Integer Time-Domain Pre- and Post-Filters for Low-Complexity Extension of JPEG Standard

Shohei Kubo*, Taizo Suzuki†, and Hiroyuki Kudo†
*Department of Computer Science, University of Tsukuba, Japan,
E-mail: kubo@imagelab.cs.tsukuba.ac.jp
† Faculty of Engineering, Information and Systems, University of Tsukuba, Japan,
E-mail: {taizo, kudo}@cs.tsukuba.ac.jp

***Abstract*—Time-domain lapped transform (TDLT) is a family of lapped transforms (LTs) which reduce blocking artifacts occurred by the DCT algorithm. Also, we can directly apply it to the existing JPEG framework because it is constructed by the DCT with time-domain pre- and post-filters. It obviously has higher complexity than an ordinary DCT. We present an integer approximated TDLT (IntTDLT) which can be easily implemented with only adders/shifters, i.e., no floating-point multipliers, and fewer step processing, while preserving the characteristics such as allowing the use of symmetric extension and structural one degree of regularity (1-regularity). As a result, it performs good coding, especially at a low bitrate, in spite of the low-complexity.**

I. INTRODUCTION

The well-known JPEG image compression (coding) standard [1] has been widely used for 20 years. However, it generates annoying error, i.e, blocking artifacts, at a low bitrate image coding. The problem is caused due to the block-based DCT [2] which excludes the neighboring block correlation. As for video coding such as H.26x series [3][4], integer approximated DCTs [5][6] are commonly used. The transforms not only have simple implementation but also avoid the mismatch problems in the decoder.

JPEG XR [7] has been developed as the next generation standard which provides high image quality even in a low bitrate coding. However, JPEG is still used as a de facto standard because of two factors: i) JPEG coder has already been used in the widespread application, ii) JPEG XR requires higher complexity than JPEG. JPEG XR adopts a class of time-domain lapped transform (TDLT) [8]. TDLT is a family of lapped transforms (LTs), which reduce blocking artifacts, and can be directly applied to the existing JPEG framework in case of block size 8 because it is constructed by the DCT with time-domain pre- and post-filters. Although the traditional TDLT consisting of lifting structure [9], which is called lifting-based TDLT (LiftTDLT) in this paper, cannot achieve effective butterfly-style processing which has fewer step processing.

We propose an integer approximated TDLT (IntTDLT) with only adders/shifters, i.e., no floating-point multipliers, and fewer step processing for image coding. The IntTDLT can adopt symmetric extension [10] to avoid boundary error even if the coefficients are approximated by integers. Similarly, it satisfies one degree of regularity (1-regularity) [11] which is an important property for image coding in filter bank theory. In addition, since all coefficients in our transform consist of only integers, we can always obtain the identical reconstructed data even under any circumstance of decoder. Moreover, it can achieve effective butterfly-style processing for low-complexity unlike LiftTDLT. We show the validity of the proposed IntTDLT in its application to JPEG standard as an example.

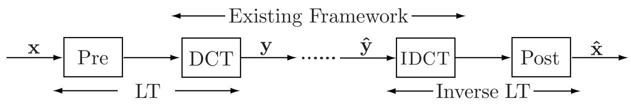

Fig. 1. Processing flow of TDLT.

Notations: We use bold-faced lowercase and uppercase characters to denote vectors and matrices, respectively. $\mathbf{I}$, $\mathbf{J}$, $\cdot^{-1}$, and $|\cdot|$ indicate an identity matrix, a reversal identity matrix, an inverse matrix, and the determinant of a matrix, respectively.

II. REVIEW

A. Time-Domain Lapped Transform (TDLT)

TDLT [8] is a family of LTs and uses the time-domain pre-filter of DCT inputs and the post-filter of IDCT outputs. The pre- and post-filters are outside the existing framework as shown in Fig. 1. The TDLT improves coding performance while achieving standard-compliance with minimal software/hardware modifications. The analysis polyphase matrix is written as

$$\mathbf{E}(z) = \mathbf{C}_M^{II} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & z^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{P}, \tag{1}$$

where

$$\mathbf{P} = \frac{1}{2} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \tag{2}$$

and $\mathbf{C}_M^{II}$ is an $M \times M$ type-II DCT matrix. The $M/2 \times M/2$ nonsingular matrix $\mathbf{V}$ holds all of the degrees of freedom in this structure which controls the properties of pre-/post-filtering.[1] The coefficients are determined by using cost function such as coding gain and stopband attenuation. Also, the

[1]If $\mathbf{V}$ is a paraunitary matrix, it is decomposed into $M(M-1)/2$ rotation matrices $\mathbf{Q}_k$ $(k = 1, 2, \cdots, M(M-1)/2)$.

