CURVED WAVELET TRANSFORM AND OVERLAPPED EXTENSION FOR IMAGE CODING

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 conventional 2-D wavelet transform for image coding is performed using a symmetric extension and 1-D filtering in the vertical and horizontal directions. In this paper, we present a *curved wavelet transform* and a method called *overlapped extension*. 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 using a small number of wavelet coefficients. The overlapped extension is proposed to prevent coding artifacts around the ends of the curves. Experimental results show that, compared with the conventional wavelet transform, the curved wavelet transform with overlapped extension significantly improves the subjective quality of decoded images, and the coding gain in PSNR can be up to 1.5 dB.

1. INTRODUCTION

Over the past decade, a number of image coders have been developed based on the 2-D wavelet transform (WT) and zero-trees [1]-[4]. These coders provide desired features and functionality with compression efficiency comparable to that of DCT-based image coders. In general, the 2-D WT in these coders is performed through 1-D filtering along horizontal and vertical straight lines. The main shortcoming of this WT is that it does not provide a compact representation of edges. In effect, the WT distributes the energy of an edge to several frequency bands. Many wavelet coefficients are thus required to properly reconstruct the edge. As a result, the conventional wavelet-based image coders produce "ringing" artifacts around edges, especially at low bit rates.

In this paper, we present a new wavelet-based image coder where the 2-D WT is performed through 1-D filtering along curves, not only along horizontal and vertical straight lines. This WT is referred to as curved wavelet transform, or *curved WT*. The basic idea behind the curved WT is that, if the curves are parallel to the edges and lines of an image, the signal consisting of the pixels along a curve mainly contains low frequency components with no or little high frequency components, and therefore can be well represented with a small number of wavelet coefficients. Although the idea is simple, its implementation for constructing an efficient image coder is challenging. The concept of applying 1-D filters in other orientations than horizontal and vertical ones appeared in [5]-[7]. In [5], an image is first partitioned into rectangular regions. Each region is then rotated and decomposed using conventional sub-band filters. In [6], wavelet filters may be applied along parallel straight lines of orientations other than horizontal and vertical, but not along curves. The geometric flows presented in [7] are indeed curved. The study in [7] was focused on the development and optimization of a new class of representation bases, called bandelet.

We propose new algorithms to implement the curved WT and to construct an efficient image coder. The algorithms include a method to determine the curves along which the filters are applied, half-pixel interpolation and full-pixel recovery for the cases where a curve passes through between two adjacent pixels, a recursive form of wavelet filters, and a signal extension method called *overlapped extension*. The curves are not constrained to be always parallel to one another. A curve may start and/or end in the interior of images. The overlapped extension is proposed to prevent the coding artifacts around the ends of curves that would be produced by the symmetric extension. The recursive form of wavelet filters is required for the inverse WT along the overlapped extended curves.

The paper is organized as follows. After the introduction, Section 2 is devoted to the curved WT, where we present its definition and constraints on the curves, a simple algorithm for determining the curves, and half-pixel interpolation and full-pixel recovery. Section 3 describes the recursive implementation of wavelet filters and the overlapped extension. Section 4 reports experimental results of image coding using the curved WT with overlapped extension. The conclusion of the paper and perspectives for future study are presented in Section 5.

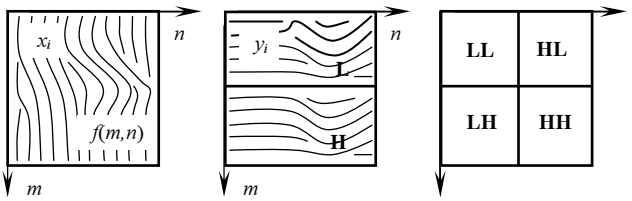

Fig.1. Concept of the curved WT. (a) Vertical curves, (b) horizontal curves, (c) four bands after the first level transform

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, for $0 \le i < I$, 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. Then, a set of horizontal curves y_i is determined for the

resulting coefficients (Fig. 1 (b)). The high-pass and low-pass filters are applied along these curves, and the outputs are sub-sampled by discarding every other column, producing four bands of coefficients LL, HL, LH, and HH (Fig. 1 (c)). The process described above achieves the first level of the curved WT. This process is then repeatedly applied to the resulting LL bands to obtain the second and higher levels of transform. 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 sub-sections, we describe these constraints, the half-pixel interpolation and full-pixel recovery for the curves passing through between two adjacent pixels, an algorithm to determine the curves, and the coding of the curves.

