

Space Adaptive Wavelet Packet Image Compression

John R. Smith and Shih-Fu Chang

Columbia University Department of Electrical Engineering and
Center for Telecommunications Research, New York, N.Y. 10027

Abstract

We present a technique for lossy image compression based on the joint-adaptive space and frequency decomposition of images. The algorithm adapts to image content by both developing wavelet packet bases for separate areas of the image and by segmenting image subbands as needed. The elements of the expansion are a two-channel filter bank and a complete and disjoint binary segmentation system. We construct the joint space and frequency library by cascading permutations of these elements. We also formulate the space and frequency operations to be commutative, which allows for the full cascade system to be organized into a graph. After the full expansion, a coding cost is assigned to all elements in the library. The best joint space and frequency basis is found by pruning the graph which indexes the library such that the embedded graph with least cost is found. Its terminal nodes correspond to the best complete basis. We show that encoding the image in its best joint space and frequency basis improves compression performance.

Keywords

wavelet packets, basis search, spatial-frequency decomposition, image compression, rate-distortion theory, feature extraction

I. INTRODUCTION

The transformation of image data is of critical importance in applications of image compression, feature extraction and noise reduction. It provides an organization of image data by which the data can be better analyzed, prioritized, quantized and/or discarded. In general, the choice of signal expansion provides a fundamental limitation within these applications. For example, the rate-distortion performance of the compression of an image using the *JPEG* algorithm is governed by the ability of the Discrete Cosine Transform (DCT) to decorrelate the image data. For some images, such as those with textured regions, the correlation between pixels may be low. In these cases, the DCT does not provide the best organization of the image data. The suitability of a particular transformation depends on the signal characteristics. In spite of the deficiencies of non-adaptive transforms, typically, the selection of the signal expansion is made non-adaptively to the image signal.

In this paper we present an image compression system which includes the adaptive selection of the signal expansion. We describe in detail the design and implementation of the system. We also propose a novel approach for the efficient expansion of an image into a joint space and spatial-frequency (*s/s-f*) library from which the best basis is selected. The expansion is produced by cascading permutations of two fundamental elements: (1) two-channel filter bank, and (2) binary segmentation. Since these operations are commutative, we organize the cascade system into a graph. The best joint space and frequency basis is found from the library by pruning the graph such that the minimum cost embedded graph is found.

A. Applications

The most common signal expansions are produced by linear transforms and filter banks. The existence of fast algorithms and hardware for their implementation makes them extremely viable for practical systems. Some applications, such as real-time video coding, necessitate low complexity at both the encoder and decoder. These applications allow for little adaptivity in the signal expansion. As a consequence, fixed transforms such as the DCT have become most common for image and video compression.

However, there are many new applications that support an asymmetric model for coding whereby the objectives at the encoder and the decoder differ greatly. For example, in digital image and video libraries the data is typically encoded once – off-line, and is stored. Real-time compression is not required and the encoder does not need to be of low complexity. The compressed data may be retrieved later for purposes of analysis, decompression and viewing. This requires that the encoded images and videos are still quickly and cheaply decompressed and possibly analyzed directly in the compressed domain. In general, digital and image libraries require new and efficient compression systems that jointly (1) decrease the code size, (2) lower the visible distortion, and (3) improve access to visual content and image features [Pic94]. Given these new applications, it is worthwhile to investigate new procedures for the adaptive decomposition of images in the design of image compression systems.

This work was supported in part by the National Science Foundation under a CAREER award (IRI-9501266), and sponsors of the ADVENT project of Columbia University.

Recently new algorithms have been proposed for adaptive transformation of images. The purpose is to derive a transform or filter bank that is customized to each image. However, it must be done in such a way that the overhead from the representation of the basis does not off-set the bit reduction attained by encoding the data in the new basis.

B. Wavelet Packets

An algorithm for adaptive selection of the best frequency, or wavelet packet basis was proposed by Coifman, Quake, Meyer and Wickerhauser [CMQW90]. The wavelet packet algorithm generates a library of orthonormal functions that are derived from a single filter kernel. The wavelet packet algorithm searches through the library to find the least cost basis which also provides the best compression. By using a filter bank the wavelet packet library is produced by cascading filtering and downsampling operations in a tree-structure. The tree also guides the search for best basis [CMQW90][Wic90][RV93]. But an important drawback of wavelet packets is that the decomposition is performed on the entire image or on fixed blocks of the image. Wavelet packets do not adapt to variations in content across the separate regions of the image or to non-stationarity.

C. Double Tree

To address the problem of non-stationarity, the double tree algorithm was proposed by Herley, Kovačević, Ramchandran and Vetterli [HKRV93]. The double tree finds the best wavelet packet basis for a hierarchy of binary segmentations of the signal. The overall basis search identifies the most efficient dyadic segmentation and corresponding wavelet packet expansions for the segments. However, the double tree does not exploit the full potential of the joint space and frequency library. The insufficiency results from the asymmetric treatment of the space and frequency operations in the tree cascade. We will show that image compression performance improves when the basis search uses a graph to better exploit the space and frequency library, rather than the double tree.

D. Space And Frequency Graph

We proposed an improvement of the double tree that treats the space and frequency operations symmetrically and results in a graph structured cascade [SC95]. Both the double tree and its dual are embedded in the space and frequency graph. The graph structured decomposition provides a more complete joint space and frequency expansion of the signal. Using the same number of nodes as the double tree library, the space and frequency graph greatly increases the number of accessible bases that represent the signal. The space and frequency graph will be described in more detail in the following sections.

II. IMAGE COMPRESSION SYSTEM

In this section we describe the building blocks of the image compression system. To solve the problem of adaptively compressing the image, we develop an optimization procedure that involves basis selection, quantization and lossless encoding. As illustrated in Figure 1, the image compression system consists of three stages: (1) transform or filter bank, (2) quantization or lossy compression, and (3) lossless compression. The goal of the transform/filter bank stage is to re-organize the data such that the latter stages produce good compression. A non-adaptive image compression system fixes the first stage such that its processing is not signal dependent, and thus the compression performance is tuned only by the last two stages. For example, the *JPEG* image compression algorithm adjusts the quantization and run-encoding of the transform data to meet the compression criteria.

