Arbitrary Shape Wavelet Transform with Phase Alignment

Jin Li and Shawmin Lei
Digital Video, Sharp Laboratories of America
5750 NW Pacific Rim Blvd., Camas, WA 98607
E-mail: {lijin,shawmin}@sharplabs.com

Abstract

In this paper, we propose a wavelet transform for arbitrary shape object. Compared with previous schemes, the proposed transform generates exactly the same number of coefficients as that of the original object. What is more, the phase of the horizontal wavelet filter is aligned so that the subsequent vertical transform is applied on coefficients with coherent phases and thus the transform efficiency is improved. We denote the scheme as the arbitrary shape wavelet transform with phase alignment (ASWP). The scheme was proposed at the MPEG4 meeting at Stockholm in July 1997, and was adopted by MPEG4 CD ver1 for wavelet transform of arbitrary shape object.

1 Introduction

As image compression becomes more and more popular, applications call for techniques that can handle not only rectangular objects, but also objects with arbitrary shape. Some typical applications include image editing, computer/video games, virtual environment, internet, etc. In response to such demands, the MPEG4 committee has established an expert group devoted specifically to the compression of arbitrary shape objects.

An arbitrary shape object consists of a shape mask and a content image. The shape mask is usually binary, indicating whether the pixels belong to an object. However, it may be of multiple values, indicating the level of transparency (i.e., the alpha channel). Coding of an arbitrary shape object involves the coding of the binary mask and the coding of the content image. If the shape mask is of multiple values, it is first decomposed into a binary shape mask and a gray scale content image and then encoded. Compression of the binary shape mask can be achieved by the modified modified READ (MMR) coding or the context arithmetic coding. Compression of the gray scale content image involves arbitrary shape object transform, quantization and entropy coding. In this contribution, we propose an effective method of transforming an arbitrary shape object into the wavelet domain.

Since the content image is not of rectangular shape, regular DCT and wavelet transform can not be applied directly. There are a number of existing approaches for transforming an arbitrary shape content image. The most popular approach is padding. Wu et al [2] proposed a padding based DCT scheme, where the image was segmented into fixed size blocks and the pixels were padded repetitively for blocks not fully occupied by the object. To reduce the number of padding pixels, Moon et al [3] proposed a systematic way of changing the block positions and tiling the blocks so that less coefficients were padded. Katata et al [4] proposed a padding based object wavelet transform (OWT) where only pixels which surround the original object and contribute significantly to the wavelet transform are padded. Padding increased the number of coefficients to be coded and thus reduced the coding efficiency. Kauff et al [6] proposed a shape-adaptive DCT (SA-DCT) which avoided the padding. When applied to a block not fully occupied by the object, SA-DCT first moved all pixels toward the upper block boundary. A variable basis DCT was applied to each column with the number of DCT basis equaled to the number of coefficients in each column. The pixels were then moved toward the left block boundary, and a similar variable basis DCT with basis corresponding to the number of coefficients in each row was applied horizontally. SA-DCT avoided padding, however, the variable basis DCT used in SA-DCT had no fast algorithms, and the neighbor pixels in the horizontal transform might not be the neighbor pixels in the original image, which reduced the transform efficiency. Li et al [5] proposed a non padding shape adaptive wavelet transform. The arbitrary shape object was first transformed in the horizontal direction and then in the vertical direction. When the data length was longer than the filter length, it was first truncated to the next available even length and transformed directly with a circular wavelet transform. The extra data point in case the data length is odd is directly copied into the low pass band. When the data length was shorter than the filter length, the Haar transform was applied. The scheme generated exactly the same number of coefficients as that of the original object, however, the implementation was complex since the scheme treated the data differently depending on its length. The transform efficiency was also not very good as the vertical transform may operate on coefficients generated by different types of wavelet filter and of different phases.

