Vibhor Kumar, Jukka Heikkonen, Jorma Rissanen and Kimmo Kaski, Minimum description length denoising with histogram models, IEEE Transactions on Signal Processing, 54 (8): 2922-2928, 2006.

Minimum Description Length Denoising With Histogram Models

Vibhor Kumar, Jukka Heikkonen, Jorma Rissanen, *Fellow, IEEE*, and Kimmo Kaski

***Abstract*—In this paper, we relax the usual assumptions in denoising that the data consist of a "true" signal to which normally distributed noise is added. Instead of regarding noise as the high-frequency part in the data to be removed either by a "hard" or "soft" threshold, we define it as that part in the data which is harder to compress than the rest with the class of models considered. Here, we model the data by two histograms: one for the denoised signal and the other for the noise, both represented by wavelet coefficients. A code length can be calculated for each part, and by the principle of minimum description length the optimal decomposition results by minimization of the sum of the two code lengths.**

***Index Terms*—Complexity, denoising, minimum description length, wavelets.**

I. Introduction

The denoising problem is to separate data $x^n = x_1, \ldots, x_n$, usually real-valued, into a "smooth" signal $\hat{x}^n = \hat{x}_1, \ldots, \hat{x}_n$ and "noise" $e^n = e_1, \ldots, e_n$ as follows:

$$x_t = \hat{x}_t + e_t.$$

The traditionally made assumption is that the data consist of a "true" underlying signal to which noise, having almost always a Gaussian distribution, is added. The problem is then regarded as one of estimating the "true" signal by $\hat{x}_t$, expressed as a linear combination of basis functions such as wavelets. Since the wavelet transform converts the data sequence x^n into an equally long coefficient sequence c^n, one way to do denoising is to retain with help of a threshold a number of the largest coefficients in absolute value, say $c_{(1)}, \ldots, c_{(k)}$, and setting the rest to zero, which by the inverse transform generates the estimate $\hat{x}^n$. Another way is to replace the "hard" threshold with a "soft" one, where a soft-threshold function shrinks the retained coefficients. Such techniques with different thresholds are discussed in [1], [4], [5], [7], [10]–[12], and [19].

The quality of the denoised signal $\hat{x}^n$ clearly depends on what we assume the noise to be and how to model it. The most often made assumption is that noise is the high-frequency part in the data and that it has a Gaussian distribution of zero mean and some variance, which must be estimated from the data. The Gaussian distribution has sometimes been replaced by the so-called generalized Gaussian distributions and used for denoising one-dimensional (1-D) and two-dimensional (2-D) data (see [1], [2], and [8]). The assumption of high-frequency noise creates the problem that the variance of noise or the generalized variance, as it were, is needed before the noise is constructed, which can be resolved by fixing the way the estimation is done.

A drastically different approach was taken in [9], which, in turn, was inspired by related but less refined ideas in [3]. In [9], no "true" underlying signal in the data was assumed. Rather, the data x^n are modeled by a Gaussian distribution with mean $\hat{x}$, determined by a linear combination of the wavelet basis functions. The variance together with the coefficients in the linear combination are the model parameters. With such a model, one can compute the number of bits required to encode both the retained coefficients, defining the denoised signal, and the noise. The minimization of their sum gives the optimal decomposition in the minimum description length (MDL) sense. The rationale for this is that the denoised signal reflects the regular features in the data that can be described with a shorter code length than the noise, which, lacking the regular features represented by the models, is harder to compress. This corresponds to intuition about noise, which should look random in light of the descriptive power of the models chosen. For those unfamiliar with the MDL principle, we mention that it is a global maximum-likelihood principle, global because it includes both the values of the parameters and their number as well as the structure where the parameters lie.

Such a criterion turns out to provide superb denoising results up to a certain noise variance level but progressively worse when the variance increases (see Fig. 2 in the last section). The bad performance with high noise variance stems from the fact that a constant that depends on the range, and hence the variance of the data, is ignored in the criterion. This problem has been recently explained with suggested solutions by Roos *et al.* [20].

In this paper, we generalize the linear-quadratic MDL denoising method [9] in such a way that we model both the wavelet coefficients, representing the denoised signal, and the rest, representing the noise, by equal bin-width histograms, with which the code lengths can be calculated; we call this method *MDL-histo* (MDL denoising with histograms). In broad terms, this is done as follows: The coefficients are separated into a number of different resolution levels, each defining its own histogram. Instead of thresholds the selection of the retained coefficients on each resolution level will be done by setting to zero the coefficients in a subset of bins, call it $\bar{S}$, while leaving the coefficients intact in the remaining bins. The code length of the retained coefficients at each resolution level can be calculated. The code

Manuscript received March 30, 2005; revised September 14, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ercan E. Kuruoglu.

The authors are with the Laboratory of Computational Engineering, Helsinki University of Technology, FIN-02015 HUT, Finland (e-mail: vibhor@lce.hut.fi; jukka.heikkonen@hut.fi; kimmo.kaski@hut.fi).

