

# Peak Transform for Efficient Image Representation and Coding

Zhihai He

**Abstract**—In this work, we introduce a nonlinear geometric transform, called *peak transform* (PT), for efficient image representation and coding. The proposed PT is able to convert high-frequency signals into low-frequency ones, making them much easier to be compressed. Coupled with wavelet transform and subband decomposition, the PT is able to significantly reduce signal energy in high-frequency subbands and achieve a significant transform coding gain. This has important applications in efficient data representation and compression. To maximize the transform coding gain, we develop a dynamic programming solution for optimum PT design. Based on PT, we design an image encoder, called the *PT encoder*, for efficient image compression. Our extensive experimental results demonstrate that, in wavelet-based subband decomposition, the signal energy in high-frequency subbands can be reduced by up to 60% if a PT is applied. The PT image encoder outperforms state-of-the-art JPEG2000 and H.264 (INTRA) encoders by up to 2-3 dB in peak signal-to-noise ratio (PSNR), especially for images with a significant amount of high-frequency components. Our experimental results also show that the proposed PT is able to efficiently capture and preserve high-frequency image features (e.g., edges) and yields significantly improved visual quality. We believe that the concept explored in this work, designing a nonlinear transform to convert hard-to-compress signals into easy ones, is very useful. We hope this work would motivate more research work along this direction.

**Index Terms**—Energy compaction, image compression, peak transform (PT), wavelet transform (WT).

## I. INTRODUCTION

THE KEY in efficient image compression is to explore source correlation so as to find a compact representation of image data. Over the past few decades, various spatial transforms, such as the Karhunen Løve transform (KLT), discrete cosine transform (DCT), and discrete wavelet transform (DWT) [1], have been developed to explore source correlation. As we know, KLT is the optimal spatial transform in removing source correlation. To elaborate, if the co-variance matrix of input source  $\mathbf{X}$  is known, we can design a KLT transform, denoted by matrix  $\mathbf{K}$ , such that the output components in  $\mathbf{Y} = \mathbf{KX}$  are uncorrelated. A theory states that KLT achieves a larger transform

coding gain than any other spatial transforms [1].<sup>1</sup> However, it should be noted that these transforms mentioned here are linear which can be represented by matrices. In image/video compression, these linear transforms, including KLT, DCT, and DWT, are used to remove source correlation. We refer to this type of statistical correlation explored by linear transforms as *linear correlation*. As we know, images (and videos) are a special type of data. They are not just 2-D arrays of pixels in a statistical sense. For example, if we randomly generate a 2-D array of data according to a given statistical distribution, the probability for this 2-D array of data to be a natural image is extremely low. This is because, besides statistical characteristics, natural images contains a lot of nonstatistical perceptual image features, such as edges, contours, patterns, structures, and objects. In other words, in images and videos, besides linear correlation, there is a significant amount of *nonlinear* source correlation presented by these perceptual image features. This type of nonlinear correlation has been left largely unexplored by linear transforms, such as KLT, DCT, and DWT.

### A. Related Work

During the past decades, researchers have been designing efficient transform tools, such as energy-compacting wavelets, to improve data compression efficiency [3]. Many state-of-the-art image compression schemes, such as SPIHT [6] and JPEG2000 [1], are built upon wavelet subband decomposition. Besides this effort, a significant body of research has been focused on developing efficient prediction schemes to explore the nonlinear source correlation which has been left largely unexplored by linear spatial transforms. For example, cross-subband parent–children dependency has been observed and explored by EZW (embedded zero-tree wavelet) [17], SPIHT [6], and many other wavelet-based image coding algorithms. Note that wavelet subband decomposition is in its nature a 1-D transform. Wavelet transform (WT) is able to efficiently capture and characterize 1-D singularity. To handle 2-D data, a common practice is to apply the WT to each row then to each column of the image. It has been observed that this type of extended 1-D WT is not able to efficiently represent 2-D curvilinear features, such as edges, object contours, etc [16]. Recently, several modified WTs which take edge flow or texture orientation into account have been developed. These transforms include curvelet [15], ridgelet [16], bandelet [7], wedgelet [8], contourlet [9], and

<sup>1</sup>However, to achieve this optimum performance, the KLT transform needs to adapt its transform matrix according to input source statistics. Therefore, in real applications, especially in image and video compression, where the input source has time-varying statistics, the KLT transform often has high computational complexity (in computing the transform matrix) and requires a significant amount of overhead bits to encode its transform matrix information.

Manuscript received October 5, 2006; revised February 26, 2007. This work was supported in part by the National Science Foundation under Grant DBI-0529082. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Giovanni Poggi.

The author is with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211 USA (e-mail: hezhi@missouri.edu).

Digital Object Identifier 10.1109/TIP.2007.896599

directionlet [10]. Some of these new transforms have shown successful applications in image denoising and enhancement [15], [16]. However, their potential in highly efficient image compression remains still unclear. One of the major reasons is that these methods cause significant over-sampling which prevents efficient data compression. Another important type of techniques, called *directional* or *orientation-adaptive wavelets*, which combine directional prediction and WT, have been developed in the literature [4], [5], [11], [12], [14]. The central idea in directional WT is to filter the image data along object structure or texture orientation, instead of filtering row by row or column by column. It has been shown that a significant performance gain can be achieved with directional WT, especially for images with significant texture patterns [4], [5], [12]. Spatial prediction is also a very important coding feature in H.264 coding of INTRA frames where image pixels are predicted by their neighbors from multiple directions [13].

### B. Overview of This Work

In this work, we propose to explore a new approach: developing a nonlinear geometric transform, called *peak transform* (PT), to assist the WT in exploring nonlinear data correlation. Conceptually speaking, the proposed PT is able to convert a hard-to-compress signal into an easier-to-compress one by exploring the nonlinear geometric source correlation within the input signal. From a transform analysis perspective, the PT is able to convert a high-frequency signal, which is often hard to be compressed by subband encoders, into a low-frequency one, which can be easily compressed, and as a result, significantly improves the transform coding gain [1]. We develop a new subband decomposition scheme, called *peak transform + wavelet transform* (PTWT) subband decomposition, for compression of generic data, including 1-D signals (e.g., speech and music) and 2-D images. Based on PTWT subband decomposition, we develop a new data compression system, called *PT encoder*, for image compression. In this paper, we will define the PT and discuss its major properties. We will develop a dynamic programming solution to find the optimum PT for a given input signal so as to maximize the transform coding gain. We will study various design issues of the PT image encoder. Our experimental results demonstrate that the PT is able to significantly improve the transform coding gain. The new PT encoder outperforms state-of-the-art image encoders, including JPEG2000 [1] and H.264 [13], by up to 2–3 dB, especially for images with a significant amount of high-frequency components.

The rest of the paper is organized as follows. In Section II, we will present the mathematical definition of PT and discuss its major properties from a data compression perspective. Section III will present the new PTWT subband decomposition scheme. In Section IV, we will present a dynamic programming solution for optimum PT design. The PT image encoder design will be discussed in Section V. In Section VI, we will discuss how the parameters of the PT encoder should be selected such that the overall rate-distortion performance of the encoder is maximized. The experimental results are presented in Section VII. Section VIII will discuss future research directions and conclude the paper.

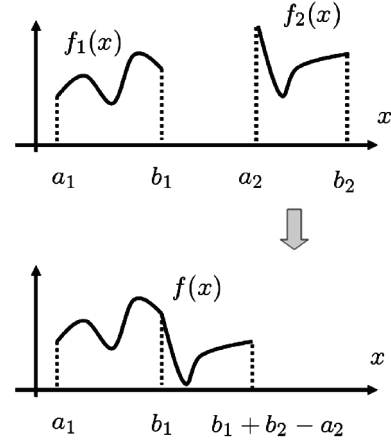


Fig. 1. Cascade of two curve segments.

## II. DEFINITION AND PROPERTIES OF PEAK TRANSFORM

In this section, we will first explain our motivation behind the idea of PT. We will then define the PT and discuss its major properties by examining several examples of PT.

### A. Motivation for Introducing Peak Transform

A well-known observation in data compression with transform coding is that: low-frequency smooth signals can be easily compressed while high-frequency ones (signals with a lot of high-frequency components) are not. Therefore, it is highly desirable to find a transform which is able to convert high-frequency signals into low-frequency ones. This will significantly improve data compression efficiency. Our research problem in this work becomes: Can we design a transform to preprocess the input signal (image data) such that the preprocessed signal has a much smaller high-frequency subband energy than the original signal? In the following sections, we will introduce the PT and show that it has this unique and important property.

### B. Definition of Peak Transform

*Definition: Curve Segment*—In this work, a curve segment is defined to be a continuous function  $f(x)$  defined over a finite interval  $[a, b]$ .

*Definition: Cascade of Curve Segments*—Given two curve segments  $f_1(x)$  and  $f_2(x)$  defined over finite intervals  $[a_1, b_1]$  and  $[a_2, b_2]$  with  $b_1 \leq a_2$ , the cascade of these two curve segments yields a new curve segment  $f(x)$  defined over  $[a_1, b_1 + b_2 - a_2]$

$$f(x) = \begin{cases} f_1(x), & x \in [a_1, b_1] \\ f_2(x) - f_2(a_2) + f_1(b_1), & x \in (b_1, b_1 + b_2 - a_2]. \end{cases} \quad (1)$$

We denote this cascading operation by

$$f(x) = f_1(x) \uplus f_2(x). \quad (2)$$

Physically, the new curve segment  $f(x)$  is obtained by joining two curve segments  $f_1(x)$  and  $f_2(x)$  with proper shifting operations as illustrated in Fig. 1.

Now we are ready to define the PT.

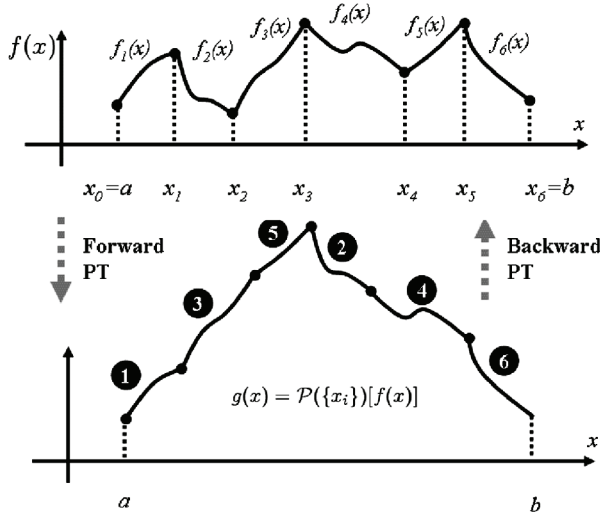


Fig. 2. Example of five-point PT.