0-8186-8821-1/98 $10.00 © 1998 IEEE

2 Arbitrary shape wavelet transform with phase alignment (ASWP)

In this work, we propose an arbitrary shape wavelet transform with phase alignment (ASWP). The proposed transform generates exactly the same number of coefficients as that of the original object. What is more, the phase of the horizontal wavelet filter is aligned. Such arrangement improves the efficiency of the subsequent vertical transform because it is applied on coefficients with coherent phases. We proposed the scheme at the MPEG4 Stockholm meeting held in July 1997. Coincidentally, Lehigh University and Oki Electronics had independently developed and proposed a similar scheme at the same meeting. The joint scheme was later adopted by MPEG4 CD version 1 for wavelet transform of arbitrary shape object.

MPEG CD version 1 only includes the ASWP with odd tap symmetrical filter. We will discuss the ASWP with both the odd and even tap filter in this paper.

2.1 Wavelet transform with symmetrical boundary extension

The implementation of ASWP is facilitated by the wavelet transform with symmetrical boundary extension. If the wavelet filter is symmetrical (i.e., of linear phase), symmetrical signal extension can be applied so that the number of wavelet transformed coefficients is exactly the same as that of the original data. There are two extension approaches which correspond to the filter with even and odd tap length, respectively. Let us first consider the case that the filter is an odd tap symmetrical wavelet filter. Both the low and high pass odd tap filters are symmetrical with regard to the center of the filter. During the filter operations, they are centered at alternate even and odd index positions, as shown in Figure 1. The wavelet decomposition may start with either low or high pass decomposition, which is denoted as phase 0 or phase 1 filtering, respectively. Let the original and transformed signal be denoted by $x(n)$ and $y(n)$ with $n=0,1,...,N-1$, respectively. The boundary extension first mirrored the data to form signal $x'(n)$ of length $2N$-2:

$$x'(n)=\begin{cases} x(n) & n=0,1,\ldots,N-1 \\ x(2N-2-n) & n=N,\ldots,2N-3 \end{cases} . \tag{1}$$

Then, $x'(n)$ is periodically repeated to form signal $\tilde{x}(n)$ of period $2N$-2:

$$\tilde{x}(n)=\sum_j x'[n-(2N-2)j]. \tag{2}$$

The extension of $\tilde{x}(n)$ is equivalent to mirror $x(n)$ repetitively along two boundary points 0 and N-1, with boundary points themselves excluded in the mirroring operation. Because of the symmetry of the extended data and the filter, the resultant decomposed wavelet coefficient $\tilde{y}(n)$ is also a period $2N$-2, symmetrically mirrored signal. Therefore, it is sufficient to store only a length N segment of $\tilde{y}(n)$, which constitutes the transform result $y(n)$. In reconstruction, $y(n)$ is symmetrically extended to form $\tilde{y}(n)$, which is then used to reconstruct $\tilde{x}(n)$. A length N segment of $\tilde{x}(n)$ is just the original signal $x(n)$. An example of symmetrical extension with odd tap filter with phase 0 and data length 5 is demonstrated in Figure 1.

Let us consider the second case that the filter is an even tap symmetrical wavelet filter. Boundary extension of the even tap filter is different than the odd tap case because that the high pass even tap filter is anti-symmetrical while the low pass filter is still symmetrical. Moreover during the filter operations, both the low and high pass filters are centered at the same position, between a pair of pixels in the original data, as shown in Figure 2. The filter may be centered between index 0 and 1, or between index -1 and 0, which corresponds to phase 0 or phase 1 filtering, respectively. In the boundary extension, the signal is first mirrored symmetrically to form signal $x'(n)$ of length $2N$:

$$x'(n)=\begin{cases} x(n) & n=0,1,\ldots,N-1 \\ x(2N-1-n) & n=N,\ldots,2N-1 \end{cases} , \tag{3}$$

which is then periodically repeated to form signal $\tilde{x}(n)$ of period $2N$:

$$\tilde{x}(n)=\sum_j x'[n-2Nj]. \tag{4}$$

The extension of $\tilde{x}(n)$ can also be considered as mirroring $x(n)$ repetitively along two points -0.5 and N-0.5, with the boundary points 0 and N-1 included in the mirroring operation. Filtering the periodic signal $\tilde{x}(n)$ results in a periodic $2N$ coefficient $\tilde{y}(n)$. Because of the symmetry of the data and the low pass filter and the anti-symmetry of the high pass filter, the low pass coefficients are symmetrical, and the high pass coefficients are anti-symmetrical with regard to mirror points -0.5 and N-0.5. Moreover, whenever the center of the filter is at points either -0.5 or N-0.5, the high pass coefficient will be zero and there will be only low pass coefficient at the boundary point. There are still N independent coefficients after filtering, which constitute the filtered coefficient $y(n)$ and are still the same as the original signal points. Depending on the phase and data length, there may be 0, 1 or even 2 fewer high pass coefficients than the low pass. In reconstruction, $y(n)$ is first extended to form $\tilde{y}(n)$, where the low pass coefficients are symmetrically mirrored and the high pass coefficients are anti-symmetrically mirrored along points -0.5 and N-0.5. Inverse filtering $\tilde{y}(n)$ and truncating a length N segment reconstructs the original signal $x(n)$. An example of symmetrical extension with even tap filter with phase 1 and data length 4 is demonstrated in Figure 2.

2.2 One-dimensional ASWP

We first explain the 1-D single scale ASWP algorithm. Let the object mask be denoted by *m(i)*. We assume that the mask is binary, with '1' for object pixel and '0' for non object pixel. The ASWP is again different between the odd tap filtering and the even tap filtering. We first illustrate 1-D ASWP with odd tap symmetrical filter, with detail procedure follows:

Step 1. Segment identification

Each segment is identified from the 1-D mask *m(i)*. Let the start and end index of each segment be *idx_st* and *idx_ed*, and let the length of the segment *be len=idx_ed-idx_st+1*.

Step 2. Arbitrary shape wavelet transform with phase alignment

Each segment is independently extended and filtered. The symmetrical extension of odd tap filter discussed in Section 2.1 is applied. The phase of the filter is fixed with regard to the original data index, i.e., the low pass filter is always applied at even index *2i*, and the high pass filter is always applied at odd index *2i+1*, with results stored at index *i* of the low and high pass band, respectively. Depending on the parity of the start index, the segment is decomposed with phase 0 filtering (even *idx_st*) or phase 1 filtering (odd *idx_st*). The fixed filter phase will be beneficial for the 2D transform. The decomposition generates the same number of low and high pass coefficients for even length segment, and generates one more low (even *idx_st*) or high (odd *idx_st*) pass coefficient for odd length segment.

An example of 1-D odd tap ASWP is shown in Figure 3. The length of the data is 16 and there are two segments. The first segment starts at odd index 3 with length 4, the second one starts at odd index 11 with length 3. Each segment is independently extended and filtered. In this example, since both segments start with odd index, they are decomposed with phase 1 filtering, i.e., starting with the high pass filter. The decomposition results in 2 low and 2 high pass coefficients for the first segment, and 1 low and 2 high pass coefficients for the second segment. They are stored at index 2-3, 6 of the low pass band and index 1-2, 5-6 of the high pass band. Both segments can be perfectly reconstructed from the transform coefficients.

The operation procedure of 1-D even tap ASWP is similar to that of the odd tap case. The segment is first identified and the start, end, and length of the segment are recorded. Each segment is extended and filtered independently. The boundary extension and filtering follows the rule of symmetrical extension with even tap filter discussed in Section 2.1. The phase of the filter is again fixed with regard to the original data index, and the center of the low and high pass filtering is always located between index *2i* and *2i+1*. The resultant coefficients are stored at index *i* of the low and high pass band, respectively. Depending on the parity of start index *idx_st*, the wavelet transform of the segment may start with either phase 0 (even *idx_st*) or phase 1 (odd *idx_st*). In case that the start index *idx_st* or the end index *idx_ed* is odd, the center of the filter coincides with the segment boundary, which results in one more low pass coefficient as the high pass coefficient will be zero at the boundary. Depending on the parity of the start index *idx_st* and the parity of the segment length *len*, there may be 0, 1, or 2 more low pass coefficients than that of high pass.

An example of one-dimension even tap ASWP is shown in Figure 4. The data sample is the same as that of Figure 3. Since both segments start with odd index, they are decomposed with phase 1 filtering, i.e., the center of the filter crosses the left segment boundary. The decomposition results in 3 low and 1 high pass coefficients for the first segment, and 2 low and 1 high pass coefficients for the second segment. The resultant coefficients are stored at index 1-3, 5-6 of the low pass band and index 2 and 6 of the high pass band. Both segments can be perfectly reconstructed from the transform coefficients.