A. Stage 1 – Basis Selection

Improved compression can be attained by selecting the transform adaptively to the signal. Some fixed signal bases are supported by statistical image models. The justifications are as follows: (1) images generally have high pixel correlation – DCT decorrelates image data well, (2) images have most energy at low frequencies – the wavelet transform provides increasing frequency resolution at lower frequencies, (3) images are non-stationary – block-based transforms compensate for long-term non-stationarity. However, in many cases real images differ significantly from the models. As such, the non-adaptive transforms provide only suboptimal organization of the image data. The incorporation of a basis selection mechanism for the first stage improves the potential for compression in the system.

B. Stage 2 – Lossy Compression

Once the transform or subband coefficients are generated, the goal of stage 2 is to discard and de-emphasize visually insignificant information. This is achieved by quantizing the transform coefficients. We use an efficient procedure for

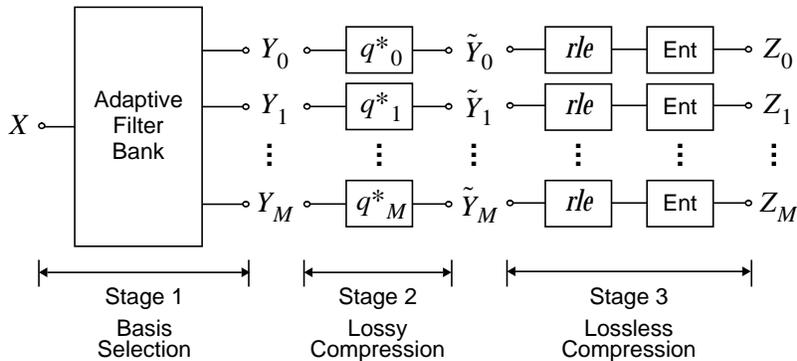


Fig. 1. Image compression system: stage 1 – image joint expansion and segmentation; stage 2 – selection of quantizers q_i^* , and quantization of signal expansion coefficients; stage 3 – lossless encoding using run length encoding (rle) and entropy encoding.

obtaining the optimal subband quantizers. The fundamental idea is that when the subbands are encoded independently, the quantizers should be selected such that all subbands operate at the same level of rate-distortion trade-off. The procedure for finding jointly the optimal rate-distortion trade-off and best basis was proposed by Ramchandran and Vetterli [RV93]. We use the algorithm to find the best space and frequency basis and quantizers. We also use uniform scalar quantizers for the DC subbands and deadzone scalar quantizers for other subbands. The deadzone quantizers work very well in producing runs of zeros which are encoded efficiently in the next stage [VK95].

C. Stage 3 – Lossless Compression

In stage 3 the quantized coefficients are encoded losslessly. We use run length encoding and entropy encoding. The goal of run length encoding is to encode the zeros jointly as they appear in runs. The direction of the scan influences the compression performance. Choices for scan direction include vertical, horizontal and zig-zag scan. Depending on the dominant frequencies corresponding to a particular subband, one scan direction may be better suited for extracting long runs of zeros. After run length encoding, the data is entropy encoded for which we use Huffman coding. We note that it is also possible to search for runs across subbands. This shows potential for improving both the lossy and the lossless compression components. For example, a common device is the *EOB* codeword which is used to denote that all data remaining in a scan is zero. It is used in both *JPEG* and the Wavelet Zero-Tree [Sha92] to terminate scans across subbands. The technique is essential for attaining good performance in these image compression systems.

D. Image Compression Optimization

In the image compression system, the compression of images is treated as an optimization problem. The goal is to encode the image using a minimum of bits that provide a minimum distortion in the reconstructed image. Each stage of the image compression system may be optimized independently, which often leads to good, yet overall suboptimal results. For example, Coifman and Wickerhauser [CW92] proposed an efficient scheme for selection of the best wavelet packet basis by minimizing information cost. However, an additional procedure would be needed to find the optimal encoding within that basis given a total bit constraint. The simultaneous optimization over multiple stages can attain better results. For example, Ramchandran and Vetterli’s algorithm, which simultaneously finds the best wavelet packet basis and quantizers, shows more favorable results [RV93].

E. Overall Design

We formulate the compression of images as the following optimization problems: (1) joint optimization over all stages, and (2) joint optimization over stages 1-2. In addition, we utilize a new and extensive library to provide the large number of potential bases. The elementary building blocks of the adaptive image expansion are (1) the two-channel quadrature mirror filter bank (QMF) filter bank, and (2) a disjoint and complete binary segmentation. Cascading permutations of these two units produces the expansion of the image in space and spatial-frequency. The building blocks of the signal expansion system are described in the next section.

III. FILTER BANK

The expansion of the image into a set of basis functions is accomplished using a discrete-time filter bank. A perfect reconstruction filter bank reproduces the image exactly. The QMF filter bank provides nearly perfect reconstruction.

The idea is to represent a signal using a set of orthonormal basis functions, $\{\phi_k\}$ such that,

$$x[n] = \sum_{k \in \mathcal{Z}} \langle \phi_k, x \rangle \phi_k[n], \quad n \in \mathcal{Z}, \quad \text{where } \langle \phi_k[n], \phi_l[n] \rangle = \delta[k-l] \quad (1)$$

The two channel filter bank is illustrated in Figure 2. It implements the orthonormal signal expansion when even shifts of the analysis and synthesis filters are related to the basis functions by,

$$\begin{aligned} h_0[2k-n] &= g_0[n-2k] = \phi_{2k}[n], \\ h_1[2k-n] &= g_1[n-2k] = \phi_{2k+1}[n]. \end{aligned}$$

The output of the analysis section, $Y_i(z)$, where $i = 0, 1$, is given by,

$$\begin{pmatrix} Y_0(z^2) \\ Y_1(z^2) \end{pmatrix} = \frac{1}{2} \begin{pmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{pmatrix} \begin{pmatrix} X(z) \\ X(-z) \end{pmatrix} \quad (2)$$

Finally, the signal is reconstructed by synthesis filters, G_i , using

$$\tilde{X}(z) = G_0(z)Y_0(z^2) + G_1(z)Y_1(z^2).$$

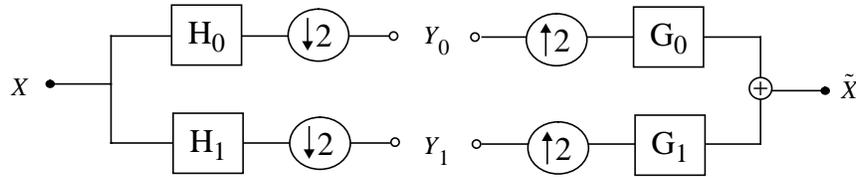


Fig. 2. Two channel filter bank, analysis filters H_i and synthesis filters G_i reconstruct $\tilde{X} = X$.