2.1. Constraints on the Curves

In general, a curve x_i in the first set must be a single-valued function of the vertical coordinate m, $x_i(m)$, i.e., it does not contain any horizontal segment, and does not have any branches. It can be of any length, but must be continuous without gaps. Such a curve is called a *vertical curve* in this paper. If a curve passes between two adjacent pixels, we can use the nearest full pixels or interpolate the half-pixel values that provide better results. Every pixel (or half-pixel) in an image $f(m, n)$ must lie on one of the vertical curves. With these constraints, the output of filtering along all the vertical curves can be sub-sampled by discarding every other row, and the coefficients L(m, n) (and H(m, n)) have a rectangular region of support. Similarly, a curve of the second set y_i is called a *horizontal curve*. It must be a continuous, single-valued function of the horizontal coordinate n. Under these constraints, a multi-level curved WT results in a dyadic decomposition of the image.

In order to reduce the computational and coding cost, the following three 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 x_i, the discrete orientations are $\pi/4$, $3\pi/8$, $\pi/2$, $5\pi/8$, and $3\pi/4$. For horizontal curves y_i, the orientation are $-\pi/4$, $-\pi/8$, 0, $\pi/8$, and $\pi/4$, as shown in Fig. 2. (ii) The image (or wavelet coefficients) to be transformed is divided into blocks of $K\times K$ pixels. Within each block, all curves are straight-line segments of the same orientation. (iii) Curves y_i consist of two identical groups, one for L(m, n) and the other for H(m, n).

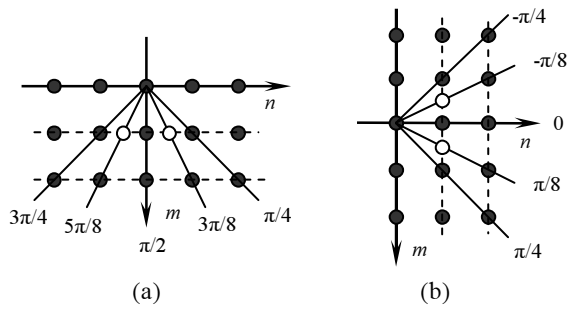

Fig. 2. Orientations of the line segments constituting the curves, (a) for vertical curves, (b) for horizontal curves. The solid circles stand for full-pixels and the empty ones for half-pixels.

2.2. Half-Pixel Interpolation and Full-Pixel Recovery

From Fig. 2, it can be seen that the line segments of orientations $-\pi/8$, $\pi/8$, $3\pi/8$, or $5\pi/8$ may pass between two pixels. As stated earlier we can use the nearest full pixels or, for better results, interpolate the half-pixel positions. In the latter case, since these half-pixel values are the ones that are coded and transmitted, the full-pixel values need to be recovered at the decoder.

The half-pixel values are interpolated using a half-band filter. The accurate recovery of the full-pixel values at the decoder is, however, difficult. After the inverse curved WT, the available data on a row (or a column) may include some full-pixels and some half-pixels. To recover the missing full-pixels, the value of some half-pixels that are not available may be required. These required half-pixels are first linearly interpolated. Then, the interpolated and the originally available half-pixels are used to recover the missing full-pixels through the half-band filter. This recovery is not perfect; the difference between the original value and the recovered value of a full-pixel is called full-pixel recovery error.