978-616-361-823-8 © 2014 APSIPA

TDLT in Eq. (1) is represented by lifting structures to obtain a transform with higher energy compaction capability and faster implementation. In this paper, let LiftTDLT be TDLT-III whose pre-filter $\mathbf{P}$ is shown in Fig. 2.

B. Integer Transform

By approximating the floating-point multipliers of a transform by integers, the multipliers can be replaced by several different shifters and sum of them (adders) [12]. The total number of adders and shifters is evaluated by using the binary representation of the integer multiplier. For example, the multiplier by $5 = (101)_2$ can be implemented by one adder and one shifter. Thus, the arithmetic complexity and dynamic range of the data are significantly reduced compared with floating-point multipliers. As a result, it achieves the simpler circuit and the faster implementation.

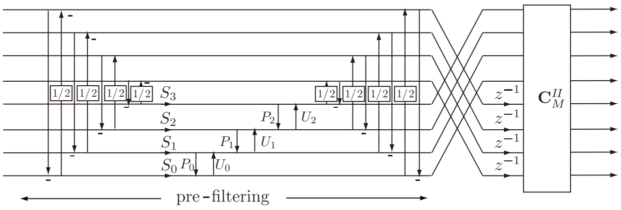

Fig. 2. Lattice structure of LiftTDLT.

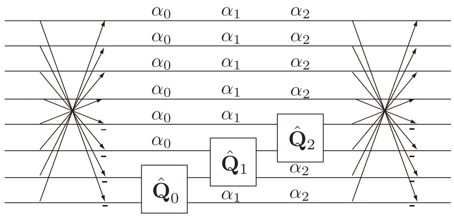

Fig. 3. Pre-filter of forward IntTDLT.

III. Integer Approximated TDLT (IntTDLT)

The purpose of this paper is to reduce blocking artifacts with low-complexity as much as possible under the integer approximation. If the transform coefficients are not integers but floating-point values, we cannot obtain the lossless image due to the quantization. Also, if the floating-point representations of coefficients are different between decoders, we cannot obtain same decoded images. With this in mind, the proposed IntTDLT is designed such that all the coefficients consist of only integers.

A. Integer Approximation of $\mathbf{V}$ *for Forward Transform*

We derive an integer approximation of $\mathbf{V}$ and explain its characteristics in this section. We define the approximation as (Fig. 3)

$$\hat{\mathbf{Q}}_k = \mathrm{round}\{\alpha_k \mathbf{Q}_k\} \quad \text{for } k = 1, 2, ..., (M-2)/2 \tag{3}$$

to convert the floating-point coefficients of a 2×2 arbitrary matrix $\mathbf{Q}_k$ into integers by a scaling factor α_k. Additionally, we eliminate the redundancy using easy matrix manipulations as

$$\begin{bmatrix} ak_1 & bk_2 \\ ck_1 & dk_2 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} ak_1 & bk_1 \\ ck_2 & dk_2 \end{bmatrix} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{5}$$

where a, b, c, d, k_1, and k_2 are arbitrary integers. If scaling factors in common appear in all passes, they are removed for low-complexity.

As an example, we refer to Table V in [8] and set $P_0 = 0$, $P_1 = -1/4$, $P_2 = -1/2$, $U_0 = 1/4$, $U_1 = 1/2$, $U_2 = 3/4$, $S_0 = S_1 = S_2 = S_3 = 1$ in Fig. 2. The parameters achieve so low-complexity. The 2×2 matrix $\mathbf{Q}_k$ $(k = 0, 1, 2)$ of the pre-filter is rewritten as

$$\begin{aligned} \mathbf{Q}_0 &= \begin{bmatrix} 1 & U_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ P_0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1/4 \\ 0 & 1 \end{bmatrix} \\ \mathbf{Q}_1 &= \begin{bmatrix} 1 & U_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ P_1 & 1 \end{bmatrix} = \begin{bmatrix} 7/8 & 1/2 \\ -1/4 & 1 \end{bmatrix} \\ \mathbf{Q}_2 &= \begin{bmatrix} 1 & U_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ P_2 & 1 \end{bmatrix} = \begin{bmatrix} 5/8 & 3/4 \\ -1/2 & 1 \end{bmatrix}. \end{aligned} \tag{6}$$

