 & 30.2337 & 31.1349 \\
& 30 & 29.1571 & 30.0284 \\
\textbf{Pepper} & 10 & 33.8042 & 34.9889 \\
& 15 & 32.4095 & 33.5840 \\
& 20 & 31.0567 & 32.1671 \\
& 25 & 29.4453 & 30.5232 \\
& 30 & 28.5040 & 29.6090 \\
\textbf{Fingerprint} & 10 & 32.1949 & 33.4628 \\
& 15 & 29.9283 & 31.0979 \\
& 20 & 28.3591 & 29.0855 \\
& 25 & 27.0496 & 27.7112 \\
& 30 & 26.0862 & 26.8245 \\
\textbf{Baboon} & 10 & 30.7353 & 32.0789 \\
& 15 & 28.7349 & 30.0299 \\
& 20 & 26.1239 & 27.6307 \\
& 25 & 24.9483 & 25.3435 \\
& 30 & 23.0463 & 24.4407 \\
\hline
\end{tabular}
\end{table}

Table III summarizes the results in Tables I and II. Note that the Figure print and Baboon images are regarded as texture images because they are dominated by texture information. Interestingly, the PSNR gain of our method over BM3D is 0.1 dB for texture images (Group 2), but only 0.02 dB for the other images (Group 1). Moreover, from Table III, we observe that the gain of BM3D over BLS-GSM is smaller on texture images (0.06 dB) than on the other images (1.0 dB). Texture images contain rich across-scales information. Since both BLS-GSM and our method exploit inter-scale coefficients for data estimation, they are more adaptive to texture images than non-texture images.
In Figures 5, 6, and 7, we perceptually compare some images that were denoised by the three methods. The perceptual quality of the images denoised by BM3D and our method is better than that of the BLS-GSM images. The highlighted regions in the figures compare certain details of the denoised images derived by all three methods with the corresponding

