Improvement of JPEG2000 Using Curved Wavelet Transform

Demin Wang, Liang Zhang, and André Vincent

Communications Research Centre Canada,
3701 Carling Avenue, Ottawa, Ontario, Canada K2H 8S2
E-mail: demin.wang@crc.ca

ABSTRACT

The wavelet transform in JPEG2000 is performed using one-dimensional (1-D) filtering in the vertical and horizontal directions. This conventional wavelet transform is not effective to represent edges and lines in images. In this paper we present a *curved wavelet transform* that improves the performance of JPEG200. The curved wavelet transform is performed using 1-D filtering along curves that are usually parallel to edges and lines in images. The pixels along these curves can be well represented by a small number of wavelet coefficients. A simple algorithm is proposed in this paper to determine the curves according to image content. Experimental results show that the curved wavelet transform can significantly improve the compression efficiency of JPEG2000, especially for images that contain sharp edges and lines. The coding gain can be up to 1.6 dB in the terms of PSNR.

1. INTRODUCTION

JPEG2000 image coding standard is based on two-dimensional wavelet transform (2-D WT) and embedded block coding with optimal truncation (EBCOT) [1]. This standard provides useful features and functionalities with compression efficiency comparable to the most efficient DCT-based image coders. The 2-D WT used in JPEG2000 is performed through 1-D filtering along each row, and then along each column in the image to be compressed. The main shortcoming of this conventional WT is that it does not provide a compact representation for edges and lines in the image. When the image is filtered in the horizontal or vertical direction, the filter often crosses edges and lines of the image. The sequence of pixels across an edge usually has a broad frequency spectrum, from low to high frequencies. The wavelet transform decomposes the energy of the pixel sequence into a large number of frequency bands. Many wavelet coefficients are therefore required to properly reconstruct the edge. As a result, JPEG2000 produces "ringing" artifacts around edges, especially at low bit rates.

This paper describes an improvement of JPEG2000 using a transform called *curved wavelet transform* (*curved WT* in short) [2]. The 2-D curved WT is performed through 1-D filtering along curves, rather than being restricted to horizontal and vertical straight lines. The curves are determined from the content of the image to be compressed. The basic idea behind the curved WT is that, if a curve is parallel to an edge and a line in the image, the sequence of pixels along the curve consists mainly of low frequency components with no or little high frequency components and can be well represented by a small number of wavelet coefficients. Therefore, the curved WT can provide a more compact representation for edges and lines than the conventional WT and improve the compression efficiency of JPEG2000. In [2], the curved WT is implemented using convolution wavelet filters and overlapped extension, whereas in this paper it is implemented with the lifting filter structure.

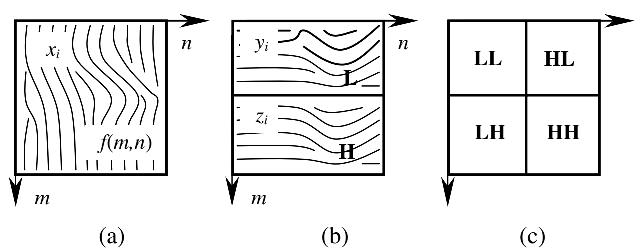


Fig.1. Curved WT. (a) Vertical curves, (b) horizontal curves, (c) four bands after the first level transform.

2. CURVED WAVELET TRANSFORM

The concept of the curved WT is illustrated in Fig. 1. The first step of the curved WT is to determine a set of vertical curves x_i, as shown in Fig. 1 (a). Low-pass and high-pass wavelet filters are applied separately along the curves, and the outputs of these filters are sub-sampled by discarding every other row, resulting in low-pass coefficients L(m, n) and high-pass coefficients H(m, n). The filters can be any pair of 1-D wavelet filters. A set of horizontal curves y_i is then determined for the resulting coefficients L(m, n), as shown in Fig. 1 (b). The low-pass and high-pass filters are applied along these curves, and the outputs are sub-sampled by discarding every other column, producing two bands of coefficients LL and HL. In a similar way, a third set of curves z_i is determined for coefficients H(m, n) and is used to decompose the H coefficients into the LH and the HH coefficients. The process described above achieves the first level transform of the curved WT. This process is repeatedly applied to the resulting LL bands to obtain the second and higher level transforms.

