

# NONLINEARLY-ADAPTED LAPPED TRANSFORMS FOR INTRA-FRAME CODING

*Dan Lelescu*

DoCoMo Communications Labs USA  
181 Metro Drive, San Jose, CA. 95110  
Email: lelescu@docomolabs-usa.com

## ABSTRACT

The use of block transforms for coding intra-frames in video coding may preclude higher coding performance due to residual correlation across block boundaries and insufficient energy compaction, which translates into unrealized rate-distortion gains. Subjectively, the occurrence of blocking artifacts is common. Post-filters and lapped transforms offer good solutions to these problems. Lapped transforms offer a more general framework which can incorporate coordinated pre- and post-filtering operations. Most common are fixed lapped transforms (such as lapped orthogonal transforms), and also transforms with adaptive basis function length. In contrast, in this paper we determine a lapped transform that non-linearly adapts its basis functions to local image statistics and the quantization regime. This transform was incorporated into the H.264/AVC codec, and its performance evaluated. As a result, significant rate-distortion gains of up to 0.45 dB (average 0.35dB) PSNR were obtained compared to the H.264/AVC codec alone.

## 1. INTRODUCTION

In this paper we will concentrate on the case of intra-coding frames in a video sequence. The statistics of predicted frame data and the mechanisms involved in producing this type of frames are very different and merit a separate treatment. The use of block transforms to independently process blocks in a frame can be a limiting factor for coding performance and usually creates undesirable blocking artifacts. To alleviate these problems, post-filtering approaches, lapped transforms (LT), and cross-block boundary intra-prediction methods were developed.

While earlier post-filtering approaches consisted of mostly ad-hoc methods which could not guarantee a consistent behavior, more recently post-filtering has been more formally posed as an image restoration problem (e.g., [1]). Post-filters have also been incorporated in video coding standards, for example the spatial post-filter used in H.264/AVC. In turn, lapped transforms provide a general context for performing both pre- and post-processing operations. The pre-processing can shape the original data to result in increased energy compaction and coding gains. In [2], a lapped orthogonal transform (LOT) was developed based on DCT basis functions, and was given a fast algorithm. For better blocking artifact removal, a lapped biorthogonal transform (LBT) was later introduced [3]. To limit the introduction of ringing artifacts, variable-length lapped transforms (VLLT) have also been developed (e.g., [4]). In [4], the non-separable lapped transforms have long basis functions taken from the LBT basis to reduce blocking artifacts, and adaptive short basis functions. In this case, the adaptation is limited since short functions do not overlap block boundaries. Adaptation in the sense of changing the lapped transform length is also discussed in [5].

Lapped transforms can be implemented (e.g., [6]) as block boundary pre- and post-filtering outside a block transform codec.

From an implementation point of view, the lapped transform can be added to a block transform codec without changing the existing block transform of the codec. The gains that a lapped transform brings compared to a block transform are diminished by the use of other complementary methods such as intra-prediction and adaptive post-filters. For increased performance, it is useful to design a signal-adapted rather than a fixed lapped transform, and evaluate the transform when incorporated in a state-of-art codec that uses techniques such as intra-prediction and post-filtering.

We present in this paper the determination and use of an adaptive lapped transform for coding intra-frames in a video codec. The basis functions of the transform are adapted, while their length remains constant. The lapped transform is implemented as the concatenation of a spatial filtering operation and a block transform. This presents the advantage that the block transform in current video codecs (i.e., H.264/AVC) can be left unchanged in the process of introducing the lapped transforms into the codec. Thus, by determining the filters, the lapped transform is specified. The filter adaptation for determining the lapped transform is done according to a nonlinear law, based on local statistics of the image as well as on the quantization regime. The filters are in general different directionally, and reflect the possibly different directional statistics of the image. Locally-adapted transforms have the ability to perform better than fixed transforms by virtue of adapting to the non-stationarity in the data. We show that the adaptation overhead associated with the lapped transform adaptation can be made small compared to other adaptive schemes. The lapped transform is implemented in the H.264/AVC codec (JM9.6), and the results of its use in this codec are evaluated. The H.264/AVC codec has its advanced features enabled (including intra-prediction and post-filtering). Using test sequences commonly accepted for video coding experiments, the nonlinearly-adaptive lapped transform produces objective rate-distortion gains of up to 0.45dB PSNR (on average 0.35dB), as well as subjective improvements in visual quality.