To achieve our purpose discussed in the beginning of this section, we set $\alpha_k = 2$, which is a simple scaling factor, in Eq. (3). This approach leads $\mathbf{Q}_k$ to the new matrix $\hat{\mathbf{Q}}_k$ as

$$\hat{\mathbf{Q}}_0 = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}, \ \hat{\mathbf{Q}}_1 = \begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}, \ \hat{\mathbf{Q}}_2 = \begin{bmatrix} 1 & 2 \\ -1 & 2 \end{bmatrix}. \tag{7}$$

Next, we eliminate the redundancy of the structure by using easy matrix manipulations as shown in Eq. (4) and (5). The scaling factor 2 is extracted from $\hat{\mathbf{Q}}_2$ as

$$\hat{\mathbf{Q}}_2 = \widetilde{\mathbf{Q}}_2 \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}. \tag{8}$$

Similarly, the scaling factor 2 extracted in Eq. (8) is moved into the next matrix $\hat{\mathbf{Q}}_1$ as

$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \hat{\mathbf{Q}}_1 = \begin{bmatrix} 4 & 2 \\ -1 & 2 \end{bmatrix}. \tag{9}$$

The same processing as Eq. (8) and (9) is repeatedly performed until $\hat{\mathbf{Q}}_0$. Finally, the scaling factors in common appeared in all passes are removed. This approach leads $\hat{\mathbf{Q}}_k$ to the new matrix $\widetilde{\mathbf{Q}}_k$ as

$$\widetilde{\mathbf{Q}}_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \ \widetilde{\mathbf{Q}}_1 = \begin{bmatrix} 4 & 1 \\ -1 & 1 \end{bmatrix}, \ \widetilde{\mathbf{Q}}_2 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}. \tag{10}$$

The structure allows a simple design and has low-complexity. The overall view of the final structure is shown in Fig. 4. The transform scaling factor $1/8$ to normalize the transformed signals is implemented on the phase before DCT.

B. Integer Approximation of $\mathbf{V}^{-1}$ for Inverse Transform

In the decoder, we should use just integer coefficients because we have the problem that reconstructed images vary due to the differences of decoders. If all coefficients of the inverse transform are integers, it is a complete system which does not depend on the decoder. Thus, we define $\widetilde{\mathbf{Q}}_k^{-1}$ as

$$\widetilde{\mathbf{Q}}_k^{-1} \triangleq |\widetilde{\mathbf{Q}}_k| \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}, \tag{11}$$

where a, b, c, and d are arbitrary integers and $|\widetilde{\mathbf{Q}}_k| \neq 0$.

For an example in Eq. (10), the matrices are defined as

$$\widetilde{\mathbf{Q}}_0^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \widetilde{\mathbf{Q}}_1^{-1} = \begin{bmatrix} 1 & -1 \\ 1 & 4 \end{bmatrix}, \widetilde{\mathbf{Q}}_2^{-1} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}. \tag{12}$$

C. Adjustment of $\widetilde{\mathbf{Q}}_k$ for Low-Complexity

If $|\widetilde{\mathbf{Q}}_k|$ is an odd number, we must introduce new scaling factors to all passes in the post-filter to satisfy complete integer approximation. This means that high-complexity is yielded in the decoder. By adjusting the determinant of 2×2 matrix $\widetilde{\mathbf{Q}}_k$ in the forward transform to the power-of-two, the IntTDLT can achieve lower complexity.

As an example, since $|\widetilde{\mathbf{Q}}_1| = 5$ which is an odd number, $\widetilde{\mathbf{Q}}_1$ and the inverse are replaced by

$$\widetilde{\mathbf{Q}}_1 = \begin{bmatrix} 3 & 1 \\ -1 & 1 \end{bmatrix}, \ \widetilde{\mathbf{Q}}_1^{-1} = \begin{bmatrix} 1 & -1 \\ 1 & 3 \end{bmatrix}. \tag{13}$$

The new post-filter is shown in Fig. 5.