**Definition:  $N$ -Point Forward Peak Transform**—A continuous function  $f(x)$  is defined over a finite interval  $[a, b]$ . This interval is partitioned into  $N + 1$  subintervals by  $N$  points,  $a < x_1 < x_2 < \dots < x_N < b$ . For convenience, we write  $x_0 = a$  and  $x_{N+1} = b$ . We refer to  $\{x_n\}$  as peaks (or breaking points). The curve segment defined over interval  $[x_{i-1}, x_i]$  is denoted by  $f_i(x)$ ,  $1 \leq i \leq N + 1$ . The  $N$ -point forward PT of  $f(x)$ , denoted by  $\mathcal{P}(\{x_i\})[f(x)]$ , is defined as

$$\mathcal{P}(\{x_i\})[f(x)] = g_o(x) \uplus g_e(x) \tag{3}$$

where

$$g_o(x) = f_1(x) \uplus f_3(x) \uplus \dots \uplus f_{2 \cdot \lfloor (N-1)/2 \rfloor + 1}(x) \tag{4}$$

and

$$g_e(x) = f_2(x) \uplus f_4(x) \uplus \dots \uplus f_{2 \cdot \lfloor N/2 \rfloor}(x) \tag{5}$$

are the cascades of all odd- and even-numbered curve segments, respectively. Here,  $2 \cdot \lfloor (N - 1)/2 \rfloor + 1$  and  $2 \cdot \lfloor N/2 \rfloor$  are, respectively, the largest odd and even integers that are less than or equal to  $N$ . Physically speaking, in forward PT, we first cascade all odd-numbered curve segments then all even-numbered curve segments and form a new curve. Fig. 2 shows an example of five-point PT. For convenience and in case of no confusion, we denote the forward PT by  $\mathcal{P}[f(x)]$ .

It can be seen that the PT only changes the order of curve segments and is reversible. The backward transform can be done by simply recascading the curve segments according to their original order. We denote the backward PT operation by  $\mathcal{P}^{-1}(\{x_n\})[\cdot]$ . In the following, we give a mathematical definition of backward PT.

**Definition: Backward Peak Transform**—Let

$$g(x) = \mathcal{P}(\{x_i\})[f(x)] \tag{6}$$

which is the output of PT of  $f(x)$  with peaks  $a < x_1 < x_2 < \dots < x_N < b$ . Let  $l_i = x_i - x_{i-1}$  which is the length of the  $i$ th interval and

$$L_o = \sum_{k=0}^{\lfloor N/2 \rfloor} l_{2k+1} \tag{7}$$

which is the total length of all odd-numbered intervals. In backward PT, the original function  $f(x)$  is constructed as follows:

$$f(x) = f_1(x) \uplus f_2(x) \uplus \dots \uplus f_{N+1}(x). \tag{8}$$

Here, if  $i$  is odd, the curve segment  $f_i(x)$  is given by

$$f_i(x) = g(x), \quad \sum_{k=0}^{\lfloor i/2 \rfloor - 1} l_{2k+1} \leq x \leq \sum_{k=0}^{\lfloor i/2 \rfloor} l_{2k+1}. \tag{9}$$

If  $i$  is even

$$f_i(x) = g(x), \quad L_o + \sum_{k=1}^{\lfloor i/2 \rfloor - 1} l_{2k} \leq x \leq L_o + \sum_{k=1}^{\lfloor i/2 \rfloor} l_{2k}. \tag{10}$$

It should be noted that, although the forward and backward PTs in the above are defined for continuous-time signals, they can be also defined for discrete-time signals in the same manner. For image data, as shown in the following section, we can simply treat each row or column of data as a piece-wise linear function with a straight line connecting two neighboring data points and apply the PT.

### C. Properties of Peak Transform

One important property of PT is that it is capable of converting a high-frequency signal into a low-frequency one if the peaks are properly selected. In other words, the PT is able to reduce the amount of high-frequency components in signals. To see this, let us first look at a toy example. As we know, 1-D signals can be approximated by piecewise linear functions. Fig. 3(A) shows a special piece-wise linear function  $f(x)$  in which all descending line segments have the the same slope  $k_1 < 0$  and all ascending line segments have the same slope  $k_2 > 0$ . We use local minimum and maximum points (breaking points) as peaks  $\{x_n\}$  for PT, as shown in Fig. 3(A) with diamonds. Fig. 3(B) shows the PT output signal  $\mathcal{P}(\{x_i\})[f(x)]$ . It can be seen that the transform output is much smoother than the original one and the high-frequency components of the original signal have been dramatically reduced. This can be demonstrated with subband decomposition. We pass the signals in Fig. 3(A) and (B) into a Daubechies (9, 7) filter bank. Fig. 4(A) and (B) shows the respective outputs of the high-pass filter. It can be seen that coefficients in high-frequency subbands with PT have much smaller magnitudes. The total energy of high-frequency subbands with PT is only 9% of that without PT. We expect that the PT has a similar behavior on other piecewise linear functions if the peaks are properly selected.

In the following example, we demonstrate this unique property of PT on more generic signals. We take the 330th row of image *Barbara* as the input signal  $f(x)$ , which is shown in Fig. 5(A). The peaks used in PT are shown in diamonds.

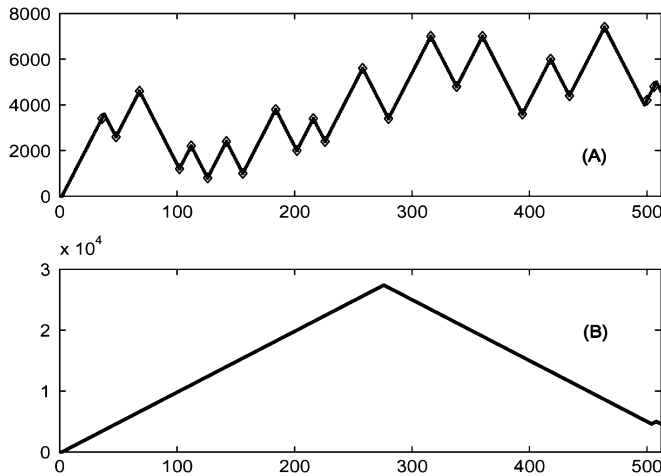


Fig. 3. PT of a piece-wise linear function: (A) original piece-wise linear function and the selected peaks shown in diamonds; (B) PT output.

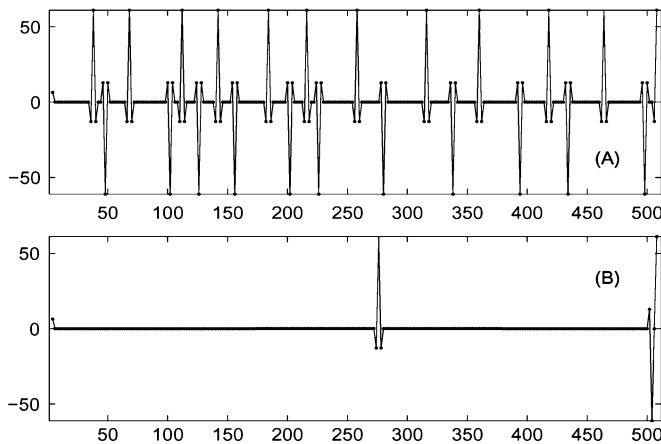


Fig. 4. (A) High-frequency components of the original signal; (B) high-frequency components of the PT output.

In Section IV, we will explain how to obtain these peaks. In this example, we first apply the forward PT to  $f(x)$  and obtain  $\mathcal{P}[f(x)]$ . Fig. 5(B) shows the PT output. As in the toy example, we pass  $\mathcal{P}[f(x)]$  to a Daubechies (9, 7) filter bank and obtain the low and high-frequency components (subbands) of  $\mathcal{P}[f(x)]$ . For easy comparison, we apply a backward PT to these two subbands to bring transform coefficients back to their original image locations. Fig. 6(A) and (B) shows the high-frequency subbands of the input signal without and with PT. According to our experiment, the energy of the high-frequency subband with PT is about 43% of that without PT. Fig. 7 shows the corresponding low-frequency subbands. We can see that they are very similar to each other.

From these two experiments, we can see that the PT has a very important property: it is able to significantly reduce high-frequency subband energy while maintaining low-frequency subband characteristics. This will have significant applications in data compression. In this work, we focus our effort on optimum PT design for 2-D image compression. The algorithms and results obtained in this work can be easily extended to 1-D cases for acoustic data or other 1-D data compression.

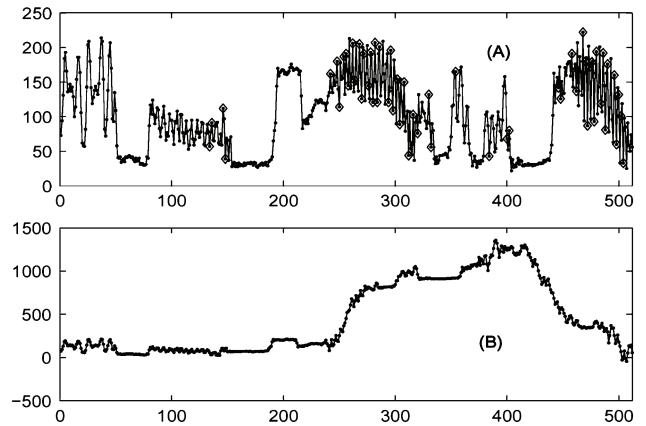


Fig. 5. PT of the 330th row of image Barbara: (A) original piece-wise linear function and the selected peaks shown in diamonds; (B) PT output.