A. Adaptive Filter Bank

Cascading the two channel filter banks produces arbitrary filter banks. The transfer function of the cascade of filters and downsamplers is given by,

$$H_r(z) = \prod_{k=0}^{K_r-1} H_{S_r(k)}(z^{2^k}),$$

and is followed with downsampling by 2^{K_r} , where $K_r =$ depth of the cascade of path r and $S_r(k) \in \{0, 1\}$ is an indicator function that selects the filter, H_0 or H_1 , at stage k in the filter path.

Since the two-channel filter bank implements an orthonormal signal expansion, the cascaded filter bank also implements an orthonormal expansion. In other words, the impulse response of filters $h_r[n]$ and their appropriate shifts also form an orthonormal basis for $l_2(z)$ [VK95]. The filter bank may be used to construct an expansion adaptively to the signal characteristics. However, in practice it is difficult to determine the best filter bank structure without first expanding the signal. One solution is to over-expand the signal using a full cascade, and then choose the paths that gives the best complete expansion. However, finding the best spatial-frequency decomposition for an image still does not compensate for non-stationarity. Images are inherently non-stationary signals. The wavelet packet algorithm adapts to the whole image and not to different regions as needed. Therefore, we extend the wavelet packet algorithm by including the next building block of the joint image expansion – segmentation.

B. Segmentation

The complete and disjoint segmentation of a signal is defined as the Schur product of the signal with the rows of a binary matrix, whereby each column in the matrix sums to exactly one. The segmentation operation is given by,

$$\mathbf{y}_k = \mathbf{I}_k \cdot \mathbf{x} \rightarrow y_k[n] = I_k[n]x[n], \quad k = 0 \dots K-1 \quad (3)$$

where $I_k[n] \in \{0, 1\}$ and \mathbf{y}_k is the segmented portion of \mathbf{x} and K is the number of segments produced. When the set $\{I_k\}$ is a complete and disjoint segmentation over the sequence of N samples it requires the following

$$I_k \in \mathcal{B}^N \quad \text{and} \quad \bigcup_{k=0}^{K-1} I_k = \mathbf{1} \quad \text{and} \quad \bigcap_{k=0}^{K-1} I_k = \mathbf{0} \quad \iff \quad \sum_{k=0}^{K-1} I_k[n] = \mathbf{1},$$

where \mathcal{B} is the set of binary numbers. The perfect reconstruction binary segmentation system is illustrated in Figure 3.

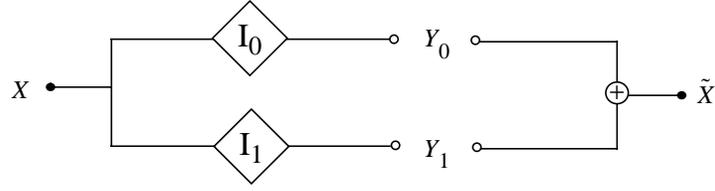


Fig. 3. Two split segmentation system – segmentation and summation give perfect reconstruction when I'_k s are disjoint and complete.

C. Filter Bank with Segmentation

By cascading the filtering and segmentation, the segmentation operator can be incorporated into the filter bank in two ways: (1) segmentation before analysis filtering, and (2) segmentation after analysis filtering. In practice, these are not equivalent because the filtering requires a border extension of the signal. Typically, with a rule-based border extension such as periodic extension, the added border data differ in the two cases. However, with the segmentations included in the filter bank, perfect reconstruction can still be maintained in both cases.

C.1 Case 1: Segmentation of Subbands ($F \rightarrow S \rightarrow S^{-1} \rightarrow F^{-1}$)

Applying the complete set of segmentation functions $\{I_k\}$ to each of the outputs of the analysis section of the filter bank produces the segmentation of the subbands. Each of the subband segments, $w_{lk}[n]$ is obtained from

$$w_{lk}[n] = I_k[n]y_l[n]. \tag{4}$$

Since the segmentation operation is invertible, the subbands can be reconstructed by summing the segments of each subband. Therefore, the perfect reconstruction property of the filter bank is maintained when the subbands are reconstructed before synthesis filtering.

C.2 Case 2: Segmentation Before Filtering ($S \rightarrow F \rightarrow F^{-1} \rightarrow S^{-1}$)

Applying the segmentation before subband decomposition creates multiple filter bank paths, one for each segment. Since each of the two channel filtering maintains perfect reconstruction of the segment, the overall system is perfect reconstruction.

D. Filtering, Segmentation and Reconstruction ($F \rightarrow S \rightarrow F^{-1} \rightarrow S^{-1}$)

Filtering and segmentation may be cascaded such that the orders within the analysis and synthesis cascades are different. This case do not guarantee that the perfect reconstruction property of the system is preserved unless careful consideration is made for the border extension used for filtering operations. But first, consider the filter bank that includes the segmentation of the subbands, as illustrated in Figure 4.

D.1 Cascade Filter Bank with Block Transform

In certain cases the filter operations do not require border extension. For example, when F implements a block transform, i.e., Haar filter bank, border extension is not necessary. In this case, the overall system maintains perfect reconstruction as follows: in the cascaded filter bank, synthesis filtering is applied to the segmented subbands, and the outputs are summed to reconstruct the signal. The analysis section consists of the cascade $F \rightarrow S$ where

$$y_l[n] = \sum_{m \in z} h_l[2n - m]x[m] \text{ and } w_{lk}[n] = I_k[n]y_l[n] = I_k[n] \sum_{m \in z} h_l[2n - m]x[m].$$

The synthesis section, $F^{-1} \rightarrow S^{-1}$, gives the reconstructed signal by

$$\tilde{x}[n] = \sum_{m \in z} \sum_{l=0}^1 \sum_{k=0}^1 g_l[n - 2m]w_{lk}[m] = \sum_{l=0}^1 \sum_{m \in z} g_l[n - 2m]y_l[m] \sum_{k=0}^1 I_k[m].$$

Since $\sum_{k=0}^1 I_k[n] = \mathbf{1}$,

$$\tilde{x}[n] = \sum_{l=0}^1 \sum_{m \in z} g_l[n - 2m]y_l[m] \xrightarrow{Z} \tilde{X}(z) = G_0(z)Y_0(z^2) + G_1(z)Y_1(z^2),$$

which is the same expression for the two channel perfect reconstruction filter bank shown previously. Therefore, a block transform and segmentation cascade system preserves the perfect reconstruction property of the filter bank.