2.3. Determination of the Curves

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

To reduce the computational complexity, we propose a simple algorithm to determine the curves that are sub-optimal. This algorithm searches for the curves that minimize the energy of high-pass wavelet coefficients within each block of $K\times K$ pixels. After multiple levels of WT with the determined curves, the energy of the image is concentrated in the low-pass coefficients at the highest level, and the high-pass coefficients of all levels contain a small amount of energy. The 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 consists of the following steps:

1) The image (or wavelet coefficients) to be transformed is divided into blocks of $K\times K$ pixels.
2) Within each block, the wavelet filters are applied along the straight lines of each allowed orientation. The lines are extended into adjacent blocks, if they exist, in order to filter the pixels near the borders of the block.
3) The energy of the resulting high-pass coefficients for each of the allowed orientations is calculated.
4) For an orientation that requires the value of half-pixels, the energy of the full-pixel recovery error is added to that of high-pass coefficients.
5) The orientation that results in the lowest energy within the block is chosen, and the straight lines in all the blocks are connected (under the constraint of having no branches) to form a set of curves.

Fig. 3 illustrates a set of vertical curves determined using this algorithm for the image *Barbara*. The curves are generally parallel to the edges and lines of orientation between $\pi/4$ and $3\pi/4$ in the image.

2.4. Coding of the Curves

The two sets of curves x_i and y_i for each level have to be transmitted to the decoder. To efficiently code these curves, an one-bit header is allocated for each set of curves. If all curves of x_i are vertical lines, or if all curves of y_i are horizontal lines, the header bit is "0" and the straight lines are not coded. Otherwise, the header is "1", and the curves are represented by the orientation of the line segments in each block and coded using arithmetic coding.

Fig.3. Illustration of curves for image *Barbara*.

3. OVERLAPPED EXTENSION

Before applying the wavelet filters to a sequence of pixels of finite length, the sequence has to be extended in order to calculate the filter outputs at the positions close to the ends of the sequence. The conventional extension method is a symmetric extension. It is known that the symmetric extension results in additional distortion in image coding [8]. From Fig. 3, we can see that some curves start and/or end in the interior of the image. If the symmetric extension is used for these curves, objectionable artifacts may appear around the ends of the curves.

To eliminate these artifacts, a curve that starts and/or ends in the interior of the image is extended to pixels of other curves, i.e., some pixels on other curves are appended to this curve and are used to calculate the wavelet coefficients of the pixels on the curve. This method is called *overlapped extension*. The forward transform with the overlapped extension is performed using the conventional non-recursive wavelet filters. The inverse transform can be performed, only after the inverse transform for the pixels on the extended part has been done, using the following recursive filters:

$$f(i) = \sum_{k=1}^{Lp} r_{i \bmod 2}(k) f(i-k) + \sum_{k=-Ln}^{0} r_{i \bmod 2}(k) F(i-k) \quad (1)$$

where $F(i)$ is a sequence of wavelet coefficients, $f(i)$ is the output of the filters. (i mod 2) is equal to 0 if i is an even number and 1 if i is an odd number. $r_0(k)$ and $r_1(k)$ are the filter coefficients, which can be obtained from the non-recursive filters. In the recursive filters, the previous outputs are used as inputs for calculating the current output.

In case the overlapped extension cannot be achieved because there are not enough previously inverse transformed pixels, the curve is extended by the symmetric extension and the conventional non-recursive filters are used for the inverse transform.

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 curved WT with overlapped extension. 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 other than vertical and horizontal orientations. In the curved WT, the block size is 32×32. The 9/7 wavelet filters [1] and the SPIHT algorithm [3] are used in both the curved and conventional WT.

