

RESEARCH

Open Access



Invertible update-then-predict integer lifting wavelet for lossless image compression

Dong Chen^{1*}, Yanjuan Li², Haiying Zhang³ and Wenpeng Gao¹

Abstract

This paper presents a new wavelet family for lossless image compression by re-factoring the channel representation of the update-then-predict lifting wavelet, introduced by Claypoole, Davis, Sweldens and Baraniuk, into lifting steps. We name the new wavelet family as invertible update-then-predict integer lifting wavelets (IUPILWs for short). To build IUPILWs, we investigate some central issues such as normalization, invertibility, integer structure, and scaling lifting. The channel representation of the previous update-then-predict lifting wavelet with normalization is given and the invertibility is discussed firstly. To guarantee the invertibility, we re-factor the channel representation into lifting steps. Then the integer structure and scaling lifting of the invertible update-then-predict wavelet are given and the IUPILWs are built. Experiments show that comparing with the integer lifting structure of 5/3 wavelet, 9/7 wavelet, and IDTT, IUPILW results in the lower bit-rates for lossless image compression.

Keywords: Integer lifting, Invertibility, Lossless image compression, Update-then-predict, Wavelet

1 Introduction

Discrete wavelet transforms and perfect reconstruction filter banks have become one of the dominant technologies in numerous areas such as signal and image processing [1–3]. The second-generation wavelets based on lifting scheme have achieved substantial recognition [4–6], which are used in the fields of signal analysis [7], image coding [8–11], palmprint identification [12], moving object detection [13], especially since their integration in the JPEG2000 standard [14–18]. The lifting scheme is an efficient and powerful tool to compute the wavelet transform. It can improve the key properties of the first-generation wavelet step by step. Moreover, it has many advantages compared to the first-generation wavelet such as in-place computation, integer-to-integer transforms, and speed.

Update-first structure is useful to build the adaptive lifting wavelet [19, 20]. G. Piella and B. Pesque-Popescu present some adaptive wavelet decompositions that can capture the directional nature of images [20]. Claypoole, Davis, Sweldens, and Baraniuk introduce a kind of nonlinear wavelet transform for image coding via lifting [21]. To

keep the stability and eliminate the propagation of error, they constructed the update-then-predict lifting wavelet using Donoho's average-interpolation [22], and they apply it to construct the nonlinear wavelet transforms. However, unfortunately it is not perfect invertible for lossless image compression using integer-to-integer structure because there is a fractional factor 1/2 in its low-pass channel (see Fig. 3), which will be discussed in detail in Section 2.2 in this paper.

Our contributions can be summarized as follows. (1) The update-then-predict lifting structure is reviewed and its limitation is given in Section 2.2. Our analysis shows that the fractional factor 1/2 destroys the perfect reconstruction property of the integer structure of update-then-predict wavelet and makes the structure is not invertible. (2) The solution method is given. To perfect the update-then-predict lifting structure, we consider some central issues such as normalization, invertibility, integer structure, and scaling lifting. We re-factor the channel representation of the previous update-then-predict lifting wavelet with normalization into lifting steps, and then the invertible update-then-predict integer lifting wavelets (IUPILWs) for lossless image compression is obtained and named in Sections 3.1 to 3.3. (3) The computational complexity analysis and comparison between IUPILWs and other methods are given in Section 3.4. Furthermore, the

*Correspondence: peakgrin@gmail.com

¹School of Life Science and Technology, Harbin Institute of Technology, No. 92 Dazhi West Street, 150001 Harbin, China

Full list of author information is available at the end of the article

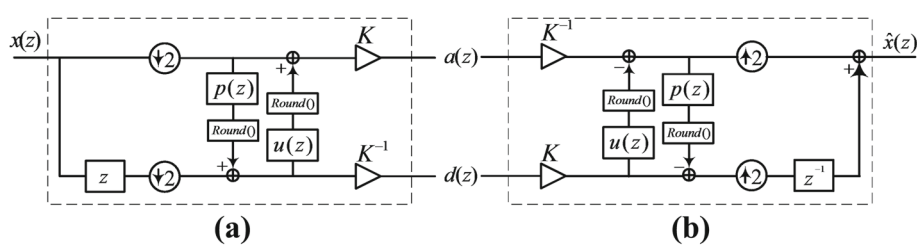


Fig. 1 **a** Analysis part of integer lifting structure. **b** Synthesis part of integer lifting structure

experimental comparison and analysis for lossless image compression are given and the advantages of our IUPILWs are introduced in Section 4.

The remainder of the paper is organized as follows. Section 2 gives a brief description of the background of integer-to-integer lifting wavelet and update-then-predict lifting wavelet. Section 3 introduces the invertible update-then-predict integer lifting wavelets with scaling lifting. According to reference [21], the channel representation of the update-then-predict lifting wavelet with normalization is given firstly. Furthermore, we re-factor the channel representation into lifting steps and then the invertibility is guaranteed. Then the integer structure and scaling lifting of the invertible update-then-predict wavelet filter banks are investigated. Finally, the computational complexity is analyzed. Sections 4 and 5 give the experiments and conclusion, respectively.

2 Integer-to-integer lifting wavelet and update-then-predict lifting wavelet

2.1 Integer-to-integer lifting wavelet

The integer-to-integer lifting wavelet transforms are proposed in [23]. Integer-to-integer wavelet transforms have important application in lossless image compression. In most cases, the wavelet filters that are used have floating point coefficients, but the images consist of integer point. This leads that the wavelet decomposition coefficients of images are floating point numbers. We know that the floating point numbers are disadvantage for the lossless compression because they need more encoding bits. Therefore, to reduce the encoding bits of lossless compression, the authors of [23] introduced the integer-to-integer lifting wavelet. The structures of integer-to-integer lifting wavelets are shown as follows (Fig. 1).