Digital Object Identifier 10.1109/TSP.2006.877635

length for the noise is obtained by fitting a second histogram to the coefficients, obtained from all the coefficients that define the original data such that the retained coefficients are set to zero. The optimal set of bins $\bar{S}$ is found by minimization of the sum of the two code lengths for the denoised signal and the noise. To find such optimal collections is a daunting computational problem, and we resort to a greedy search to give a suboptimal collection.

When all this is done, we modify the construction of the denoised signal by a process related to "soft" thresholding, [1], [19]. Although there are no thresholds in the proposed MDL-histo approach to denoising, the idea in [1] nevertheless provided the inspiration to obtain a similar effect albeit in a manner which cannot be justified within the MDL theory.

This paper is organized such that after a preliminary section on histogram modeling, we give the details of the denoising algorithm, followed by a section on applications of the method to a variety of signals with comparisons to other well known denoising techniques. Despite the several shortcuts we take to simplify the calculations, the MDL-histo technique turns out to be clearly the best in the tested 1D cases whenever the noise added to the data has a distribution that differs from the normal one. In images, the performance comparison with the best competing algorithm among those tested, BayesShrink [1], is less clear, although even with Gaussian noise a slight improvement can be detected.

II. Code Length of Data With Histograms

As a preliminary topic, we give the code length for a sequence $y^n = y_1, y_2, \ldots, y_n$ of real valued data points y_t, quantized to a common precision δ and modeled by a histogram. Let all y_t fall in the interval $[a, b]$, which is partitioned into m equal-width bins, the width given by $w = R/m$, where $R = b - a$. Let n_i data points fall in the i'th bin. Then, the code length of the data string, called stochastic complexity, relative to such histograms, is given by

$$L(y^n|w,m,\delta) = \log\binom{n}{n_1,\ldots,n_m} + \log\binom{n+m}{n} + n\log\left(\frac{w}{\delta}\right) \quad (1)$$

where the logarithms are to base 2 (see [15] and [16]). The first term is the code length for encoding the n bin indexes corresponding to the bins of $y_1, y_2, \ldots, y_n$. The second term is the code length for the integers n_i. To see this, notice that the positive indexes can be encoded as a binary string, which starts with n_1 0's and a 1, followed by n_2 0's and a 1, and so on. The string has m 1's, and its length is $n + m$. If we sort all such strings we can encode the index of any one with code length given by the logarithm of their number $\binom{n+m}{n}$. The third term gives the code length for the numbers y_t, quantized to the precision δ. Indeed, each quantized number is one of the w/δ points in its bin. If we add the code length $L(m) = \log m + 2\log\log m$ for the number m, we can minimize the conditional code length

$$\min_m\{L(y^n|w,m,\delta) + \log m + 2\log\log m\}$$

to remove the dependence on m and leave only the two numbers w and δ on which the code length is conditioned.

Now we need to consider the code length of the modified data string $\hat{y}^n$, obtained by retaining the data points in a subset S of the bins while setting the rest to zero; i.e., the points in the set of the remaining bins $\bar{S}$ are 0. Denote the indexes of the bins of the data points $\hat{y}_t$ by $(0), (1), \ldots, (|S|)$, where $|S|$ denotes the number of the bins in the set S. The bin with index (0) is added to contain all the $n_{(0)} = n - k$ 0-points, where k denotes the number of the retained points falling in the bins of S. For instance, if the first retained bin is the fifth bin out of the m bins, then $(1) = 5$. The code length for the sequence of the bin indexes is now

$$\binom{n}{n_{(1)},\ldots,n_{(|S|)},n-k} \quad (2)$$

where $n_{(j)}$ denotes the number of points falling in the bin having the index (j). Then, the code length for the string $\hat{y}^n$, given w, m, and δ, is

$$L(\hat{y}^n|m,w,\delta) = \log\binom{n}{n_{(1)},\ldots,n_{(|S|)},n-k} + \log\binom{n+|S|+1}{n} + k\log\left(\frac{w}{\delta}\right) + m. \quad (3)$$

The last term is the code length for S having 2^m subsets.

III. MDL-Histo Algorithm

We now describe the main denoising algorithm. Let $c^n = c_1, \ldots, c_n$ denote the sequence of coefficients obtained by applying a wavelet transform to the data sequence x^n to be denoised, and let R denote their range. We separate the coefficients into, say r resolution levels, $c^n = c_1^{n_1}, \ldots, c_r^{n_r}$, each having a sequence of coefficients $c_i^{n_i} = c_{i,1}, \ldots, c_{i,n_i}$, where $n_1 = n/2$, $n_2 = n/4$, and so on, with their sum $\sum_i n_i = n$. At each resolution level, an equal bin-width histogram is fitted to the coefficients with the same number of bins m of width $w = R/m$. Let H_i denote the histogram on the ith level, which contains n_i points. The number of bins determines the amount of computations needed, and it is not optimized. Here, it is selected to be $m = 7$.