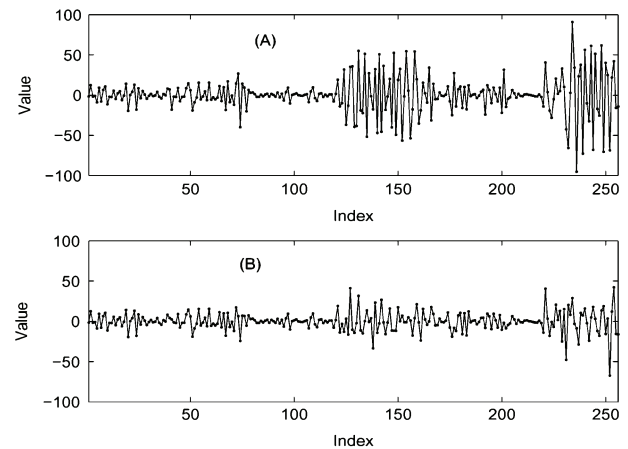


Fig. 6. (A) High-frequency subband of the original signal without PT; (B) high-frequency subband with PT.

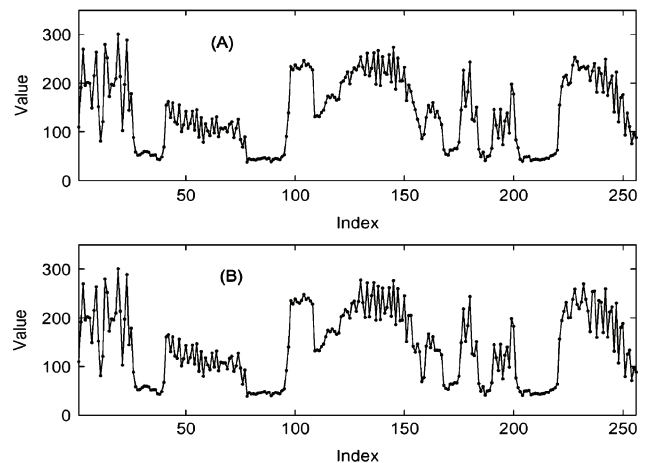


Fig. 7. (A) Low-frequency subband of the original signal without PT; (B) low-frequency subband with PT.

To successfully design an image encoder based on PT, the following issues need to be fully addressed. 1) *Optimum peak transform design*. How do we find the optimum set of peaks  $\{x_n\}$  such that the total energy of high-frequency subbands is minimized? 2) *Image encoder design*. Note that, in PT, the peak locations are overhead information which needs to be encoded

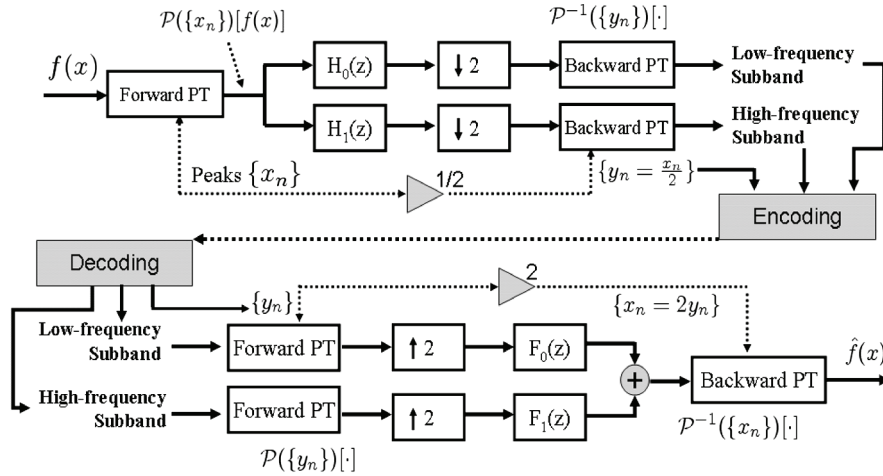


Fig. 8. PTWT subband decomposition.

and transmitted to the decoder for backward PT and image reconstruction. How to efficiently encode the peak locations and how to design an image encoding system to achieve high data compression efficiency are two open issues. 3) How do we analyze, control, and optimize the rate-distortion behavior of the encoder such that its compression performance is maximized? In the following sections, we will investigate these issues.

### III. PEAK TRANSFORM-BASED SUBBAND DECOMPOSITION AND SYNTHESIS

In this section, we will discuss how to design a data compression system based on PT. In PT-based data compression, the PT is coupled with WT and subband decomposition to minimize the signal energy of high-frequency subbands.

Let us start with 1-D input signals, which can be a row or a column of image pixels. Let  $f(x)$  be an input signal of length  $M$  where  $1 \leq x \leq M$  (as mentioned in Section II-B, we consider the discrete-time input signal as a piece-wise linear function  $f(x)$ ). As illustrated in Fig. 8, an  $N$ -point forward PT with peaks  $\{x_n\}$  is applied to the input signal  $f(x)$ . The PT output, denoted by  $\mathcal{P}(\{x_n\})[f(x)]$ , is then passed to a two-branch filter bank with a low-pass filter  $H_0(z)$  and a high-pass filter  $H_1(z)$ . An  $N$ -point backward PT with peaks  $\{y_n\}$ , denoted by  $\mathcal{P}^{-1}(\{y_n\})[\cdot]$ , is applied to both low and high-frequency subbands. It should be noted that the sizes of low and high-frequency subbands due to down-sampling are both  $M/2$ . Therefore, the original peaks cannot be used for backward PT any more. One possible approach is to force the location of each original peak  $x_n$  to be an even integer. During the backward PT of two subbands, we use  $\{y_n = x_n/2\}$  as peaks.

The backward PT is introduced here for two major reasons. First, according to our experience, the backward PT, acting as a balance to the forward PT, is able to alleviate the accumulation of quantization errors. Second, using  $y_n = x_n/2$  as peaks, the backward PT is able to bring WT coefficients back to their original image locations so as to maintain source correlation. We will explain this in detail within the context of image compression in Section V. The above procedure, which decomposes the input signal into two subbands using PT and WT, is called *PTWT subband decomposition*.

The PTWT subband decomposition procedure can be repeated for low and high-frequency subbands to further decompose the signal into more frequency subbands. The frequency subbands will be quantized, entropy encoded, and transmitted to a decoder, as illustrated in Fig. 8. The peaks  $\{y_n\}$  will be also compressed with lossless coding and sent as overhead information to the decoder. At the decoder side, the peaks  $\{y_n\}$  will be decoded. A forward PT  $\mathcal{P}(\{y_n\})[\cdot]$  with peaks  $\{y_n\}$  will be applied to low and high-frequency subbands. After subband synthesis, a backward PT  $\mathcal{P}^{-1}(\{x_n\})[\cdot]$  will be performed to obtain the reconstructed signal  $\hat{f}(x)$ . Here,  $x_n = 2y_n$ . We can see that the PTWT subband decomposition and synthesis illustrated in Fig. 8 guarantees perfect reconstruction and the only loss is caused by quantization, as in conventional wavelet-based data compression [3].

### IV. OPTIMUM PEAK TRANSFORM DESIGN USING DYNAMIC PROGRAMMING

In this section, we will formulate the optimum PT design problem and develop a dynamic programming solution to find the optimum PT.

#### A. Problem Formulation

As in conventional WT-based subband decomposition, the PTWT subband decomposition outlined in Fig. 8 will decompose the input signal into a series of subbands, which consist of one lowest-frequency subband and a number of high-frequency subbands. Here, by “*high-frequency subbands*,” we mean those subbands which have ever been filtered by the high-pass filter  $H_1(z)$ . In data compression with subband decomposition, bits are mostly used for encoding high-frequency subbands because the lowest-frequency subband becomes very small in size after several levels of subband decomposition. We observe that, if the input image, quantization, and data coding schemes (such as EZW or SPIHT) are given, the performance of subband image compression mainly depends on how efficiently the subband decomposition (or WT) is able to remove spatial source correlation and minimize the energy of high-frequency subbands. In general, a smaller high-frequency subband energy often results in less coding bits [17]. Therefore, in optimum PT design, we need

to find the PT which is able to minimize the energy of high-frequency subbands.

Let us look at the PTWT subband decomposition in Fig. 8 (the top half) in more detail. Our purpose in optimum PT design is to minimize the energy of the high-frequency subband. We denote the high-pass filtering operation by  $\mathcal{H}_1\{\cdot\}$  and the high-frequency PTWT subband after down-sampling by  $u(x)$ . Since  $u(x)$  is the high-pass filtering output of the PT data, we write

$$u(x) = \mathcal{H}_1\{\mathcal{P}(\{x_n\})[f(x)]\} \quad (11)$$

whose energy is denoted by

$$\mathcal{E}_u = \mathcal{E}_{H_1}\{\mathcal{P}(\{x_n\})[f(x)]\}. \quad (12)$$

Note that the PT is fully characterized by its peaks  $\{x_n\}$ . Therefore, the optimum PT can be formulated as: *find peaks  $\{x_n\}$  such that the high-frequency subband energy is minimized*, which can be mathematically written as

$$\begin{aligned} \min_{N, \{x_n\}} \quad & \mathcal{E}_u = \mathcal{E}_{H_1}\{\mathcal{P}(\{x_n\})[f(x)]\} \\ \text{s.t.} \quad & a < x_1 < x_2 < \dots < x_N < b. \end{aligned} \quad (13)$$

Here,  $N$  is also an optimization variable since the total number of peaks  $N$  needs to be determined. Theoretically speaking, any even-numbered pixel in the input or any even integer between 1 and  $M$  (the size of the input signal) can be chosen as a peak for PT. The reason for ‘‘even’’ is explained in Section III. Define

$$b_m = \begin{cases} 1, & \text{if } 2m \text{ is chosen as a peak} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where  $1 \leq m \leq M/2$ . The optimization problem in (13) can be rewritten as

$$\begin{aligned} \min_{\{b_1, b_2, \dots, b_{M/2}\}} \quad & \mathcal{E}_u = \mathcal{E}_{H_1}\{\mathcal{P}(\{x_n\})[f(x)]\} \\ \text{s.t.} \quad & \{x_n\} = \{2m : b_m = 1, 1 \leq m \leq M/2\}. \end{aligned} \quad (15)$$

We can see that there are  $2^{M/2}$  possible combinations of  $\{b_1, b_2, \dots, b_{M/2}\}$ . One brute-force approach to solving the optimum PT design problem in (18) is to search all of these  $2^{M/2}$  combinations and find the one which has the minimum high-frequency subband energy  $\mathcal{E}_u$ . Clearly, the computational complexity of this approach is too high. In the following section, we propose a fast and efficient dynamic programming solution for optimum PT design.