D. Application of Symmetric Extension and Structural 1-Regularity

TDLT needs no extra signals at the signal boundaries by applying the symmetric extension [10]. Let $\mathbf{x}_{M/2}$ and $\mathbf{J}\mathbf{x}_{M/2}$ be the $M/2 \times 1$ input vector signal and the reflected vector signal of $\mathbf{x}_{M/2}$, respectively. The pre-filter $\mathbf{P}$ implements the signals as follows:

$$\begin{aligned}\mathbf{P}\begin{bmatrix} \mathbf{J}\mathbf{x}_{M/2} \\ \mathbf{x}_{M/2} \end{bmatrix} &= \frac{1}{2}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{J}\mathbf{x}_{M/2} \\ \mathbf{x}_{M/2} \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix}\begin{bmatrix} 2\mathbf{x}_{M/2} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{J}\mathbf{x}_{M/2} \\ \mathbf{x}_{M/2} \end{bmatrix}. \end{aligned} \tag{14}$$

This means that pre- and post-processing at the signal boundaries can be skipped even if the matrix $\mathbf{V}$ is any matrix.

On the other hand, DCT satisfies 1-regularity condition [11] as

$$\mathbf{C}_M^{II}\mathbf{1}_M = \mathbf{C}_M^{II}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{M} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{15}$$

Since the pre-filter $\mathbf{P}$ in TDLT is a mirror mapping operator as in Eq. (14), it can just output $\mathbf{1}_M$ when the signal $\mathbf{1}_M$ is input. Thus, it satisfies structural 1-regularity condition even if the matrix $\mathbf{V}$ is any matrix.

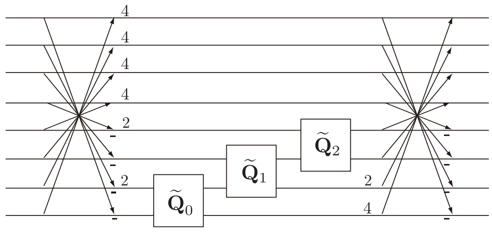

Fig. 4. Pre-filter of forward IntTDLT ($\alpha_k = 2$).

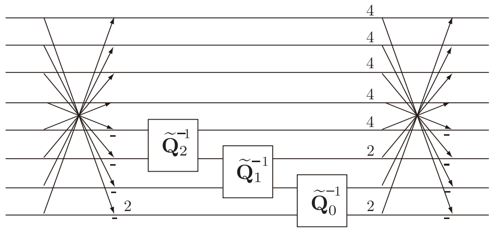

Fig. 5. Post-filter of inverse IntTDLT ($\alpha_k = 2$).

IV. Experimental Results

In this section, we compare the complexity and image coding performance of DCT in JPEG, LiftTDLT, and IntTDLT.

A. Complexity

Table I shows the comparison of the arithmetic complexity for pre-/post-filters of LiftTDLT and IntTDLT. The IntTDLT has the same number of adders and the fewer number of shifters than them of LiftTDLT. Moreover, the numbers of processing steps are shown in Table I. The pre-filter of LiftTDLT needs nine step processing, whereas that of IntTDLT needs only five step processing because the proposed method can be implemented with butterfly-style operation. This means that IntTDLT can be implemented faster than LiftTDLT.

B. Image Coding Performance

To evaluate coding performance fairly, we used structural similarity (SSIM) [13]. Peak signal-to-noise ratio (PSNR) is well-known as an objective assessment, which is used most widely, but it is unsuitable for perceptual visual quality, especially concerning blocking artifacts [14]. SSIM is an alternative complementary framework for quality assessment based on the degradation of structural information and is closer to human subjective evaluation than PSNR. The dynamic range of SSIM is [0,1] and the lager SSIM values signify the better visual quality. We used the MATLAB code at [15]. To consider a fair and practical situation, we used a JPEG framework after adjusting the transformed coefficients to the 8-bit range [0, 255]. In Fig. 6, IntTDLT and LiftTDLT showed better coding performance compared with DCT in JPEG. And IntTDLT was comparable to LiftTDLT though the coefficients are forcibly approximated by integers. Also, as demonstrated in Fig. 7, blocking artifacts were reduced and edges and textures were preserved better than those of the DCT.