We start the selection process for the retained coefficients on the first resolution level. Let S_1 be one of the 2^m subsets of the m bins on the first level having k_1 coefficients from the sequence $c_1^{n_1}$ falling in the bins of S_1. Put $m_1 = |S_1|$. Define $\hat{c}_1^{n_1}$ as the string of the points falling in the bins of S_1, and zero otherwise. The code length for $\hat{c}_1^{n_1}$ is given as in (3) by

$$L(\hat{c}_1^{n_1}|S_1,w,\delta) = \log\binom{n_1}{n_{1,(1)},\ldots,n_{1,(m_1)},n_1-k_1} + \log\binom{n_1+m_1+1}{n_1} + k_1\log\left(\frac{w}{\delta}\right) \quad (4)$$

where $n_{1,(j)}$ denotes the number of points falling in the bin of H_1 having the index (j). To clarify these indexes, let b_k denote the k'th bin of H_1 for $k = 1, 2, \ldots, 7$, and let the set S_1 of the retained non-zero bins be $S_1 = \{b_2, b_3, b_5, b_7\}$. Then, $(1) = 2$,

$(2) = 3$, $(3) = 5$, and $(4) = 7$. We also omit the fixed code length $m = 7$ for the set S_1.

Next, we calculate the code length of the residual $\bar{e}^n$. Since the wavelet transform can be inverted, the code length is obtained from the coefficients c^n, in which the coefficients retained so far are set to zero. Let $\bar{c}_1^{n_1} = c_1^{n_1} - \hat{c}_1^{n_1}$ denote the string of the residuals on the first resolution level, and put $\bar{c}^n = \bar{c}_1^{n_1}, c_2^{n_2}, \ldots, c_r^{n_r}$, where each coefficient is written to the precision δ. In this sequence, there are $n - k_1$ nonzero coefficients, provided that all the n coefficients in c^n are nonzero. Let R_1 denote the range of the coefficients in $\bar{c}^n$. We model this sequence by an equal bin-width histogram of M bins. Hence, the common bin width is given by $w_1 = R_1/M$. Since k_1 coefficients in $\bar{c}_1^{n_1}$ were set to zero the sequence $\bar{c}^n$, given S_1, w_1, δ, and M, can be encoded with the code length

$$L(\bar{c}^n|S_1, w_1, \delta, M) = \log \binom{n - k_1}{\nu_1, \ldots, \nu_M} + \log \binom{n - k_1 + M}{M} + (n - k_1) \log \left(\frac{w_1}{\delta}\right) \quad (5)$$

where ν_j is the number of points falling in the jth bin.

Writing $L(M) = \log M + 2 \log \log M$ and ignoring the code lengths for the ranges R and R_1 the code length for the data is given by

$$L(x^n|S_1, w, w_1) = L(\hat{c}_1^{n_1}|S_1, w, \delta) + L(\bar{c}^n|S_1, w_1, \delta, M) + L(M).$$

In this the precision δ does not appear at all, and the term $n \log R_1$ does not depend on S_1. The subset S_1 is then determined by the following minimization criterion:

$$\min_{S_1, M} \left\{ \log \binom{n_1}{n_{1,(1)}, \ldots, n_{1,(m_1)}, n_1 - k_1} + \log \binom{n_1 + m_1 + 1}{n_1} + \log \binom{n - k_1}{\nu_1, \ldots, \nu_M} + \log \binom{n - k_1 + M}{M} - (n-1) \times \log M + k_1 \log \left(\frac{MR}{(mR_1)} \right) + 2 \log \log M \right\}. \quad (6)$$

We continue the process for the second resolution level. For this we denote by $\hat{k}_1$ the number of retained coefficients in the optimal subset $\hat{S}_1$ found so far and by $\hat{L}_1$ the corresponding minimum code length (4). Let S_2 be a tentatively selected subset of the bins of the m-bin histogram H_2 for $c_2^{n_2}$ with k_2 coefficients falling in the bins of S_2. As above, we define $\hat{c}_2^{n_2}$ and $\bar{c}_2^{n_2}$ as the strings obtained from $c_2^{n_2}$ by setting the coefficients to zero that fall in $\bar{S}_2$ and S_2, respectively. Put $m_2 = |S_2|$. The code length for the retained coefficients on the second resolution level is given by

$$L(\hat{c}_2^{n_2}|S_2, w, \delta) = \log \binom{n_2}{n_{2,(1)}, \ldots, n_{2,(m_2)}, n_2 - k_2} + \log \binom{n_2 + m_2 + 1}{n_2} + k_2 \log \left(\frac{w}{\delta}\right) \quad (7)$$

where $n_{2,(j)}$ denotes the number of points falling in the bin of H_2 having the index (j).

