A Low-Complexity Parametric Transform for Image Compression

Saad Bouguezel
Laboratoire de Croissance et Caractérisation de Nouveaux Semiconducteurs (LCCNS)
Department of Electronics, Faculty of Technology
Ferhat Abbas University, Setif, Algeria
bouguezel_saad@yahoo.com

M. Omair Ahmad and M.N.S. Swamy
Department of Electrical and Computer Engineering
Concordia University
Montreal, Canada
omair@ece.concordia.ca, swamy@ece.concordia.ca

***Abstract*—In this paper, a one-parameter eight-point orthogonal transform suitable for image compression is proposed. An algorithm for its fast computation is developed and an efficient structure for a simple implementation valid for all possible values of its independent parameter is proposed. It is shown that an appropriate selection of the values of the parameter results in a number of new multiplication-free transforms having a good compromise between the computational complexity and performance. Applying the proposed transform to image compression, we show that it outperforms the existing transforms having complexities similar to that of the proposed one.**

***Index Terms*—Multiplication-free transform, low-complexity transform, image compression, parametric transform.**

I. Introduction

Compression of still images is essential in many applications such as multimedia, internet and mobile communication, wherein a huge amount of data has to be stored or transmitted. Transform-based compression techniques are widely used in such applications [1]-[4]. The transformation step in these techniques is the most time consuming step. One of the most popular transforms used in these techniques is the discrete cosine transform (DCT), which has a good energy compaction capability. Even though there exist fast algorithms for its computation, the multiplications required in these algorithms make them less attractive for real-time applications. Therefore, multiplication-free transforms having acceptable energy compaction capability are good alternatives for such applications. Hence, the development of very fast multiplication-free transforms has become a challenging problem [5]-[10].

Specifically, it has been shown in [7]-[9] that the transforms reported therein have a good compromise between computational complexity and performance. In addition, the complexity of these transforms is lower than that of the existing multiplication-free transforms. In this paper, we introduce a parameterized transform based on the one reported in [7], and show that it outperforms all the existing transforms with complexity similar to that of the proposed one.

II. Proposed Transform

In this section, we propose a one-parameter eight-point transform by introducing an arbitrary parameter in the transform matrix reported in [7] and performing some row permutations. The elements of the transform matrix that are replaced by the parameter are appropriately selected to ensure the orthogonality property of the resulting parametric transform. The purpose of the row permutations is to enhance the energy compaction capability of the transform. This process leads to a new transform matrix given by

$$\mathbf{C}_a = \mathbf{D}_a \mathbf{T}_a \tag{1}$$

where a is an arbitrary scalar, $\mathbf{D}_a$ is a diagonal matrix given by

$$\mathbf{D}_a = \operatorname{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{4}}, \frac{1}{\sqrt{4+4a^2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{4}}, \frac{1}{\sqrt{4+4a^2}}\right), \tag{2}$$

and

$$\mathbf{T}_a = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & a & -a & -1 & -1 & -a & a & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ a & -1 & 1 & -a & -a & 1 & -1 & a \end{bmatrix} \tag{3}$$

It is interesting to note that $\mathbf{C}_a$ is orthogonal for all possible values of a, i.e., $\mathbf{C}_a{}^{-1} = \mathbf{C}_a{}^{t} = \mathbf{T}_a{}^{t}\mathbf{D}_a$, where t denotes the transpose operation.

