Integer DCT–II by Lifting Steps

G. Plonka, M. Tasche

Abstract

In image compression, the discrete cosine transform of type II (DCT–II) is of special interest. In this paper we use a new approach to construct an integer DCT–II first considered in [14]. Our method is based on a factorization of the cosine matrix of type II into a product of sparse, orthogonal matrices. The construction of the integer DCT of length 8 works with lifting steps and rounding–off. We are especially interested in the normwise error and the componentwise error when the integer DCT is compared with the exact DCT-II.

1 Introduction

The discrete cosine transform of type II (DCT–II) has found a wide range of applications in signal and image processing (see [16,17]), especially in image compression. It has become the heart of international standards in image compression such as JPEG and MPEG. In some applications, the input data consist of integer vectors or integer matrices. Then the output of DCT–II algorithm consists no longer of integers. For lossless coding it would be of great interest to be able to characterize the output completely again with integers. Lossless coding schemes are hardly based on integer DCT–II which have been studied in recent years (see [2,3,4,5,7,8,9,12,14,15,18,19]). Especially, integer DCT–II of length 8 and 16 and integer wavelets (see e.g. [1]) have been proposed.

An integer DCT–II of length n is a nonlinear, left–invertible mapping which acts on $\mathbf{Z}^n$ and approximates the classical DCT–II of length n. Integer DCT–II possesses some features of the classical DCT–II, whereas its computational cost is not higher than in the classical case. As known the JPEG compression algorithm divides the source image into 8×8 blocks. Then the two–dimensional DCT–II of each block is performed and the result is compressed by a quantization step. Hence $n = 8$ is the most interesting case which we consider in detail. Note that in the JPEG–2000 proposal [11], the use of the integer DCT–II for lossless image coding is recommended. One method for developing an integer DCT–II algorithm is based on the idea to approximate the components of the cosine matrix C_n^{II} of order n by dyadic rationals (see e.g. [18]), paying attention that the symmetry relations are kept. This method destroys the orthonormality of the matrix and the challenge is to find an invertible approximation $\tilde{C}_n$ of C_n^{II} such that its inverse $\tilde{C}_n^{-1}$ again only consists of dyadic rationals. For this reason, suitable approximations of the cosine matrix C_n^{II} of order $n = 8$ or $n = 16$ have only been given.

In [3,5,7,9,19], different factorizations of the transform matrix into products of so–called lifting matrices and simple matrices are applied. Here a lifting matrix is a matrix whose diagonal elements are 1, and only one nondiagonal element is nonzero. Simple matrices are permutation matrices or sparse matrices whose nonzero entries are only integers or half integers. Then the noninteger entries of the lifting matrices are rounded to dyadic rationals, and the inverse matrix factors are easy to determine. This method has the advantage that it works for arbitrary radix–2 lengths.

Due to the rounding of matrix entries, these two methods can lead to high errors, if one compares the integer DCT–II output with the classical DCT–II result, especially if the range for the components of the input vector is large. Explicit error estimates for these algorithms have not been considered.

In this paper, we use a new approach to integer DCT–II introduced in [14]. Note that we are not building integer DCT–II in integer arithmetic. Thus the computations are still done with floating point numbers, but the result is guaranteed to be an integer and the invertibility is preserved. Our algorithms are based on a factorization of C_8^{II} (see [13]) into sparse orthogonal matrices of simple structure. By suitable permutations, each matrix factor can be transferred to a block–diagonal matrix, where every block is an orthogonal matrix of order 2. Now the idea for construction of integer DCT–II of length 8 is very simple. For each block R_2 of order 2 and for arbitrary $\mathbf{x} \in \mathbf{Z}^2$, find a suitable integer approximation of $R_2\mathbf{x}$ such that this process is left–invertible.

The paper is organized as follows. In Section 2 we introduce the cosine matrix of type II and present a factorization of C_8^{II} into products of sparse, orthogonal matrices. In Section 3, we present integer transforms of length 2. Applying the lifting technique (see e.g. [6]) and rounding–off, we construct an integer approximation of $R_2\mathbf{x}$ for arbitrary $\mathbf{x} \in \mathbf{Z}^2$ and estimate the truncation error. Further, we give some properties of the corresponding nonlinear mapping which allow the conjecture that vanishing components of the exact vector $2\,C_8^{II}\mathbf{x}$ with $\mathbf{x} \in \mathbf{Z}^8$ are preserved in the proposed integer DCT–II algorithm.

The results of Sections 2 and 3 are applied to integer DCT–II of length 8 (in Section 4) and to two–dimensional integer DCT–II of size 8 × 8 (in Section 5). We present algorithms for the integer DCT–II and estimate the normwise error as well as the componentwise error when the integer DCT–II result is compared with the exact DCT–II. Our worst case estimates for the absolute error show that the proposed integer DCT–II algorithm is very close to the exact DCT–II and hence preserves the features of frequency decorrelation. In particular, a detailed consideration of the componentwise error will lead us to better error estimates than given in [14]. The numerical results illustrate the performance of our new integer DCT–II algorithm.