The remainder of the paper is organized as follows. The determination of the nonlinearly-adapted lapped transforms is presented in Section 2. Experimental results are given in Section 3. Section 4 concludes this paper.

## 2. NONLINEARLY-ADAPTED LAPPED TRANSFORMS

The block diagrams of an intra-frame transform coder and decoder incorporating a nonlinearly-adapted lapped transform (NLT) are shown in Fig. 1. The lapped transform is determined by the concatenation of cross-block boundary pre- and post-filtering with the forward and inverse block transform, correspondingly. The design of the NLT in this case requires the determination of the NLT pre- and post-filters.

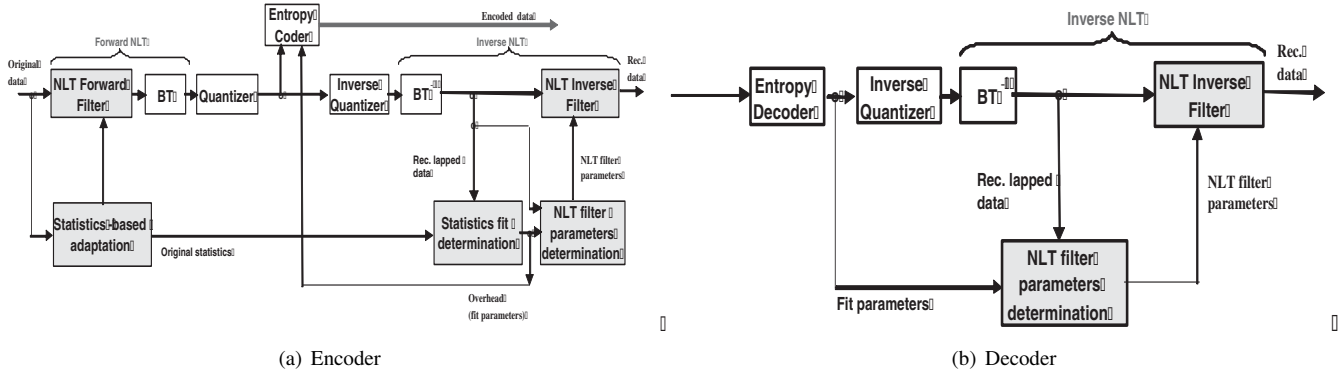


Fig. 1. NLT for intra-frame coding.

By means of providing a road map for the determination and use of the NLT, let us summarize at a higher level the most general process taking place in the NLT codec (Fig. 1).

Let us consider first the NLT encoder. For an original frame which will be coded as an intra-frame, the following steps are taken for determining the NLT:

1. Determine and store cross-boundary original statistics for all blocks in current frame.
2. Determine the NLT pre-filters using the original statistics and quantization step  $q$ .
3. Apply NLT pre-filters to all block boundaries, followed by the conventional transform coding of the frame.
4. Obtain the reconstructed, but still lapped frame, by inverse quantization and inverse block transform.
5. Determine cross-block boundary statistics of the reconstructed lapped frame.
6. Determine a parametric fit between the original statistics, and corresponding reconstructed lapped frame statistics. The parameters constitute the overhead of the NLT (sent to entropy coder).
7. Obtain an estimate of the original statistics by using the fit parameters in conjunction with the reconstructed lapped statistics.
8. Determine the NLT inverse filters using the estimates of the original statistics.
9. Apply NLT inverse filtering.

The operation of the NLT in the decoder implements steps 7–9 above, with both the fit parameters and the reconstructed lapped data being decoded from the compressed stream.

### 2.1. Forward NLT determination

The NLT is directionally separable since the NLT filters as well as the block transform are separable. In the encoder, the NLT filtering takes place in both directions (horizontal and vertical), and is followed by the application of the block transform to the lapped frame. In this context, the determination of the NLT is described for the one-dimensional case.