### B. A Dynamic Programming Solution

Our dynamic programming solution is based on the following two important observations. The first observation will reduce search space of the optimization problem while the second observation will lead us to a dynamic programming solution.

1) *Observation A:* The objective of PT is to rearrange the input data on a segment basis and make the connection between segments smoother so as to significantly reduce its high-frequency subband energy around these connecting points (peaks). This can be easily seen from the example shown in Fig. 3. Therefore, it is more desirable to choose those locations with large high-frequency energy levels as peaks and we hope the PT is

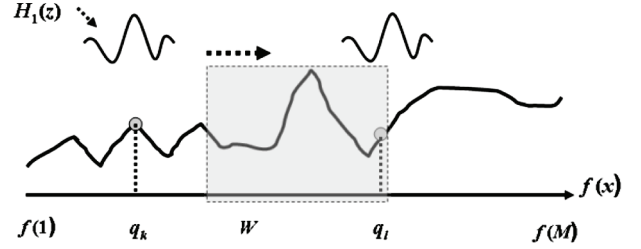


Fig. 9. High-pass filtering of an input signal.

able to significantly reduce the energy levels at these locations. To do this, we can apply the high-pass filter  $H_1(z)$  to the input  $f(x)$ . Let  $\mathcal{H}_1[f(m)]$  be the filtering output at location  $m$  or the signal response to the high-pass filter  $H_1(z)$ , as shown in Fig. 9. We can then choose peaks from the following candidate set

$$\mathcal{Q} = \{2m : |\mathcal{H}_1[f(2m)]| > \delta, 1 \leq m \leq M/2\} \quad (16)$$

where  $\delta$  is a threshold. The value of  $\delta$  will be determined later. Note that here we make sure that all candidate peaks are even integers. Let  $L = |\mathcal{Q}|$  be the size of the candidate set  $\mathcal{Q}$ . Now, the optimum PT design problem reduces to: *choose peaks  $\{x_n\}$  from  $\mathcal{Q}$  such that the high-frequency subband energy is minimized*. Let

$$b_l = \begin{cases} 1, & \text{if } q_l \in \mathcal{Q} \text{ is chosen as a peak} \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

The optimization problem in (15) can be rewritten as

$$\begin{aligned} \min_{\{b_1, b_2, \dots, b_L\}} \quad & \mathcal{E}_u = \mathcal{E}_{H_1}\{\mathcal{P}(\{x_n\})[f(x)]\} \\ \text{s.t.} \quad & \{x_n\} = \{q_l : b_l = 1, 1 \leq l \leq L\}, b_l \in \{0, 1\}. \end{aligned} \quad (18)$$

Here, the size of search space is reduced to  $2^L$ . Clearly, the smaller the  $\delta$ , the larger the search space.

2) *Observation B:* We also observe that the optimum PT design problem has the following two properties which allow us to solve this problem using a dynamic programming approach.

First, the decision on  $b_l$ , either choosing  $q_l$  as peak or not, only changes the high-frequency subband energy within a neighborhood of  $q_l$  and the neighborhood size is determined by the length of the high-pass filter  $H_1(z)$ . This implies that, if two locations  $q_k$  and  $q_l$  are separated from each other by a distance larger than the filter length (denoted by  $W$ ) of  $H_1(z)$ , as illustrated in Fig. 9, the decisions on choosing  $q_k$  and  $q_l$  as peaks or not are independent of each other. The decision on  $q_l$  only depends on the status and decisions of previous candidate peaks within the window of  $[q_l - W, q_l]$ .

Second, the objective function  $\mathcal{E}_u$  is the high-frequency subband energy, which is the summation of the energy of all  $M/2$  coefficients in the subband. Therefore, if  $\{b_1^*, b_2^*, \dots, b_L^*\}$  minimizes the total energy of the whole subband indexed by  $[1, M/2]$ , it should also minimize the subband energy within each subinterval of  $[1, M/2]$ . For example,  $\{b_1^*, b_2^*, \dots, b_k^*\}$ ,  $k < L$  must minimize the subband energy within interval  $[1, q_k]$ . Otherwise, we just replace  $\{b_1^*, b_2^*, \dots, b_k^*\}$  with the optimum one.

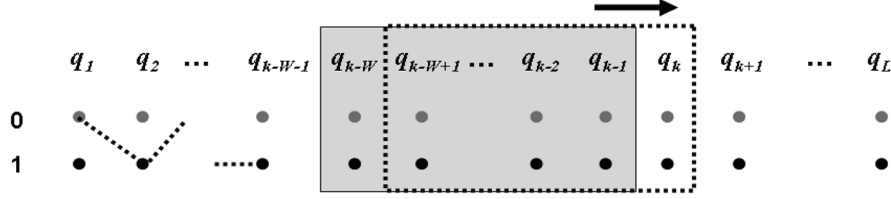


Fig. 10. Window-based dynamic programming for optimum PT.

We can see that these two important properties of optimum PT design allow us to solve this problem using dynamic programming, which will be explained in detail in the following section.

### C. Window-Based Dynamic Programming for Optimum Peak Transform

As in conventional dynamic programming, the problem-solving process is divided into  $L$  stages. Here,  $L$  is the total number of candidate peaks. At each stage, the decision parameter  $b_l$  has two states:  $b_l = 1$  which means  $q_l$  is chosen as a peak; and  $b_l = 0$  which means  $q_l$  is not chosen; In conventional dynamic programming, the decision at stage  $l$  only depends on the result of its previous stage. However, in optimum PT design, as discussed in the above section, the decision on a candidate peak  $q_l$  depends on several previous candidate peaks within the window  $[q_l - W, q_l]$ . Therefore, in this work, we propose a window-based dynamic programming method to solve the optimum PT design problem. It should be noted that the number of candidate peaks within the window  $[q_l - W, q_l]$  is not fixed since  $\{q_l\}$  might not be uniformly distributed: some regions might have more candidate peaks than others. This will introduce some additional complexity into dynamic programming. To simplify the problem, we assume that the decision on  $q_l$  only depends on its previous  $W$  candidate peaks  $\{q_{l-W}, \dots, q_{l-1}\}$ . In this way, the number of candidate peaks in the dependence window is fixed at  $W$ . The value of  $W$  should be chosen based on the length of the high-pass filter  $H_1(z)$ . For example, if we use the Daubechies (9, 7) filter bank, the high-pass filter has a length of 7. In this case, as we can see from the experimental results in Section VII,  $W = 5$  will be sufficient.

Let us introduce one more notation. Given a binary vector  $\{b_1, b_2, \dots, b_l\}$ , we choose peaks for PT from candidates  $\{q_1, q_2, \dots, q_l\}$  according to this binary vector. We then perform PTWT subband decomposition on the input signal  $f(x)$  within interval  $[0, q_l]$ . We denote the corresponding high-frequency subband energy by  $\mathcal{E}\{b_1, b_2, \dots, b_l\}[f(x)]$ .

Suppose that, in the current stage, the decision window is at candidate peaks  $\{q_{k-W}, \dots, q_{k-1}\}$ , as illustrated in Fig. 10. Each candidate peak has two possibilities: either being selected as a peak or not. Therefore, in total, within this window, there are  $2^W$  possibilities denoted by  $\{[b_{k-W}, \dots, b_{k-1}] | b_{k-i} \in \{0, 1\}, 1 \leq i \leq W\}$ . We index these  $2^W$  possibilities by integers

$$v = \mathcal{B}(b_{k-W}, \dots, b_{k-1}) \quad (19)$$

where the function  $\mathcal{B}(\dots)$  converts binary numbers into an integer with  $b_{k-W}$  and  $b_{k-1}$  as the least and most significant bits, respectively. Suppose that, for each possibility  $v$ , we have already known the optimum decisions on previous candidate peaks, denoted by  $\{b_1^v, b_2^v, \dots, b_{k-W-1}^v\}$  and the corresponding minimum high-frequency subband energy is denoted by

$$E_{[b_{k-W}, \dots, b_{k-1}]} = \mathcal{E}\{b_1^v, b_2^v, \dots, b_{k-W-1}^v, b_{k-W}, \dots, b_{k-1}\}[f(x)]. \quad (20)$$

In total, there are  $2^W$  energy values.

Suppose now we are shifting the decision window to the next candidate peak  $q_k$ , as illustrated in Fig. 10. For each possibility  $[b_{k-W+1}, \dots, b_k]$ , we need to find the optimum decisions on its previous candidate peaks  $\{q_1, \dots, q_{k-W}\}$ . Note that for each possibility in this new window, there are two cases:  $b_{k-W} = 0$  and  $b_{k-W} = 1$ . For the case of  $b_{k-W} = 0$ , from the previous stage, we already know the optimum decision on previous candidate peaks  $\{q_1, \dots, q_{k-W-1}\}$ , which is denoted by  $\{b_1^{v0}, b_2^{v0}, \dots, b_{k-W-1}^{v0}\}$ , where

$$v0 = \mathcal{B}(b_{k-W} = 0, b_{k-W-1}, \dots, b_{k-1}). \quad (21)$$

Similarly, for the case of  $b_{k-W} = 1$ , from the previous stage, we also already know the optimum decision on previous candidate peaks  $\{q_1, \dots, q_{k-W-1}\}$ , which is denoted by  $\{b_1^{v1}, b_2^{v1}, \dots, b_{k-W-1}^{v1}\}$ , where

$$v1 = \mathcal{B}(b_{k-W} = 1, b_{k-W-1}, \dots, b_{k-1}). \quad (22)$$

What we need to do is to compare these two cases and determine which one has a smaller high-frequency subband energy, either  $b_{k-W} = 0$  or  $b_{k-W} = 1$ . Let

$$E^0 = \mathcal{E}\{b_1^{v0}, b_2^{v0}, \dots, b_{k-W-1}^{v0}, b_{k-W} = 0, b_{k-W+1}, \dots, b_k\}[f(x)] \quad (23)$$

$$E^1 = \mathcal{E}\{b_1^{v1}, b_2^{v1}, \dots, b_{k-W-1}^{v1}, b_{k-W} = 1, b_{k-W+1}, \dots, b_k\}[f(x)] \quad (24)$$

If  $E^0 \leq E^1$ , we use  $[b_1^{v0}, b_2^{v0}, \dots, b_{k-W-1}^{v0}, b_{k-W} = 0]$  as the optimum decision on previous candidate peaks and  $E^0$  as the minimum energy. Otherwise, we use  $[b_1^{v1}, b_2^{v1}, \dots, b_{k-W-1}^{v1}, b_{k-W} = 1]$  and  $E^1$ , respectively.