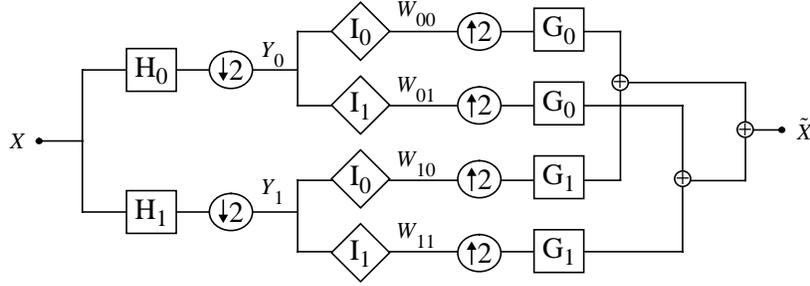


Fig. 4. Perfect reconstruction filter bank incorporating segmentation of subbands, analysis filters H_l , complete and disjoint indicator functions I_k , and synthesis filters G_j , where $l, k, j \in \{0, 1\}$.

D.2 Cascade Filter Bank with Arbitrary Filters

In the more general case, the filters H_l and G_j have length $M \geq 2$. Since the signal input into each filter is finite, say length N , the convolution produces an output signal of length $L = M + N - 1$. This increases the amount of signal data, which is undesirable in the application of image compression. To alleviate this problem, a finite length input signal is extended at its borders by $M/2 - 1$ data points, and the filter output is truncated to N points [KV89]. To minimize the distortion resulting from border extension, the same extension rule is used for both synthesis and analysis filtering. However, in the case of the filter bank of Figure 4, each synthesis filtering operation does not correspond to one analysis filtering operation. As a result, there is no simple solution for matching the synthesis and analysis border extension. For example, in Figure 4, the extended data for W_{00} is different from the extended data for the input X . Using the same extension rule generates different extended border data for W_{00} and X , which degrades the reconstruction property of the filter bank.

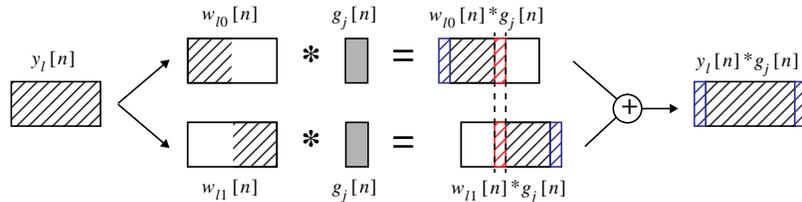


Fig. 5. Overlap-add method of signal convolution using independent convolution of sections of the signal.

Therefore, we propose an alternate solution for border extension for synthesis filtering. This includes both matching synthesis border extension to that of analysis filtering and using border extension with zeros. The technique is related to the *overlap-add method* of obtaining a linear convolution with an infinite length signal [OS75][VK95]. Using overlap-add, the signal is broken up into non-overlapping sections and each is extended at its borders with zeros. When each section is filtered, the output data expands by $M - 1$, where M is the filter length. The outputs of filtering all the sections are then realigned such that they overlap with neighboring sections by a total of $M - 1$ data points. By summing these sections together, the effect of breaking the signal into segments before filtering is eliminated. This is illustrated in Figure 5, and is formulated as follows: let each segment of y_l have only N nonzero points, then segment w_{lk} can be expressed as,

$$w_{lk}[n] = \begin{cases} y_l[n] & \text{if } kN \leq n \leq (k+1)N - 1 \\ 0 & \text{otherwise} \end{cases}$$

Then $y_l[n]$ is equal to the sum of the $w_{lk}[n]$'s and is given by $y_l[n] = \sum_{k=0}^{K-1} w_{lk}[n]$, for K segments. Then the convolution of $y_l[n]$ with $g_j[n]$ is equal to the sum of the $w_{lk}[n]$ convolved with $g_j[n]$, as desired,

$$y_l[n] * g_j[n] = \sum_{k=0}^{K-1} w_{lk}[n] * g_j[n]. \quad (5)$$

By using overlap-add in synthesis filtering, the effect of segmenting the subbands is eliminated. However, since the original input signal, $x[n]$, is of finite length, overlap-add is used only on segment $w_{lk}[n]$'s borders that correspond to the interior of $x[n]$. When the border of a segment coincides with a border of the unsegmented signal, the extension is matched to that used for analysis filtering.

This completes the description of the building blocks of the adaptive signal expansion. The two-channel filter bank is used to produce the orthonormal signal expansion and the segmentation bank divides its input into non-overlapping and complete sections. Filtering and segmentation can be arbitrarily cascaded in the analysis and synthesis systems. When the overlap-add rule is used for synthesis, the cascade orders of the analysis and synthesis do not need to match in order to maintain perfect reconstruction. This was shown for the two-channel filter bank with binary segmentation system above. Armed with these powerful operators, we now construct a system for the arbitrary and adaptive image expansion that includes decomposition in both space and spatial-frequency.

IV. TREE AND GRAPH STRUCTURED DECOMPOSITION

As mentioned above, the building blocks for the joint space and frequency decomposition consist of expansions using the two-channel filter bank and binary segmentation. Since we are developing the compression system for images, which are 2-D signals, we note here that we augment these basic building blocks for image expansion. For spatial-frequency decomposition, we utilize a 2-D signal, 4-channel filter bank by cascading two 2-channel filter banks and by transposing filter channel outputs. For segmentation, we utilize a quad-tree spatial segmentation by cascading binary segmentations and by transposing the outputs of the segmentation operators. In other words, the 2-D filtering and segmentation used here are separable and composed of independent operations on the rows and columns of the image. In many of the descriptions and illustrations that follow, a 1-D signal is shown for simplicity. However, the results apply directly to the 2-D image case when using the augmented building blocks. We now present high-level diagrams that describe the joint signal expansion. The notation in Figure 6 is used to illustrate the joint expansions in space and spatial-frequency. As shown in Figure 6, each straight-line in the high-level notation corresponds to a tree.

