Adaptive Wavelet Coding for Still Images

Uygar Sümbül
uygar@stanford.edu

Abstract

Wavelet transforms play an important role in image compression and they can be implemented very flexibly using the lifting structure without trading invertibility. The advent of optimal update formulation in a lifting structure given the prediction operator [1] makes wavelet transforms more amenable to adaptive schemes; only the prediction operator has to be adapted, and the optimal update operator follows automatically. In this paper, we will explore this idea of introducing adaptivity into wavelet transforms for still image compression using the optimal update formulation.

1 Introduction

The discrete wavelet transform(DWT) is widely used in natural image compression applications. This compression stems from the fact that the DWT tends to express such images using just a few coefficients of large magnitude, which is a direct consequence of the inherent structure of natural images. Such images have most of their energy in the low frequency bands, and correspondingly, they have almost negligible frequency content in higher frequencies. This notion of compactness suggests that more efficient representations are possible, which is exactly what DWTs use. It may be helpful to consider the DWT as a prediction-error decomposition, where the prediction part represents a low-pass filtered version of the image and the error part corresponds to the remaining high-pass components. From this point of view, the lifting concept bears a natural role in the implementation of wavelet transforms. Once a low-pass filtered version of the image is obtained, it is further iterated to maximize the compression gain. Another advantage of the DWT is scalability, which has become a very desirable feature over the last years for a variety of applications. This property of the DWT is a direct consequence of the iterated implementation.

In this paper, we use lifting to design our customized DWTs that adapt the image to be compressed. Claypoole *et. al* have used a similar in a series of papers [2, 3, 4]. The rationale behind our scheme is to adapt the lifting structure to minimize the energy of prediction residuals(errors/high-frequency components).

The rest of the paper is organized as follows: In section 2, we first review the basic lifting structure and then provide a direct extension of this structure for two dimensional signals. In section 3, the optimal update formulation in lifting structures will be provided, based on B. Girod and S. Han's recent work. We will present a rate-distortion controlled adaptive scheme using the optimal update operation in section 4. Different ways of sending the adaptation information will also be discussed in this section. In section 5, test results of our scheme will be provided. We close in section 6 with a brief discussion of the work done.

2 The Lifting Structure and Two-dimensional Signals

Lifting is a method to construct DWTs and its power stems from the great flexibility it offers; one can use any nonlinear or space-varying predictor and update without disturbing the invertibility of the overall transform. The basic structure is composed of a splitting step, a prediction operation, an update operation, and a scaling operation. Here, the splitting part divides the original signal into two parts; namely the "even" and "odd" parts, where any reasonable decomposition is acceptable. Figure 1 demonstrates this basic lifting structure.

One can readily extend this basic structure to accomodate two dimensional signals: The structure should consist of two substructures to be meaningful for further iterations. Also, there should be another even-odd decomposition step before the second substructure. Figures 2 and 3 describe particular choices for the decompositions, the rationale behind which will hopefully become apparent in the subsequent sections.

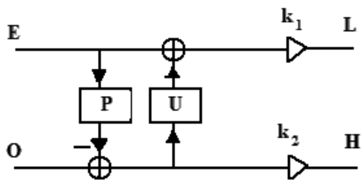

Figure 1: The basic lifting structure

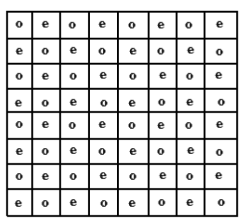

Figure 2: Even-odd decomposition, first part

3 Optimal Update in Lifting Structures

Recently, B. Girod and S. Han have obtained the optimal update operator of a lifting module for a given prediction operator [1]. Here, optimality is guaranteed only when *both* of the operators are linear. The optimality criterion is the mean-squared error. When the prediction operator, P, is given, the optimal update operator, U, satisfies the following relation [1]:

$$U = (I + P^T P)^{-1} P^T. \tag{1}$$

For our purposes, the resulting prediction and update operators are very large, but sparse matrices. Therefore, for a practical implementation, sparse matrix methods have to be employed. The problematic part, however, is taking the inverses in both the lifting modules in the encoder and the inverse lifting modules in the decoder. The solutions for both of the problems essentially follow from the technique outlined in [1], and is not repeated here.

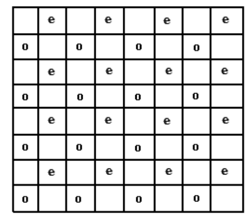

Figure 3: Even-odd decomposition on the even pixels of the first part

4 Introducing Adaptivity Using the Optimal Update Formulation

Adaptive techniques are used in a wide variety of applications. One drawback of such techniques is that efficient adaptation becomes harder as the number of parameters to be adapted increases. From this point of view, the optimal update operator formula provided in the previous section makes DWTs amenable to adaptive techniques since this formula removes the need to adapt the update step; adaptation in the prediction step will be enough.

Another drawback of adaptive techniques in our context is the fact that adaptation decisions need to be sent separately as side information. The hope is that the gain will dominate and the overall bit rate will be less to reconstruct an image with the same quality.