The use of the proposed transform matrix $\mathbf{C}_a$ in a compression technique consists of merging the diagonal matrix $\mathbf{D}_a$ into the quantisation/dequantisation matrix and performing the transformation step by the matrix $\mathbf{T}_a$; such a merging has also been suggested in many papers, including [6]-[10]. Therefore, the matrix $\mathbf{T}_a$ is the only source of arithmetic operations in the transformation step of an image compression procedure that is based on the proposed transform. The direct transformation of an eight-element row or column vector using $\mathbf{T}_a$ requires 36 additions and 8 multiplications. In order to reduce this computational complexity, we develop a fast algorithm valid for any value of the parameter a by decomposing $\mathbf{T}_a$ into a product of sparse matrices in the form

$$\mathbf{T}_a = \mathbf{P}\mathbf{Q}_a\mathbf{R}\mathbf{S} \tag{4}$$

where the permutation matrix $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{5}$$

$$\mathbf{Q}_a = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{7}$$

and

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{8}$$

The structure corresponding to the decomposition given by (4) is illustrated in Fig. 1. It is clear from this structure that the proposed transform requires only 18 additions and 2 multiplications. This computational complexity can be reduced for some special values of the arbitrary parameter a. For example, if $a = 2^l$, where l is a positive or negative integer, then the two multiplications become just bit-shift operations. In this case, the transform requires only 18 additions and 2 shifts. If $a = 1$, then no shift operation is necessary. In the case of $a = 0$, the complexity reduces to just 16 additions. In this case, we have introduced some row permutations in the transform matrix $\mathbf{T}_a$ given by (3) to enhance the energy compaction capability considered in the subsequent section. Therefore, the resulting structure for $\mathbf{T}_0$ is given in Fig. 2.

III. Application to Image Comression

In this section, we apply the proposed transform to image compression and show that it outperforms the existing transforms having complexities similar to that of the proposed one. For this purpose, we carry out an experiment similar to that in [5], [7]-[10] and implement the JPEG image compression technique for the case of an 8×8 matrix by employing the transform matrices reported in [7]-[9], and the proposed transform matrix $\mathbf{T}_a$ for different values of a, in order to study the effect of the parameterization on the performance, so that a choice can be made for a tradeoff between performance and complexity.

It is well-known that a transform having a greater ability to compact most of the energy in a smaller number of low-frequency transform coefficients is more attractive for image compression applications. Based on this fact, a comparison of the performance of the different transforms is carried out using the PSNR as a function of the number of consecutive transform coefficients retained for the reconstruction of the original image.

We first compare the proposed transform matrices $\mathbf{T}_{0.5}$ and $\mathbf{T}_1$ with those reported in [7] and [9], respectively. The difference between the PSNR obtained by using $\mathbf{T}_{0.5}$ and that obtained by using the transform matrix in [7] is given in Fig. 3.a for the 256×256 Cameraman image and in Fig. 3.b for the 256×256 Lena image. It is clear from these figures that there is an improvement in the performance when the number of transform coefficients is greater than fifteen and can reach up to about 0.5 dB when the compression ratio is two (i.e., the number of transform coefficients is 32). These improvements are achieved without an increase in the computational complexity. Similar results are shown in Fig. 4 for $\mathbf{T}_1$ and the transform in [9]. In this case, there is only a slight improvement without increasing the complexity.

We now compare the performance of the proposed transforms $\mathbf{T}_{0.5}$, $\mathbf{T}_1$ and $\mathbf{T}_0$ requiring 18 additions plus 2 bit-shifts, 18 additions, and 16 additions, respectively, with that of [8], which requires 18 additions and has a good compromise between the computational complexity and performance . The PSNR obtained by the various transforms for the two test images is given in Fig. 5 in terms of the number of transform coefficients retained for reconstructing the original image. It is clear from this figure that the proposed transform $\mathbf{T}_{0.5}$ outperforms all the other transforms. Figs. 6 and 7 show clearly the improvements achieved by the proposed transforms $\mathbf{T}_{0.5}$ and $\mathbf{T}_1$ compared to that reported in [8]. The proposed transform $\mathbf{T}_1$ performs better than that in [8], especially for a compression ratio that is greater than four. It is noted that compression ratios in lossy compression are generally greater than four. It is clear from Fig. 5 that the

PSNR obtained by the proposed transform $\mathbf{T}_0$ is lower than that obtained by the other transforms for compression ratios that are lower than ten. However, it is higher than that obtained by [8] for higher compression ratios and close to that obtained by the proposed transform $\mathbf{T}_{0.5}$. Therefore, the proposed transform $\mathbf{T}_0$ would be attractive for applications requiring higher compression ratios since it has the lowest computational complexity. Some numerical results are given in Tables I and II. We have carried out simulations using other test images and the results are similar to those presented in this paper. Due to lack of space, we are not able to include all the results in this four-page paper.