This time, the coefficients of the residual are $\bar{c}^n = \bar{c}_1^{n_1}, \bar{c}_2^{n_2}, c_3^{n_3}, \ldots, c_r^{n_r}$. Let their range be R_2. We model the residuals by an equal bin-width histogram with a variable number M bins and the bin width $w_2 = R_2/M$. As above, we get the criterion to be minimized, as follows:

$$\min_{S_2, M} \left\{ \log \binom{n_2}{n_{2,(1)}, \ldots, n_{2,(m_2)}, n_2 - k_2} + \log \binom{n_2 + m_2 + 1}{n_2} + \log \binom{n - \hat{k}_1 - k_2}{\nu_1, \ldots, \nu_M} + \log \binom{n - \hat{k}_1 - k_2 + M}{M} - (n-1) \times \log M + k_2 + (\hat{k}_1 + k_2) \log \left(\frac{MR}{(mR_2)} \right) + 2 \log \log M \right\}. \quad (8)$$

Notice that the criterion (8) to minimize S_2 does not include the optimized code length $L(\hat{c}_1^{n_1}|\hat{S}_1, w, \delta)$, (4), obtained on the first layer, since by our greedy algorithm we do not let it affect the minimization. The process is repeated for the subsequent resolution levels until the number $\hat{k} = \hat{k}_1 + \cdots, \hat{k}_r$ of the retained coefficients for all resolution levels are determined, which gives the sequence $\hat{c}^n = \hat{c}_1^{n_1}, \ldots, \hat{c}_r^{n_r}$. If we take the inverse transform of $\hat{c}^n$ we get the denoised signal. We call this the *rigid MDL-histo* denoising algorithm.

We summarize the major steps in the algorithm.

1) Start with the wavelet transformed sequence of coefficients on r resolution levels, $c^n = c_1^{n_1}, \ldots, c_r^{n_r}$.
2) Recursively on resolution levels $i = 1, \ldots, r$ fit an m-bin histogram H_i to the coefficients $c_i^{n_i}$, and select a tentative partition of the bins into two sets S_i and $\bar{S}_i$. Put $m_i = |S_i|$. Denote by $n_{i,(j)}$ the number of points falling in the bin of H_i having the index (j). The bins in S_i contain k_i retained coefficients on this resolution level. Write the two sequences of the retained coefficients and the rest as $\hat{c}^{n_i}$ and $\bar{c}^{n_i}$, respectively.
3) Fit a second histogram with M bins of width R_i/M to the coefficients of the residual sequence $\bar{c}^n = \bar{c}_1^{n_1}, \ldots \bar{c}_i^{n_i}, c_{i+1}^{n_{i+1}}, \ldots, c_r^{n_r}$, where the first i residual strings are obtained by setting the already optimized retained coefficients to zero.
4) Find the optimal S_i by the minimization criterion, as follows:

$$\min_{S_i, M} \left\{ \log \binom{n_i}{n_{i,(1)}, \ldots, n_{i,(m_i)}, n_i - k_i} + \log \binom{n_i + m_i + 1}{n_i} + \log \binom{n - \sum_{j=1}^{i-1} \hat{k}_j - k_i}{\nu_{i,1}, \ldots, \nu_{i,M}} + \log \binom{n + M - \sum_{j=1}^{i-1} \hat{k}_j - k_i}{M} - (n-1) \log M + \left(\sum_{j=1}^{i-1} \hat{k}_j \right) \log \left(\frac{MR}{(mR_i)} \right) + k_i \log \left(\frac{MR}{(mR_i)} \right) + 2 \log \log M \right\} \quad (9)$$

where $\sum_{j=1}^{i-1} \hat{k}_j$ denotes the number of retained coefficients in the so far optimized sets S_j, $j < i$. For $i = 1$ this sum is zero, and $\nu_{i,j}$ is the number of coefficients falling in the j'th bin of the M-bin histogram fitted to the string $\bar{c}^n$.

5) The denoised signal is obtained by taking the inverse transform of the sequence $\hat{c}^n = \hat{c}_1^{n_1}, \ldots, \hat{c}_r^{n_r}$ of all the retained coefficients.

Next, we describe a softer version of the algorithm, to be called *soft1*, where the nonretained coefficients in the string $c^n - \hat{c}^n$ instead of having set to zero will be attenuated in a certain manner. Such techniques do not conform with the MDL principle without further arguments, and the only reason we discuss them is to see if they improve denoising as they do in the other denoising techniques. The results turn out to be mixed as seen in the next section.

There are a number of ways to introduce such a modification and next we describe one of them. Sort all the coefficients in c^n in descending order by the absolute value $|c_{(1)}| \geq \cdots \geq |c_{(n)}|$, which induces the 1-1 map $i \mapsto (i)$, or in words, '$c_{(i)}$ in the original sequence $c^n = c_1, c_2, \ldots c_n$ is the ith largest coefficient. Let $\hat{C}$ and $\bar{C}$ be the sets of the retained and the nonretained nonzero coefficients in c^n, respectively. We construct a string $\tilde{c}^n = \tilde{c}_1, \ldots, \tilde{c}_n$, which consists of the retained coefficients of $\hat{C}$ left intact and the remaining coefficients attenuated as follows:

If $c_{(i)} \in \bar{C}$, then $\tilde{c}_{(i)} = e^{-\alpha i/n} c_{(i)}$,

else, $\tilde{c}_{(i)} = c_{(i)}$.

For instance, let the largest nonretained coefficient be the seventh largest coefficient $c_{(7)}$ of them all. Then, it gets attenuated by the factor $e^{-\alpha 7/n}$, while if the smallest nonretained coefficient is the last c_n, then it gets attenuated most by the factor $e^{-\alpha}$. The denoised signal $\hat{x}^n$ is the inverse transform of the sequence $\tilde{c}^n$.