Figure 1 denotes the analysis part and synthesis part of integer lifting structure. In Fig. 1a, the “Round()” operations are given following the steps prediction $p(z)$ and update $u(z)$, respectively. However, the scaling factors K and K^{-1} ($K \neq 1$) make the approximate coefficients $a(z)$ or detail coefficients $d(z)$ are not the integer point numbers, then make the structure is not integer-to-integer.

One solution method we can imagine is omitting the scaling factors K and K^{-1} in Fig. 1a, b. If the scaling

factors are omitted, the approximate coefficients $a(z)$ and detail coefficients $d(z)$ are all integer point numbers via lifting wavelet transform. Therefore, it seems that the integer-to-integer lifting is achieved. However, the problem is whether the structure obtained by omitting the scaling factors is a kind of wavelet filter with normalization. Obviously the answer is no. The reason is that the lifting wavelets are usually obtained by factoring the traditional wavelets, and the scaling factors are the important parts of the factoring. If we omit the scaling factors, the structure of the traditional wavelet is also destroyed. The function of the scaling factors is to keep the same energy for the coefficients in different scale. “Keeping the same energy” is important to image compression, it can make the encoding algorithm using less bits to encode the wavelet coefficients. Therefore, the method by omitting the scaling factors is not a good choice.

Another solution method is to lift the scaling factors, which is introduced in [4]. We will review the lifting of scaling factors and build our invertible update-then-predict integer lifting wavelet filter bank with scaling lifting in Section 3.3.

2.2 Update-then-predict lifting wavelet with normalization

The update-then-predict lifting wavelets are introduced in reference [21] by Claypoole, Davis, Sweldens, and

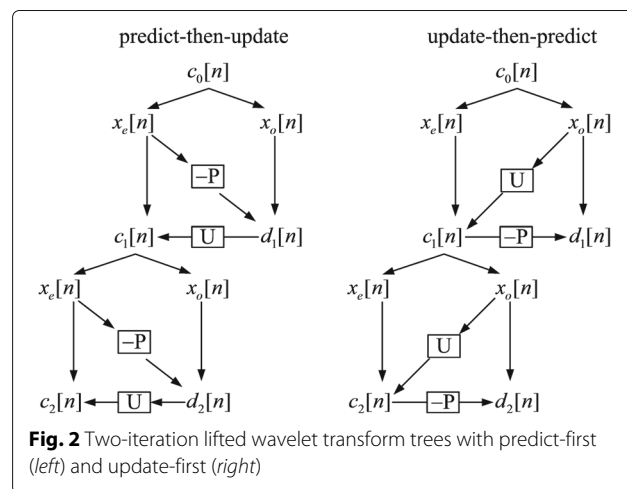


Fig. 2 Two-iteration lifted wavelet transform trees with predict-first (left) and update-first (right)

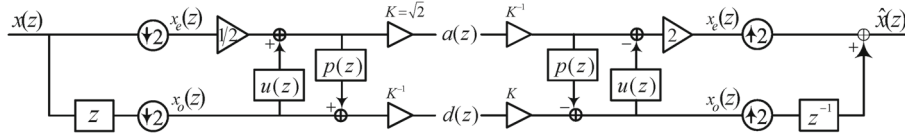


Fig. 3 Update-then-predict lifting wavelet filter bank with normalization

Baraniuk. To ensure the stability of the wavelet transform for the image coding, the authors introduced the update-then-predict lifting structure and applied them to design the nonlinear wavelet. In [21], the authors discussed the advantages of the update-then-predict lifting structure. That is, comparing with the predict-then-update lifting structure, it has more stability and synchronization.

See Fig. 2, when predicting first, the prediction must be performed prior to construction of the approximate coefficients and iteration to the next scale. When updating first, the prediction operator is outside the loop. The approximate coefficients can be iterated to the lowest scale, quantized, and reconstructed prior to the predictions. They analyzed that for the update first, the transform is only iterated on the low pass coefficients $c[n]$, all $c[n]$ throughout the entire pyramid linearly depend on the data and are not affected by the nonlinear predictor. Therefore, the update-then-predict lifting structure is effective for the building of nonlinear lifting wavelets.

We give the structure of update-then-predict lifting wavelet filter banks with normalization in Fig. 3.

In Fig. 3, the update filter consists of a fixed value, that is, $u(z) = 1/2$. According to reference [21], the prediction filters of update-then-predict lifting structure of $(1, N)$ are obtained and shown in Table 1.

To build the integer-to-integer lifting structure of Fig. 3, we can replace the analysis part and synthesis part using the structure in Fig. 1a, b. However, the factor 1/2 in Fig. 3 must be remained and it is an obstacle for the implementation of integer-to-integer. For example, considering the situation there is a “Round()” operation after fractional factor 1/2, then after multiplying by the fractional factor 1/2, the integer values 7 and 8 have the same “Round()” value 4, but we cannot reconstruct the original integer values 7 and 8 using the same value 4 in the

Table 1 Prediction filters of update-then-predict lifting wavelets (UPLWs)

N	z^{-k}						
	z^{-3}	z^{-2}	z^{-1}	z^0	z^1	z^2	z^3
1				-1			
3			$\frac{1}{8}$	-1	$-\frac{1}{8}$		
5		$-\frac{3}{128}$	$\frac{11}{64}$	-1	$-\frac{11}{64}$	$\frac{3}{128}$	
7	$\frac{5}{1024}$	$-\frac{11}{256}$	$\frac{201}{1024}$	-1	$-\frac{201}{1024}$	$\frac{11}{256}$	$-\frac{5}{1024}$

synthesis part. That is, the factor 1/2 destroys the perfect reconstruction property of the integer structure of update-then-predict wavelet and makes the structure is not invertible. Therefore, we will discuss how to preserve the perfect reconstruction property of the update-then-predict lifting wavelet and then give the design of the invertible update-then-predict lifting wavelet in Section 3.