IV. Conclusion

In this paper, a one-parameter eight-point transform has been proposed by introducing an arbitrary parameter in the transform matrix reported in [7] and performing some row permutations. We have shown that the transform is orthogonal and has an efficient simple structure for all possible values of the parameter. It has also been shown that a number of new multiplication-free transforms having a good compromise between the computational complexity and performance can easily be derived from the proposed transform by an appropriate selection of the values of the parameter, showing the importance of the proposed parameterization in achieving different performances with different complexities, and providing an option for the user to choose between performance and complexity. Specifically, the transform obtained by setting the value of the parameter to zero requires only 16 additions and has been found to have a good performance in the case of higher compression ratios.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Regroupement Stratégique en Microélectronique du Québec (ReSMiQ).

References

[1] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video and Audio Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[2] T. Chang, C. Kung, and C. Jen, "A simple processor core design for DCT/IDCT," *IEEE Trans. Circuits Syst. for Video Technology,* vol. 10, pp. 439–447, April 2000.

[3] M. Lin, L. Dung, and P.Weng, "An ultra low power image compressor for capsule endoscope," *BioMed.Eng. Online*, vol. 5, no. 14, Feb. 2006, 10.1186/1475-925X-5-14.

[4] A. Puri, X. Chen, and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard," *Signal Processing: Image Commun.*, vol. 19, pp. 793–849, Oct. 2004.

[5] Tarek I. Haweel, "A new square wave transform based on the DCT," *Signal Processing*, vol. 81, pp. 2309-2319, 2001.

[6] Krisda Lengwehasatit, and Antonio Ortega, "Scalable Variable Complexity Approximate Forward DCT," *IEEE Trans. Circuits Syst. for Video Technology*, vol. 14, pp. 1236-1248, Nov. 2004.

[7] Saad Bouguezel, M. Omair Ahmad, and M.N.S. Swamy, "Low-Complexity 8×8 Transform for Image Compression," *Electronics Letters*, vol. 44, Oct. 2008.

[8] Aye Aung, Boon Poh Ng, and Susanto Rahardja, "Conjugate Symmetric Sequency-Ordered Complex Hadamard Transform," *IEEE Trans. Signal Processing*, vol. 57, pp. 2582-2593, July 2009,

[9] Saad Bouguezel, M. Omair Ahmad, and M.N.S. Swamy, "A fast 8×8 transform for image compression," *IEEE International Conference on Microelectronics* (ICM'2009), Marrakech, Morocco, Dec. 2009, pp. 74-77.

[10] Saad Bouguezel, M. Omair Ahmad, and M.N.S. Swamy, "A novel transform for image compression," *IEEE International Midwest Symposium on Circuits and Systems* (MWSCAS'2010), Seattle, USA, August 2010, pp. 509-512.

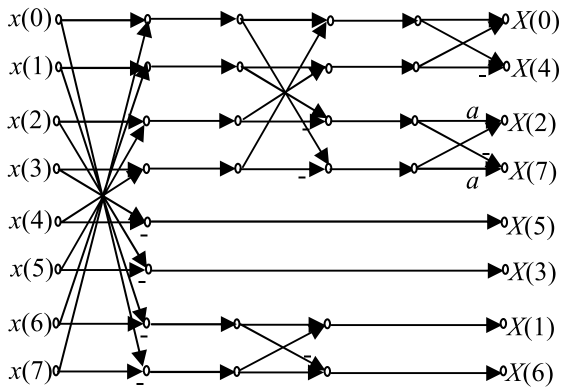

Fig. 1. Fast algorithm for the proposed transform $\mathbf{T}_a$.

