ECE437 Project Report

Wavelet–domain Statistical Modeling using HMMs

Juan Liu

May 15, 1998

Abstract

Wavelet–based techniques has long been known in signal/image processing literature, and applied into various applications such as denoising and compression. As a separate framework, hidden Markov model (HMM) has been widely used to provide formal statistical models. This projects follows the paper by M. Crouse *et al* [1], trying to make the connection between these two separate concepts. The wavelet representation of a signal often has the properties of clustering and persistence across scales. By investigating these properties we develop a wavelet–domain HMM to model the wavelet coefficients. An efficient Expectation Maximization (EM) algorithm is developed for fitting the HMMs to observational wavelet coefficients. After the model is obtained, we can use this model into the framework of signal estimation, and many other applications as well.

1 Introduction

In the signal and image processing literature, wavelet–based techniques has captured a lot of attention. Wavelet transform represents signal in a coarse–to–fine manner, which is intuitively natural. The wavelet representation has the remarkable properties such as sparsity, locality, multiresolution, clustering and persistence, etc. These properties have led wavelet–based techniques to various applications, such as estimation, detection, compression, etc. It is thus important to find a proper model for the wavelet coefficients, since the performance of applications often depends heavily on the validity of the model.

Separate from the development of wavelet–based methods, hidden Markov model (HMM) has been proposed and widely used in applications such as speech processing, computer vision and artificial intelligence. Many real world random processes can be successfully modeled using HMM. The success of HMM, from what I believe, is due to two facts: first, the natural world process often has dependencies which could be modeled as a Markovian; second, HMM allows some flexibility within the model in the sense that the model parameters are dictated by the data itself. In other words, using the observation data, we adjust the parameters of the model to obtain the best fit between the data and the model.

This project is aiming at making the connection between these two concepts, which are both well–known and well–developed but separate. More precisely, the goal of this project is:

- to develop wavelet–domain data models using HMMs;
- to implement the wavelet–domain data models in applications such as signal estimation.

This report is organized as follows: in Sec. 2, we briefly explain the wavelet transform and its properties. Compression literature provides various probabilistic model for the wavelet coefficients. Examples are provided in this section. In Sec. 3, we review the structure of Hidden Markov Models, its main characteristic, and reveal the natural connection between the wavelet–domain data and HMMs. Models investigating various dependencies between wavelet coefficients are proposed. In Sec. 4, the standard problems of HMMs are addressed and solutions are provided. The EM algorithms are adapted train the model.

Once trained, HMMs provide an excellent approximation of true joint probability of the wavelet coefficients. It is reasonable to believe that using this model, we ought to obtain good performance in various applications. Sec. 5 presents the implementation of wavelet–domain HMMs in signal estimation

applications. Results are shown.

2 Wavelet Transform

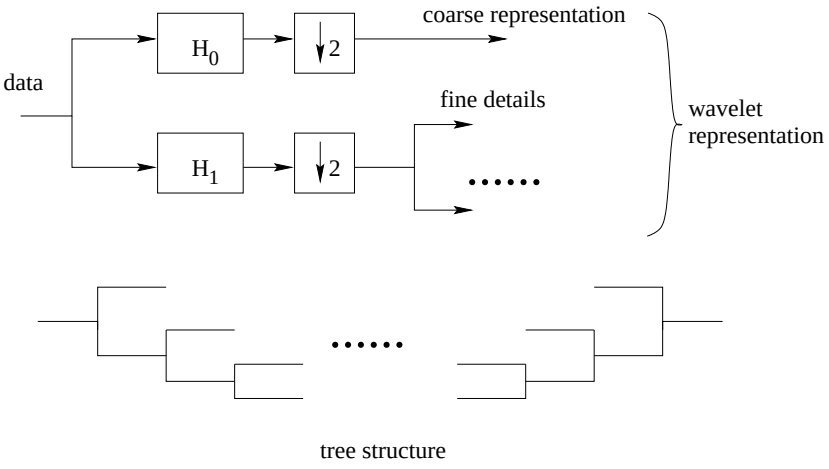


Figure 1: Illustration of filter banks

Wavelet representation is a coarse–to–fine representation of the data. One way of obtaining the wavelet representation is to iteratively pass the data through a filter bank, as shown in Fig. 1. The upper branch produces the coarse version of the signal, while the lower branch produces fine details. Wavelet transform can be regarded as a multiresolutional time–frequency analysis. It is well–known that wavelet representation has the attractive properties:

- **Locality.** Wavelet basis is localized simultaneously in time and in frequency.
- **Sparsity.** The wavelet representation of real–world signals tend to be sparse. In the fine scales, most wavelet coefficients are insignificant. Only a few coefficients are large in amplitude. This property has been widely used in state–of–the–art coders, and estimation as well.
- **Clustering.** If a particular wavelet coefficient is large/small, then its neighbors are very likely to be also large/small.
- **Persistence.** Large/small values of wavelet coefficients tends to propagate across scales. In other words, in the tree structure of wavelet decomposition, if a parent node has a large/small value, its children is very likely to be also large/small.

These properties can be conveniently shown in Fig. 2. The first image is the original *Lena* image. The second is its wavelet decomposition. In each subbands, we can notice the locality. Most coefficients in the fine scales are small. Only a few coefficients are large, corresponding to boundaries and textures of the image, for example, the contour of the hat, and feathers on the hat. These large coefficients coming into clusterings, because for real–world images, boundaries and textures are clustered. Furthermore, we can notice the persistence across scale.

Figure 2: Data and its wavelet representation

The above properties of wavelet coefficients has been exploited by the compression community. For example, the zero–tree coder proposed by Shapiro [2] exploit the property of persistence across scales. The EQ coder proposed by LoPresto *et al* [3] assumes that the wavelet coefficients in fine scales are Gaussian distributed with slowly varying variance field. This is in fact using the clustering property. These properties can also be used in other signal or image processing tasks.

3 Modeling wavelet coefficients using HMMs

The wavelet coefficients have many remarkable properties, as we discussed in the last section. How to model the wavelet coefficients become an important issue, because the performance that can be achieved in practical applications depends heavily on the model. The simplest model that we can come up with is that the wavelet coefficients are assumed to be independent of each other. This

model is based on the property that the wavelet transform nearly De-correlates the signal, so after the transform there is not much correlation remained. To match the non–Gaussian statistics, we consider a M–state **independent Gaussian Mixture** model which consists of:

- a discrete random state variable S taking the values $s \in 1, 2, ..., M$ according to some probability mass function $P(S = s)$, which $s \in 1, 2, ..., M$.
- the observation probability distribution function associated to each states $p(w|S = s)$. Often, the observation pdfs are considered as Gaussian, and characterized by their means and variances.

In most applications, the value of a wavelet coefficient w is observed, while the state S is not (hidden state). The marginal distribution of w is:

$$p_W(w) = \Sigma_{s=1}^{M} P_S(S = s) p_{W|S}(w|S = s).$$

It is shown by Crouse *et al* that a two–state ($M = 2$) Gaussian mixture model is a good approximate to the marginal distribution of individual wavelet coefficients. The two–state model is convenient and effective. By increasing the number of states M, the fit to the observational statistics can be better, but in practice, it is seldom used. Furthermore, for the simplicity of illustration, we use two–state models, but all the models and corresponding algorithms can be extended to M–state models in a straight–forward way.

This model is simple. It represents the first order statistics of wavelet coefficients. However, this model is very coarse and naive, in the sense that it neglects the important properties of wavelet coefficients such as clustering and persistence. Ideally, we should use models employing these properties. The intra–scale dependency (clustering) and inter-scale dependency (persistence across scales) can lead us to better models.

To include the intra–scale dependency, we modify the independent Gaussian mixture model. Each coefficients are in two hidden states, "high" or "low", but the states of two neighboring coefficients are not independent. As the clustering property implies, if one coefficient is in one state, then its neighbors are very likely to be in the same state. We can model this dependency as Markovian. The wavelet state variables are linked across time/location by chains. To characterize this **Hidden Markov Chain** model, we need the probability mass function for the first wavelet coefficient in the scale $P_{S_1}(s)$; the state transition probabilities $P(S_{i+1} = m|S_i = n)$ for $m, n \in 1, 2, ..., M$; and the observation pdfs associated to every M states.