In this way, at each stage of our dynamic programming, we always keep  $2^W$  possibilities within the decision window; and for each possibility, we always record the optimum decision on its previous peak candidates and the corresponding minimum energy values. Initially, the decision window starts from  $[q_1, \dots, q_W]$  with no previous peaks. We then shift the window stage by stage by performing the above updating procedure.

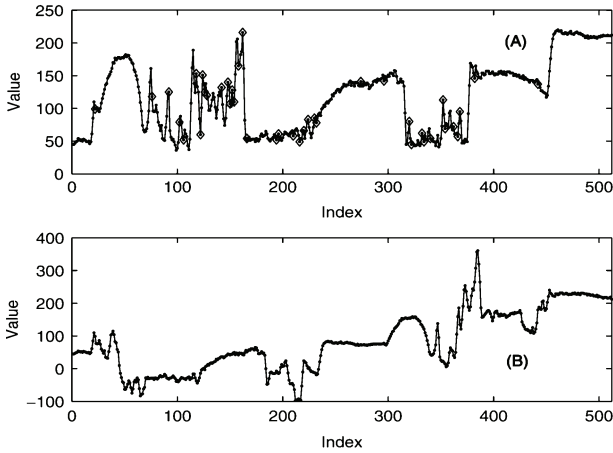


Fig. 11. (A) 370th row of Image Lena and the optimum peaks for PT; (B) PT output.

When it reaches to the last candidate peak  $q_L$ , we still have  $2^W$  possibilities. We simply find the possibility that has the minimum high-frequency subband energy as the optimum solution. The corresponding decision on candidate peaks forms the optimum peaks for PT.

#### D. Experimental Results on Optimum Peak Transform Design

We have implemented the window-based dynamic programming algorithm for optimum PT in MATLAB. The MATLAB codes and test data for optimum forward and backward transform can be downloaded from our web page [18]. Our extensive simulations show that this algorithm is robust and efficient.

In the following experiment, we use the 370th row of image *Lena* ( $512 \times 512$ ) as the test signal  $f(x)$ , as shown in Fig. 11(A). The optimum set of peaks determined by the window-based dynamic programming algorithm are also shown in Fig. 11(A) with diamonds. In total, there are 68 candidate peaks and 40 of them are selected as peaks for PT. Here, the peak threshold  $\delta$  is set to be 16 and the decision window size  $W$  is 5. Fig. 11(B) shows the PT output. Fig. 12 compares the high-frequency subbands with and without PT. We can see that the high-frequency subband with PT in Fig. 12(B) has a smaller energy than that in Fig. 12(A) without PT. To numerically evaluate this energy reduction performance, we define a PT gain as

$$G_{PT} = \frac{\mathcal{E}_{H1}[f(x)]}{\mathcal{E}_{H1}^{PT}[f(x)]} \quad (25)$$

where  $\mathcal{E}_{H1}^{PT}[f(x)]$  and  $\mathcal{E}_{H1}[f(x)]$  represent the high-frequency subband energy of  $f(x)$  with and without PT, respectively. In this experiment, the PT gain is  $G_{PT} = 1.64$ . In other words, the PT reduces the high-frequency subband energy by nearly 40%. Fig. 13(A) and (B) shows the low and high-frequency subbands using wavelet subband decomposition and PTWT subband decomposition, respectively. It can be seen that, as in conventional subband decomposition, the PTWT low-frequency subband is also similar to the input signal.

In Fig. 14(A), we plot the PT gain for each row of image *Barbara*. Fig. 14(B) shows the corresponding number of peaks selected by the dynamic programming algorithm for PT. It can be seen that, for most image rows, the PT gain is larger than 1.6.

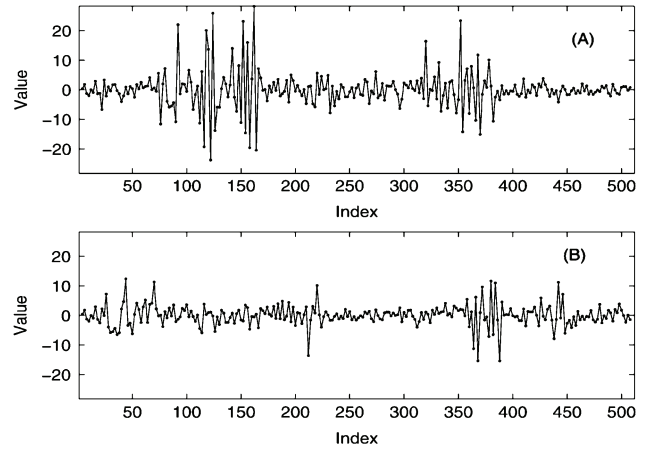


Fig. 12. (A) High-frequency subband without PT; (B) high-frequency subband with PT.

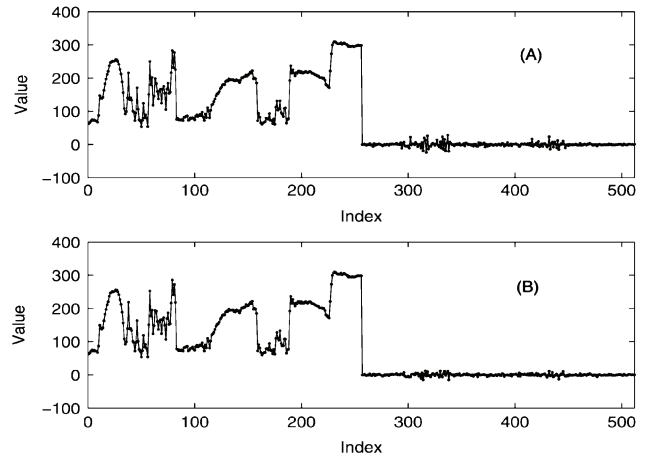


Fig. 13. (A) Low and high-frequency subbands without PT; (B) low and high-frequency subbands with PT.

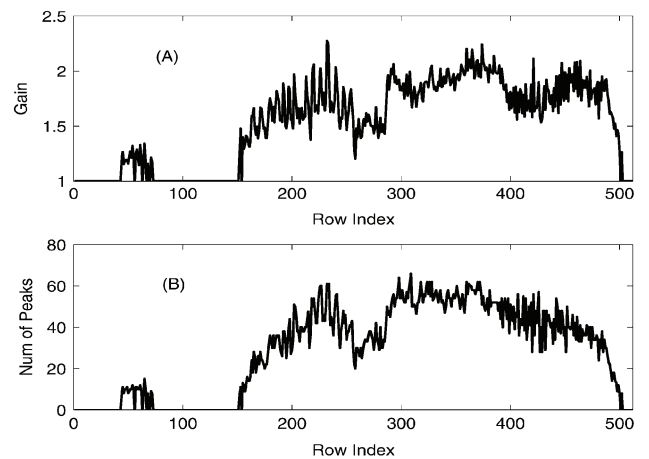


Fig. 14. (A) PT gain for each row of image *Barbara*; (B) corresponding number of peaks used for PT.

For some image rows, the transform gain is even larger than 2. This implies that the PT reduces the high-frequency subband energy by about 40%–55% (note that, for some image rows, the PT gain is 1 and the number of peaks is zeros; this is because these image rows are already very smooth and there is no need to



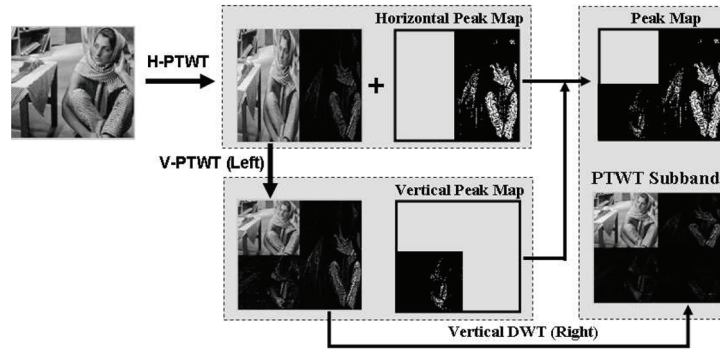


Fig. 15. One level of PTWT subband decomposition of images.

apply any PT). Our simulations over other test signals yield similar results, especially for input signals with a significant amount of high-frequency components.

## V. PEAK TRANSFORM-BASED IMAGE ENCODER DESIGN

In this section, we will study how to design an efficient image compression system based on PT. First, we will discuss how to apply the PTWT subband decomposition to 2-D images. Second, we will study how the overhead information, i.e., the locations of peaks, could be efficiently encoded.

### A. Subband Decomposition of Images Using Peak Transform

In this section, we extend the 1-D PTWT subband decomposition presented in Section III to 2-D PTWT subband decomposition of images. Fig. 15 shows one-level PTWT subband decomposition of image *Barbara*. We first apply the 1-D PTWT subband decomposition described in Fig. 8 to every image row. This will create two image subbands: low and high frequency subbands, denoted by  $\mathbf{L}$  and  $\mathbf{H}$ , respectively. In addition, for each row, we have a set of peaks  $\{x_n\}$  used in PT. Note that  $x_n = 2 \times y_n$  is an even integer. For each row, we use a binary *peak map* to indicate the locations of peaks. Fig. 15 (top-middle) shows the horizontal peak map for all image rows.

Now, we are ready to explain the need of inverse PT in PTWT subband decomposition as mentioned in Section III. Note that during horizontal PTWT subband decomposition, different image rows use different peaks for PT. Therefore, the image correlation in the vertical direction might be destroyed. For example, Fig. 16 shows one-level PTWT subband decomposition of image *Barbara* if the inverse PT is not performed. We can see that the low-frequency subband does not like the original image at all and the vertical image correlation is destroyed after horizontal PTWT subband decomposition. Therefore, by performing the inverse PT, we are able to bring transform coefficients back to their original image locations so as to preserve the source correlation and prepare for the next stage of subband decomposition along the vertical direction.