For one-dimensional data blocked into segments of size  $M$  samples, the cross-boundary overlap at each end is taken to be  $K$  samples. We take  $K = M/2$ . Therefore, a 1D lapped transform is applied to segments of length  $L = M + 2K$ . Then, the 1D lapped transform matrix is of size  $M \times L$ , having  $M$  basis functions of length  $L$ . The lapped transform is actually implemented by a filtering operation applied to cross-block boundary vectors of length

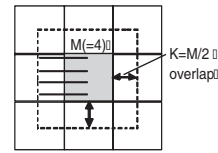


Fig. 2. NLT filtering

$2K$  ( $K$  samples on each side of a boundary, Fig. 2), followed by the block DCT.

The determination of the NLT filters is illustrated for the case  $M = 4$  and overlap  $K = M/2 = 2$ . Let us determine the following form for the NLT pre-filter  $\mathbf{P}$ :

$$\mathbf{P} = \begin{bmatrix} 1 + a & b & -b & -a \\ c & 1 + d & -d & -c \\ -c & -d & 1 + d & c \\ -a & -b & b & 1 + a \end{bmatrix} \quad (1)$$

where  $a, b, c, d$  are the parameters of the filter. Denoting by  $\mathbf{x} = [x(0) \ x(1) \ x(2) \ x(3)]^T$  a  $2K \times 1$  cross-boundary vector, the application of this filter results in a vector  $\mathbf{y}$  of size  $2K \times 1$ :

$$\mathbf{y} = \mathbf{P}\mathbf{x}, \quad (2)$$

where for example

$$y(0) = x(0) + a[x(0) - x(3)] + b[x(1) - x(2)]. \quad (3)$$

Let us define the element of adaptation for the NLT filtering as the set of cross-block boundary samples based on which the local statistics are computed and the NLT filter is determined. We will consider next the case where the element of adaptation consists of a cross-block boundary vector, in which case the NLT filter is adapted for each such vector. More coarsely, this element can be taken to contain all cross-boundary vectors overlapping two blocks (e.g., four vectors per boundary for  $M = 4$ ).

The parameters of the filter  $\mathbf{P}$  are functions of statistics of the cross-boundary samples being filtered (e.g., standard deviation), and the quantization step  $q$ , as shown next. Let us denote by  $a_0, b_0, c_0, d_0$  the parameters of the filter  $\mathbf{P} = \mathbf{P}_0$  corresponding to a fixed (baseline) lapped transform. This transform can be one determined based on the optimization of a coding gain criterion using assumptions about the statistics of the data (e.g., AR(1) process). In our case,

these are the parameters implementing an LBT ( $\rho = 0.95$ ) transform, where  $\rho$  is the correlation coefficient of the AR model. These parameters are fixed and known to both encoder and decoder.

Then, let the parameters  $a, b, c, d$  have the following form:

$$a = \frac{a_0}{\alpha_1}, \quad b = \frac{b_0}{\alpha_2}, \quad c = \frac{c_0}{\alpha_1}, \quad d = \frac{d_0}{\alpha_2}. \quad (4)$$

The filtering is adjusted based on the distance of the samples from the block boundary (see for example Eq. (3)).

Particular cases of lapped transforms can be obtained by properly setting the values of  $\alpha_1, \alpha_2$ . For  $\alpha_1, \alpha_2 \rightarrow \infty$ , the NLT filters become the identity matrix (and thus the lapped transform reduces to the original block transform of the codec), while for  $\alpha_1 = \alpha_2 = 1$ , the NLT reduces to the fixed lapped transform described by the parameters  $a_0, b_0, c_0, d_0$ .

Further,  $\alpha_1, \alpha_2$  have a nonlinear dependency on local data statistics (in this case, the standard deviation of the samples being filtered), and quantization step  $q$ , as follows:

$$\alpha_1 = \frac{2(1+\sigma)^{n_1}}{q^{m_1}}, \quad \alpha_2 = \frac{2(1+\sigma)^{n_2}}{q^{m_2}} \quad (5)$$

where  $n_1, m_1, n_2, m_2$  are exponents determined experimentally.

The dependencies shown in Eqs. (4), (5) cause the filtering strength to be increased as the variance of the data to be filtered decreases. At the same time, the filtering strength is increased as the quantization used in the codec becomes coarser. Typically this dependency is made weaker than the first one. In practice, the two adaptations work in concert to select filtering strength in areas of the image having different characteristics (e.g., relatively smooth areas coarsely quantized, or higher frequency areas also under coarse quantization).