To capture the inter–scale dependency, similarly, we consider the persistence property. As it implies, if a parent node is in a particular state, then its children is very likely to be in the same state. We can model this dependency using a Markov tree. This model is addressed as **Hidden Markov Tree (HMT)** model. More precisely, we have three type of parameters to characterize the model:

- the probability of the tree root S_1 — $P(S_1 = m)$, for $m \in 1, 2, ..., M$. (This is similar to the initial state probability π.)
- the transitional probability between a child node i and its parent node $p(i)$ — $\epsilon_{i,p(i)}^{m,r} = P(S_i = m|S_{p(i)} = r)$. (This is analogous to transitional probability matrix A.)
- the mean and the variance of the wavelet coefficient w_i given its state $S_i = m$ — $\mu_{i,m}$ and $\sigma_{i,m}^2$, respectively. (This is similar to B, the observation probability.)

In this report, when implementation is concerned, we concentrate on the HMT model.

4 HMT Framework

There are three standard problems associated with the wavelet–domain HMT models:

1. **Likelihood determination:** Given the model $\lambda = (A, B, \pi)$, how to compute the probability of observing the sequence $W = w_1 w_2 ... w_T$. The value of the likelihood shows how well the model matches the observational data.
2. **State Estimation:** Given the model $\lambda = (A, B, \pi)$ and the observation sequence $W = w_1 w_2 ... w_T$, how to determine the underlying hidden state sequence $S = S_1 S_2 ... S_T$ corresponding to each observation.
3. **Training:** Given one or more observations of a signal, how to determine the approximate model $\lambda = (A, B, \pi)$ for the signal. By Training we hope to optimize the fit between the model and the data.

Solutions to the first two problems are standard: we use the forward–backward procedure to efficiently compute the likelihood probability; and use Viterbi algorithm to determine the most probable hidden states. For training, EM (Expectation Maximization) algorithm is used. The specific steps in EM algorithms has to be adapted to the HMT model we discussed in Sec. 3.

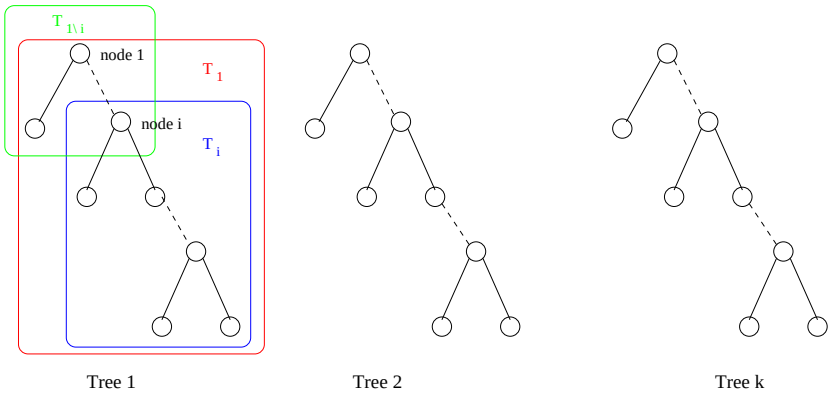


Figure 3: The notation of trees

Fig. 3 illustrate the tree structure of wavelet coefficients. As an example, we consider the case that a signal with 256 data samples is decomposed for four levels, so that there are 32 wavelet coefficients in the very coarse subband. Using the tree structure, we can organize the wavelet coefficients into 32 trees, each rooted at one wavelet coefficients in the very coarse band. These 32 trees should all contribute when defining the tree–structured probabilistic model. For convenience, we define every node in the very coarse band as "node 1", and call the tree rooted at it as "T_1". For the 32 trees, we call them $T_1^{(1)}$, $T_1^{(2)}$,..., $T_1^{(k)}$, ..., $T_1^{(K)}$, with $K = 32$. Each subtree rooted at node i is labeled as T_i, and every subtree rooted at node 1 ending at node i is labeled as $T_{1\backslash i}$. All these labellings are shown in the figure.

Before we come to any details of the algorithm, we explain the notation. The states are denotes as $S_i = m$, with $m \in 1, 2, ..., M$. The probabilistic model is denotes by λ. As shown in the last paragraph, T denotes the wavelet coefficient tree, and of course that is our observation sequence. The following variables are defined.

- forward variable: $\alpha_i(m) = P(T_{1\backslash i}, S_i(m)|\lambda)$;
- backward variable: $\beta_i(m) = P(T_i|S_i = m, \lambda)$;
- two–step backward variable: $\beta_{p(i)\backslash i}(m) = P(T_{p(i)\backslash i}|S_{p(i)} = m, \lambda)$.