In order to obtain a dyadic decomposition like that produced by the conventional WT, there must be some constraints on the curves. In the following subsections, we describe these constraints, the algorithms used to determine the curves, and the implementation of curved WT via the lifting structure.

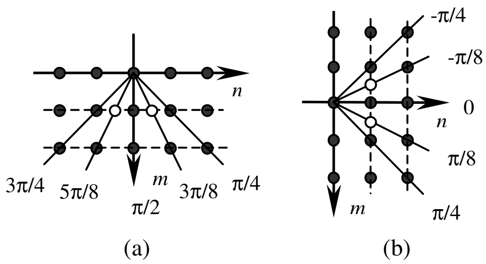


Fig. 2. Allowed orientations of the line segments constituting the curves, (a) for vertical curves, (b) for horizontal curves. The solid circles denote full-pixels and the empty ones denote half-pixels.

Fig.3. Some of the vertical curves determined for image *Barbara.*

2.1. Constraints on the Curves

In order that the curved WT provides a dyadic decomposition, a vertical curve is defined by a continuous single-valued function of the vertical coordinate m. This means that a vertical curve does not contain any horizontal straight-line segment. The pixels along a vertical curve come from a set of successive rows of the image to be transformed, with one and only one pixel per row. A curve may connect with others and may pass between two adjacent pixels. Each pixel in the image to be filtered has to associates with at least one curve. Similarly, a horizontal curve is defined by a continuous single-valued function of the horizontal coordinate n.

To reduce the computational cost and the coding cost for the curves, the following additional constraints are imposed on the curves in our implementation: (i) All curves consist of line segments with a few discrete orientations. For vertical curves, the allowed orientations are π/4, 3π/8, π/2, 5π/8, and 3π/4. For horizontal curves, the allowed orientation are -π/4, -π/8, 0, π/8, and π/4 shown in Fig. 2. (ii) For each set of curves, the image (or wavelet coefficients) to be filtered is divided into blocks of $K\times L$ pixels. All curves within a block are straight-line segments of the same orientation. (iii) The third set of curves z_i is identical with the second set of curves y_i.

2.2. Determination of the Curves

Ideally, for a given image and target bit rate, the best curves could be determined through rate-distortion optimization. This would require a large amount of calculations because any change affecting a curve also impacts other curves at the same and higher levels of transform.

To reduce the computational complexity, we propose a simple algorithm to determine the curves. This algorithm searches for the curves that minimize the energy of high-pass wavelet coefficients within each block of $K\times L$ pixels. After the curved WT with the curves determined in this manner, the energy of the image is concentrated in the low-pass coefficients at the highest level, and the high-pass coefficients at all levels contain a small amount of energy. A low energy of high-pass coefficients means that the high-pass coefficients require a small number of bits to code. Since the number of the high-pass coefficients is much larger than that of the low-pass coefficients, reducing the bits required by the high-pass coefficients can effectively reduce the total number of bits required for the image. The proposed algorithm for determining each set of the curves consists of the following steps:

1) The high-pass filter is applied to the image (or wavelet coefficients) along the straight lines of each allowed orientation.
2) The filtered image is divided into blocks of $K\times L$ pixels.
3) In each block, the energy of the resulting high-pass coefficients is calculated for each of the allowed orientations.
4) The orientation that results in the lowest energy is chosen, and the line segments of this orientation within the block are kept.
5) The kept line segments are connected with those of the adjacent blocks to form a set of curves.

Fig. 3 illustrates some of the vertical curves that are determined using the proposed algorithm for the image *Barbara*. The curves are generally parallel to the edges and lines of orientations between π/4 and 3π/4 in the image.

2.3. Implementation with lifting structure

It is known that wavelet filters can be implemented with a lifting structure. For the 5/3 and 9/7 wavelet filters, each lifting step involves three pixels. When one of the wavelet filters is applied along a column in an image, a pixel is predicted or updated using the two adjacent pixels on the column at each lifting step [1].

The curved WT can be easily implemented using the lifting structure. For simplicity, we only describe the implementation with vertical curves in the following. Suppose that a pixel is on a vertical curve. At each lifting step, this pixel is predicted or updated with its two adjacent pixels that are on the same curve. One of the adjacent pixels is on the row above the pixel and the other on the row below the pixel. For the example shown in Fig. 4 (a), pixels *a*, *b*, and *c* are on the same vertical curve $x(m)$. Pixel *a* is predicted or updated by $a + \lambda(b + c)$, where λ is the lifting coefficient.