Discussion

Although the simplicity of the calculations is not the main objective in denoising it is clear that our algorithm requires a lot more calculations than the other techniques. The main time consuming tasks are the search through 128 subsets of the bins of the retained coefficients on each resolution level to find the optimum and the optimization of the number of bins in the histogram for the residuals. There are a number of ways to speed up the calculations. For instance, instead of optimizing the number of bins M for each layer and test each integer from 1 to n, in reality we tested M only for every fifth integer from 1 to $n/2$. It turned out that the values ranging from 30 to 100 gave good results in all the data tested. Similarly, we found experimentally that the value of α in our *soft1* algorithm giving good results is $n/(n - \hat{k})$. In our MATLAB implementation, for a 256×256 pixel grayscale image on an ordinary workstation with the number of bins $m = 7$ it takes typically less than one minute to get the denoised image.

A most important way to improve the results is to increase the number of bins m from seven, which would permit a finer separation of the retained coefficients from the rest. This however would rapidly increase the calculations for finding the optimum even with the greedy search, which finds the optimum separately for each resolution level rather than jointly for them all. On the other hand there is a way to pick the retained coefficients for all the resolution levels jointly—albeit nonoptimally. We could restrict the selection of the bins such that at each level i we take m_i bins of the largest coefficients in absolute value, and find the optimal collection of r integers $\{k_1, \ldots, k_r\}$. Another rather important way is to fit variable bin-width histograms on each resolution level to the retained coefficients, where the break points are calculated from the empirical fact that the distribution of the coefficients can well be approximated by a Laplace distribution.

Finally, we have modeled the retained coefficients on each resolution level as independent of the other levels, while certainly there is a strong dependency between them. For instance, the location of the retained coefficients on the first level affects their position in the second level, and so on. Such improvements, however, are applicable to all the different denoising algorithms, and we do not study them.

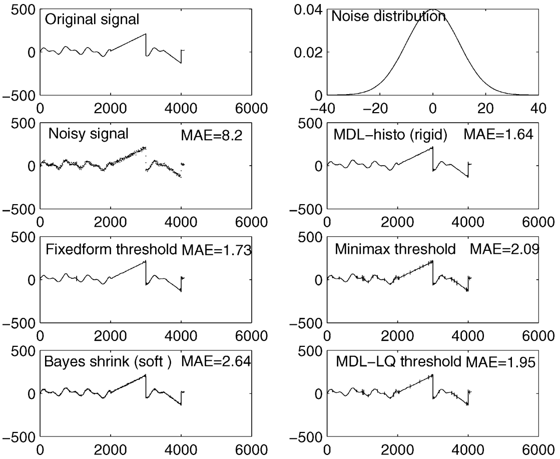

Fig. 1. Comparison of different methods for denoising a particular signal with added Gaussian noise.

IV. Experimental Results

In this section, we test our MDL-histo method by applying it to data where the original signal is both 1-D and 2-D mixed with different types of noise, and the results are compared with other denoising methods. In all the experiments, we have applied Daubechies (db5) [17] wavelet basis functions.

Fig. 1 illustrates a time series signal of 4096 sample points consisting of two parts, a sinusoidal and a ramp, to which low variance Gaussian noise of zero mean and standard deviation 10 was added. The tested denoising techniques are BayesShrink [1], Minimax thresholding [6], Fixedform thresholding [6], and Rissanen's linear-quadratic MDL-LQ method [9]. In addition to comparing the results visually, we have also calculated the mean absolute error (MAE) measure. We find that Rissanen's MDL-LQ denoising gives good performance, but it clearly has some peak-like leftovers. The Fixedform, BayesShrink, and MDL-histo *rigid* methods give similar performance with a slight edge in favor of MDL-histo.

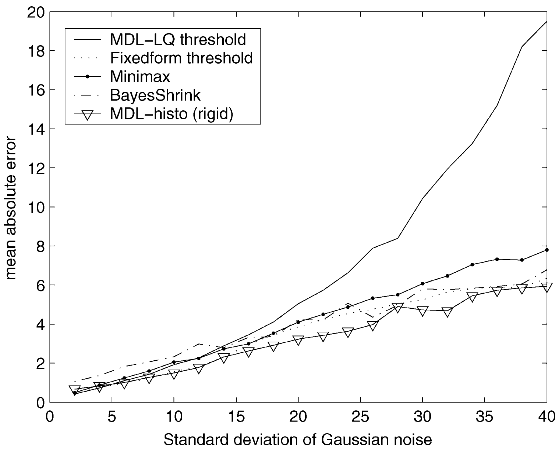

Fig. 2. Comparison of different methods for denoising the signal in Fig. 1 with added zero-mean Gaussian noise of increasing variance.