The first two variables are used in computing the likelihood using the forward–backward procedure and in training the model. The last variable is useful in computing the transitional probability.

We manipulate the α's and β's to calculate the state probabilities and the likelihood functions. Using the basic probabilistic manipulation, we obtain:

$$P(S_i = m, T_1|\lambda) = \alpha_i(m)\beta_i(m);$$

$$P(S_i = m, S_{p(i)} = n, T_1|\lambda) = \alpha_{p(i)}(n)\beta_{p(i)\setminus i}(n)\beta_i(m)\epsilon^{nm}_{i,p(i)}.$$

The likelihood of the entire tree T_1 is

$$p(T_1|\lambda) = \Sigma^M_{m=1}P(S_i = m, T_1|\lambda) = \Sigma^M_{m=1}\beta_i(m)\alpha_i(m).$$

The desired conditional probabilities are then:

$$P(S_i = m|T_1, \lambda) = \frac{\alpha_i(m)\beta_i(m)}{\Sigma^M_{n=1}\alpha_i(n)\beta_i(n)};$$

$$P(S_i = m, S_{p(i)=n}|T_1, \lambda) = \frac{\alpha_{p(i)}(n)\beta_{p(i)\setminus i}(n)\beta_i(m)\epsilon^{nm}_{i,p(i)}}{\Sigma^M_{n=1}\alpha_i(n)\beta_i(n)}.$$

Thus we have obtained the desired likelihood functions. These are also useful in the training procedure. Based on the values of the likelihood functions, Viterbi algorithm is employed to estimate the underlying hidden states. Since Viterbi algorithm is fully covered in class, we omit further explanation in this report.

Since wavelet coefficients are grouped into trees, namely $T_1^{(1)}$, $T_1^{(2)}$, ..., $T_1^{(K)}$, and they comply exactly the same HMT model, it is correct to regard all these wavelet trees as separate observations of the same model. In other word, the underlying HMT model produces K separate phenomena as we experiment K times. In the EM algorithm, to implement the E(Expectation) step, we independently apply the likelihood determination procedure to each of the K trees, meaningly calculate $P(S_i^{(k)} = m|T_1^{(k)})$ and $P(S_i^{(k)} = m, S_{p(i)}^{(k)} = n|T_1^{(k)})$ for each tree. The expected value of the state probabilities are then:

$$P_{S_i}(m) = \frac{1}{K}\Sigma^K_{k=1}P(S_i^{(k)} = m|T_1^{(k)})$$

and

$$\epsilon^{nm}_{i,p(i)} = \frac{1}{KP_{S_{p(i)}}(m)}\Sigma^K_{k=1}P(S_i^{(k)} = n, S_{p(i)}^{(k)} = m|T_1^{(k)}).$$

The expectation here is rather the average over the all K available wavelet coefficient trees. Notice that all the K trees contributes in dictating the parameter of the HMT model.

After getting the estimates of state probabilities, we can do the M(Maximization) step rather easily, because the maximum likelihood estimates for the Gaussian means and variances can be calculated based on simply statistics. More precisely,

$$\mu_{i,m} = \frac{1}{KP_{S_i}(m)} \Sigma_{k=1}^{K} w_i^{(k)} P(S_i^{(k)} = m | T_1^{(k)});$$

$$\sigma_{i,m}^2 = \frac{1}{KP_{S_i}(m)} \Sigma_{k=1}^{K} (w_i^{(k)} - \mu_{i,m})^2 P(S_i^{(k)} = m | T_1^{(k)}).$$

Note that the means and variances are simply the weighted empirical means and variances, with probability as the weights.

By now, we have completed the training procedure. Once the model is trained, it is an good appropriate to the true underlying model, with the fit to the data been optimized. We can implement this model in various applications. The next section will illustrate an example.

5 HMT Model in Signal Estimation Application

Suppose the signal is corrupted by additive white Gaussian noise with zero-mean and variance σ_n^2. In the wavelet domain, if the transform is orthonormal, we can consider the equivalent problem that the wavelet coefficients of the signal are corrupted by the Gaussian noise with zero–mean and the same variance. The estimation problem can now be expressed as:

$$w_i = y_i + n_i,$$

where w_i, y_i and n_i are wavelet coefficients of the observation, the signal and the noise, respectively.