3 Invertible update-then-predict integer lifting wavelets with scaling lifting

In this section, the polyphase representation and channel representation of the update-then-predict wavelet in Fig. 3 are given firstly. Secondly, the invertible update-then-predict lifting wavelet is obtained by re-factoring the channel representation into lifting steps. Then the integer structure of the invertible update-then-predict lifting wavelet with scaling lifting are constructed. Finally, the computational complexity is analyzed.

3.1 Channel representation of the update-then-predict wavelet filter bank

The polyphase representation is a particularly convenient tool to build the connection between lifting representation and channel representation [4]. We give the polyphase representation and channel representation of the update-then-predict lifting wavelet filter bank in Figs. 4 and 5, respectively.

The polyphase representation of a filter h is given by

$$\tilde{h}(z) = \tilde{h}_e(z^2) + z^{-1}\tilde{h}_o(z^2)$$

$$h(z) = h_e(z^2) + z^{-1}h_o(z^2)$$

where h_e contains the even coefficients, and h_o contains the odd coefficients:

$$\tilde{h}_e(z) = \sum_k \tilde{h}_{2k}z^{-k} \text{ and } \tilde{h}_o(z) = \sum_k \tilde{h}_{2k+1}z^{-k}$$

$$h_e(z) = \sum_k h_{2k}z^{-k} \text{ and } h_o(z) = \sum_k h_{2k+1}z^{-k}$$

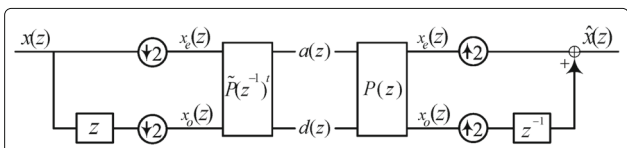


Fig. 4 Polyphase representation of update-then-predict wavelet filter bank

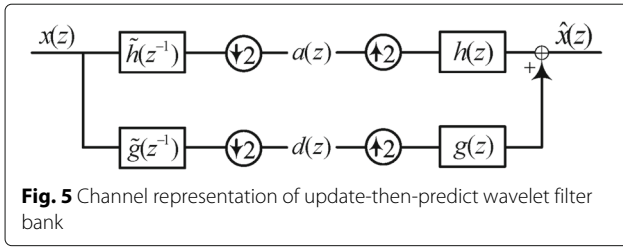


Fig. 5 Channel representation of update-then-predict wavelet filter bank

We assemble the polyphase matrix as

$$\tilde{P}(z) = \begin{bmatrix} \tilde{h}_e(z) & \tilde{g}_e(z) \\ \tilde{h}_o(z) & \tilde{g}_o(z) \end{bmatrix} \text{ and } P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix}$$

According to the polyphase representation (see Fig. 4), the perfect reconstruction condition of wavelet filter bank is given by

$$P(z)\tilde{P}(z^{-1})^t = I \tag{1}$$

The relationship equations between polyphase representation and lifting representation are given by

$$\tilde{P}(z) = \begin{bmatrix} \tilde{h}_e(z) & \tilde{g}_e(z) \\ \tilde{h}_o(z) & \tilde{g}_o(z) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{4}p(z^{-1}) \\ \sqrt{2} \cdot u(z^{-1}) & \frac{\sqrt{2}}{2}(1 + p(z^{-1})u(z^{-1})) \end{bmatrix} \tag{2}$$

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} = \begin{bmatrix} \sqrt{2} + \sqrt{2}p(z)u(z) & -2\sqrt{2} \cdot u(z) \\ -\frac{1}{\sqrt{2}}p(z) & \sqrt{2} \end{bmatrix} \tag{3}$$

Therefore, the relationship equations between channel presentation and lifting representation can be given by

$$\tilde{h}(z) = \tilde{h}_e(z^2) + z^{-1}\tilde{h}_o(z^2) = \sqrt{2} \left(\frac{1}{2} + z^{-1}u(z^{-2}) \right) \tag{4}$$

$$\tilde{g}(z) = \tilde{g}_e(z^2) + z^{-1}\tilde{g}_o(z^2) = \frac{1}{\sqrt{2}} \left(z^{-1} + p(z^{-2}) \cdot \left(\frac{1}{2} + z^{-1}u(z^{-2}) \right) \right) \tag{5}$$

$$h(z) = h_e(z^2) + z^{-1}h_o(z^2) = \frac{1}{\sqrt{2}} (2 - p(z^2) \cdot (z^{-1} - 2u(z^2))) \tag{6}$$

$$g(z) = g_e(z^2) + z^{-1}g_o(z^2) = \sqrt{2} (z^{-1} - 2u(z^2)) \tag{7}$$

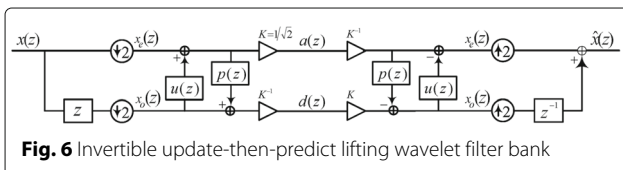


Fig. 6 Invertible update-then-predict lifting wavelet filter bank

Table 2 Prediction filters of invertible update-then-predict lifting wavelet filter bank

N	z^{-k}						
	z^{-3}	z^{-2}	z^{-1}	z^0	z^1	z^2	z^3
1				$-\frac{1}{2}$			
3			$\frac{1}{16}$	$-\frac{1}{2}$	$-\frac{1}{16}$		
5		$-\frac{3}{256}$	$\frac{11}{128}$	$-\frac{1}{2}$	$-\frac{11}{128}$	$\frac{3}{256}$	
7	$\frac{5}{2048}$	$-\frac{11}{512}$	$\frac{201}{2048}$	$-\frac{1}{2}$	$-\frac{201}{2048}$	$\frac{11}{512}$	$-\frac{5}{2048}$