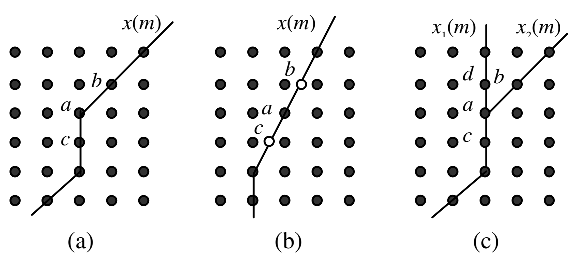


Fig. 4. Three situations in the lifting implementation of curved WT. (a) the normal situation, (b) requiring half-pixel interpolation, (c) intersection of two or more curves.

There are two special situations to handle in the implementation. The first one involves half-pixel interpolation. As shown in Fig. 2, the line segments of orientations -π/8, π/8, 3π/8, and 5π/8 pass through between two pixels. Half-pixel interpolation is required for these line segments. For example, if pixel *a* is on a line segment (part of a curve) of orientation 5π/8, as shown in Fig. 4 (b), *b* and *c* are between two horizontally adjacent pixels. In this case, the values of the half-pixels *b* and *c* are interpolated and used to predict or update the value of pixel *a*.

The second special situation concerns the intersections of two or more curves. If two or more curves pass through a pixel, as pixel *a* in Fig. 4 (c), this pixel has more than one adjacent pixel on the row above and/or below. In this case, one adjacent pixel has to be selected from those on the row above and one from those on the row below. The selection in our implementation is based on the position of these adjacent pixels. The order of the selection is the pixel on the same column first if it is on one of the curves, second the pixel on the left column, third that on the right column, fourth the half-pixel on left side, and at last the half-pixel on the right side. For the example in Fig. 4 (c), pixel *c* and *d* are selected for predicting or updating pixel *a*.

3. IMPROVEMENT OF JPEG2000

The curved WT has been used to improve the compression efficiency of JPEG2000. In the improved coder, the conventional WT is replaced by the curved WT. In addition to the resulting wavelet coefficients, two sets of curves x_i and y_i for each transform level are also encoded and transmitted to the decoder.

To efficiently encode the curves, a one-bit header is allocated for each set of curves. If all curves in a set of vertical (horizontal) curves are vertical (horizontal) straight lines, the header bit is “0” and the straight lines are not encoded. Otherwise, the header is “1” and the set of curves is encoded. As described in Section 2, a set of curves consists of line segments and the line segments in each block are of the same orientation. Therefore, the curves are represented by the orientation of the line segments in each block and encoded using arithmetic coding.

For transmitting the curves to the decoder, a marker segment of curves is defined and added to the tile headers in the code stream [1]. This marker segment consists of two bytes indicating the start of the marker segment, two bytes specifying the length of the marker segment, one bit indicating that the first set of curves for every level is vertical or horizontal curves, and then the codes of all the curves for the tile.

4. EXPERIMENTAL RESULTS

In following, the coding results of two well-known grayscale images, *Barbara* and *Lena*, are presented to show the performance of the improved coder. Both the images have a resolution of 512×512 pixels, 8 bits per pixel (bpp). *Barbara* contains a large amount of line features of various orientations, whereas *Lena* contains large regions where the luminance varies smoothly and a few edges of other than vertical and horizontal orientations. These images have transformed using both 9/7 and 5/3 wavelet filters.

Fig. 5 shows the PSNR of the decoded images at various bit rates. As expected, the improved coder outperforms JPEG2000, and the 9/7 wavelet filters outperforms the 5/3 filters in term of PSNR. For image *Barbara*, the improved coder achieves significantly higher PSNR than JPEG2000. The average PSNR gains are 1.01 dB with 9/7 filters and 1.38 dB with the 5/3 filters. The highest gain is 1.62 dB that is obtained with the 5/3 filters at the bit rate of 0.3 bits per pixel (bpp). For image *Lena*, the average PSNR gains of the improved coder are small, 0.016 dB with the 9/7 filters and 0.20 dB with the 5/3 filters, because the image contains a few edges.