Wiener filtering is well–known in signal estimation literature. If we use the simple model that the signal coefficient is iid Gaussian with zero–mean and variance σ_y^2, then wiener filter is simply a one–tap filter. The estimated signal has the following simple form:

$$E[Y_i|W_i] = \frac{\sigma_y^2}{\sigma_y^2 + \sigma_n^2} w_i.$$

In implementation, σ_y^2 is often unknown. In practice, we can use the empirical variance as an estimate of σ_y^2, and proceed with Wiener filtering method.

Now we have the HMT model. Intuitively we would hope to get better estimation result because we now have a more deliberately designed model. After the training, the hidden states of each wavelet

for *Lena*		
Noise variance^2	Wiener Filtering	HMM Filtering
25	12.96	10.53
49	19.64	15.93
100	29.28	26.69
for *Barbara*		
Noise variance^2	Wiener Filtering	HMM Filtering
25	20.52	15.97
49	35.54	27.83
100	60.85	47.53

Table 1: MSE results of HMT based estimation compared with Wiener Filtering

coefficient can be estimated. Based on the estimated states, we modify Wiener filtering as following:

$$E[y_i^{(k)}|W] = \Sigma_m P(S_i^{(k)} = m|W)\frac{\sigma_{im}^2}{\sigma_n^2 + \sigma_{im}^2} w_i^{(k)}.$$

Note this is the weighted version of Wiener filtering, with the likelihood of each state as the weight. The final signal estimate is computed by the inverse wavelet transform of the estimated wavelet coefficients.

We experiment this estimation scheme on both 1–D signals and images. For 1–D signal, we use representative signals such as: Doppler (continuous with a large range of frequency components); Blocks (discontinuous with all frequency components); and Bumps (δ–like functions). The denoised results are shown in Fig. 4. The one in the left columns are noisy signals. The ones on the right are denoised signals. The improvement in SNR is about 6dB.

For 2–D images, we experiment on standard images such as *Lena* and *Barbara*. Results in the form of MSE are reported in Table 1. Fig. 5 shows the noisy and the denoised *Lena* image. We can notice that the visual quality improvement.

Good practical results suggest that the HMT model is an appropriate model for the real–world signals. This is not surprising, because the model is investigating the across-scale dependency, and the fit between the observational data and the model has been optimized. This HMT model could also be implemented into other applications such as signal classification, detection of abrupt change, etc.

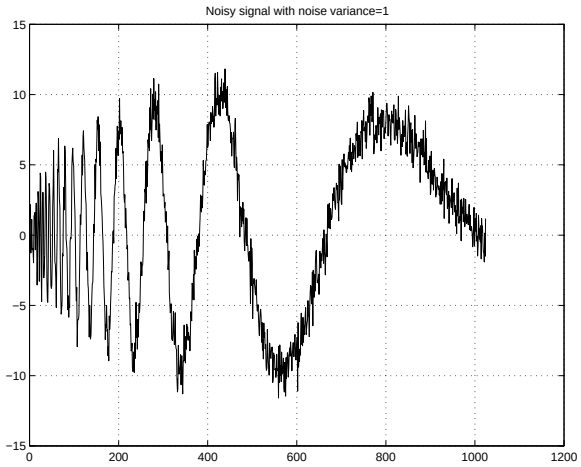
References

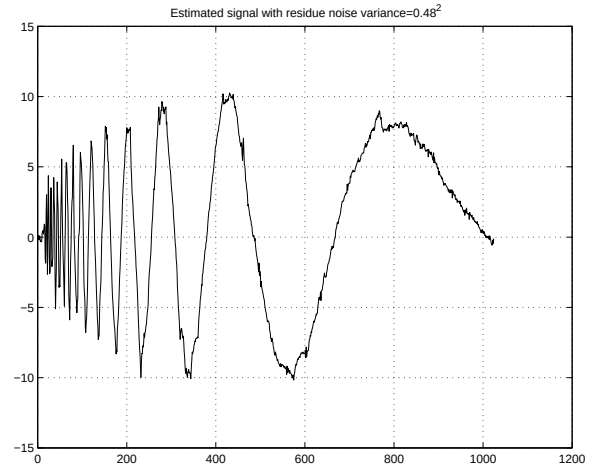
[1] M. Crouse, R. Nowak, and R. Baraniuk, "Wavelet–based Statistical Signal Processing using Hidden Markov Models," to appear in *IEEE Trans. Sig. Proc.*.