For example, in Table 1, Let $N = 3$, there is

$$p(z) = \frac{1}{8}z^{-1} - 1 - \frac{1}{8}z \tag{8}$$

Substitute $u(z) = \frac{1}{2}$ and Eq. (8) into (4), (5), (6), and (7), we have

$$\tilde{h}(z) = \frac{\sqrt{2}}{2} \cdot (1 + z^{-1}) \tag{9}$$

$$\tilde{g}(z) = \left(\frac{1}{16}z^2 + \frac{1}{16}z - \frac{1}{2} + \frac{1}{2}z^{-1} - \frac{1}{16}z^{-2} - \frac{1}{16}z^{-3} \right) / \sqrt{2} \tag{10}$$

$$h(z) = \left(-\frac{1}{8}z^2 + \frac{1}{8}z + 1 + z^{-1} + \frac{1}{8}z^{-2} - \frac{1}{8}z^{-3} \right) / \sqrt{2} \tag{11}$$

$$g(z) = \sqrt{2} \cdot (-1 + z^{-1}) \tag{12}$$

Therefore, the channel representation of the update-then-predict wavelet filter bank is obtained. In the next section, we will construct the invertible update-then-predict lifting wavelet filter bank by re-factoring the channel representation into lifting steps.

3.2 Re-factoring channel representation into lifting steps

The channel representation of wavelet filter bank can be factored into lifting steps using Euclidean algorithm [4]. In this section, we factor the synthesis low-pass filter $h(z)$ (see Eq. (11)) into lifting steps and then the synthesis polyphase matrix $P^{\text{new}}(z)$ is obtained. Furthermore, the conjugate transpose matrix $\tilde{P}^{\text{new}}(z^{-1})^t$ of analysis polyphase matrix can be given. Therefore, the factor 1/2

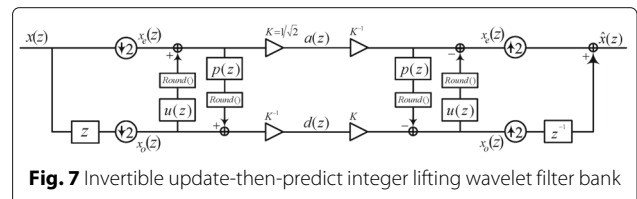


Fig. 7 Invertible update-then-predict integer lifting wavelet filter bank

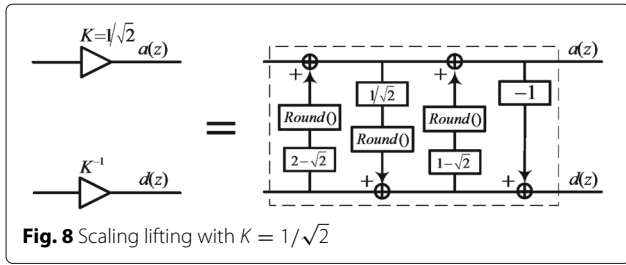


Fig. 8 Scaling lifting with $K = 1/\sqrt{2}$

in Fig. 3 is gone and the invertible update-then-predict lifting wavelet filter bank is built.

The synthesis low-pass filter $h(z)$ in Eq. (11) can be rewritten as

$$\begin{aligned}
 h(z) &= h_e(z^2) + z^{-1}h_o(z^2) \\
 &= \frac{1}{\sqrt{2}} \cdot \left(-\frac{1}{8}z^2 + 1 + \frac{1}{8}z^{-2}\right) + z^{-1} \cdot \frac{1}{\sqrt{2}} \\
 &\quad \times \left(\frac{1}{8}z^2 + 1 - \frac{1}{8}z^{-2}\right) \quad (13)
 \end{aligned}$$

Therefore, we have

$$h_e(z) = \left(-\frac{1}{8}z + 1 + \frac{1}{8}z^{-1}\right) / \sqrt{2} \quad (14)$$

$$h_o(z) = \left(\frac{1}{8}z + 1 - \frac{1}{8}z^{-1}\right) / \sqrt{2} \quad (15)$$

According to Euclidean algorithm, we have

$$\begin{aligned}
 \begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} &= \begin{bmatrix} \left(-\frac{1}{8}z + 1 + \frac{1}{8}z^{-1}\right) / \sqrt{2} \\ \left(\frac{1}{8}z + 1 - \frac{1}{8}z^{-1}\right) / \sqrt{2} \end{bmatrix} \\
 &= \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{16}z + \frac{1}{2} - \frac{1}{16}z^{-1} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix} \quad (16)
 \end{aligned}$$

Observe that

$$\begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & q_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_i(z) & 1 \end{bmatrix} \quad (17)$$

Using the first equation of (17) in case i is odd and the second in case i is even yields:

$$\begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{1}{16}z + \frac{1}{2} - \frac{1}{16}z^{-1} & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix} \quad (18)$$

Therefore

$$\begin{aligned}
 P^{\text{new}}(z) &= \begin{bmatrix} h_e(z) & g_e^{\text{new}}(z) \\ h_o(z) & g_o^{\text{new}}(z) \end{bmatrix} \\
 &= \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{1}{16}z + \frac{1}{2} - \frac{1}{16}z^{-1} & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (19)
 \end{aligned}$$

Then considering Eq. (1), there is

$$\begin{aligned}
 \tilde{P}^{\text{new}}(z^{-1})^t &= P^{\text{new}}(z)^{-1} \\
 &= \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{1}{16}z^{-1} - \frac{1}{2} - \frac{1}{16}z & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad (20)
 \end{aligned}$$

Considering Eq. (20) and Fig. 4, the update filter and prediction filter can be obtained

$$u^{\text{new}}(z) = 1 \quad (21)$$

$$p^{\text{new}}(z) = \frac{1}{16}z^{-1} - \frac{1}{2} - \frac{1}{16}z \quad (22)$$

Therefore, the new update-then-predict lifting wavelet filter bank is given in Fig. 6.