In both the encoder and decoder an additional process takes place which is related to the determination of the inverse NLT, as shown in Fig. 1. This process is described next.

## 2.2. Reconstruction-NLT determination

The inverse NLT is implemented in a similar manner to the forward NLT, through the inverse block transform and the NLT inverse filtering operation (see Fig. 1). The form of the reconstruction NLT filter  $\mathbf{Q}$  can be obtained by a direct inverse of the forward NLT filter matrix  $\mathbf{P}$  in Eq. (1). Equivalently, it is illustrative to use a result obtained in [6] to write the filter  $\mathbf{P}$  in the following form:

$$\mathbf{P} = \frac{1}{2} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \quad (6)$$

where  $\mathbf{J}$  is the mirrored (left-right) identity  $\mathbf{I}$ , all sub-matrices are of size  $K \times K$  ( $=M/2 \times M/2$ ), and  $\mathbf{U}$  is the matrix of independent parameters of filter  $\mathbf{P}$ .

To obtain the filter  $\mathbf{P}$  in Eq. (1) we make:  $\mathbf{U} = \mathbf{I} + \mathbf{U}_0./\Theta$  where  $/$  signifies element-to-element division, and the matrices have the form:

$$\mathbf{U}_0 = \begin{bmatrix} d_0 & c_0 \\ b_0 & a_0 \end{bmatrix} \quad \text{and} \quad \Theta = \begin{bmatrix} \alpha_2 & \alpha_1 \\ \alpha_2 & \alpha_1 \end{bmatrix} \quad (7)$$

Then, the inverse filter  $\mathbf{Q}$  is obtained as:

$$\mathbf{Q} = \frac{1}{2} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \quad (8)$$

In either case considered above, for  $M = 4$  the inverse filter

matrix  $\mathbf{Q}$  has the following form:

$$\mathbf{Q} = \frac{1}{2} \begin{bmatrix} 1 + K(1 + \bar{d}) & -K\bar{b} & K\bar{b} & 1 - K(1 + \bar{d}) \\ -K\bar{c} & 1 + K(1 + \bar{a}) & 1 - K(1 + \bar{a}) & K\bar{c} \\ K\bar{c} & 1 - K(1 + \bar{a}) & 1 + K(1 + \bar{a}) & -K\bar{c} \\ 1 - K(1 + \bar{d}) & K\bar{b} & -K\bar{b} & 1 + K(1 + \bar{d}) \end{bmatrix} \quad (9)$$

where  $K = 1/[(1 + \bar{a})(1 + \bar{d}) - \bar{b}\bar{c}]$ . The parameters  $a, b, c, d$  have been replaced by their estimates  $\bar{a}, \bar{b}, \bar{c}, \bar{d}$  since the original data statistics are not available in the decoder. It would be impractical for the encoder to send as overhead these parameters for each element of adaptation. Therefore, methods for estimating the local original statistics used for the adaptation of the forward NLT filter are needed. For example, it is possible to learn in the encoder the parameters of a fit (e.g., least squares) between the original statistics of the data to be filtered and the corresponding statistics of the reconstructed lapped data (prior to the inverse NLT filtering). This is done in the statistics fit determination block shown in Fig. 1. The parameters of this fit represent the adaptation overhead that is transmitted to the decoder. Once the parameters of this fit (typically, a polynomial of low order) are determined, they can be used in both the encoder and decoder along with the reconstructed lapped data, to compute an estimate of the original local data statistics. This estimate is then used to determine the inverse NLT filter as in Eq. (9).