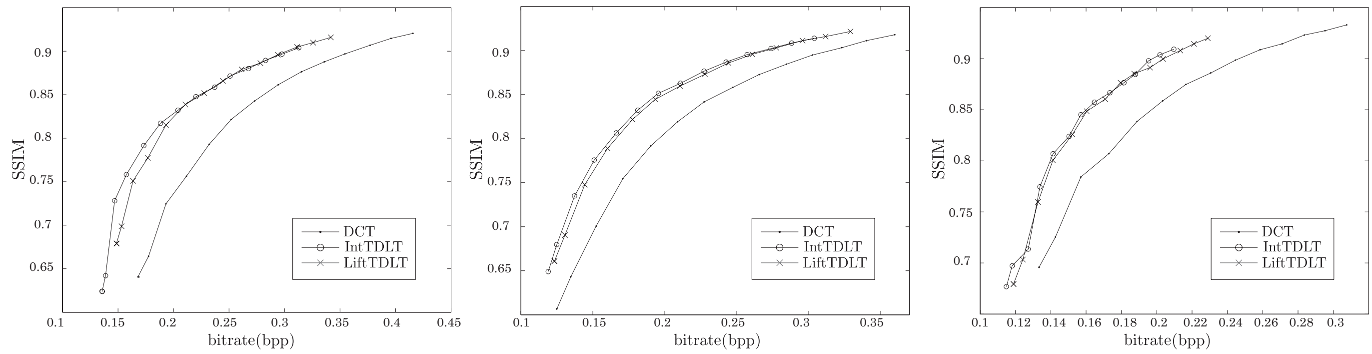

Fig. 6. Comparison of SSIM with low bitrate : (left) Barbara, (middle) Goldhill, (right) Lena.

TABLE I
COMPARISON OF NUMBERS OF ADDERS, SHIFTERS, AND STEPS AMONG LIFTTDLT AND INTTDLT

	Forward Transform		Inverse Transform	
	LiftTDLT	IntTDLT	LiftTDLT	IntTDLT
adders	22	22	22	22
multipliers	0	0	0	0
shifters	13	9	13	9
steps	9	5	9	5

V. CONCLUSIONS

We have presented the theory, design, and implementation of the IntTDLT. It achieved faster implementations with only adders/shifters and fewer step processing while preserving the nice characteristics such as allowing the use of symmetric extension and structural 1-regularity. As a result, we showed good performance as an extension of JPEG standard. Especially, it reduced blocking artifacts occurred by JPEG in a low bitrate situation.

ACKNOWLEDGMENT

This work was supported by a JSPS Grant-in-Aid for Young Scientists (B), Grant Number 25820152.

REFERENCES

[1] W. Pennebaker and J. Mitchell, *JPEG, Still Image Data Compression Standard*, NY: Van Nostrand, 1993.
[2] K. R. Rao and P. Yip, *Discrete Cosine Transform Algorithms*, Academic Press, 1990.
[3] T. Wiegand, G. J. Sullivan, S. Member, G. Bjøntegaardntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.
[4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[5] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, July 2003.
[6] B.K. Mohanty P.K. Meher, S. Y. Park and K. S. Lim, "Efficient integer DCT architectures for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 168–178, Jan. 2014.
[7] F. Dufaux, G. J. Sullivan, and T. Ebrahimi, "The JPEG XR image coding standard," *IEEE Signal Process. Mag.*, vol. 26, no. 6, pp. 195–199, 204, Nov. 2009.
[8] T. D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre- and post-filtering," *IEEE Trans. Signal Process.*, vol. 6, no. 6, pp. 1557–1571, June 2003.
[9] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Proc. of SPIE San Diego*. CA, July 1995, vol. 2569, pp. 1–12.
[10] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Englewood Cliffs, NJ: Prentice-Hall, 1995.
[11] W. Dai and T. D. Tran, "Regularity-constrained pre- and post-filtering for block DCT-based systems," *IEEE Trans. Image Process.*, vol. 51, no. 10, pp. 2568–2581, Oct. 2003.
[12] Y. J. Chen, S. Oraintara, T. D. Tran, K. Amaratunga, and T. Q. Nguyen, "Multiplierless approximation of transforms with adder constraint," *IEEE Trans. Signal Process.*, vol. 9, no. 11, pp. 344–347, Nov. 2002.
[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[14] B. Girod, *What's wrong with mean squared error*, Digital Images and Human Vision, A. B. Watson, Ed. Cambridge, MA: MIT Press, 1993, pp. 207-220.
[15] "The SSIM index for image quality assessment," [Online]. Available: https://ece.uwaterloo.ca/ z70wang/research/ssim/.