When the length of the segment is 1, i.e., a single pixel, we simply copy the pixel to the low (for odd tap filtering with even *idx_st* or for even tap filtering) or high (for odd tap filtering with odd *idx_st*) pass band.[1]

2.3 Two-dimensional ASWP

In 2-D ASWP, the object is first horizontal filtered with 1-D ASWP for each row and then vertically filtered for each column, as shown in Figure 5. Since the phase of the wavelet filter is always fixed, the vertical transform is applied on the coefficients of coherent phases of the horizontal transform. Such phase alignment of ASWP improves the transform efficiency. Multi-scale 2-D ASWP can be achieved by recursively decomposing the LL subband of each scale. The implementation of ASWP is consistent with the rectangular wavelet transform with symmetrical boundary extension. In fact, the rectangular wavelet transform is a special case of ASWP with the object being the entire image.

3 Experimental Results

Experiments have been conducted to compare ASWP with a number of arbitrary shape object transform schemes, which include the padding based DCT

[1] This differs from the implementation of MPEG4 CD version 1, which multiplies the single pixel by a factor of $\sqrt{2}$ and always copies it to the low pass band.

(PADDCT), the shape-adaptive DCT (SA-DCT)[3] and the object wavelet transform(OWT)[4]. The test images are the first frame Akiyo, Coast, Sean, and Weather of CIF and QCIF format. The test objects are the woman in Akiyo, the man in Sean, the big boat in Coast, and the woman in Weather. Objects are coded at 5k, 10k, 15k bits per frame for QCIF format, at 15k, 30k, 50k for CIF format. All schemes encode the binary shape mask with the context arithmetic coding (CAE) scheme and the shape bitrate is included in the result. PADDCT and SA-DCT transformed coefficients are further compressed by a 2-D run level huffman coding, the OWT and ASWP coefficients are compressed by the MZTE, both specified in MPEG4 VM7. The comparison results are shown in Table 1. It is observed that ASWP outperforms OWT, a padding based arbitrary shape object wavelet transform scheme for about 0.1-2.6dB over all test images, with an average of 0.8dB. Since the coefficient coding schemes of the OWT and ASWP are the same, the gain confirms that ASWP reduces the number of coefficients to be coded and improves the coding efficiency by eliminating the padding. It can also be observed that ASWP outperforms SA-DCT for an average of 0.8dB, and outperforms PADDCT for an average of 0.2dB for our test images. ASWP thus demonstrates itself to be an efficient scheme for the wavelet transform of arbitrary shape object.

4 References

[1] C. M. Brislawn, "Preservation of subband symmetry in multirate signal coding", *IEEE Trans. on Signal Processing*, vol. 43, no. 12, pp. 3046-3050, Dec. 1995.

[2] Z. Wu and T. Kanamaru, "Block-based DCT and wavelet selective coding for arbitrarily shaped images", *Visual Communication and Image Processing'97*, SPIE Vol. 3024, pp. 658-665, Jan. 1997, San Jose, CA.

[3] J. Moon, G. Park, S. Chun and S. Choi, "Shape-adaptive region partitioning method for shape-assisted block-based texture coding", *IEEE Trans. on Cir. & Sys. for Video Technology*, vol. 7, no.1, pp.240-246,Feb. 1997.

[4] H. Katata, N. Ito, T. Anno and H. Kusao, "Object wavelet transform for coding of arbitrarily shaped image segments", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp.235-237, Feb. 1997.

[5] W. Li and S. Li, "Shape-adaptive discrete wavelet transform for coding arbitrarily type shaped texture", *Visual Communication and Image Processing'97*, SPIE Vol. 3024, pp. 1046-1056, Jan. 1997, San Jose, CA.

[6] P. Kauff, B. Makai, S. Rauthenberg, U.Golz, J. Lameillieure and T. Sikora, "Functional coding of video using a shape-adaptive DCT algorithm and an object-based motion prediction toolbox", *IEEE Trans. on Circuits and systems and Video Technology*, vol. 7, no. 1, pp.181-196, Feb. 1997.