Denote by  $\sigma_o$  the standard deviation of an original vector of cross-boundary samples, and by  $\sigma_r$  the standard deviation of the corresponding reconstructed but still lapped vector of samples obtained after decoding, and prior to the inverse NLT filtering stage. Let the dependency between corresponding sets of such variables (for all block boundaries in a frame) be described and parameterized by a function  $f$  (e.g., obtained as a result of a least squares fit between the two sets). Then, using this fit, an approximation of  $\sigma_o$  can be obtained, denoted by  $\tilde{\sigma}_o$ :

$$\tilde{\sigma}_o = f(\sigma_r). \quad (10)$$

Hence, the inverse NLT filter parameter determination block in Fig. 1 can take the reconstructed lapped data the decoder generates, determine  $\sigma_r$ , and use the adaptation overhead consisting of the parameters of  $f$  to compute the parameters  $\bar{a}, \bar{b}, \bar{c}, \bar{d}$  of the NLT inverse filter as follows:

$$\bar{a} = \frac{a_0}{\bar{\alpha}_1}, \quad \bar{b} = \frac{b_0}{\bar{\alpha}_2}, \quad \bar{c} = \frac{c_0}{\bar{\alpha}_1}, \quad \bar{d} = \frac{d_0}{\bar{\alpha}_2}. \quad (11)$$

with

$$\bar{\alpha}_1 = \frac{2(1+\tilde{\sigma}_o)^{n_1}}{q^{m_1}}, \quad \bar{\alpha}_2 = \frac{2(1+\tilde{\sigma}_o)^{n_2}}{q^{m_2}} \quad (12)$$

and  $\tilde{\sigma}_o$  is computed as in Eq. (10). The parameters  $n_1, m_1, n_2, m_2$  have the same values as those used in the NLT forward filter.

## 3. EXPERIMENTAL RESULTS

In this section we show the results of incorporating the NLT into the H.264/AVC video codec (JM9.6) for intra-frame coding. The H.264 codec was operated in its high-complexity mode (advanced features enabled). Intra-prediction was active, as well as the H.264 post-filter applied after the inverse NLT. The NLT filtering in the encoder is applied to the original frame (and thus before the intra prediction). The NLT works with the 4x4 integer transform of H.264 ( $M = 4$ ), and the overlap is  $K = 2$  samples. The element of adaptation for the NLT was taken to be each cross-boundary vector of size  $2K = 4$ . Also, the estimates of the original statistics used for the inverse NLT determination were taken to be equal to the corresponding reconstructed lapped statistics (thus, the overhead is zero in this case, and

$f$  is the identity in Eq. (10)). It is expected that even larger gains will be obtained by using the statistics fit parametrization described in section 2 and therefore a better estimate of the original local statistics. The overhead is negligible in this case (only a few fit parameters per intra frame). An integer implementation was used for the NLT filtering.

The experiments used well-known test sequences at QCIF and CIF resolution (see sequences in Fig. 4), with a frame skip parameter as mandated by video standards test conditions (frame skip allows an increased spread of coded frames through a sequence). 60 frames from each sequence were encoded as intra-frames. The quantization steps were selected as  $QP = \{36, 34, 32, 30, 28, 24, 20, 16\}$  and the corresponding rate-distortion curves were determined for H.264 and H.264+NLT. We show in Fig. 3 only four such distortion curves due to space limitations. However, we also include the rate-distortion results for all the sequences, expressed using the BDRate and BDPSNR measures [7] routinely used in video standards. Essentially, these average-based measures eliminate the "sweet-spot" in rate-distortion results and make the comparison process more fair. The BDRate and BDPSNR were computed using the subset  $QP = \{36, 32, 30, 24\}$  such as to cover the range from 30-40dB PSNR. The individual NLT gains for all sequences as well as the average gain are shown in Fig. 4.