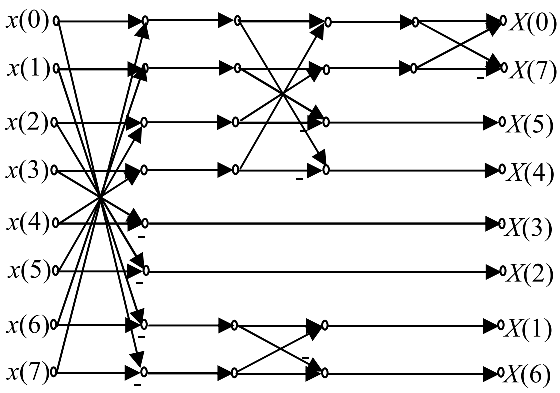

Fig. 2. Fast algorithm for the proposed transform $\mathbf{T}_0$.

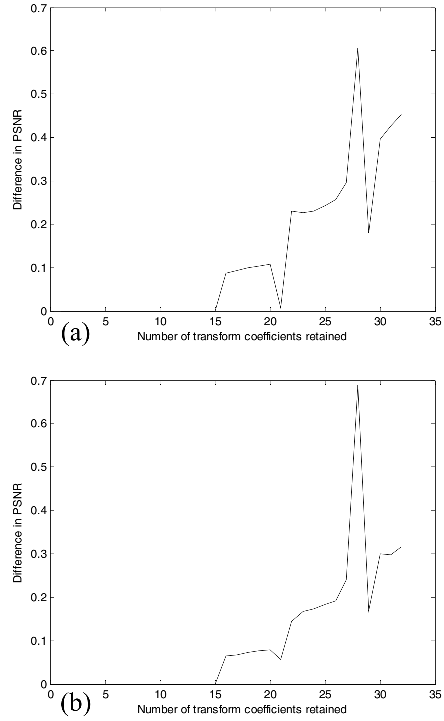

Fig. 3. PSNR obtained by $\mathbf{T}_{0.5}$ minus that obtained by [7] in the case of (a) cameraman and (b) Lena.

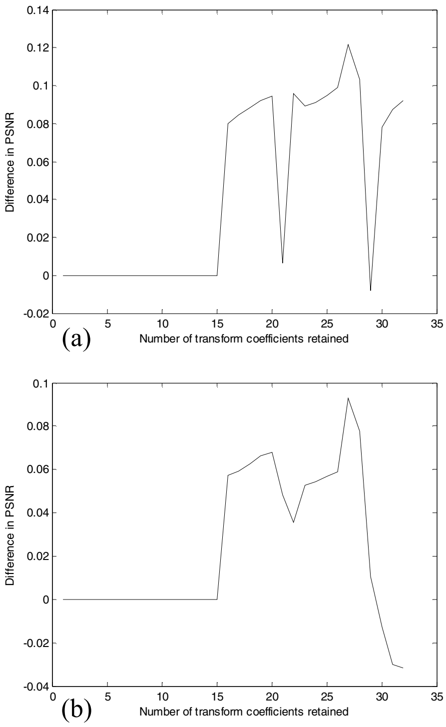

Fig. 4. PSNR obtained by $\mathbf{T}_1$ minus that obtained by [9] in the case of (a) cameraman and (b) Lena.

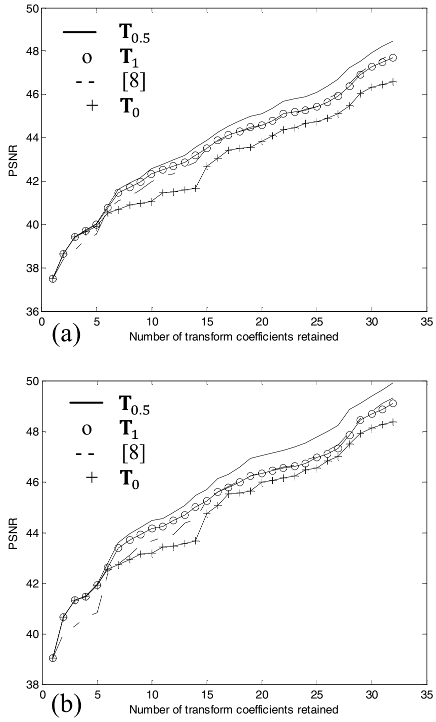

Fig. 5. PSNR obtained by different transforms in the case of (a) cameraman and (b) Lena.