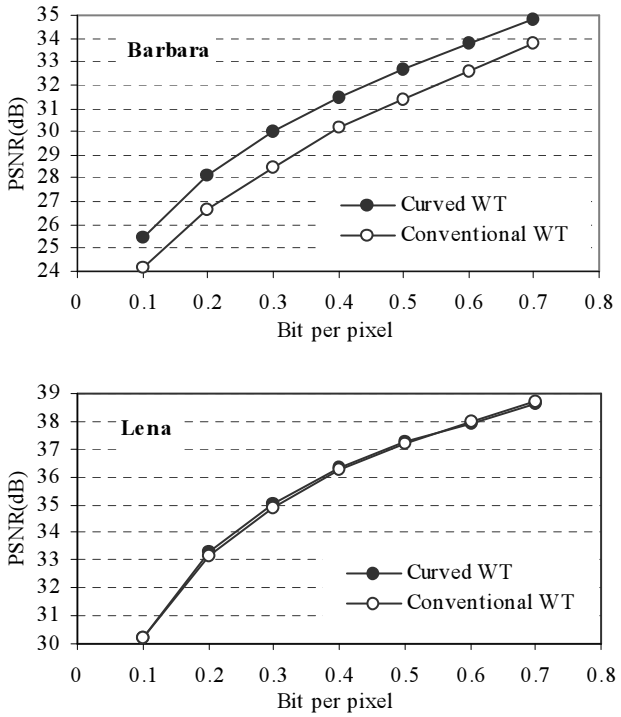

Fig. 4. PSNR of decoded images using the curved WT with overlapped extension and the conventional WT.

Fig. 4 shows the PSNR of the decoded images at various bit rates. For *Barbara*, the curved WT with overlapped extension achieves a significantly higher PSNR and subjective quality (shown in Fig. 5) than the conventional WT. The PSNR difference is 1.29 dB on average and is up to 1.51 dB at a bit rate of 0.3 bpp. For *Lena*, the curved WT produces an average PSNR only slightly higher than that of the conventional WT. However, the subjective quality of the image obtained using the curved WT is better than that obtained using the conventional WT. Comparing the images shown in Fig. 6 (b) and (c), one can see that the edges of the hat

brim and the mirror frame in the image of curved WT are smooth, whereas those in the image of conventional WT have significant aliasing.

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 on the orientation of edges and lines to produce a compact representation of an image. A simple algorithm for determining the curves is proposed. The resulting curves are usually parallel to edges and lines in the image. To eliminate the artifacts resulting from the symmetric extension, an overlapped extension and a recursive form of wavelet filter are proposed. The overlapped extension extends the curve using pixels on other curves. The recursive wavelet filters are required for the inverse wavelet transform. Experimental results show that, compared with the conventional WT, the curved WT with overlapped extension significantly improves the subjective quality of decoded images and the coding gain in PSNR can be up to 1.5 dB. The performance of the proposed image coder can be further improved by using optimal curves and a better full-pixel recovery method.

REFERENCES

[1] M. Antonini, *et al.*, "Image coding using wavelet transform," *IEEE Trans. Image Processing,* vol. 1, pp. 205-220, April 1992.

[2] J. M. Shapiro, "Embeded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing,* vol. 41, pp. 3445-3462, Dec. 1993.

[3] 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. 243-250, June 1996.

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

[5] D. Taubman and A. Zakhor, "Orientation adaptive subband coding of images," *IEEE Trans. Image Processing,* Vol. 3, pp. 421-436, July. 1994.

[6] P. L. Dragotti, *et al.*, "Discrete directional wavelet bases for image compression," *Proceedings of SPIE Vol. 5150, Visual Communications and Image Processing 2003*, pp. 1287-1295, 2003.

[7] E. Le Pennec and S. Mallat, "Geometrical image compression with bandelets," *Proceedings of SPIE Vol. 5150, Visual Communications and Image Processing 2003*, pp. 1273-1286, 2003.

[8] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consumer Electronics*, Vol. 46, No. 4, pp. 1103-1127, Nov. 2000.

Fig. 5. Portions of decoded *Barbara* images at 0.2 bpp, (a) original image, (b) the proposed coder, (c) the conventional WT and SPIHT.

Fig. 6. Portions of decoded *Lena* images at 0.2 bpp (a) original image, (b) the proposed coder, (c) the conventional WT and SPIHT.