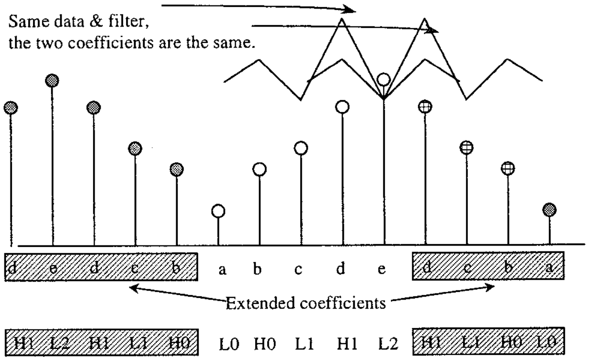

Figure 1. Wavelet decomposition with symmetrical extension, (odd tap filter, phase 0)

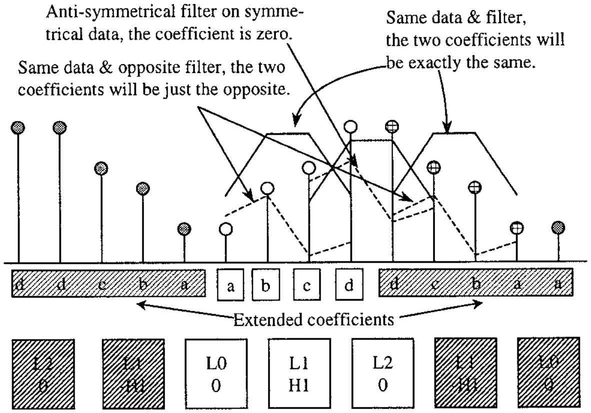
Anti-symmetrical filter on symmetrical data, the coefficient is zero.
Same data & filter, the two coefficients will be exactly the same.
Same data & opposite filter, the two coefficients will be just the opposite.
d d c b a
a b c d
d c b a a
Extended coefficients
L1 0
L1 -H1
L0 0
L1 H1
L2 0
L1 -H1
L0 0

Figure 2. Wavelet decomposition with symmetrical extension, (even tap filter, phase 1)