In the second step, we apply the 1-D PTWT subband decomposition to each column of the  $\mathbf{L}$  subband. This will decompose the  $\mathbf{L}$  subband into two subbands, denoted by  $\mathbf{LL}$  and  $\mathbf{LH}$ , respectively. As in the first step, a peak map, called *vertical peak map*, for this  $\mathbf{L}$  subband image is also generated, as shown in

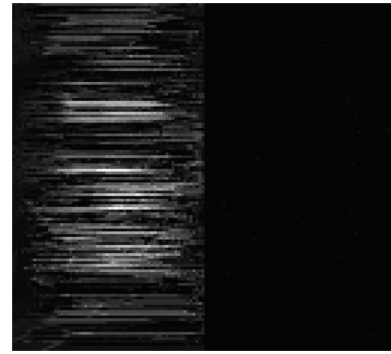


Fig. 16. One level of PTWT subband decomposition of image *Barbara* without inverse PT.

Fig. 15 (bottom-middle). In the third step, we apply a vertical DWT (without PT) to further decorrelate the  $\mathbf{H}$  subband, as shown in Fig. 15 (bottom-right). Note that, here, we use DWT instead of PTWT. This is because the performance gain of PT on the  $\mathbf{H}$  subband is very limited. The horizontal and vertical peak maps are combined into one single peak map, as shown in Fig. 15 (top-right). This concludes one-level of 2-D PTWT subband decomposition.

Certainly, this 2-D PTWT subband decomposition can be repeated for the  $\mathbf{LL}$  subband so as to obtain a dyadic subband decomposition of the image. Fig. 17(B) shows a three-level PTWT subband decomposition of image *Barbara*. For comparison, we also show the three-level subband decomposition using DWT only in Fig. 17(A). It can be seen that the PT significantly reduces the energy of high-frequency subbands. More specifically, the transform coefficients in Fig. 17(B) have much smaller magnitudes than those in Fig. 17(A). The corresponding peak map is shown in Fig. 17(C). In the following section, we will discuss how this binary peak map can be efficiently encoded.

### B. Coding of Peak Map

In PTWT-based image compression, we need to encode the subband data, as illustrated in Fig. 17(B), as well as the peak map, and send them to the decoder for image reconstruction. A number of algorithms have been developed in the literature, such as EZW (embedded zero-tree wavelet) [17], SPIHT (set partitioning in hierarchical trees) [6], JPEG2000 [1], to encode image subband data. In this work, we use the encoder developed

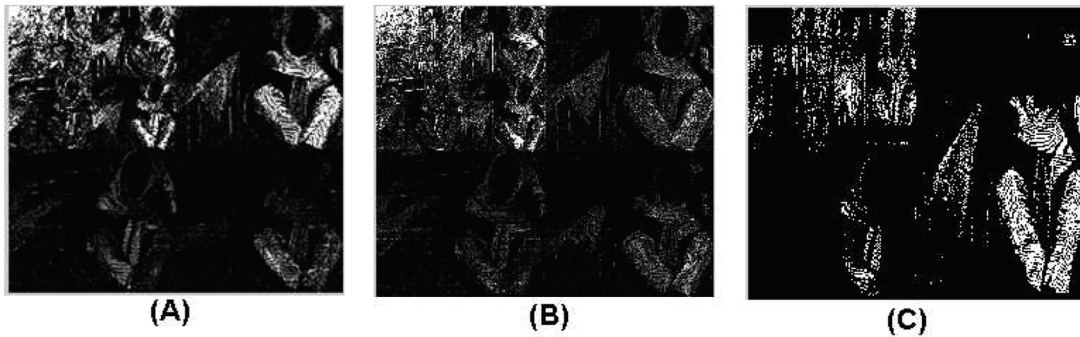


Fig. 17. (A) Three-level subband decomposition of *Barbara*with WT; (B) three-level PTWT subband decomposition; (C) peak map.

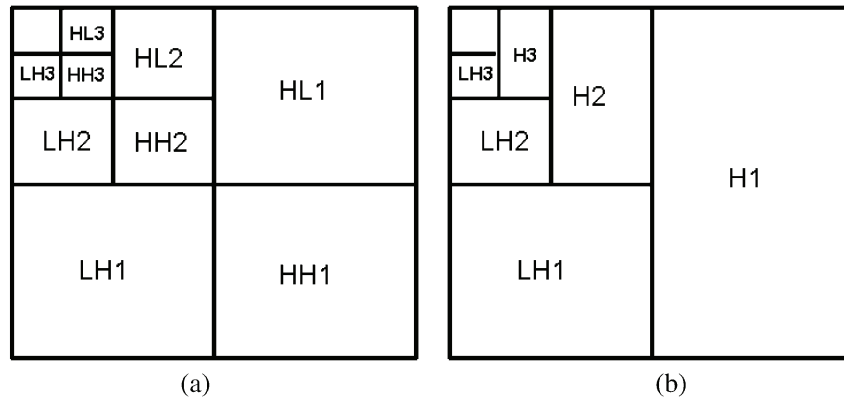


Fig. 18. (a) Labeling of image subbands; (b) subband labeling for peak map.

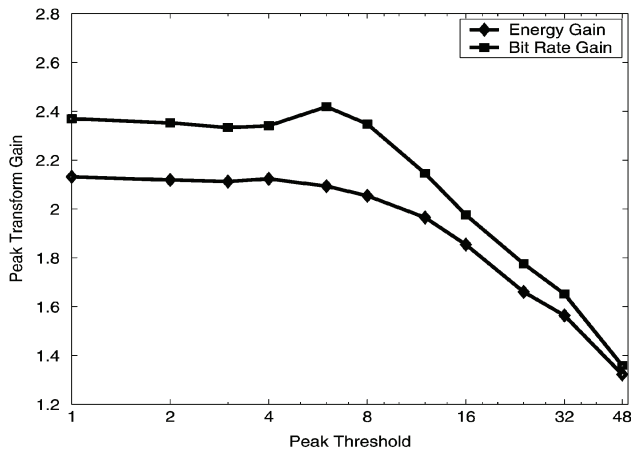


Fig. 19. PT gain versus peak threshold.

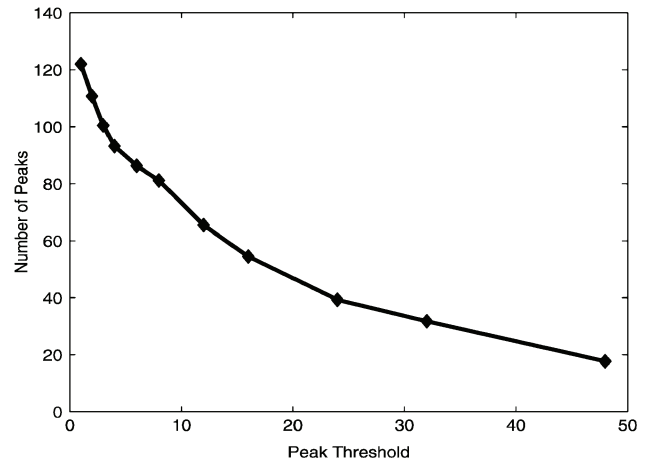


Fig. 20. Number of peaks versus peak threshold.

in our previous work to encode the subband data [19]. This encoder uses a similar encoding mechanism as SPIHT with a similar performance. However, its computational complexity and implementation cost are much lower [19].

1) *Overview of Peak Map Coding:* The binary peak map coding has to be lossless. In this work, we observe that the peak map, as illustrated in Fig. 17(C), bears a significant amount of correlation with the subband data. This is because, as discussed in Section IV-B1, the peaks are chosen from those locations with large high-frequency energy levels. In this work, we propose to exploit this correlation so as to reduce the coding bit rate of peak map. Note that the subband data is encoded and is

available at the decoder. Therefore, we can use the subband data to predict the peak map. More specifically, at the encoder side, the image is decomposed level by level (from level 1 to level  $\Lambda$ ) into a series of frequency subbands, as illustrated in Fig. 18. Correspondingly, at the decoder side, the image should be reconstructed level by level in a reversed order (from level  $\Lambda$  to level 1). While we are reconstructing an image subband at level  $\lambda, 1 \leq \lambda \leq \Lambda - 1$ , image subband data at levels  $\lambda + 1, \dots, \Lambda$  are already available. This implies that we can use the reconstructed image subband data in higher levels to predict the peak map. Conceptually, this approach is similar to motion prediction in video coding [23].



Fig. 21. Test images: (a) Lena; (b) Barbara; (c) NBA2; (d) Football2.

To elaborate, the peak map is a binary image with 1s indicating the positions of peaks. We will use a context adaptive arithmetic coder [20] to encode this binary peak map. As we know, the key in context adaptive arithmetic coding is to estimate the probability of symbols. In the following, we will develop a scheme to predict the probability of 1s within each region of the peak map using image subband data. This probability information will be then used by the context adaptive arithmetic coder. Two major issues will be discussed: a) linear prediction of peak map and b) a similarity metric for peak map prediction.

2) *Linear Prediction*: Our central idea in prediction is to use a linear combination of the image subbands at higher levels, which have already been reconstructed, to predict the symbol (either 0 or 1) probability at each pixel location of the peak map at the current level. In the following, we use an example to explain our idea. Fig. 18(a) shows the image subbands and their labels. Fig. 18(b) shows the subband<sup>2</sup> labeling of peak map. In this example, we encode the H1 subband of peak map which is represented by a binary matrix  $\mathbf{B} = [b_{uv}]$  of size  $U \times V$ . We can use the LH1, LH2, HH2, and HL2 image subbands for prediction. To do this, we up-sample these subbands with bilinear interpolation to the size of  $\mathbf{B}$  and denote these up-sampled image subbands by  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_J$ . Here,  $\mathbf{X}_j, 1 \leq j \leq J$ , is a matrix of transform coefficients of size  $U \times V$ . A typical value of  $J$  ranges from 2 to 5. Let

$$\mathbf{Y} = [y_{uv}]_{U \times V} = \lambda_1 \mathbf{X}_1 + \lambda_2 \mathbf{X}_2 + \dots + \lambda_J \mathbf{X}_J, \quad (26)$$

which is a linear combination of these image subbands. Here, the parameters  $\{\lambda_j\}$  need to be determined. We define a reference binary image  $\hat{\mathbf{B}} = [\hat{b}_{uv}]_{U \times V}$  as

$$\hat{b}_{uv} = \begin{cases} 1, & \text{if } |y_{uv}| > 1 \\ 0, & \text{otherwise.} \end{cases} \quad (27)$$

We denote this thresholding operation by  $\mathcal{N}(\cdot)$  and write

$$\hat{\mathbf{B}} = \mathcal{N}(\lambda_1 \mathbf{X}_1 + \lambda_2 \mathbf{X}_2 + \dots + \lambda_J \mathbf{X}_J). \quad (28)$$

During prediction, we need to select the parameters  $\{\lambda_j\}$  such that the similarity between the reference binary image  $\hat{\mathbf{B}}$  and peak map  $\mathbf{B}$  is maximized or the dissimilarity between them is minimized. To this end, we need to introduce a metric to describe the similarity between two binary images, which will be explained the following section.