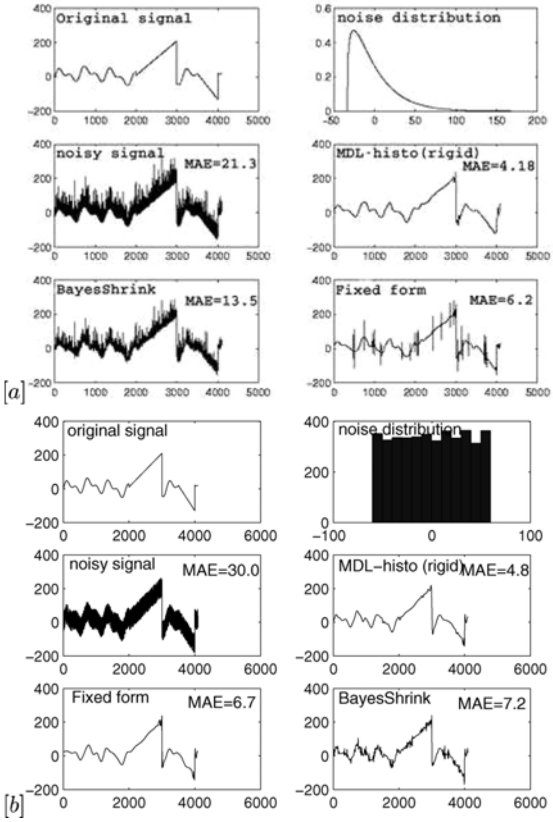

Fig. 3. Comparison of different denoising methods for non-Gaussian noise. (a) Comparison of methods on gamma distributed zero mean noise. (b) Comparison of methods on uniformly distributed noise.

Fig. 2 illustrates the MAE results of the application of the tested methods to denoise the same signal as in Fig. 1 with added Gaussian noise of increasing variance. In this case, the MDL-LQ method gives the best performance for small variance but the worst performance for the variance exceeding 15. Our MDL-histo method appears to give excellent results for all variance levels, beating the BayesShrink, Minimax and Fixedform thresholding methods.

Fig. 3 illustrates two cases where the added noise has a non-Gaussian distribution. In the upper part, noise was generated with an asymmetric gamma distribution of zero mean, of the kind observed in many real-world situations like in spectroscopic measurements [13] and electrical noise [14]. Clearly, the best result, both visually and in the MAE score, is obtained with MDL-histo. We also find that the Fixedform algorithm beats BayesShrink.

The lower part illustrates the case where noise has a symmetric uniform distribution. This time, there is not much difference in the visual appearance of the denoised curve obtained with MDL-histo and Fixedform algorithms, although the former has a lower MAE score. Of the algorithms tested, the BayesShrink performed worst. Apparently, since the symmetric uniform distribution can be approximated reasonably well with a Gaussian, the difference in the performance of the algorithms compared is less drastic than in the previous case.

After the above 1-D data, we applied the denoising algorithms to image data, to which either Gaussian noise (Fig. 4) or speckled noise (Fig. 5) was added. In both cases, we compared our *soft1* algorithm with BayesShrink, VisuShrink, [4], and Fixedform [19] algorithms, all with soft thresholding. Although in images the visual appearance is often considered a better measure of merit than the MAE values, we show both.

In case of Gaussian noise (see Fig. 4), the MAE values of our *soft1* and BayesShrink algorithms are close. The visual appearance of the denoised image with *soft1* is smoother than that with BayesShrink, but the edges are not quite as sharp. On the whole, the quality of the two denoised images is comparable and better than that obtained with the other two algorithms.

The noisy image in Fig. 5 was generated by adding a multiplicative noise to the image pixels $I(i,j)$, thus $J(i,j) = I(i,j) + I(i,j) \times U(i,j)$, where $U(i,j)$ has a uniform distribution with zero mean and variance $V = 0.4$. Our *soft1* algorithm gave clearly the smallest MAE score, but since it distorts the image somewhat while removing the speckled noise, it is a matter of judgment whether one would prefer the denoised image obtained with BayesShrink, which is sharper while leaving more of the speckled noise. This time, the VisuShrink algorithm performed quite well, with the denoised image only slightly more blurred than ours. While the result of Fixedform was visually the worst, its MAE turned out to be better than that of BayesShrink and VisuShrink.

V. Conclusion

The proposed MDL-histo technique is distinguished by its ability to remove noise no matter how it is distributed. In particular, there is no need to specify a zero-zone around low magnitude wavelet coefficients as done conventionally. Although the