In Fig. 6, the update filter $u(z)$ and prediction filter $p(z)$ are given in Eqs. (21) and (22), respectively. Note that it is a result obtained where $N=3$ in Table 1 (see Eq. (8)). Therefore, considering Table 1 and repeating the same construction as before, the prediction filters of the new invertible update-then-predict lifting wavelets can be given in Table 2.

Comparing Table 2 with Table 1, we know that the slight difference is each value in Table 2 is the half of the corresponding value in Table 1. That is, $p^{\text{new}}(z) = \frac{1}{2}p(z)$, where $p(z)$ is the prediction filter in original update-then-predict lifting wavelets, and $p^{\text{new}}(z)$ is the prediction filter in our invertible update-then-predict lifting wavelets. Another difference between these two lifting structure are $u^{\text{new}}(z) = 2 \times u(z)$ and the factor $1/2$ is omitted in our invertible update-then-predict lifting structure.

Comparing Fig. 6 with Fig. 3, we note that there are some differences between them. First, the factor $1/2$ in Fig. 3 is omitted in Fig. 6. This means the invertibility of the update-then-predict integer lifting wavelet can be

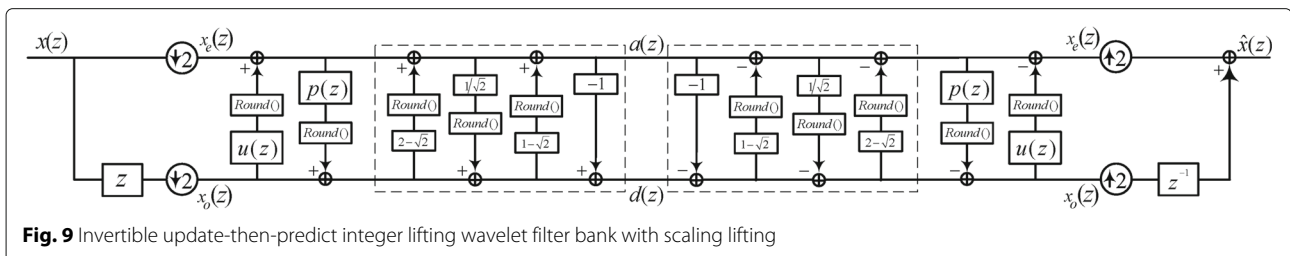


Fig. 9 Invertible update-then-predict integer lifting wavelet filter bank with scaling lifting

guaranteed. Second, the update filter and prediction filter are different. Finally, the scaling factor K is different, $K = \sqrt{2}$ in Fig. 3, but $K = 1 / \sqrt{2}$ in Fig. 6.

Considering Fig. 1, the integer structure of the invertible update-then-predict lifting wavelet bank in Fig. 6 is given as follows (Fig. 7).

Comparing Figs. 7 and 6, we know that just the operations “Round()” are added and followed prediction filter $p(z)$ and update filter $u(z)$. The operations “Round()” ensure the invertible of the prediction step and update step. Note that the above structure is not completely invertible because the scaling factors K and K^{-1} are included in it. Therefore, we will discuss the scaling lifting (focus to K and K^{-1}) in the next section.

3.3 Invertible update-then-predict integer lifting wavelet filter bank with scaling lifting

The invertible lifting wavelet can be built using integer-to-integer method. But as mentioned above, the scaling factors K and K^{-1} can cause an issue for invertible lifting. Daubechies and Sweldens introduced a method (scaling lifting) to factorize the scaling factors (K and K^{-1}) into four lifting steps [4]. The scaling lifting is shown as follows.

$$S(z) = \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 - 1/K \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ K & 1 \end{bmatrix} \begin{bmatrix} 1 & 1/K^2 - 1/K \\ 0 & 1 \end{bmatrix} \quad (23)$$

Let $K = 1 / \sqrt{2}$,

$$S(z) = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 - \sqrt{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1/\sqrt{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 - \sqrt{2} \\ 0 & 1 \end{bmatrix} \quad (24)$$

Where $S(z)$ denotes the matrix of scaling factors. Therefore, according to Eq. (24), the scaling lifting of analysis part which merged with integer-to-integer can be given in the right part of Fig. 8. Similarly, the scaling lifting of

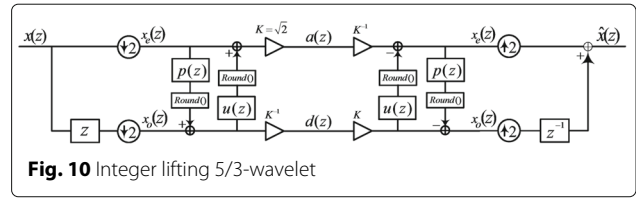


Fig. 10 Integer lifting 5/3-wavelet

synthesis part can be obtained by slipping the signs and reversing the operations.

Now, we can replace the scaling factors K and K^{-1} in Fig. 7 using the right part of Fig. 8. Then the structure of real invertible update-then-predict integer lifting wavelets (IUPIWLs) with scaling lifting can be given in Fig. 9.