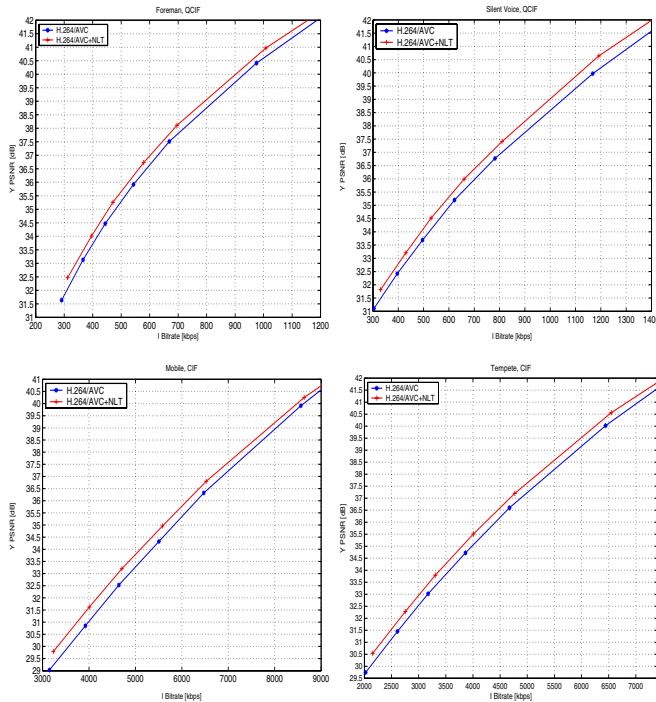


Fig. 3. Rate-distortion curves for H.264/AVC+NLT vs. H.264/AVC.

#### 4. CONCLUSION

The use of the determined nonlinearly-adapted lapped transform for the coding of intra-frames in a state-of-art video codec such as the H.264/AVC brings significant rate-distortion gains. The lapped transform's features include its nonlinear adaptation to both local image statistics and quantization regime, as well as its low overhead. The experimental gains are obtained while enabling features of the H.264/AVC codec whose purpose overlaps with that of the NLT (such as intra-prediction for reducing cross-boundary correlation, and the H.264 post-filter). As further advantages, the lapped trans-

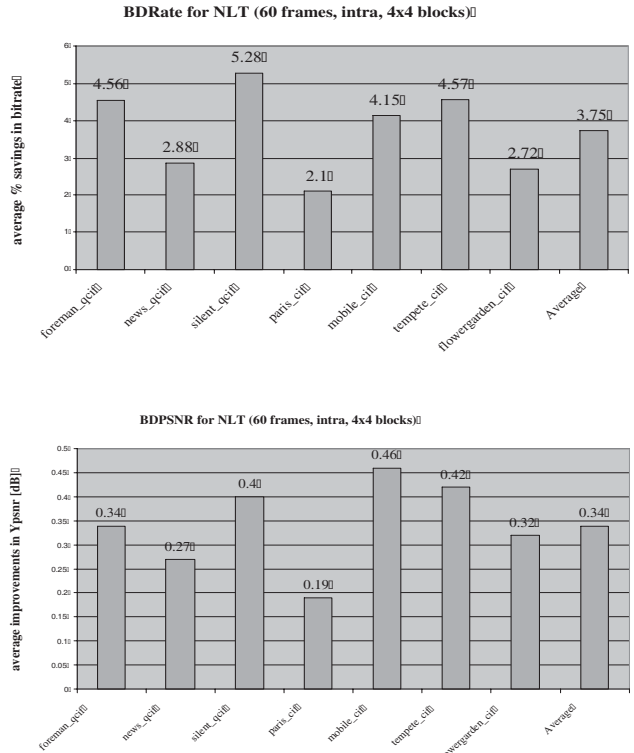


Fig. 4. BDRate savings and BDPSNR gains for H.264/AVC+NLT vs. H.264/AVC.

form has an integer implementation, and is incorporated in the codec while keeping unchanged the efficient integer block transform in H.264/AVC.

#### 5. REFERENCES

- [1] J. Mateos, A.K. Katsaggelos, and R. Molina, "A Bayesian approach for the estimation and transmission of regularization parameters for reducing blocking artifacts," *IEEE Transactions on Image Processing*, vol. 9, pp. 1200–1215, July 2000.
- [2] H. S. Malvar and D. H. Staelin, "The LOT: transform coding without blocking effects," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, pp. 553–559, April 1989.
- [3] H. S. Malvar, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Transactions on Signal Processing*, vol. 46, pp. 1043–1053, April 1998.
- [4] T. Tanaka, *Lapped transforms and their applications in image processing*, Ph.D. thesis, Tokyo Institute of Technology, February 2002.
- [5] T.J. Klausutis and V.K. Madiseti, "Adaptive lapped transform-based image coding," *IEEE Signal Processing Letters*, vol. 4, pp. 245–247, September 1997.
- [6] T. D. Tran, "Lapped transform via time-domain pre- and post-processing," in *Information Science and Systems*, 2001.
- [7] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," in *ITU-T SG16/Q.6, Document VCEG-M33*, 2001.