# IMAGE INTERPOLATION USING WAVELET-BASED HIDDEN MARKOV TREES

*Kentaro Kinebuchi, D. Darian Muresan, and Thomas W. Parks*

School of Electrical and Computer Engineering
Cornell University
kkineb,darian,parks@ee.cornell.edu

## ABSTRACT

Hidden Markov trees in the wavelet domain are capable of accurately modeling the statistical behavior of real world signals by exploiting relationships between coefficients in different scales. The model is used to interpolate images by predicting coefficients at finer scales. Various optimizations and post-processing steps are also investigated to determine their effect on the performance of the interpolation. The interpolation algorithm was found to produce noticeably sharper images with PSNR values which outperform many other interpolation techniques on a variety of images.

## 1. INTRODUCTION

Wavelet based interpolation methods offer the possibility of preserving sharp edges since edge representation decay across scale [7] can be used to preserve edge information during interpolation. Other wavelet based methods for interpolation across scale can be found in [1, 3, 6]. In this paper we model the statistical relationship between coefficients at coarser scales using a hidden Markov tree to predict the coefficients on the finest scale. Our approach is based on recent work using hidden Markov models in denoising [4] and classification [5].

## 2. INTERPOLATION

The approach to image interpolation, which we call prediction of image detail, can be explained with the help of Figure 1. In Figure 1 the high resolution image is represented as the signal $X$ at the input to the filter bank. We assume that the low resolution, more coarsely sampled image is the result of a low-pass filtering operation followed by decimation to give the signal $A$. The low-pass filter, $L$, represents the effects of the image acquisition system. If we were able to filter the original high-resolution signal with the high pass filter $H$ to obtain the detail signal $D$ in Figure 1, and if we had a perfect reconstruction filter bank, it would then be possible to reconstruct the original image. We do not

This work was supported by NSF MIP9705349, TI, and Kodak

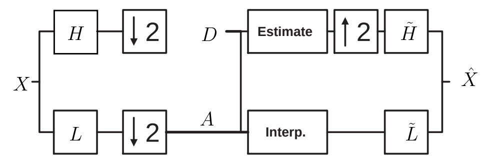


**Fig. 1**. Problem Formulation

have access to the detail signal $D$, however, so we estimate what it must be. The estimation is done based on a Hidden Markov Model.

## 3. MARKOV MODEL

The main distinction between a Markov and hidden Markov model is that in the latter case we do not observe the state a particular process is in. Recall that a Markov model consists of states of a process and its state transition probabilities. We always know the particular state of a process and the probability that it will be in another state the next instant, which incidentally is only dependent on its current state. With a hidden Markov model, on the other hand, we do not explicitly know the state of the process. The state is associated with some other probability distribution which is the one we observe. This results is a largely probabilistic inference for what state the process is in. A Markov tree is a structure where the state a node is in is only dependent on its immediate parent and any children it may have.

The wavelet coefficients were modeled as a hidden Markov tree with each coefficient was taken as a node in the tree with a certain hidden state. The parent of a particular node would be the coefficient at the next coarsest scale directly before it in what would amount to the same spatial location. The node also has a hidden state with a probability distribution associated with it, which is what we observe. We can see that the Markov framework effectively models the persistence property of wavelets since relationships between coefficients across scale are captured. The other property to be modeled, that of the non-Gaussian distribution of coefficients is done by the mixture probability of the hidden states. Since each state has a different probability density associated with it we can easily see that this will result in a mixture density. The probability densities themselves were chosen to be Gaussian with different means and variances according to state. The result of such a mixture probability is, in our case, the eventual probability distribution of the wavelet coefficients looking like a Laplacian distribution. The model used here is the same as that given in [4] and [5], and for convenience the same notation will be used here.

The Expectation Maximization (EM) algorithm was used to train the Markov tree model. The algorithm essentially works by finding the set of parameters which would most likely result in the set of observed wavelet coefficients. In this particular implementation the algorithm takes as input the wavelet coefficients and produces the state transition probabilities, and the means and variances for each different state for each coefficient. The parameters are enough to fully model the hidden Markov tree but additional information is also produced, which are the state probabilities for the wavelet coefficients. This information, though not necessary for the model itself, is highly useful as will be discussed later. The EM algorithm works by successively iterating model parameters until a specified error is achieved. The algorithm is implemented by splitting the problem into the E and M steps. In the E step, the joint pdf for the hidden state variables is calculated. In the M step, we set the model parameters to the ones which maximize the expectation of the parameters calculated in the E step. The process is repeated until a convergence error criterion is reached. An in-depth description of the EM algorithm is given in [4]. An important question to ask at this point is whether the parameters are all different within a scale. At first glance we can see that since we are training the transition probabilities, means, and variances, that to get the lowest error rates it would be best to have different parameters for each node in the tree. However, as the wavelet transform of a signal is local in both space and frequency, we can see that this would result in a model of very little use since even training on a slightly shifted signal would result in a different set of parameters. The logical thing is to therefore 'tie' the parameters within scales such that there are the same parameters for each scale. As we shall see later, tying is not always used as there are cases in which we would like to know the state of individual coefficients.