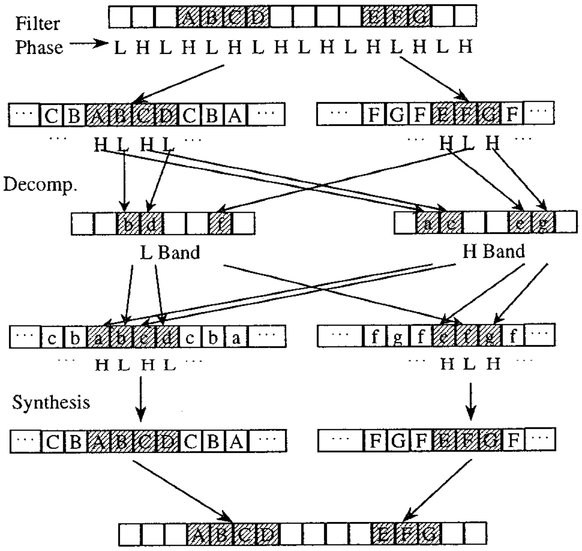

Figure 3. 1-D odd tap arbitrary shape wavelet transform with phase alignment

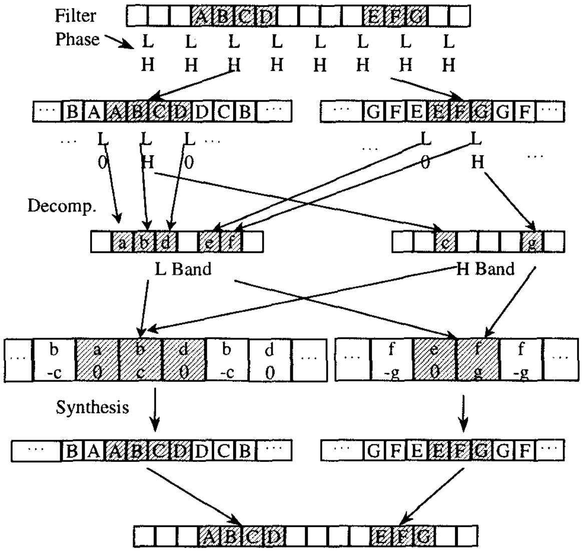

Figure 4. 1-D even tap arbitrary shape wavelet transform with phase alignment

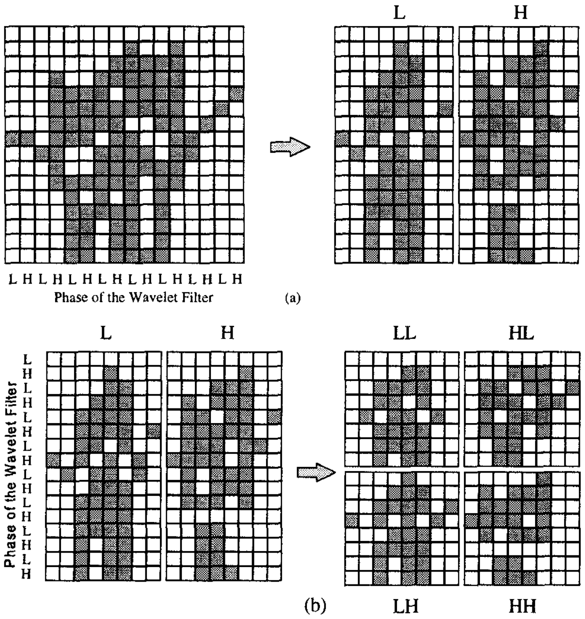
L
H
L H L H L H L H L H L H L H
Phase of the Wavelet Filter
(a)
L
H
L H L H L H L H L H L H L H L H
Phase of the Wavelet Filter
LL
HL
(b)
LH
HH

Figure 5. 2-D arbitrary shape wavelet transform using odd tap filter with (a) 1-D horizontal decomposition of each row and (b) 1-D vertical decomposition of each column

Table 1 Comparison between ASWP, SA-DCT, PADDCT and OWT for images Akiyo, Coast, Sean and Weather

ASWP	QCIF		CIF	
Image	Bit Rate (bits)	PSNR (dB)	Bit Rate (bits)	PSNR (dB)
Akiyo	4934	28.1	15157	31.0
	10118	33.0	29645	35.4
	14590	36.1	50765	40.0
Coast	4962	26.0	15036	26.9
	10098	35.1	30036	35.0
	14770	42.3	47668	42.1
Sean	5010	26.3	14900	28.7
	9906	31.1	29564	33.0
	15242	34.7	48700	37.0
Weather	5007	28.5	14976	30.5
	9999	33.2	30336	35.8
	15375	37.2	49544	40.2
SA-DCT Akiyo	4984	27.1	15056	29.5
	10016	31.8	30712	34.7
	15480	35.3	51880	39.3
Coast	5064	25.9	15064	27.3
	10192	34.5	29360	35.0
	13912	39.7	49200	41.8
Sean	4944	26.3	14824	28.0
	9800	30.6	30888	33.2
	14336	34.0	49896	37.4
Weather	5096	27.3	14896	29.2
	9960	31.6	30584	34.8
	14184	34.6	47080	39.0
PADDCT Akiyo	4968	28.1	14744	30.3
	10112	32.6	29152	35.1
	15712	36.1	47008	39.0
Coast	5048	27.0	14792	28.7
	9976	34.3	28144	35.2
	15616	40.6	45848	40.8
Sean	5032	26.6	15168	28.8
	9984	31.3	30032	33.6
	14464	34.5	52824	38.4
Weather	5039	28.1	14968	30.3
	10423	33.0	29464	35.4
	15144	35.6	51192	40.4
OWT Akiyo	5036	27.9	15013	30.4
	10308	32.7	30349	34.7
	15316	36.0	53389	38.9
Coast	5033	24.9	15222	25.9
	10377	32.9	31038	33.5
	15913	39.7	54278	39.9
Sean	5017	25.8	15194	28.1
	10017	30.5	30218	32.6
	15073	33.6	50962	36.8
Weather	5039	28.1	15333	30.2
	10423	33.0	31197	35.4
	15839	36.9	52509	40.1