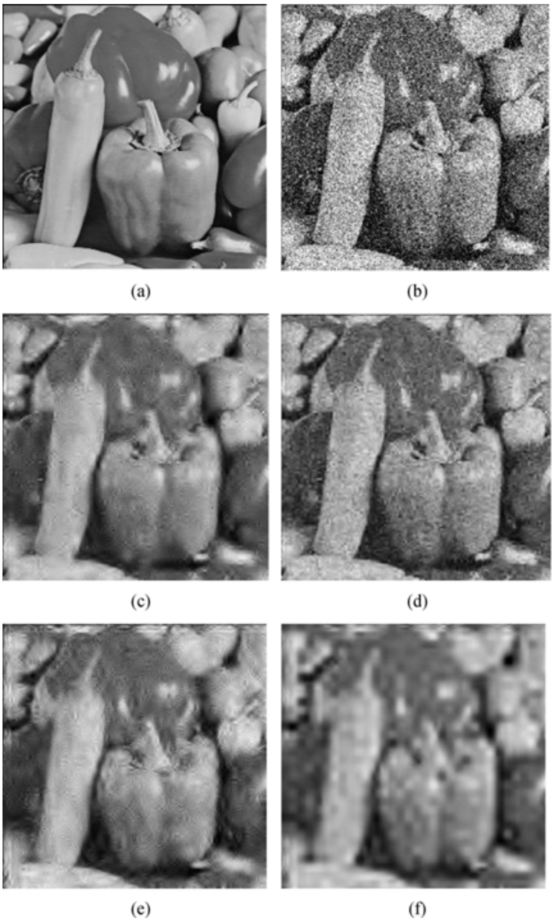

Fig. 4. Comparison of different denoising methods on an image with added Gaussian noise. (a) Original 256×256 pepper image rescaled to interval [0,1]. (b) Noisy image with Gaussian noise of variance 0.04. (c) MDL-histo *soft1* with residual MAE $= 0.0528$. (d) BayesShrink with soft thresholding; residual MAE $= 0.0545$. (e) VisuShrink with soft thresholding; residual MAE $= 0.0594$. (f) Fixedform with soft thresholding; residual MAE $= 0.058$.

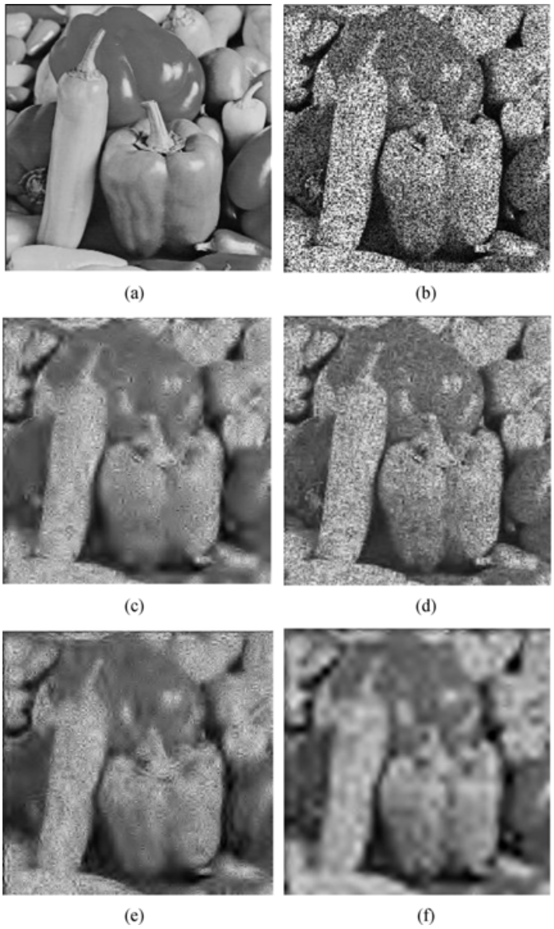

Fig. 5. Comparison of different denoising methods on an image with speckled noise. (a) original image as in Fig. 4, (b) noisy image with speckled noise, (c) MDL-histo *soft1* variant with residual $\mathrm{MAE} = 0.072$, (d) BayesShrink with soft thresholding; residual $\mathrm{MAE} = 0.0826$, (e) VisuShrink with soft thresholding; residual $\mathrm{MAE} = 0.0866$, and (f) Fixedform with soft thresholding; residual $\mathrm{MAE} = 0.078$.

setting in this paper was in the wavelet domain, the idea can be extended to other transforms which rely on energy compaction.

For image data the performance of MDL-histo is comparable with the best of the tested algorithms. However, when the noise distribution differs from the Gaussian, in particular in 1-D data, the MDL-histo algorithm is found to excel.

There are a few problems to be studied in the future, the most important being to find a method to select the bins of the retained coefficients for a larger number of bins. This would permit separation of noise even from high frequency data.

REFERENCES

[1] S. G. Chang, B. Yu, and M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” *IEEE Trans. Image Process.*, vol. 9, pp. 1532–1546, Sep. 2000.

[2] E. Simoncelli and E. Adelson, “Noise removal via Bayesian wavelet coding,” in *Proc. IEEE Int. Conf. Image Processing*, Sep. 1996, vol. 1, pp. 379–382.

[3] H. Krim and I. C. Schick, “Minimax description length for signal denoising and optimized representation,” *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 898–908, Apr. 1999.

[4] D. L. Donoho and I. M. Johnstone, “Ideal spatial adaptation by wavelet shrinkage,” *Biometrika*, vol. 81, pp. 425–455, 1994.

[5] ——, “Adapting of unknown smoothness via wavelet shrinkage,” *J. Amer. Stat. Assoc.*, vol. 90, no. 432, pp. 1200–1224, Dec. 1995.