In Fig. 9, the update filter $u(z)$ is given in Eq. (21), that is, $u(z) = 1$. The prediction filters $p(z)$ are given in Table 2. The integer lifting and scaling lifting are achieved by using the matrix factoring (see Eq. (24)) and rounding-off operations. The structure in Fig. 9 is perfect invertible, which means the processes from signal $x(z)$ to $a(z)$ and $d(z)$, the process from $a(z)$ and $d(z)$ to the reconstruction signal $\hat{x}(z)$ are all lossless, and the result $\hat{x}(z) = x(z)$ can be obtained. We name the above new update-then-predict wavelet family as invertible update-then-predict integer lifting wavelets (IUPIWLs), and we will do some experiment comparisons between IUPIWLs and the integer lifting structure of 5/3 wavelet, 9/7 wavelet, and iDIT for lossless image compression in Section 4.

3.4 Computational complexity

In this section, we discuss the computational complexity of IUPIWLs, integer lifting 5/3-wavelet, integer lifting 9/7-wavelet, and iDIT based on the lossless image compression. The unit we use to analyze the computation complexity is the cost, measured in number of multiplications, additions, and roundings. Besides, the scaling lifting (see Fig. 8) step can give four multiplications, four additions, and three rounds. For image compression, we suppose the size of image is $m \times n$, where m is the height of the image and n is the width of the image.

Table 3 Cost of analysis part (IUPIWLs)

Item	No. of multiplication	No. of addition	No. of rounding	Sum
$u(z)$	1	0	0	1
Round after $u(z)$	0	0	1	1
+ after $u(z)$	0	1	0	1
$p(z)$	N	N-1	0	2N-1
Round after $p(z)$	0	0	1	1
+ after $p(z)$	0	1	0	1
Scaling lifting	4	4	3	11
Sum	N+5	N+5	5	2N+15

Table 4 Cost of analysis part (integer lifting 5/3-wavelet)

Item	No. of multiplication	No. of addition	No. of rounding	Sum
$p(z)$	2	1	0	3
Round after $p(z)$	0	0	1	1
+ after $p(z)$	0	1	0	1
$u(z)$	2	1	0	3
Round after $u(z)$	0	0	1	1
+ after $u(z)$	0	1	0	1
Scaling lifting	4	4	3	11
Sum	8	8	5	21

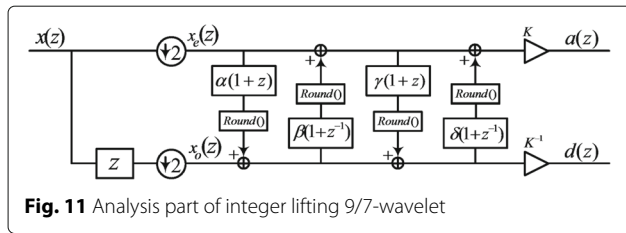


Fig. 11 Analysis part of integer lifting 9/7-wavelet

For IUPIWLs shown in Fig. 9, we give the cost of its analysis part in Table 3.

Also considering the synthesis part of IUPIWLs, the size of image, the row and column lifting, we obtain the number of multiplications, additions, and rounding for IUPIWLs is $2 \times (2N + 15) \times m \times n$.

The structure of integer lifting 5/3-wavelet is shown as follows (Fig. 10)

For integer lifting 5/3-wavelet, its prediction filter is $p(z) = -\frac{1}{2} - \frac{1}{2}z$ and its update filter is $u(z) = \frac{1}{4} + \frac{1}{4}z^{-1}$. Therefore, we give the cost of its analysis part in Table 4.

Also considering the synthesis part of integer lifting 5/3-wavelet, the size of image, the row and column lifting, we obtain the number of multiplications, additions, and rounding for integer lifting 5/3-wavelet is $2 \times 21 \times m \times n$.

The structure of integer lifting 9/7-wavelet is shown as follows (Fig. 11)

We give the cost of its analysis part in Table 5.

Also considering the synthesis part of lifting 9/7-wavelet, the size of image, the row and column lifting, we obtain the number of multiplications, additions, and rounding for integer lifting 9/7-wavelet is $2 \times 31 \times m \times n$.

For the iDTT in [11], for each 8×8 block, the number of multiplications is 8×8 and the number of additions is

Table 5 Cost of analysis part (integer lifting 9/7-wavelet)

Item	No. of multiplication	No. of addition	No. of rounding	Sum
$\alpha(1+z)$	2	1	0	3
Round after $\alpha(1+z)$	0	0	1	1
+ after $\alpha(1+z)$	0	1	0	1
$\beta(1+z)$	2	1	0	3
Round after $\beta(1+z)$	0	0	1	1
after $\beta(1+z)$	0	1	0	1
$\gamma(1+z)$	2	1	0	3
Round after $\gamma(1+z)$	0	0	1	1
+ after $\gamma(1+z)$	0	1	0	1
$\delta(1+z)$	2	1	0	3
Round after $\delta(1+z)$	0	0	1	1
+ after $\delta(1+z)$	0	1	0	1
Scaling lifting	4	4	3	11
Sum	12	12	7	31

Table 6 Cost of IUPIWLs, integer lifting 5/3, integer lifting 9/7, and iDTT

Wavelets	Cost (multi., add., and roundings)
IUPIWLs	$2 \times (2N + 15) \times m \times n$
Integer lifting 5/3	$2 \times 21 \times m \times n$
Integer lifting 9/7	$2 \times 31 \times m \times n$
iDTT	$4 \times m \times n$

$(8 \times 8 - 1)$. Therefore, the number of multiplications and additions for iDTT is shown as follows.

$$2 \times (8 \times 8 + 8 \times 8 - 1) \times \frac{m}{8} \times \frac{n}{8} \approx 4 \times m \times n$$

Therefore, we can summarize the cost of IUPIWLs, integer lifting 5/3-wavelet, integer lifting 9/7-wavelet, and iDTT for lossless image compression in Table 6.