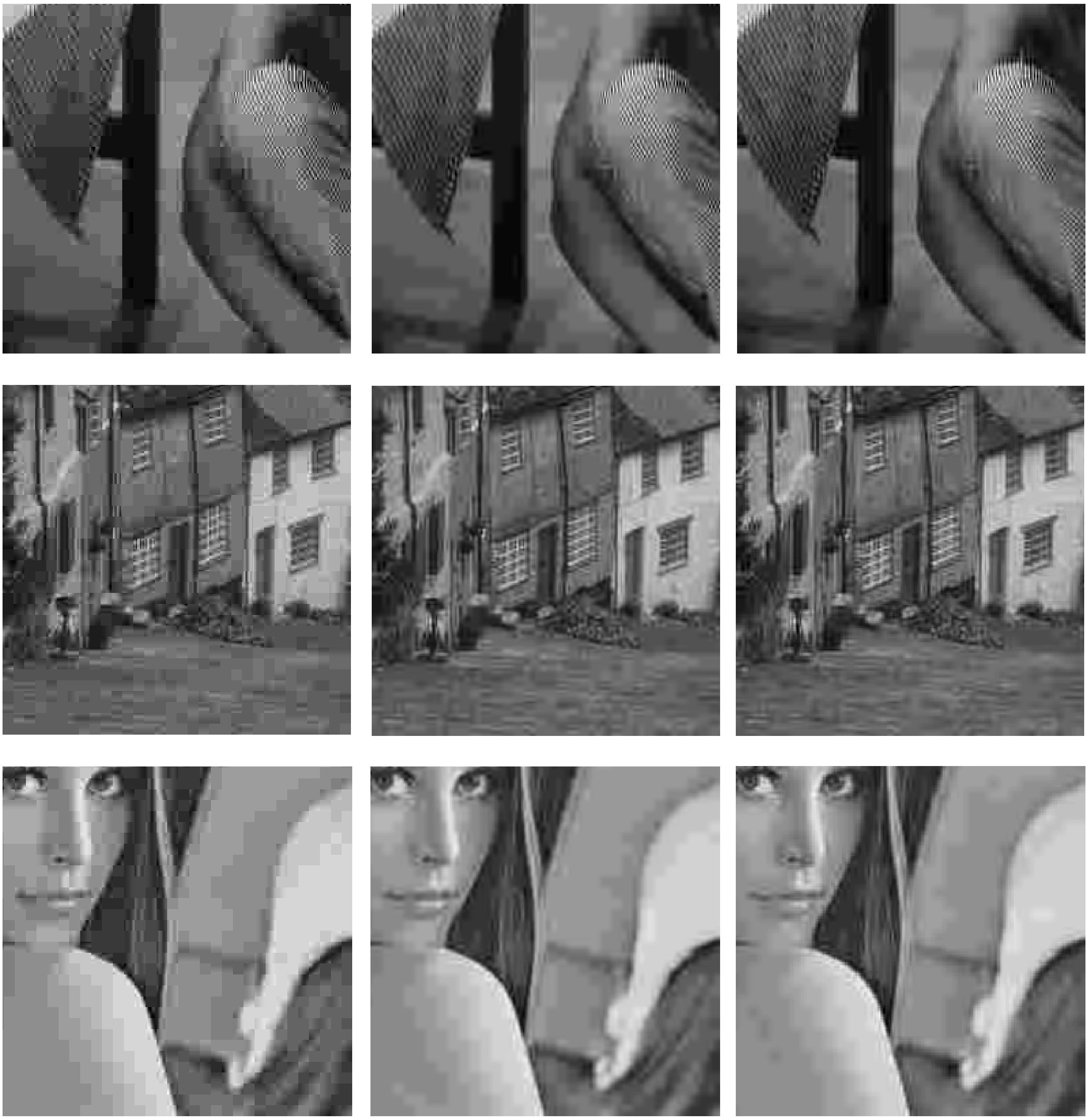

Fig. 7. Comparison of a particular area of decoded images: (top) Barbara: DCT (0.252[bpp]), LiftTDLT (0.245[bpp]), IntTDLT (0.251[bpp]), (middle) Goldhill: DCT (0.247[bpp]), LiftTDLT (0.244[bpp]), IntTDLT (0.244[bpp]), (bottom) Lena: DCT (0.203[bpp]), LiftTDLT (0.204[bpp]), IntTDLT (0.202[bpp]).