3) *Similarity Metric for Peak Map Prediction*: Note that the purpose of our prediction is to accurately estimate the probability of 1s in  $\mathbf{B}$  such that its arithmetic coding bit rate is mini-

mized. Therefore, in this work, we propose to use the probability of 1s at each location to measure the similarity between binary images  $\mathbf{B}$  and  $\hat{\mathbf{B}}$ . More specifically, let  $\Omega_{uv}(\mathbf{B})$  be the fraction of 1s within a squared window with size  $\Delta$  centered at location  $[u, v]$  in binary image  $\mathbf{B}$ . Likewise, we can also define  $\Omega_{uv}(\hat{\mathbf{B}})$  for the reference binary image  $\hat{\mathbf{B}}$ . Here,  $\Omega_{uv}(\mathbf{B})$  and  $\Omega_{uv}(\hat{\mathbf{B}})$  approximate the probabilities of 1s at location  $(u, v)$  in images  $\mathbf{B}$  and  $\hat{\mathbf{B}}$ , respectively. We define the dissimilarity between  $\mathbf{B}$  and  $\hat{\mathbf{B}}$  as

$$\Omega(\mathbf{B} - \hat{\mathbf{B}}) = \frac{1}{UV} \sum_{u=1}^U \sum_{v=1}^V |\Omega_{uv}(\mathbf{B}) - \Omega_{uv}(\hat{\mathbf{B}})|. \quad (29)$$

It can be seen that  $\Omega(\mathbf{B} - \hat{\mathbf{B}})$  represents the average difference in probability of 1s between  $\mathbf{B}$  and  $\hat{\mathbf{B}}$ . If  $\hat{\mathbf{B}}$  is the same or very close to  $\mathbf{B}$ , the value of  $\Omega(\mathbf{B} - \hat{\mathbf{B}})$  is 0 or very small. With this metric, the prediction problem can be formulated as

$$\min_{\{\lambda_1, \lambda_2, \dots, \lambda_J\}} \Omega(\mathbf{B} - \hat{\mathbf{B}}) = \Omega(\mathbf{B} - \mathcal{N}(\lambda_1 \mathbf{X}_1 + \lambda_2 \mathbf{X}_2 + \dots + \lambda_J \mathbf{X}_J)). \quad (30)$$

In this work, we just use brute-force search to find the optimum solution, denoted by  $\{\lambda_1^*, \lambda_2^*, \dots, \lambda_J^*\}$  at the encoder side since the value of  $J$  is small (we use  $J = 4$  in our experiments). Certainly, fast search or optimization algorithms can be designed in the future to speed up the prediction process.

Once  $\{\lambda_1^*, \lambda_2^*, \dots, \lambda_J^*\}$  are determined, the optimum reference image is given by

$$\hat{\mathbf{B}}^* = \mathcal{N}(\lambda_1^* \mathbf{X}_1 + \lambda_2^* \mathbf{X}_2 + \dots + \lambda_J^* \mathbf{X}_J) \quad (31)$$

which will be used as context during the context-adaptive encoding of of peak map  $\mathbf{B}$  [20]. The values of  $\{\lambda_1^*, \lambda_2^*, \dots, \lambda_J^*\}$  will be encoded and sent to the decoder. At the decoder side, after the image subbands are decoded, we can also construct the same reference image  $\hat{\mathbf{B}}^*$  for arithmetic decoding of the peak map.

## VI. TRADEOFF BETWEEN PEAK TRANSFORM GAIN AND OVERHEAD INFORMATION BITS

As discussed in Section IV-B1, we choose those pixels with high-frequency energy levels larger than a threshold  $\delta$  as candidate peaks. We then use the dynamic programming approach presented in Section IV-A to determine which subset of candidate peaks minimize the high-frequency subband energy or maximize the PT gain. If we increase the value of threshold  $\delta$ , the total number of candidate peaks will decrease. Therefore, the number of overhead information bits used to encode the peak

<sup>2</sup>For convenience, we call a block region of the peak map as a subband.

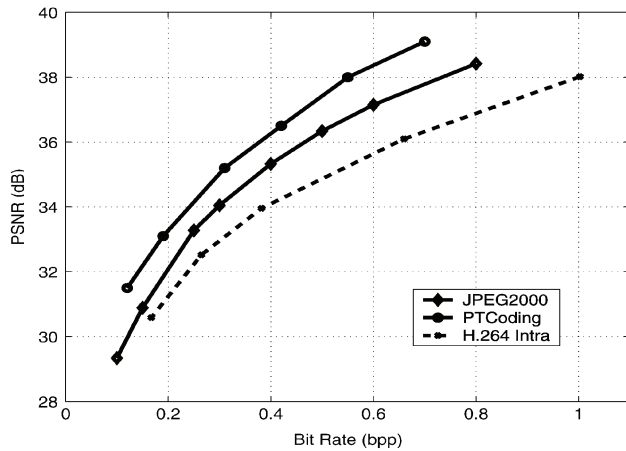


Fig. 22. Compression performance comparison with JPEG2000 and H.264 (INTRA) on image *Lena*.

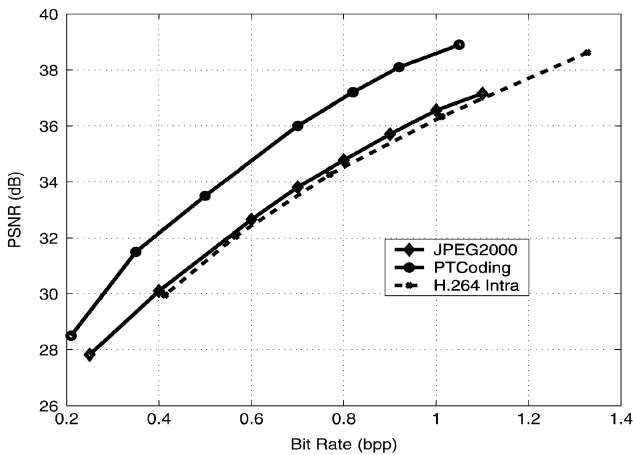


Fig. 23. Compression performance comparison with JPEG2000 and H.264 (INTRA) on image *Barbara*.

map will be reduced. However, the energy reduction by PT will be less, which results in more coding bits for the subband data. This leads to a tradeoff between PT gain and overhead information bits; and implies that we need to determine the optimum value of  $\delta$  to maximize the overall coding efficiency.

To determine the optimum value of  $\delta$ , we need an accurate rate-distortion model to characterize the rate-distortion behavior of the PT encoder [21], [22]. This is a challenging task and needs a significant amount of research efforts. In this work, instead of pursuing an optimum solution, we propose an empirical approach which is simple yet efficient. According to our simulation experience, we observe that the optimum threshold  $\delta$  is related to the quantization step size of transform coefficients. More specifically, the following empirical formula is used in this work to determine the value of peak threshold:

$$\delta = \frac{1}{2}QP \quad (32)$$

where  $QP$  is the quantization step size of transform coefficients. To justify this empirical formula, in the following, we take image *Barbara* as an example. In this experiment, we set the quantization step size to be  $QP = 16$  and try different peaks thresholds  $\delta$ . Fig. 19 shows the PT gain defined in (25) for each  $\delta$  in a solid line with diamonds. Fig. 19 also shows

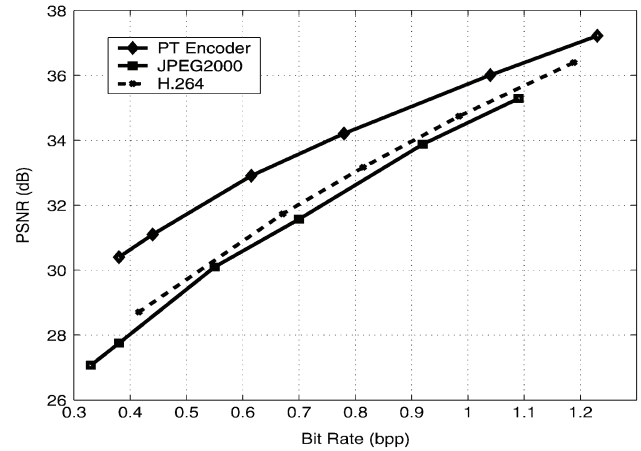


Fig. 24. Compression performance comparison with JPEG2000 and H.264 (INTRA) on image *Football*.

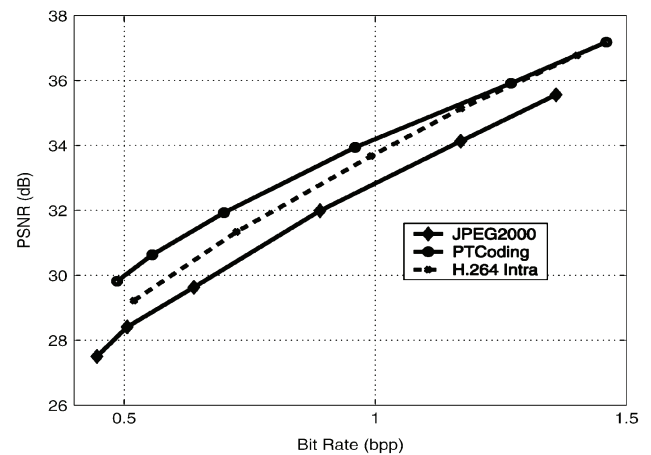


Fig. 25. Compression performance comparison with JPEG2000 and H.264 (INTRA) on image *NBA*.