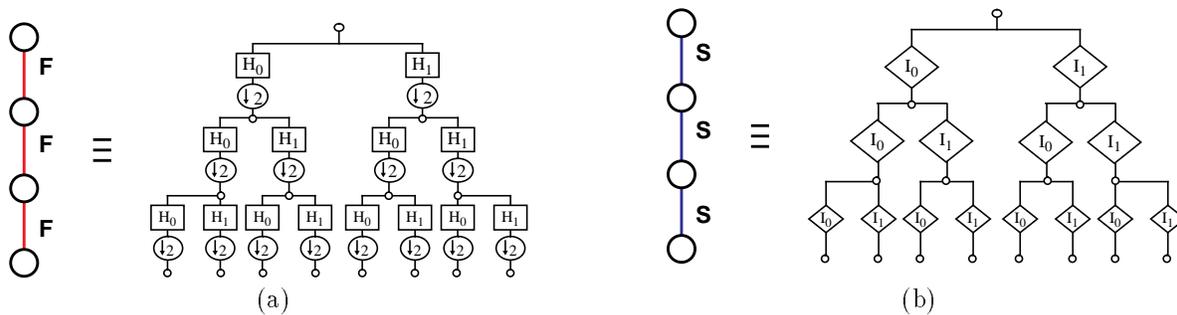


Fig. 6. High-level notation for expansions (a) tree-structured filter bank, (b) hierarchical segmentation.

A. Single Tree

By building the full cascade system for each class of operator, space and spatial-frequency respectively, a single tree structured decomposition is obtained. The single trees can be grown to arbitrary depth, $D \leq N \log_2 N$, where N is the length of the signal. A single tree built from filter bank units implements the tree-structured wavelet transform, while the single tree built from the segmentation operators implements the quad-tree segmentation.

A.1 Tree-Structured Wavelet Transform

The tree-structured wavelet transform, or wavelet packet tree, is constructed by recursively passing the output of each analysis filter channel through another instance of the filter bank. This is illustrated in Figure 6(a) and 7(a). The wavelet packet tree has the advantage of attaining the complete hierarchy of segmentations in frequency. By selectively choosing the frequency segments, the signal can usually be represented at lower storage cost than the original signal. The wavelet packet compression produces its coding gain by choosing the frequency segmentation that best captures a frequency spectrum that is not flat. However, as mentioned, the frequency spectrum for the signal as a whole is analyzed. Wavelet packets cannot compensate for or take advantage of signal nonstationarity.

A.2 Quad-Tree Spatial Segmentation

The full quad-tree spatial segmentation is obtained by cascading the segmentation operations such that each segmentation output is again segmented completely. This is illustrated in Figure 6(b) and 7(b). The quad-tree segmentation can be used to best compensate for image non-stationarity. By selectively choosing from the hierarchy of spatial segments, the image can be represented at lower cost. For images quad-tree spatial segmentation does not have the same energy packing ability as the filter bank. However, many images have largely diverse content in different spatial regions, and the independent treatment of regions can provide some gain.



Fig. 7. Using notation above, (a) tree-structured wavelet transform or wavelet packet tree, (b) hierarchical segmentation.

B. Double Tree

The weaknesses of the single trees – the tree-structured wavelet transform and binary segmentation was first addressed in [HKRV93]. The authors proposed a double tree which combines signal binary segmentation and wavelet packets. The double tree consists of the complete hierarchy of dyadic segmentations and the full wavelet packet decomposition of each segment. The double tree is illustrated in Figure 8(a). The diagram shows that the double tree does not segment any of the frequency F nodes. In other words, the outputs of the filter banks are not segmented spatially. This limits the performance of the double tree. For example, since frequency decomposition provides much of the energy compaction, branching from the root node towards the right, S , reduces the potential number of frequency iterations for the entire signal.

C. Dual Double Tree

The dual of the double tree also combines segmentation and wavelet packets. The dual double tree is produced by first growing a single wavelet packet tree followed by the full segmentation of all subbands. The dual double tree is illustrated in Figure 8(b). Its implementation is less complex than the double tree since it requires fewer frequency operations.

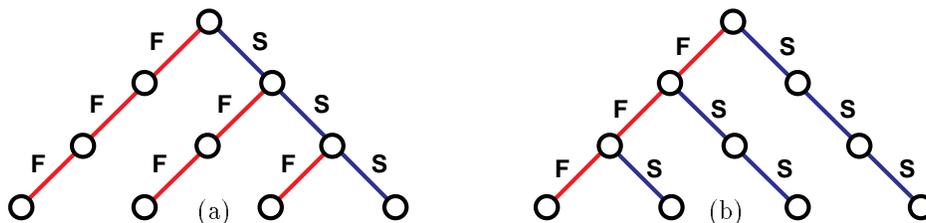


Fig. 8. (a) double tree, (b) dual double tree

D. Joint Adaptive Space and Frequency Tree

Both the double tree and the dual double tree provide an asymmetric treatment of space and frequency. As a result, neither tree attains the full joint decomposition in space and frequency. In order to produce the full joint decomposition, all nodes in the trees need both a frequency and segmentation branch. In other words, the outputs at each stage must be split by frequency and by segmentation. This creates the much larger tree structure which is illustrated in Figure 9(a). This full space and frequency tree combines both the double tree and the dual double tree into a single structure.