The encoder that we implemented is shown in figure 4. The prediction operators that we chose uses a combination of the neighboring even pixels in predicting the odd pixels. The combination is chosen according to the adaptation information. The location of the neighboring even pixels with respect to the odd pixel are north, east, south and west for the first even/odd decomposition block, and northeast, southeast, southwest and northwest for the second even/odd decomposition block. Such a decomposition is chosen to maintain similarity between the two substages of a lifting module and with the hope that better predictions will result from using the nearest and symmetrically located pixels. Obviously, better predictions lead to smaller prediction residuals and, correspondingly, higher PSNR rates. The outputs of the DWT are then fed into a SPIHT coder. The SPIHT(Set Partitioning In Hierarchical Trees) algorithm is particularly suitable for encoding DWT outputs and has become immensely popular [5]. The corresponding decoder scheme is also provided in figure 5. The decoder is almost a mirror image

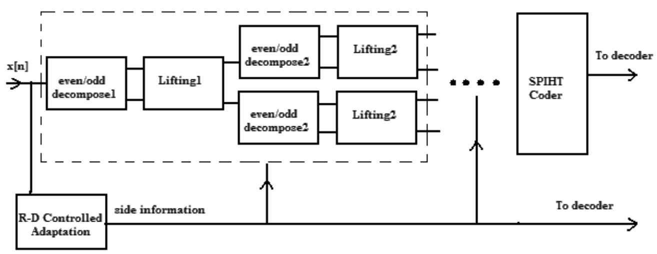

Figure 4: The encoder - Horizontal dots represent iteration on the low-pass output

of the encoder, except for the Rate-Distortion controlled adaptation block, which resides at the encoder.

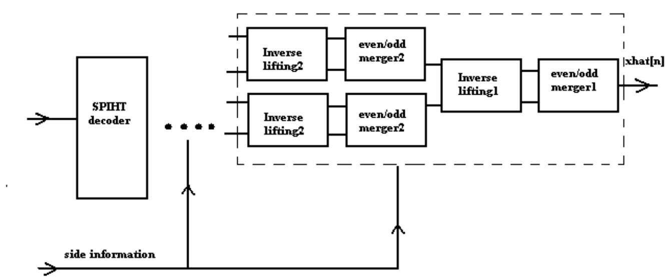

Figure 5: The decoder - Horizontal dots represent iteration

One plausible way of sending side information is to use the edge map of the underlying image. Since most of the high-frequency content of an image resides at the edges, one can avoid large prediction residuals if the predictions are done by taking the edge information into account. Furthermore, the edge map could be useful during the subsequent iterations if a proper way of scaling the edge map is used. Another advantage is that rate-distortion optimization could be handled easily together with adaptation by choosing the threshold parameter of the particular edge finder algorithm. In practice, however, we found that this idea is not as promising as it seems as a result of the following shortcomings:

- Entropy coding the edge map directly and sending the resulting stream as side information creates an overhead, which cannot be compensated by the gain afforded by adaptivity. Therefore, edge maps of the subsampled and resized image are used. This, however, means that the map cannot be used in the upper iterations since the map contains binary data. Subpixel accuracies cannot be reached by the classical interpolation methods.
- Scaling the edge map and using it in the subsequent iterations is not useful. The reason is again the fact that classical interpolation is not possible in this case.
- Sending the edge map in a pixel by pixel manner creates an overhead that cannot be compensated. Sending information for a group of pixels will more efficient, which resembles the rationale behind vector quantization.

The first two concerns suffer from the same problem: The edge map is too fragile to be scaled. A one-pixel error in determining the position of the edge leads to huge distortion rates in DWT application, while such an error is hardly ever a problem for computer vision purposes. An approach that could solve these two concerns could be inventing smart computer vision algorithms so that whether a pixel contains an edge or not is determined accurately. However, this will almost surely be a computationally demanding method, added on top of our already complex scheme. We chose not to follow this path due to lack of time, and instead tried to directly send the prediction information for a group of pixels as discussed below.

We decided to specify 9 different edge situations to describe a 4×4 block, where the number 9 is chosen heuristically. The need for such a choice will be better appreciated when one considers the total number of edge situations in a 4×4 block: $2^{16} = 65536$. The hope is that we will be able to choose the meaningful ones. The decision among the chosen situations is performed at the encoder, in the R-D controlled adaptation block, using the famous Lagrangian cost function formulation:

$$J = D + \lambda R, \tag{2}$$

where R represents the total rate, and D represents the distortion. One of the 9 situations is the empty block, where no pixels are chosen to be edges. Taking the properties of natural images into account, one can easily say that this situation should be the most encountered, and hence should incur the least rate. In this case, all four of the neighboring even pixels are used in prediction, whereas only the same-colored neighboring even pixels are used in the remaining cases. Once again, adaptation does not only use the distortion resulting from using these template blocks, it also takes the resulting rate into account and decides accordingly via the Lagrangian cost function. The template blocks describing the edge situation in a 4×4 block are provided in figure 6.