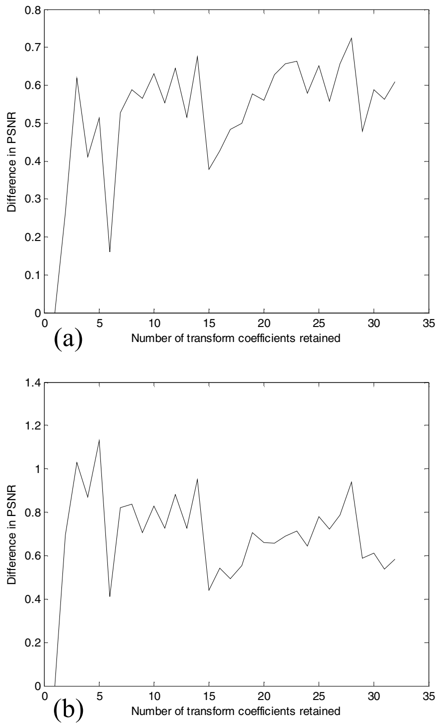

Fig. 6. PSNR obtained by $\mathbf{T}_{0.5}$ minus that obtained by the transform in [8] in the case of (a) cameraman and (b) Lena.

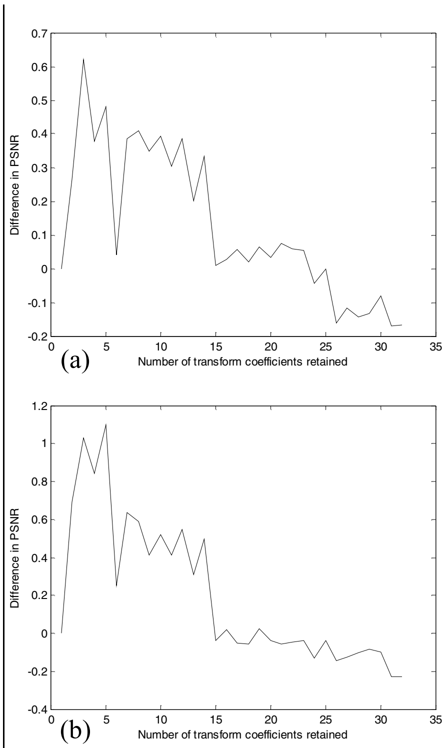

Fig. 7. PSNR obtained by $\mathbf{T}_1$ minus that obtained by the transform in [8] in the case of (a) cameraman and (b) Lena.

Table 1: PSNR obtained by different 8×8 transform matrices for Cameraman image

Number of coefficients retained	$\mathbf{T}_{0.5}$ (18 adds + 2 shifts	$\mathbf{T}_1$ (18 adds)	$\mathbf{T}_0$ (16 adds)	[8] (18 adds)
5	40.03	40.00	39.92	39.52
10	42.57	42.33	41.07	41.94
20	45.12	44.59	43.85	44.56
30	47.93	47.26	46.32	47.34

Table 2: PSNR obtained by different 8×8 transform matrices for Lena image

Number of coefficients retained	$\mathbf{T}_{0.5}$ (18 adds + 2 shifts	$\mathbf{T}_1$ (18 adds)	$\mathbf{T}_0$ (16 adds)	[8] (18 adds)
5	41.93	41.90	41.89	40.80
10	44.48	44.16	43.16	43.65
20	47.03	46.33	45.97	46.37
30	49.39	48.68	48.11	48.77