E. Space And Frequency Graph

By looking more closely at the series of branchings in the full space and frequency tree, Figure 9(b), we see that we can reduce the tree into an equivalent graph structure [SC95]. The payoff comes from recognizing commutativity in F and S branchings. As mentioned above, the filter bank and segmentations may be arbitrarily cascaded and perfect reconstruction is maintained even if the synthesis path has a different order than the analysis path. This commutativity is provided by using the overlap-add method in synthesis filtering. This observation can be used to greatly reduce the

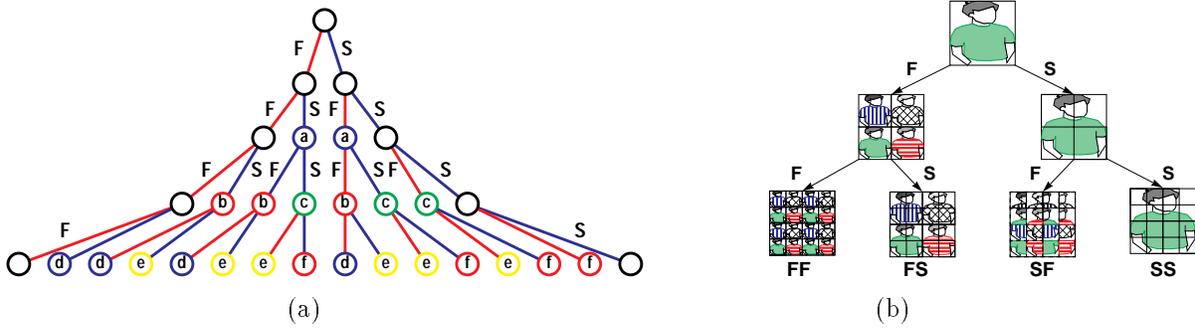


Fig. 9. (a) full space and frequency decomposition tree, (b) Image joint space and frequency expansion.

size and complexity of the joint expansion. The full adaptive tree contains many redundant nodes. Since the output of the $S \rightarrow F$ cascade is equivalent to the output of the $F \rightarrow S$ cascade, the data does not need to be produced twice. As illustrated for the image in Figure 9(b), the set of children generated from the $S \rightarrow F$ cascade is the same as that in the $F \rightarrow S$ cascade, but the order is different. When the full adaptive tree is modified to collect the redundant nodes, the space and frequency graph structure is produced, see Figure 10(a). Inspection reveals that the double tree, Figure 8(a), the dual double tree, Figure 8(b), and the space and frequency graph, Figure 10(a), have identical nodes, but with different connectivity. The space and frequency graph offers the maximum connectivity between nodes.

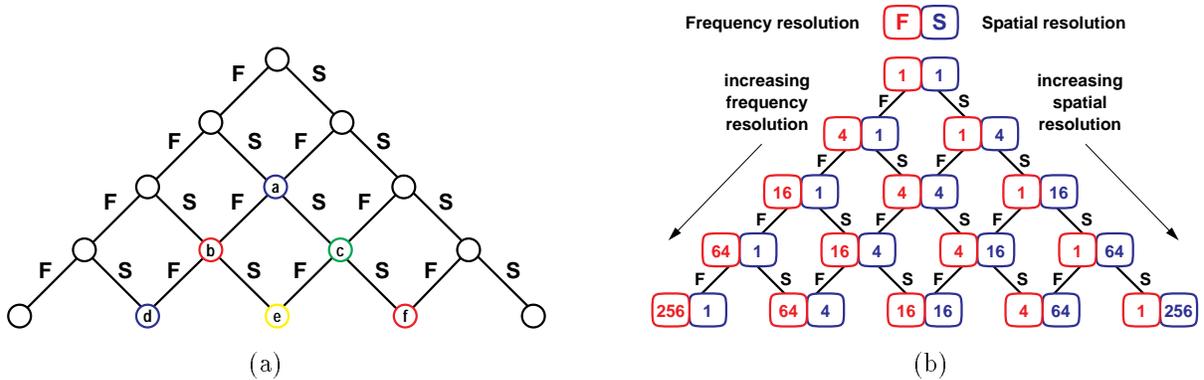


Fig. 10. (a) space and frequency graph, (b) Space and frequency resolution trade-off with graph depth.

V. SPACE AND FREQUENCY LIBRARY

The space and frequency graph provides a very convenient structure for the joint decomposition of the image in space and spatial-frequency. It requires only the basic building blocks of a two-channel filter bank and binary segmentation. The graph is a natural mechanism for accessing and indexing the nodes in the joint expansion. But several interesting questions remain, such as, can a more complete joint decomposition be found and what is the physical significance to each node in the graph? To answer these questions we analyze more closely the expansion produced by the space and frequency graph.

A. Space and Frequency Resolution Trade-off

As illustrated in Figures 10(b), the space and frequency graph explores the trade-off in the joint resolution in space and spatial-frequency. The data at each node of the graph corresponds to a particular spatial position and spatial-frequency range. Fundamentally, the joint resolution is limited by the uncertainty principle, and is bounded by the relations $\Delta_x^2 \cdot \Delta_u^2 \geq \frac{1}{4\pi}$ and $\Delta_y^2 \cdot \Delta_v^2 \geq \frac{1}{4\pi}$ where $[\Delta_x^2, \Delta_y^2]$ gives resolution in space and $[\Delta_u^2, \Delta_v^2]$ gives resolution in spatial-frequency. In the space and frequency graph, the frequency resolution is doubled with each filter bank operation and spatial resolution is doubled with each segmentation. Therefore, to answer the first question, the space and frequency graph includes only a quaternary hierarchy of decomposition in space and spatial-frequency. The graph cannot produce

any arbitrary resolution in space and spatial-frequency even within the bounds of the uncertainty principle. This manifests itself physically in that an arbitrary segment cannot be extracted from either the spatial plane or the spatial-frequency plane. In other words, the segmentations must be made on quad-tree boundaries and spatial-frequency subbands are extracted as quad-tree regions in the spatial-frequency plane.

B. Space and Frequency Library

There are two notions of libraries in the signal expansion: (1) the library of orthonormal basis functions utilized in the full wavelet packet expansion, as discussed earlier, and (2) the library consisting of all the nodes in space and frequency graph. In order to select the nodes from the space/spatial-frequency library that best represent the image, the nodes must be accessed, analyzed and compared. For one, each node in the graph can be indexed by its graph position. Alternatively, since each node represents a region in space and spatial-frequency, a node can also be indexed using a description of the region. For example, in the 1-D version of the graph – the joint time-frequency graph, a four-index transform of the 1-D signal is produced. Each node represents a region in the time-frequency plane that can be indexed by four parameters: i = time resolution, j = frequency resolution, k = position in time and l = position in frequency. The time-frequency library and the embedded graph is shown in Figure 11(a). For 2-D signals, the expansion similarly implements an eight-index transform, which includes x and y directions in space and spatial-frequency.