Now we test the time cost of above wavelet filter banks using the 512×512 gray-scale Barbara image. Here we just consider the sum of time cost of analysis part and synthesis part of lifting wavelet filter banks, therefore, the time cost can be given in Table 7.

4 Experiments

In this section, the bit-rates of image lossless compression are compared between the integer lifting structure of 5/3 wavelet, 9/7 wavelet, iDTT, and the invertible update-then-predict integer lifting wavelets (IUPIWLs). For the lossless image compression, the bit-rates (bit/pixel) are important. The lower bit-rate means higher compression ratio. The calculation of bit-rates for lossless image compression is given as follows.

$$bitRates = \frac{\text{total number of bits in final code file}}{\text{total number of pixels in original image}}$$

For example, for a 512×512 8-bit gray-scale image, letting the “final code file” equal to the “original image”, then the value of “bitRates” is “8”. It means that the encoding for each pixel of the original image consists of 8-bit. Obviously, the small value of “bitRates” means the less encode bits for each pixel of original image.

In this experiment, 18 512×512 8-bit gray-scale images are chosen and EBCOT coding algorithm [24] is employed to test the integer lifting structure of 5/3 wavelet, 9/7

Table 7 Time cost of IUPIWLs, integer lifting 5/3, integer lifting 9/7, and iDTT

Wavelets	Time cost (unit: ms)
IUPIWLs	483 (N=5)
Integer lifting 5/3	469
Integer lifting 9/7	516
iDTT	47

Table 8 Bit-rates (bit/pixel) for lossless image compression (18 images)

Image	5/3-wavelet	9/7-wavelet	iDTT	IUPILW-(1, 1)	IUPILW-(1, 3)	IUPILW-(1, 5)
Baboon	6.029224	6.065701	6.053941	6.217995	6.040749	5.986946
Barbara	5.040646	5.079731	5.186513	5.468422	5.100750	5.037720
Bike	5.690926	5.806850	5.622491	5.440147	5.614979	5.626175
Bridge	5.922947	5.956379	5.951451	6.127350	5.930901	5.877102
Couple	5.135845	5.166874	5.159809	5.358124	5.142929	5.056225
Crowd	4.457260	4.482311	4.541467	5.007710	4.518784	4.421471
Elaine	5.208218	5.285564	5.266576	5.374374	5.202930	5.139389
Goldhill	5.104885	5.141258	5.133493	5.332794	5.114780	5.034573
Lake	5.385849	5.409073	5.403192	5.592735	5.395832	5.316547
Lena	4.532589	4.611752	4.567688	4.858097	4.558022	4.514534
Man	4.909214	4.934772	4.942688	5.237564	4.936646	4.869335
Milkdrop	4.106567	4.187325	4.148641	4.324787	4.112820	4.023701
Peppers	4.872906	4.954685	4.902747	5.090134	4.877186	4.823788
Plane	4.323769	4.370487	4.367616	4.650158	4.347626	4.250137
Portofino	5.170605	5.233532	5.213912	5.324055	5.172428	5.094902
Woman1	5.010029	5.086510	5.045542	5.231766	5.014275	5.101509
Woman2	3.596012	3.690968	3.664954	4.023357	3.628143	3.543446
Zelda	4.255531	4.357315	4.312983	4.630722	4.276470	4.230957

wavelet, iDTT, and IUPILWs. The bit-rates of image compression can be given using integer lifting structure of 5/3 wavelet, 9/7 wavelet, iDTT, and the IUPILWs proposed in Section 3. The results are shown in Table 8.

Table 8 shows the bit-rates using the integer lifting structure of 5/3 wavelet, 9/7 wavelet, iDTT, and IUPILWs, respectively. Compared with the integer lifting structure of 5/3 wavelet, 9/7 wavelet, and iDTT, and IUPILW-(1, 5) gets the lowest bit-rates, which means IUPILW-(1, 5) has the best performance for lossless image compression.

The test also have been done for the eight images of the ISO 12640-1 corpus (gray scaled, size 2048×2560, N1-Portrait, N2-Cafeteria, N3-Fruit Basket, N4-Wine and Tableware, N5-Bicycle, N6-Orchid, N7-Musicians,

N8-Candle). The results are shown in Table 9. In Table 9, we observed that 5/3 wavelet has the bit-rates between IUPILW-(1, 3) and IUPILW-(1, 5), 9/7 wavelet, and iDTT have the bit-rates between IUPILW-(1, 1) and IUPILW-(1, 3). Obviously, IUPILW-(1, 5) gets the lowest bit-rates.

One of the reasons why the IUPILW-(1, 5) has the better performance than the predict-then-update lifting wavelet may be the update-then-predict structure can reduce the errors during the wavelet decomposition. Update-first means the approximate coefficients will be obtained firstly during each decomposition-level, and then the approximate coefficients of next decomposition-level will be obtained using the approximate coefficients of the current decomposition level. It means that the errors will not

Table 9 Bit-rates (bit/pixel) for lossless image compression (corpus ISO 12640-1)

Image	5/3-wavelet	9/7-wavelet	iDTT	IUPILW-(1, 1)	IUPILW-(1, 3)	IUPILW-(1, 5)
N1	4.424217	4.493907	4.462199	4.656300	4.444999	4.346868
N2	5.273573	5.308329	5.302399	5.537801	5.293032	5.188928
N3	4.291140	4.393228	4.339916	4.478541	4.298331	4.214809
N4	4.606598	4.703839	4.659298	4.734381	4.603954	4.519682
N5	4.591891	4.645296	4.624030	4.811606	4.603206	4.509288
N6	3.681239	3.807993	3.719131	3.817874	3.678956	3.586124
N7	5.473100	5.574008	5.504893	5.563710	5.465703	5.400206
N8	5.751989	5.808060	5.770165	5.924475	5.756146	5.665746

spread between the approximate coefficients. However, for the predict-first lifting structure, the detail coefficients must be get using the approximate coefficients of the upper level, then computing the approximate coefficients of the current level using these detail coefficients. Therefore, for the predict-first structure, the errors will spread between detail coefficients and approximate of the same decomposition level.