### Table II: Average PSNR Results for Image Set II

<table>
<thead>
<tr>
<th>Image</th>
<th>Noise Level</th>
<th>BLS-GSM PSNR</th>
<th>BM3D PSNR</th>
<th>Ours PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0 512 x 512</td>
<td>10</td>
<td>31.6061</td>
<td>32.52</td>
<td>32.6212</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>29.0063</td>
<td>30.3213</td>
<td>30.4613</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>28.2891</td>
<td>28.8595</td>
<td>28.9214</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>27.7514</td>
<td>27.7379</td>
<td>27.8307</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>25.6629</td>
<td>25.9243</td>
<td>25.9303</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>24.2951</td>
<td>24.5322</td>
<td>24.5588</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>23.3451</td>
<td>23.401</td>
<td>23.4062</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>22.602</td>
<td>22.5878</td>
<td>22.5874</td>
</tr>
<tr>
<td>F7 512 x 512</td>
<td>10</td>
<td>37.7804</td>
<td>37.3043</td>
<td>37.4956</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>35.5299</td>
<td>35.6216</td>
<td>35.6782</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>24.2851</td>
<td>24.3322</td>
<td>24.3568</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>23.3451</td>
<td>23.401</td>
<td>23.4062</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>22.602</td>
<td>22.5878</td>
<td>22.5874</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>21.7442</td>
<td>21.6195</td>
<td>21.6415</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>20.8213</td>
<td>20.6102</td>
<td>20.6102</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>20.0994</td>
<td>20.4042</td>
<td>20.406</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>19.3782</td>
<td>19.549</td>
<td>19.5687</td>
</tr>
<tr>
<td>p3 512 x 512</td>
<td>10</td>
<td>26.305</td>
<td>26.382</td>
<td>26.573</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>24.5449</td>
<td>24.48</td>
<td>24.545</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>22.8854</td>
<td>22.8154</td>
<td>22.8295</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>21.7442</td>
<td>21.6195</td>
<td>21.6415</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>20.8213</td>
<td>20.6102</td>
<td>20.6102</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>20.0994</td>
<td>20.4042</td>
<td>20.406</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>19.3782</td>
<td>19.549</td>
<td>19.5687</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>18.657</td>
<td>18.749</td>
<td>18.7687</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>17.9362</td>
<td>18.046</td>
<td>18.0687</td>
</tr>
<tr>
<td>w6 512 x 512</td>
<td>10</td>
<td>25.5041</td>
<td>25.564</td>
<td>25.5986</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>24.5449</td>
<td>24.48</td>
<td>24.545</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>22.8854</td>
<td>22.8154</td>
<td>22.8295</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>21.7442</td>
<td>21.6195</td>
<td>21.6415</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>20.8213</td>
<td>20.6102</td>
<td>20.6102</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>20.0994</td>
<td>20.4042</td>
<td>20.406</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>19.3782</td>
<td>19.549</td>
<td>19.5687</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>18.657</td>
<td>18.749</td>
<td>18.7687</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>17.9362</td>
<td>18.046</td>
<td>18.0687</td>
</tr>
<tr>
<td>F6 512 x 512</td>
<td>10</td>
<td>22.408</td>
<td>23.146</td>
<td>23.324</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>20.7213</td>
<td>20.5213</td>
<td>20.5213</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>19.9555</td>
<td>19.9042</td>
<td>19.9042</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>19.0555</td>
<td>18.9784</td>
<td>18.9784</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>18.1555</td>
<td>18.0784</td>
<td>18.0784</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>17.2555</td>
<td>17.1784</td>
<td>17.1784</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>16.3555</td>
<td>16.2784</td>
<td>16.2784</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>15.4555</td>
<td>15.3784</td>
<td>15.3784</td>
</tr>
<tr>
<td>F8 512 x 512</td>
<td>10</td>
<td>25.5401</td>
<td>25.564</td>
<td>25.5986</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>24.5449</td>
<td>24.48</td>
<td>24.545</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>22.8854</td>
<td>22.8154</td>
<td>22.8295</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>21.7442</td>
<td>21.6195</td>
<td>21.6415</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>20.8213</td>
<td>20.6102</td>
<td>20.6102</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>20.0994</td>
<td>20.4042</td>
<td>20.406</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>19.3782</td>
<td>19.549</td>
<td>19.5687</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>18.657</td>
<td>18.749</td>
<td>18.7687</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>17.9362</td>
<td>18.046</td>
<td>18.0687</td>
</tr>
<tr>
<td>S2 512 x 512</td>
<td>10</td>
<td>25.5401</td>
<td>25.564</td>
<td>25.5986</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>24.5449</td>
<td>24.48</td>
<td>24.545</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>22.8854</td>
<td>22.8154</td>
<td>22.8295</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>21.7442</td>
<td>21.6195</td>
<td>21.6415</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>20.8213</td>
<td>20.6102</td>
<td>20.6102</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>20.0994</td>
<td>20.4042</td>
<td>20.406</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>19.3782</td>
<td>19.549</td>
<td>19.5687</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>18.657</td>
<td>18.749</td>
<td>18.7687</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>17.9362</td>
<td>18.046</td>
<td>18.0687</td>
</tr>
</tbody>
</table>

### Table III: Summary of Our PSNR Gain

<table>
<thead>
<tr>
<th>Image Group</th>
<th>Average PSNR Gain versus BLS-GSM</th>
<th>Average PSNR Gain versus BM3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1 (First seven images in Table I)</td>
<td>1.0239</td>
<td>0.0207</td>
</tr>
<tr>
<td>Group 2 (Fingerprint + Balloon + Images in Table II)</td>
<td>0.2922</td>
<td>0.1268</td>
</tr>
<tr>
<td>All Images in Tables I and II</td>
<td>0.9403</td>
<td>0.0594</td>
</tr>
</tbody>
</table>

6. Conclusion

The Bayesian formula indicates that the denoising problem is essentially a prior probability modeling and estimation task. In this paper, we present constructive data-adaptive procedure that derives a hidden graph structure from the wavelet coefficients. The graph is then used to model the prior probability of the original image for denoising purposes. Moreover, we show that if the network is a spanning tree, the standard BP algorithm can estimate MAP.
efficiently. We compare our denoised results with those derived by other approaches, including BM3D, and demonstrate that our method yields a better PSNR and better perceptual quality on the textured areas of an image. Extending our method to content sensitive wavelet patches is an issue that merits future study. We will also investigate ways to speed up our algorithm’s execution time.

7. Acknowledgment

The authors would like to thank the reviewers for their insightful comments, which have helped to significantly improve the quality of this paper.

References