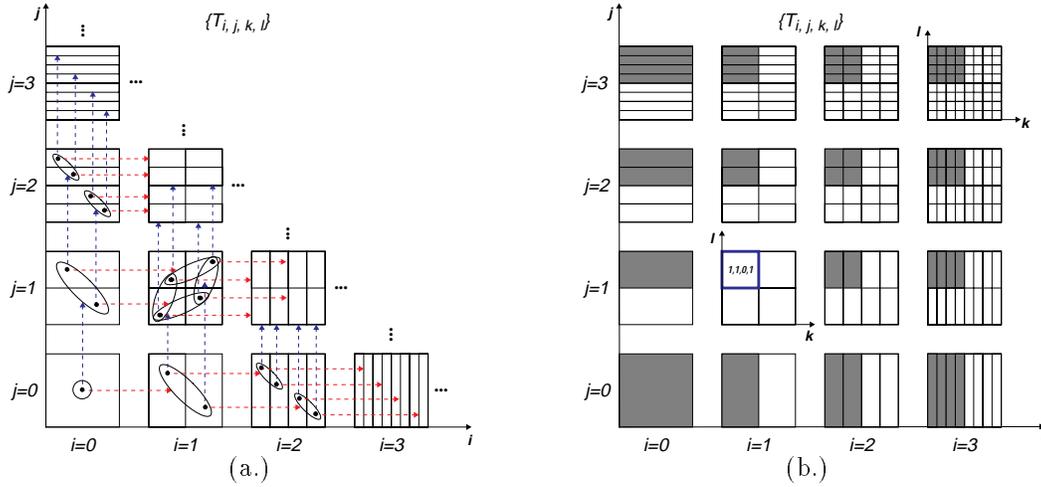


Fig. 11. Time-frequency library, (a) connectivity provided by time-frequency graph, (b) selection of node $t_{1,1,0,1}$ excludes all darkened nodes from the basis.

C. Basis Sets

For 1-D signals, a basis set corresponds to a tiling of the time-frequency plane [HKRV93]. For images, it corresponds to a partitioning of the joint 4-D space and spatial-frequency space. To generate a basis, the nodes in the library must be selected such that the image can be reconstructed completely and non-redundantly. Using the indexing notation of the time-frequency plane, this requires that nodes should be selected to form the basis $\{T_{i,j,k,l}^*\}$ such that

$$\bigcup_{t_{i,j,k,l} \in \{T_{i,j,k,l}^*\}} t_{i,j,k,l} \supseteq l^2(z) \iff \sum_{t_{i,j,k,l} \in \{T_{i,j,k,l}^*\}} 2^{-i-j} = 1.$$

If the nodes do not overlap, this guarantees that the basis is complete by requiring that the time-frequency plane is completely covered by the selected nodes. In order to ensure that the basis is non-redundant – no overlap, each node included in the basis necessarily excludes other nodes. For example, this is illustrated in Figure 11(b). When node $t_{1,1,0,1}$ is included in the basis, in order to have no overlap with other nodes, all the darkened nodes must be excluded from the basis. This requires that,

$$t_{i,j,k,l} \in \{T^*\} \iff t_{i',j',k',l'} \notin \{T^*\}, \text{ where } i', j' \in z, k2^{i'-i} \leq K < k2^{i'-i+1} \text{ and } l2^{j'-j} \leq L < l2^{j'-j+1}.$$

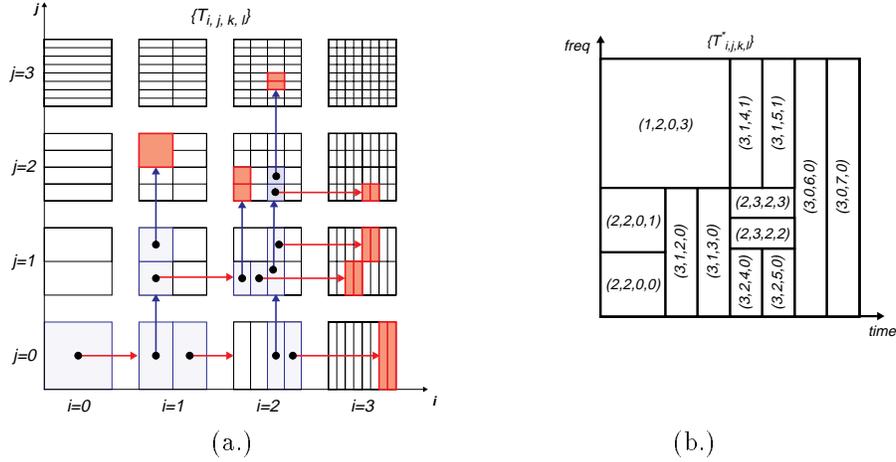


Fig. 12. (a) Basis selection from dyadic time/frequency library, (b) corresponding time/frequency tiling.

The restrictions on node membership in a basis can be translated into equivalent requirements on the graph. Any basis, or complete and non-redundant tiling of the time-frequency plane, corresponds to the set of terminal nodes of an embedded graph. For example, Figure 12(a) indicates a selection of nodes from the library that give the tiling of the time-frequency plane in Figure 12(b). Notice that the full graph has been pruned to produce the graph in Figure 12(a). The light shaded nodes are intermediate nodes in the pruned graph, the dark nodes are the terminal nodes and the unshaded nodes have been pruned from the graph. We also point out that the basis shown in Figure 12 is not accessible from either the wavelet packet tree, the spatial quad-tree, the double tree or the dual double tree. The completeness and non-redundancy requirements correspond to a recursive selection at each node, starting from the root node, of either (1) termination, (2) frequency branching or (3) segmentation.

VI. OPTIMIZATION, BASIS SEARCH AND ADAPTIVE PRUNING

We formulate the basis selection problem as finding the optimal pruning of the space and frequency graph. The pruning technique was first suggested for wavelet packets in [CMQW90] and was also used in [RV93]. These previous works applied the pruning algorithm to a tree with two decision points at each node: prune *vs.* not prune. The difference here is that the data structure is a graph with a three-way decision at each node: (1) terminate, (2) branch by frequency, or (3) branch by segmentation. However, the tree algorithm is easily extended to the graph. The algorithm works as follows: first, a coding cost value is assigned to each node. Then, by iterating over the full graph, the least cost embedded sub-graphs from each node are found. The final structure left after all pruning is the graph with the lowest total cost.