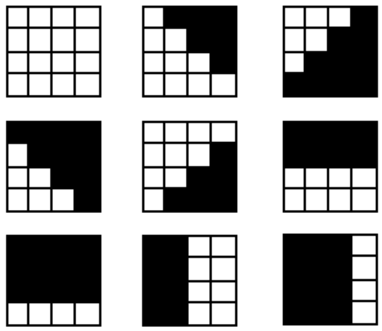

Figure 6: Template blocks used in adaptation

5 Results

We tested our scheme on the 512×512 boats image. The results show that no improvement could be obtained; only a very marginal increase in the PSNR-rate performance, which is not even apparent in the unzoomed plot.(Please see figure 7.) For all purposes, such an improvement(on the range of 0.01 dB) is clearly negligible. We suspect that one or more of the following items may be helpful in obtaining a better curve, which we would have more carefully tried, had we had more time before the project deadline:

- The edge templates may be chosen more carefully. Both the number of templates used and their contents may be altered to achieve a better performance.
- During our tests, we use the adaptation information only for the first iteration, again due to the fact that edge information is very fragile to scaling. However, if the number of templates used is increased, a healthier decision on the subsequent iteration may be possible, increasing the PSNR.
- The operation point for the adaptation block, λ, may not be at its optimum. However, this should not result in considerable gains/losses.
- A more radical approach to be able to use adaptation information in the subsequent iterations may be to employ smart computer vision algorithms. Working with 4×4 blocks may be suitable for the immediately following iteration. However, larger blocks(even higher computational complexity) is needed to use the available information in the subsequent iterations.

We also note that during our simulations we did not use the adaptive arithmetic encoding stage after the SPIHT coder. According to [5], this will result in an improvement of 0.3-0.5 dB. Yet, this is not relevant for our purposes since the improvement will effect both of the curves.

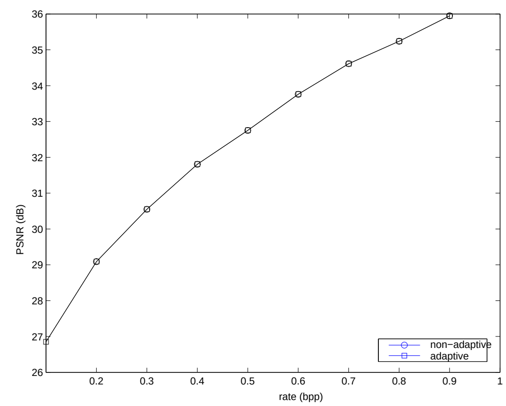

Figure 7: Test results on 512×512 boats.tiff

6 Conclusions

We investigated the feasibility of adaptively changing the lifting modules in a DWT implementation for compression purposes. What makes this startegy attractable for us is the rather beautiful optimal update formulation [1], which eliminates the need to adapt the update operator once the prediction operator is adapted.

We have seen that almost no improvement was obtained. This is in conjunction with the comments of Claypoole *et. al*, who published a series of papers on this subject. They were also trying to use edge information as the side information required for adaptation, but they did not use the optimal update formulation as they were not aware oof it. This has both a negative effect, which is obvious, and a positive effect, which is the fact that they are not constrained by linear operators; simple and effective nonlinear operators such as median filtering are allowed in their scheme.

As discussed in the previous section, there are a few ways to improve the current performance, which appeal to the fact that being able to use the side information in different scales should result in an extra gain. We believe that further improvements may be obtained this way. However, it is quite questionable that this scheme will ever be favorable to the corresponding non-adaptive one, considering the extra complexity introduced.

References

[1] B. Girod and S. Han, "Optimum motion-compensated lifting," (to be submitted).

[2] R. L. Claypoole, G. M. Davis, W. Sweldens, and R. G. Baraniuk, "Nonlinear wavelet transforms for image coding via lifting," *IEEE Trans. Image Processing,* vol. 12, no. 12, pp. 1449–1459, 2003.

[3] R. L. Claypoole, R. G. Baraniuk, and R. D. Nowak, "Lifting construction of non-linear wavelet transforms," in *Proc. of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis,* October 1998.

[4] R. L. Claypoole, R. G. Baraniuk, and R. D. Nowak, "Adaptive wavelet transforms via lifting," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing,* May 1998.

[5] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on Set Partitioning in Hierarchical Trees," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 6, no. 3, pp. 243–250, June 1996.

Addendum - Project Log

April 1 - April 20: Reading the referred articles and general reference books on DWT, the lifting structure, and adaptivity in this context

April 20 - May 10: Implementing the core of the code; the lifting module, the inverse lifting module, construction of prediction and update operators, edge map based adaptation, several helper codes

May 10 - May 20: Experimenting with edge map based adaptation; using different edge maps, experimenting with the threshold parameter of the edge map, scaling the edge map, predicting the edge map from subsampled edge map, introducing the rate-distortion optimization block

May 20 - May 24: Using template blocks to send prediction information directly as side information, choosing the number and content of template blocks, preparing the final report