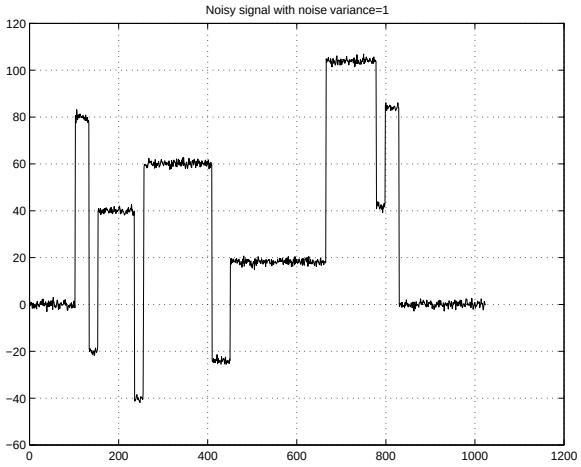
[2] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Trans. Sig. Proc.*, Vol. 41, No. 12, pp. 3445—3462, 1993.

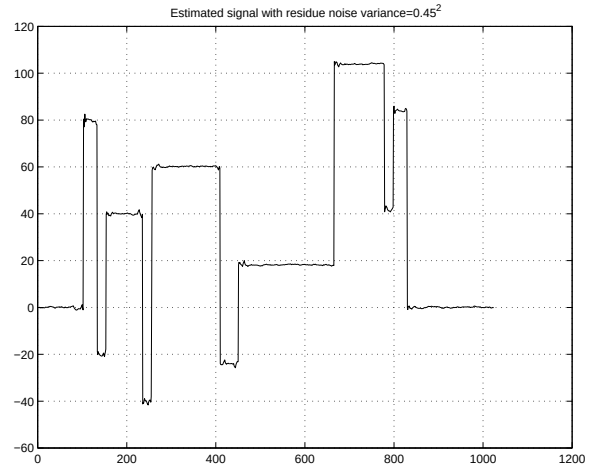
[3] S. LoPresto, K. Ramchandran and M. T. Orchard, "Image Coding based on Mixture Modeling of Wavelet Coefficients and a Fast Estimation–Quantization Framework," *Data Compression Conference 97*, Snowbird, Utah, 1997.

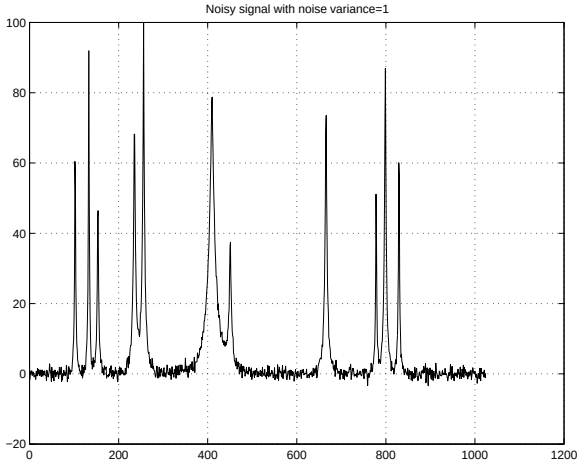
[4] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Processing," *Proceedings of IEEE* , Vol. 77, No. 2, pp. 257—286, 1989.











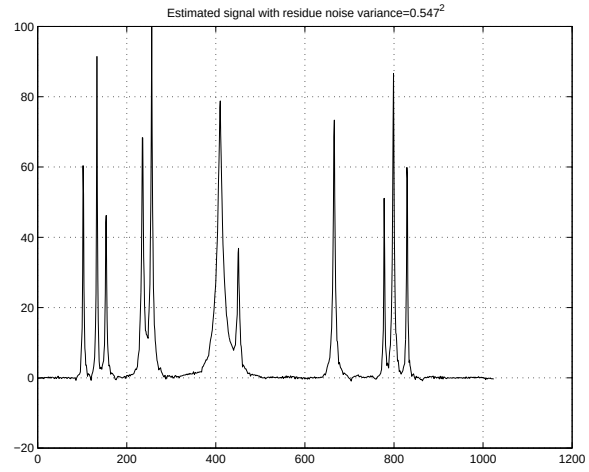


Figure 4: Estimation result for 1–D signals

Figure 5: Denoising results for Lena