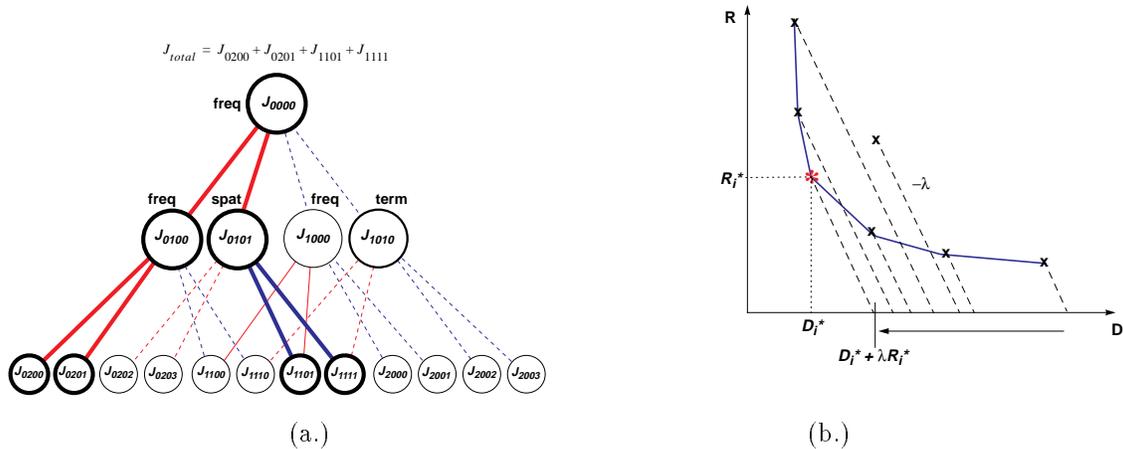


Fig. 13. (a) adaptive pruning, (b) rate-distortion operating point computation.

A. Cost Functions

In order to produce the basis with the lowest total coding cost, the cost function should indicate the cost of coding each node. One restriction is that the cost function is additive. Coifman and Meyer [CW92] suggested the entropy measure as an appropriate cost function. Ramchandran and Vetterli [RV93] suggested that a rate-distortion additive cost function is more suited to the compression problem. This two-sided measure better matches the goal of finding the optimal rate-distortion point for compression. We assign rate-distortion costs to each node in the graph by quantizing the data using the set of quantizers. This produces a set of rate-distortion points for each node. The procedure for pruning the graph involves both identification of the optimal rate-distortion points and the pruning decisions, and is illustrated in Figure 13. For a detailed description of the algorithm, refer to [RV93]. A brief description follows: the algorithm sweeps through a series of rate-distortion operating points λ , and the least cost quantizer for each node is selected as the one that minimizes $J^* = D_i^* + \lambda R_i^*$, see Figure 13(b). Next the least cost embedded graph is found by pruning, and the total rate is compared to the budget, see Figure 13(a). If the total rate exceeds the budget, λ is adjusted and the procedure is repeated.

VII. COMPRESSION EXAMPLES

The image compression was applied to several test images, see table below. The results show an increase in performance over ordinary wavelet packets and the double tree. This results from the increase in the number of potential bases that are accessible from the space and frequency graph compared to the tree methods. As shown in the table, the bases from the space and frequency library that are optimal for image compression cannot be reached using the wavelet packet tree or the double tree.

Image compression results on Barbara image

| | JPEG | wavelet | spatial quad-tree | wavelet packet | double tree | adaptive graph |
|-------------------|--------|---------|----------------------|-------------------|----------------|-------------------|
| 0.5bpp, $SNR_p =$ | 28.3db | 29.5db | 19.1db | 32.7db | 32.7db | 33.0db |
| 1.0bpp, $SNR_p =$ | 33.1db | 34.6db | 26.5db | 37.1db | 37.1db | 37.7db |
| 2.0bpp, $SNR_p =$ | 38.9db | 40.7db | 34.7db | 43.0db | 43.0db | 43.8db |

VIII. CONCLUSION

We presented a new algorithm for image compression that uses an adaptive joint space and frequency image expansion. The expansion is produced by building a graph that cascades both filter bank and segmentation operations. The nodes in the graph form a space and frequency library from which the best basis is selected. The library includes other expansions, such as wavelet, wavelet packet and double tree decompositions. The space and frequency graph offers the most rich expansion, and image compression performance increases when using the best basis selected from the space and frequency library.

REFERENCES

- [CMQW90] R. R. Coifman, Y. Meyer, S. Quake, and M. V. Wickerhauser. Signal processing and compression with wave packets. Technical report, Yale University Department of Math technical report, April 1990.
- [CW92] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *I.E.E.E. Transactions on Information Theory*, March, vol. 38, no. 2 1992.
- [HKRV93] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Tilings of the time-frequency plane: Constructions of arbitrary orthogonal bases and fast tiling algorithms. *I.E.E.E. Transactions on Signal Processing*, December 1993.
- [KV89] G. Karlsson and M. Vetterli. Extension of finite signals for sub-band coding. *Signal Processing*, vol. 17 1989.
- [OS75] A. V. Oppenheim and R. W. Schaffer. *Digital Signal Processing*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975. bookshelf, used for DSP.
- [Pic94] R. W. Picard. Content access for image/video coding 'the fourth criterion'. Technical report, MIT Media Laboratory and Modeling Group, technical Report No. 295, October 1994.
- [RV93] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *I.E.E.E. Transactions on Image Processing*, June 1993.
- [SC95] J. R. Smith and S.-F. Chang. Frequency and spatially adaptive wavelet packets. In *I.E.E.E. International Conference on Acoustics, Speech and Signal Processing*, May 1995.
- [Sha92] J. M. Shapiro. An embedded wavelet hierarchical image coder. In *I.E.E.E. International Conference on Acoustics, Speech, and Signal Processing 1992*, March 1992.
- [VK95] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice Hall PTR, Englewood Cliffs, N.J., 1995. bookshelf, used for signal processing.
- [Wic90] M. V. Wickerhauser. Picture compression by best-basis sub-band coding. Technical report, Yale University Department of Math technical report, January 1990.