## 4. IMPLEMENTATION

The problem of image interpolation lends itself very nicely to the hidden Markov tree structure. More specifically, the problem was formulated as one of predicting the HL, LH, and HH bands of an image and then taking the inverse wavelet transform, resulting in a picture of twice the resolution. Since our hidden Markov tree requires training, a simple question would be what to train the model on. It can even be expected that there might be a universal set of parameters which work well on most real world images as described in [4], [5]. However, in [4] and [5] the focus was on denoising whose formulation uses a hidden Markov tree but is less sensitive to slight differences in parameters. The reason behind this lies in the fact that denoising is performed using state transition probabilities to find the conditional mean estimates of the wavelet coefficients. On the other hand, for the present interpolation method, the state transition probabilities themselves are directly used in prediction. This makes the method more sensitive.

Two similar methods with important differences were investigated for predicting the coefficients at the finest scale. The first formulation for predicting coefficients in finest bands in our case is:

$$P(w_i^k) = \frac{1}{\sigma_{i.m}^k\sqrt{2\pi}} e^{\frac{(w_i^k - \mu_{i,m}^k)^2}{2\sigma_{i,m}^k}} \tag{1}$$

Where $w_i^k$ are the wavelet coefficients at the finest scale k which we are trying to predict. Variables $\sigma_{i,m}^k$, $\mu_{i,m}^k$ are obtained from the training set once we find the expected value of the state:

$$E(S_i^k) = \sum_M mP(S_i^k = m) \tag{2}$$

with

$$P(S_i^k = m) = \sum_M P(S_i^k = m|S_i^{k-1} = n)P(S_i^{k-1} = n) \tag{3}$$

In the second method, we find the expected value of the state of the parent state, and then use the state transition probabilities to arrive at the child state:

$$E(S_i^{k-1}) = \sum_M mP(S_i^{k-1} = m) \tag{4}$$

$$P(S_i^k = m|S_i^{k-1} = n) \tag{5}$$

A random number generator is used together with the state transition probabilities, to arrive at the predicted state. Looking at these two methods, we can see that the main difference lies in the way the states of the coefficients we want to predict are found. One uses a deterministic approach which will yield the same value for the states each time while the other uses a stochastic approach. The values for the coefficients themselves will inherently be different each time whether we use the deterministic or stochastic approach as we are generating them according to a Gaussian probability distribution. Essentially we can see that we are modeling the exponential decay of the wavelet coefficients using the variance of our Gaussian distribution. In other words, looking at the variances obtained from training with tying from our model, we see that the magnitude of the variances decay exponentially as we go down to finer scales.

The extent to which a certain set of parameters trained on an image are generalizable was investigated as there are definitely some interesting implications. On first thought, it can be expected that a set of parameters will work well on images with similar statistics in the wavelet domain. However, for a very large number of states, it can be expected that the set of parameters will start to be more image specific. Before continuing it is more instructive to outline the precise steps taken in the interpolation scheme. To interpolate an image:

1. Create a hidden Markov tree of the image statistics of a relatively similar image. The model will consist of the state transition probabilities, the state means and variances, and the state probabilities. Training should be done using tying within scale. Tying within scale refers to the option where the same set of parameters is used for every node of the Markov tree within a scale. For example, with tying all the nodes in the HL band of a image will have the same state transitions probabilities, etc. The reason for using tying is twofold. Firstly, the wavelet transform is sensitive to shifts and hence without tying the parameters would be different for a slightly shifted image. Secondly, we would like to capture only the characteristics of the wavelet transform previously mentioned, and not any image specific statistical dependencies. In this step we obtain $\sigma_{i,m}^k$, $\mu_{i,m}^k$ and $P(S_i^k = m|S_i^{k-1} = n)$ used in (2).

2. Obtain information about all sign changes occurring in the training set wavelet coefficients from parent to child in the hidden Markov tree. For example, look at the sign of the wavelet coefficient associated with a node in the HL band and look at the sign of the wavelet coefficient associated with its parent. Count the total number of such sign changes and obtain an empirical sign change probability for use later.

3. Take the wavelet transform of the image to be interpolated.

4. Iterate the EM algorithm for a single iteration for the Markov tree found during training. Tying must not be used here because the state probabilities will be used later for prediction. Here we obtain $P(S_i^{k-1} = n)$ used in (3)

5. Use the appropriate equations according to which method is being chosen and find the states of the coefficients in the finest scale.

6. Use the Gaussian probability distribution to randomly generate a value for the wavelet coefficient.

7. Using the probabilities of sign change, check the signs of the coefficients and make any appropriate changes.

8. Introduce a post-processing step which consists of a Gaussian low-pass filter followed by a simple sharpening mask to remove any stray noise.