[6] D. L. Donoho, “Progress in wavelet analysis and WVD: A ten minute tour,” in *Progress in Wavelet Analysis and Applications*, Y. Meyer and S. Roques, Eds. Paris, France: Editions Frontiers, 1993, pp. 109–128.

[7] Abarmovich, T. Sapatinas, and B. W. Silverman, “Wavelet thresholding via a Bayesian approach,” in *J. R. Statist. Soc.*, 1998, vol. 60, B, pp. 725–749.

[8] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[9] J. Rissanen, “MDL denoising,” *IEEE Trans. Inf. Theory*, vol. 46, no. 7, pp. 2537–2543, Nov. 2000.

[10] G. Nason, "Choice of the threshold parameter in wavelet function estimation," in *Wavelets in Statistics*, A. Antoniades and G. Pooenheim, Eds. Berlin, Germany: Springer-Verlag, 1995.

[11] Y. Wang, "Function estimation via wavelet shrinkage for long-memory data," *Ann. Statist.*, vol. 24, pp. 466–484, 1996.

[12] H. Chipman, E. Kolaczyk, and R. McCulloch, "Adaptive Bayesian wavelet shrinkage," *J. Amer. Stat. Assoc.*, vol. 92, no. 440, pp. 1413–1421, 1997.

[13] J. Ojanen, T. Miettinen, J. Heikkonen, and J. Rissanen, "Robust denoising of electrophoresis and mass spectrometry signals with minimum description length principle," *Fed. Eur. Biochem. Soc. Lett.*, vol. 570, pp. 107–113, 2004.

[14] M. Gupta*, Electrical Noise: Fundamentals and Sources*. New York: IEEE Press, 1977.

[15] P. Hall and E. J. Hannan, "On stochastic complexity and nonparametric density estimation," *Biometrika*, vol. 75, pp. 705–714, Dec. 1988.

[16] J. Rissanen, T. P. Speed, and B. Yu, "Density estimation by stochastic complexity," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 315–323, Mar. 1992.

[17] I. Daubechies, "Ten lectures on wavelets," presented at the CBMS-NSF Regional Conference Series in Applied Mathematics Philadelphia, PA, 1992.

[18] Lim and S. Jae*, Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1990, pp. 536–540.

[19] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, pp. 613–627, May 1995.

[20] T. Roos, P. Myllymãki, and H. Tirri, "On the behavior of MDL denoising," in *Proc. Artificial Intelligence Statistics 2005 (AISTATS-05)*, pp. 309–316.

Vibhor Kumar received the M.Tech. degree in computer technology from the Electrical Engineering Department of the Indian Institute of Technology (IIT), Delhi, in 2001.

He was a Research Assistant at University of Helsinki, Finland. Since June 2003, he has been working as a Researcher at the Laboratory of Computational Engineering of Helsinki University of Technology, Finland. His research interest include neural networks, signal denoising, signal processing for biomedical signals and image processing for computer tomography.

Jukka Heikkonen received the M.Sc. and Dr.Tech. degrees in information technology at the Lappeenranta University of Technology, Finland, in 1991 and 1994, respectively.

Since 1996, he has been working at the Laboratory of Computational Engineering of Helsinki University of Technology, Finland, where he is currently acting a Senior Research Scientist. His research covers statistical modeling, information theory, and data analysis in variety of application fields.

Jorma Rissanen (F'97) received the Honorary Doctorate degree from the Technical University of Tampere, Finland, in 1992.

Currently, he is Professor Emeritus in the same university. He is a Fellow of Helsinki Institute for Information Technology and associated with the Technical University of Helsinki.

Dr. Rissanen is a Foreign Member of Finland's Academy of Science and Letters. Further, he is an Associate Editor of the *IMA Journal of Mathematical Control and Information* and of the EURA *Journal of Signal Processing and Bioinformatics*. His two main achievements are the invention of arithmetic coding and the introduction of the minimum description length principle with the associated idea of stochastic complexity. For these works he has received a number of awards, including the following: the IEEE Information Theory Society Golden Jubilee Award for Technological Innovation "for the invention of arithmetic coding" in 1998; the IEEE Richard W. Hamming medal "for fundamental contributions to information theory, statistical inference, control theory, and the theory of complexity" in 1993; the IBM Corporate Award "for the MDL/Predictive MDL principles and stochastic complexity" in 1991; the Best Paper Award from the IEEE Information Theory Group in 1986, which covered all papers in the world published in information theory during the preceding two-year period; and the Best Paper Award from the International Federation of Automatic Control (IFAC) in 1981.

Kimmo Kaski received the M.Sc. degree and the Licenciate in Technology degree from the Department of Electrical Engineering at Helsinki University of Technology, Finland, in 1973 and 1977 and the Ph.D. degree from the Theoretical Physics Department of Oxford University, Oxford, U.K., in 1981.

Since 1996, he has been Professor in computational engineering in the Laboratory of Computational Engineering of Helsinki University of Technology, Finland, where he is currently an Academy Professor in Computational Science and Engineering of the Academy of Finland. His research interests include computational science, complex systems, computational intelligence, and high-performance computing (parallel processing).