For further demonstrating the performance of the improved coder, Fig. 6 shows portions of the decoded *Barbara* images obtained with the 9/7 filters at 0.2 bpp. In the image obtained with the improved coder, the lines on the clothes are well reconstructed and the edges of the arm are smooth. However, many lines in the image produced by JPEG2000 are lost or distorted, and there are serious "ringing" artifacts around the edges. Fig. 7 shows portions of the decoded *Lena* images at 0.1 bpp. The PSNR difference between the two decoded images is only 0.11 dB. However, the subjective quality of the image obtained using the improved coder is obviously better than that obtained using JPEG2000. The edges of the hat brim and the mirror frame in the image from the improved coder are smooth, whereas those in the image from JPEG2000 have significant "ringing" artifacts.

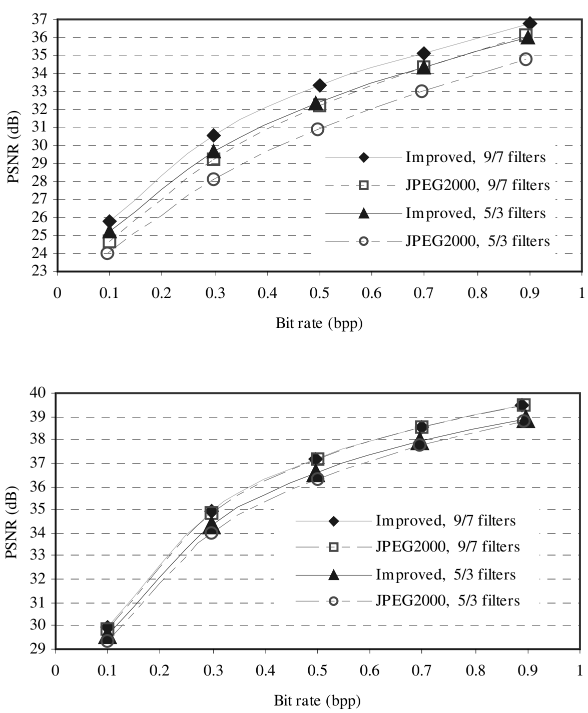
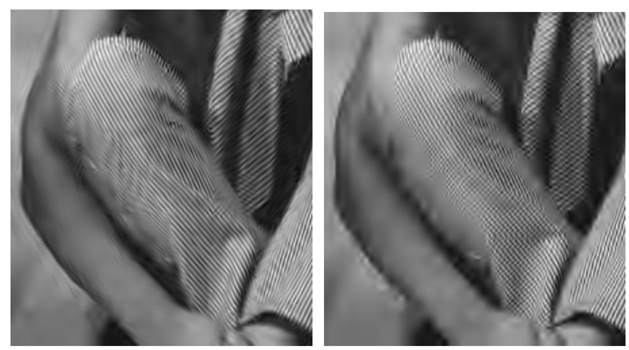


Fig. 5. PSNR obtained using the improved coder and the standard JPEG2000 for images *Barbara* and *Lena*.

CONCLUSIONS

In this paper, we present the curved wavelet transform in which 1-D wavelet filters are applied along curves, not only along horizontal and vertical straight lines. The curved wavelet transform can exploit information from the orientation of edges and lines to provide a compact representation for images. A simple algorithm for determining the curves is proposed. The resulting curves are usually parallel to edges and lines in the image to be compressed. The curved wavelet transform was used to improve JPEG2000 by replacing the conventional wavelet transform. Experimental results show that the effectiveness of the curved WT depends on image content. It provides significant improvement for images that contain a fair amount of sharp edges and lines. Compared with the standard JPEG2000, the improved image coder with the curved WT can provide a coding gain of over 1.62 dB in terms of PSNR.



(a) (b)

Fig. 6. Portion of decoded *Barbara* images at 0.2 bpp, (a) the improved coder (28.67dB), (b) JPEG2000 (27.25dB).

(a) (b)

Fig. 7. Portion of decoded *Lena* images at 0.1 bpp, (a) the improved coder (29.97dB), (b) JPEG2000 (29.86dB).

REFERENCES

[1] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Boston, 2002.

[2] D. Wang, L. Zhang, and A.Vincent, "Curved Wavelet Transform and Overlapped Extension for Image Coding," *Proceedings ICIP 2004*, Singapore, Oct. 24-27, 2004.

[3] M. D. Adams, http://www.ece.uvic.ca/~mdadams/jasper.