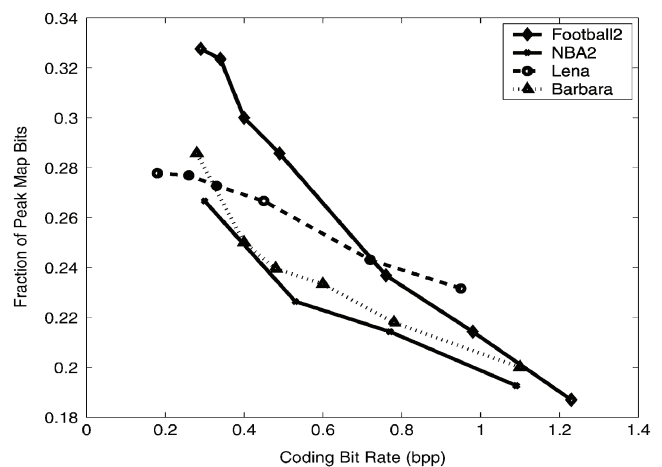


Fig. 26. Fraction of peak map bits versus coding bit rate for each test image.

the bit rate gain for each  $\delta$  in a solid line with squares. Here, the bit rate gain is the ratio between the coding bit rates of image encoders with and without PT. It can be seen that, as  $\delta$  increases, the PT gain decreases. Fig. 20 shows the average number of peaks per image row for each peak threshold  $\delta$ . As

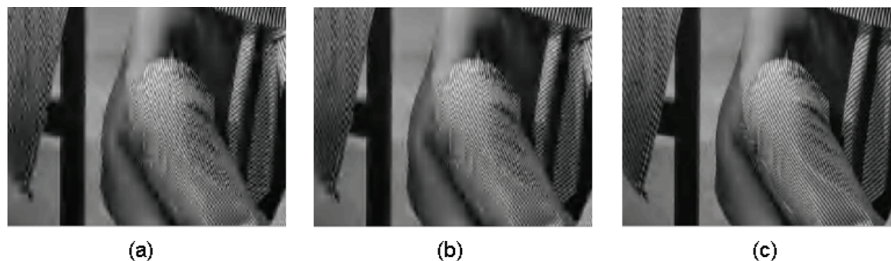


Fig. 27. Subjective picture quality comparison on image *Barbara* coded at 0.4bpp: (a) JPEG2000; (b) H.264 INTRA; (c) PT encoding.

$\delta$  increases, more peaks are used for PT, which implies that more bits will be used to encode the peak map. We can see that  $1/2QP = 8$  is a reasonable choice for the peak threshold  $\delta$ .

This empirical formula is simple. Our experimental results presented in the following section will show that it is also efficient. In our future work, we shall study the rate-distortion modeling problem for PT encoding and develop a systematic approach to find the optimum peak threshold.

## VII. EXPERIMENTAL RESULTS

In this section, we evaluate the data compression performance of the proposed PT image encoder and compare its performance with state-of-the-art image encoders, include JPEG2000 and H.264 INTRA coding. The JPEG2000 image encoder we used in this performance evaluation is the Jasper JPEG2000 encoder [24]. In H.264 encoding, we use JM 9.0 H.264 video encoder (INTRA frames only) with CABAC (context adaptive arithmetic coding) [25]. The test images (grayscale) of size  $512 \times 512$  are shown in Fig. 21. To deal with grayscale images in H.264, we set the chrominance components (Cb and Cr) of each frame to be a constant 128. Figs. 22—25 show the PSNR (peak signal-to-noise ratio) performance of the PT encoder in comparison with JPEG2000 and H.264 image coding over all test images. It can be seen that the PT encoder consistently outperforms the other two image encoders on all test images. For image *Barbara*, the PT encoder outperforms JPEG2000 by up to 2.3 dB. Fig. 26 shows the fraction of bits used for encoding peak map as a function of coding bit rate for each test image. We can see that, as the overall coding bit rate increases, the fraction of peak map bits decreases (however, it should be noted that the number of peak map bits does increase). Fig. 27 shows the images of *Barbara* encoded at 0.4 bpp by JPEG2000, H.264INTRA, and the proposed PT encoder. It can be seen that, with PT encoding, the image has a much better visual quality with enhanced edge information. This is because the PT encoder is able to efficiently preserve high-frequency image features using PT and peak map.

## VIII. CONCLUDING REMARKS, DISCUSSION, AND FURTHER RESEARCH DIRECTIONS

The major contribution of this work is that we have introduced a nonlinear geometric transform, called PT, which is able to convert high-frequency signals into low-frequency ones. Coupled with WT and subband decomposition, the PT is able to significantly reduce signal energy in high-frequency subbands. This has significant applications in data compression of 1-D signals (e.g., speech and acoustic signals) and 2-D images. We have developed an fast and efficient dynamic solution to find optimum (or suboptimum) PT to minimize the high-frequency

subband energy or maximize the transform coding gain. We have also studied how to design an image compression system based on PT. Our experimental results show that the proposed PT encoder outperforms the state-of-the-art image encoders, including JPEG2000 and H.264 (INTRA).

In our future work, the following issues can be further investigated. First, as discussed in Section VI, the rate-distortion behaviors of different modules of the PT encoder need to be analyzed and modeled so that we can find an optimum tradeoff between them to maximize the overall coding efficiency. For example, the peak selection during dynamic programming can be coupled with peak map encoding. Some peaks can be dropped if they do not contribute much to the transform coding gain; however, they cost a lot of bits to encode. Second, in Section V-B, we have developed a simple yet efficient way to encode the binary image of peak map. However, this is not the final solution. How to efficiently encode a binary image using prediction from reference images is still an open issue. Third, our current PT design and PT encoding are still computationally intensive. In the future, we will study how to optimize the algorithm and speed up the encoding process.

In this work, we studied one specific application of PT on lossy image compression. It can be also applied to other scenarios of data compression, such as lossless image coding and acoustic data compression. Since the PT has changed our way to filter and process signals, we can also find its applications in other signal processing tasks, such as image denoising and enhancement, depending on how the peaks are selected and the PT is performed.

From an even broader perspective, the concept proposed in this work—designing a nonlinear transform to convert hard-to-compress signals into easy ones—is very useful. The encouraging results in this work motivate us to find more generic nonlinear transforms or geometric transforms to explore the inherent correlation among the source data so as to represent and code the data more efficiently. We hope this paper could inspire new approaches and more research work along this direction.

## REFERENCES

- [1] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer, 2002.
- [2] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 4, pp. 205–20, Apr. 1992.
- [4] W. Ding, F. Wu, and S. Li, "Lifting-based wavelet transform with directionally spatial prediction," presented at the Picture Coding Symp., San Francisco, CA, Dec. 2004.

- [5] C.-L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform via directional lifting and bandeletization," presented at the IEEE Int. Conf. Image Processing, Atlanta, GA, Oct. 2006.
- [6] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on Set Partitioning in Hierarchical Trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–50, Jun. 1996.
- [7] E. L. Pennec and S. Mallat, "Bandelet representations for image compression," in *Proc. Int. Conf. Image Processing*, Oct. 2001, vol. 1, pp. 12–15.
- [8] D. Donoho, "Wedgelets: Nearly-minimax estimation of edges," *Ann. Statist.*, vol. 27, pp. 353–382, 1999.
- [9] M. N. Do and M. Vetterli, J. Stoeckler and G. V. Welland, Eds., *Contourlets, in Beyond Wavelets*. New York: Academic, 2003.
- [10] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Directionlets: Anisotropic multidirectional representation with separable filtering," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1916–1933, Jul. 2006.
- [11] D. Taubman and A. Zakhor, "Orientation adaptive subband coding of images," *IEEE Trans. Image Process.*, vol. 3, no. 4, pp. 421–437, Jul. 1994.
- [12] D. Wang, L. Zhang, A. Vincent, and F. Speranza, "Curved wavelet transform for image coding," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2413–2421, Aug. 2006.
- [13] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–516, Jul. 2003.
- [14] P. Vanderghynst and J.-F. Gobbers, "Directional dyadic wavelet transforms: Design and algorithms," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 363–72, Apr. 2002.
- [15] J.-L. Starck, E. J. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–84, Jun. 2002.
- [16] M. N. Do and M. Vetterli, "The finite ridgelet transform for image representation," *IEEE Trans. Image Process.*, vol. 12, no. 1, pp. 16–8, Jan. 2003.
- [17] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–462, Dec. 1993.
- [18] Peak Transform Web Page [Online]. Available: <http://videonet.ece.missouri.edu/research/peaktransform.htm>
- [19] Z. He, T.-H. Yu, and S. K. Mitra, "Blockwise zero mapping image coding," in *Proc. Inf. Conf. Image Processing*, Sep. 2000, vol. 3, pp. 170–173.
- [20] P. G. Howard and J. S. Vitter, "Arithmetic coding for data compression," *Proc. IEEE*, vol. 82, no. 6, pp. 857–865, Jun. 1994.
- [21] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.
- [22] Z. He and S. K. Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1221–1236, Dec. 2001.
- [23] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*. New York: Wiley, Aug. 2003.
- [24] Jasper JPEG2000 Encoder [Online]. Available: <http://www.ece.uvic.ca/mdadams/jasper/>
- [25] H.264 JM Codec [Online]. Available: <http://iphome.hhi.de/suehring/tml/>



**Zhihai He** received the B.S. degree in mathematics from Beijing Normal University, Beijing, China, the M.S. degree in mathematics from the Institute of Computational Mathematics, Chinese Academy of Sciences, Beijing, in 1994 and 1997, respectively, and the Ph.D. degree in electrical engineering from the University of California, Santa Barbara, in 2001.

In 2001, he joined Sarnoff Corporation, Princeton, NJ, as a Member of Technical Staff. In 2003, he joined the Department of Electrical and Computer Engineering, University of Missouri, Columbia, as an Assistant Professor. His current research interests include image/video processing and compression, network transmission, wireless communication, computer vision analysis, sensor network, and embedded system design.

Dr. He received the 2002 IEEE Transactions on Circuits and Systems for Video Technology Best Paper Award, and the SPIE VCIP Young Investigator Award in 2004. He is a member of the Visual Signal Processing and Communication Technical Committee of the IEEE Circuits and Systems Society, and serves as Co-Chair, Technical Program Committee member, and Session Chair for a number of international conferences.