5 Conclusions

A new update-then-predict integer lifting wavelet family for lossless image compression is built and named in this paper. It is a perfect invertible update-then-predict structure and compared with the integer lifting structure of 5/3 wavelet, 9/7-wavelet, and iDTT, IUPILW-(1, 5) results in the lower bit-rates for lossless image compression.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Nos. 61300098, 61303080), the Fundamental Research Funds for the Central Universities (No. DL13BB02), and Self-Planned Task (No. SKLRS201407B) of State Key Laboratory of Robotics and System (HIT).

Authors' contributions

DC built the theoretical framework of this paper. YL and HZ drew all the figures and provide funding support. DC, YL, and WG finished the experimental section in this paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

¹School of Life Science and Technology, Harbin Institute of Technology, No. 92 Dazhi West Street, 150001 Harbin, China. ²School of Information and Computer Engineering, North-East Forestry University, No. 26 Hexing Street, 150040 Harbin, China. ³Software School of Xiamen University, 361005 Xiamen, China.

Received: 6 July 2016 Accepted: 30 December 2016

Published online: 14 January 2017

References

- M Vetterli, C Herley, Wavelets and filter banks: theory and design. *IEEE Trans. Signal Process.* **40**, 2207–2232 (1992)
- A Ellmauthaler, CL Pagliari, EAB da Silva, Multiscale image fusion using the undecimated wavelet transform with spectral factorization and nonorthogonal filter banks. *IEEE Trans. Image Process.* **22**, 1005–1017 (2013)
- A Rehman, Y Gao, J Wang, Z Wang, Image classification based on complex wavelet structural similarity. *Signal Process. Image Commun.* **28**, 984–992 (2013)
- I Daubechies, W Sweldens, Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.* **4**, 247–269 (1998)
- X Yang, Y Shi, L Chen, Z Quan, The lifting scheme for wavelet bi-frames: Theory, structure, and algorithm. *IEEE Trans. Image Process.* **19**, 612–623 (2010)
- M Vrankic, D Sersic, V Susic, Adaptive 2-D wavelet transform based on the lifting scheme with preserved vanishing moments. *IEEE Trans. Image Process.* **19**, 1987–2004 (2010)
- NCF Tse, LL Lai, Wavelet-based algorithm for signal analysis. *EURASIP J. Adv. Signal Process.* **1**, 1–10 (2007)
- T Suzuki, M Ikehara, TQ Nguyen, Generalized block-lifting factorization of M-channel biorthogonal filter banks for lossy-to-lossless image coding. *IEEE Trans. Image Process.* **21**, 3220–3228 (2012)
- Z Zuo, X Lan, L Deng, S Yao, X Wang, An improved medical image compression technique with lossless region of interest. *Optik* **126**, 2825–2831 (2015)
- L Zhang, B Qiu, Edge-preserving image compression using adaptive lifting wavelet transform. *Int. J. Electron.* **102**(7), 1190–1203 (2015)
- B Xiao, G Lu, Y Zhang, W Li, G Wang, Lossless image compression based on integer Discrete Tchebichef Transform. *Neurocomputing* **214**, 587–593 (2016)
- X Wang, J Liang, M Wang, On-line fast palmprint identification based on adaptive lifting wavelet scheme. *Knowl.-Based Syst.* **42**, 68–73 (2013)
- CH Hsia, JM Guo, Efficient modified directional lifting-based discrete wavelet transform for moving object detection. *Signal Process.* **96**, 138–152 (2014)
- C Christopoulos, J Askelöf, M Larsson, Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard. *IEEE Signal Process. Lett.* **7**, 247–249 (2000)
- C Christopoulos, A Skodras, T Ebrahimi, The JPEG2000 still image coding system: An overview. *IEEE Trans. Consum. Electron.* **46**, 1103–1127 (2000)
- LJ Rodríguez, FA Llinàs, MW Marcellin, FAST rate allocation for JPEG2000 video transmission over time-varying channels. *IEEE Trans. Multimed.* **15**, 15–26 (2013)
- H Oh, A Bilgin, MW Marcellin, Visually lossless encoding for JPEG2000. *IEEE Trans. Image Process.* **22**, 189–201 (2013)
- LJ Rodríguez, FA Llinàs, MW Marcellin, Visually lossless strategies to decode and transmit JPEG2000 imagery. *IEEE Signal Process. Lett.* **21**, 35–38 (2014)
- HJAM Heijmans, B Pesque-Popescu, G Piella, Building nonredundant adaptive wavelets by update lifting. *Appl. Comput. Harmon. Anal.* **18**, 252–281 (2005)
- G Piella, B Pesque-Popescu, Combining seminorms in adaptive lifting schemes and applications to image analysis and compression. *J. Math. Imaging Vision.* **25**, 203–226 (2006)
- RL Claypoole, GM Davis, W Sweldens, RG Baraniuk, Nonlinear wavelet transforms for image coding via lifting. *IEEE Trans. Image Process.* **12**, 1449–1459 (2003)
- DL Donoho, *Smooth wavelet decompositions with blocky coefficient kernels. in Recent Advances in Wavelet Analysis.* (Academic Press, New York, 1993), pp. 259–308
- AR Calderbank, I Daubechies, W Sweldens, BL Yeo, Wavelet transforms that map integers to integers. *Appl. Comput. Harmon. Anal.* **5**, 332–369 (1998)
- D Taubman, High performance scalable image compression with EBCOT. *IEEE Trans. Image Process.* **9**, 1158–1170 (2000)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com