An important observation is that the algorithm will give us different results every time we interpolate an image. The hope however, is that the results we are given will be acceptable every or most of the time such that the interpolation is useful.

There are two post-processing steps used in this algorithm, both of which contribute to higher visual quality and PSNRs. The first step is that of sign changes. In our model, the coefficients are modeled as a Gaussian mixture density. From empirical evidence, and as also mentioned in [3], the means of those Gaussian densities tend to be very close to zero and the main distinction between states tend to be in variance. After all, this was also the initial assumption where a Laplacian distribution was being model by a Gaussian mixture density with zero means and different variances. We can now see that an inherent drawback in this model is that the Gaussian distribution of zero mean is symmetric around the origin and hence there is no way to keep track of sign changes. This is only really a problem for when this model is used in interpolation. For applications such as denoising in [4], we already have the coefficients in the finest scale so sign information is more or less implicit. The easiest way to correct this to obtain empirical probabilities of sign change and apply them as a post-processing step. It is expected that if the parent of a particular node is in a high state with a large magnitude coefficient, then the node will also probably have a large coefficient of the same sign. This is in line with the persistence property of wavelets. As a last step, the interpolated image is convolved with a 3 x 3 Gaussian low-pass filter mask to remove stray noise. Since the filter has the effect of slightly blurring the image, it is then convolved with a 3 x 3 high boost filter.

## 5. RESULTS

Overall, the algorithm was found to outperform most other interpolation schemes. This was especially true for images which tend to lend themselves to wavelet methods. Such images have clear edges delineated by smooth regions. Images with large texture regions do not perform as well as was also observed in [3] for another wavelet based method. The output of our interpolation process for Lena is Fig. 3, where the predicted details are the HL, LH and HH. For the rays image we can see similar performance to the method in [3], except for the center region which, as expected, does not display a completely uniform pattern as can be seen in Fig. 2.

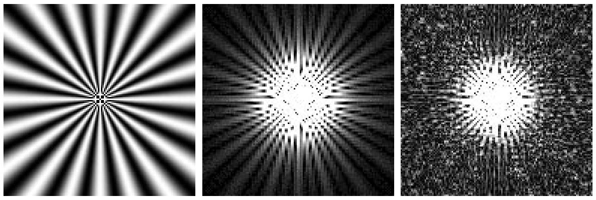

**Fig. 2**. Interpolation Performance. Left is the original image, center is the difference between the original image and the cubic interpolated image and right is the difference between the original image and the wavelet based interpolation

## 6. CONCLUSION

One of the main problems with the model used was the Gaussian mixture model for the wavelet coefficients. For denoising applications the model is fine but for interpolation the main shortcoming is the inherent inability to keep track of the sign of coefficients. This stems from the fact that the variance is being modeled which when combined with a mean of 0, cannot model whether a coefficient is positive or negative. This shortcoming was partially reduced by the post-processing step which flipped the signs according to empirical probabilities obtained from training. This is definitely not the best method as it does not take into account the whole Markov tree structure. A possible remedy to be investigated would be to use a distribution which does take into account sign information. Other possible improvements include tying on smaller parts of the sub-band during training and then deciding to use different state transition probability matrices. Indeed, there are truly many possible improvements and enhancements which can be done on the model itself as well as the post-processing step used later on. More than anything our results up to now show that this method of interpolation has definite potential and performs reasonably well even at this level of refinement.

**Fig. 3**. Interpolation of Lena. Originally, the image had zero detail coefficients. Top image, shows the detail coefficients added after interpolation. Bottom image shows the interpolated image.

## 7. REFERENCES


[1] D. D. Muresan and T. W. Parks, "Prediction of Image Detail," *IEEE International Conference on Image Processing,* September 2000.

[2] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian Tree-Structured Image Modeling Using Hidden Markov Models," *submitted to IEEE Transactions on Image Processing,* March 2000.

[3] W. K. Carey, Daniel B. Chuang and Sheila S. Hemami, "Regularity-Preserving Image Interpolation," *IEEE Transactions on Image Processing,* Vol. 8, No. 9 (Sept. 1999), 1293-1297.

[4] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-Based Signal Processing Using Hidden Markov Models," *IEEE Transactions on Signal Processing (Special Issue on Wavelets and Filterbanks),* April 1998, 886-902.

[5] H. Choi and R. G. Baraniuk, "Image Segmentation Using Wavelet Domain Classification," *Proc. SPIE Technical Conference on Mathematical Modeling, Bayesian Estimation, and Inverse Problems,* April 1998.

[6] S. Grace Chang, Zoran Cvetkovic, and Martin Vetterli, "Resolution Enhancement of Images Using Wavelet Transform Extrema Interpolation," *IEEE ICASSP,* (May 8-12, 1995), 2379-2382.

[7] Stephane Mallat and Sifen Zhong, "Characterization of Signals from Multiscale Edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 14, No. 7 (July. 1992), 710-732.