2 Factorization of Cosine Matrix

Let $n \geq 2$ be a given integer. In the following, we consider the *cosine matrix of type* II *with order* n which is defined by

$$C_n^{II} := \sqrt{\tfrac{2}{n}} \left(\epsilon_n(j) \cos \tfrac{j(2k+1)\pi}{2n} \right)_{j,k=0}^{n-1}, \tag{2.1}$$

where $\epsilon_n(0) := \sqrt{2}/2$ and $\epsilon_n(j) := 1$ for $j \in \{1, \ldots, n-1\}$. In our notation a subscript of a matrix denotes the corresponding order. Observe that these matrices are orthogonal (see e.g. [16], pp. 13 – 14; [17]). The *discrete cosine transform of type* II (DCT–II) *with length* n is a linear mapping of $\mathbb{R}^n$ onto $\mathbb{R}^n$, which is generated by C_n^{II}. In [13], simple split–radix algorithms are proposed for these transforms of radix–2 length n. They are based on factorization of C_n^{II} into a product of sparse, orthogonal matrices. In this paper, we want to restrict ourselves to $n = 8$ and use the orthogonal factorization of C_8^{II} in order to present an integer DCT–II of length 8, which is very close to the original DCT–II and maps integer vectors to integer vectors (see also [14]). Naturally, this integer DCT–II is not longer a linear mapping.

First, we introduce some notations. Let I_n denote the identity matrix and $J_n := (\delta(j + k - n + 1))_{j,k=0}^{n-1}$ the counteridentity matrix, where δ means the Kronecker symbol. Blanks in a matrix indicate zeros. The direct sum of two matrices A, B is defined to be a block–diagonal matrix $A \oplus B := \text{diag}\,(A, B)$. For even $n \geq 4$, P_n denotes the *even–odd permutation matrix* (or 2*–stride permutation matrix*) defined by

$$P_n\mathbf{x} := (x_0, x_2, \ldots, x_{n-2}, x_1, x_3, \ldots, x_{n-1})^T, \qquad \mathbf{x} = (x_j)_{j=0}^{n-1}.$$

An orthogonal factorization of the cosine matrix C_8^{II} looks as follows (see [13]):

$$C_8^{II} = B_8\,(I_4 \oplus A_4(1))\,(C_2^{II} \oplus C_2^{IV} \oplus C_2^{II} \oplus C_2^{II})\,(T_4(0) \oplus T_4(1))\,T_8(0) \qquad (2.2)$$

with the bit reversal matrix $B_8 := P_8^T(P_4 \oplus P_4)$, determined by

$$B_8\mathbf{x} = (x_0, x_4, x_2, x_6, x_1, x_5, x_3, x_7)^T, \quad \mathbf{x} = (x_j)_{j=0}^7,$$

$$A_4(1) = \tfrac{1}{\sqrt{2}}\begin{pmatrix} \sqrt{2} & & & \\ & 1 & & 1 \\ & 1 & & -1 \\ & & \sqrt{2} & \end{pmatrix},\ T_4(1) = \begin{pmatrix} \cos\frac{\pi}{16} & & & \sin\frac{\pi}{16} \\ & \cos\frac{3\pi}{16} & \sin\frac{3\pi}{16} & \\ & -\sin\frac{3\pi}{16} & \cos\frac{3\pi}{16} & \\ \sin\frac{\pi}{16} & & & -\cos\frac{\pi}{16} \end{pmatrix},$$

$$T_4(0) = \tfrac{1}{\sqrt{2}}\begin{pmatrix} I_2 & J_2 \\ I_2 & -J_2 \end{pmatrix}, \qquad T_8(0) = \tfrac{1}{\sqrt{2}}\begin{pmatrix} I_4 & J_4 \\ I_4 & -J_4 \end{pmatrix},$$

and with

$$C_2^{II} = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad C_2^{IV} := \begin{pmatrix} \cos\frac{\pi}{8} & \sin\frac{\pi}{8} \\ \sin\frac{\pi}{8} & -\cos\frac{\pi}{8} \end{pmatrix}.$$

Let us denote the five *orthogonal* matrix factors of C_8^{II} in (2.2) in this order by

$$C_8^{II} = B_8\,A_8(0,1)\,T_8(0,1,0,0)\,T_8(0,1)\,T_8(0). \qquad (2.3)$$

Note that the factorization (2.3) implies a fast algorithm for computing the DCT–II of length 8 with 11 multiplications and 29 additions (see [13]). This algorithm is very similar to that of C. Loeffler et al. [10].

3 Integer Transforms of Length 2

Considering the factorization (2.3) of C_8^{II}, we observe that each of the matrix factors possesses at most two nonzero entries in one row. The main idea to obtain an integer DCT–II is now as follows. First we consider rotation matrices of the form

$$R_2(\omega) := \begin{pmatrix} \cos\omega & \sin\omega \\ -\sin\omega & \cos\omega \end{pmatrix}, \qquad \omega \in (0, \tfrac{\pi}{4}].$$

For the selected case $n = 8$ we need $R_2(\omega)$ for $\omega \in \{\frac{\pi}{16}, \frac{\pi}{8}, \frac{3\pi}{16}, \frac{\pi}{4}\}$ only.

For $R_2(\omega)$ and for arbitrary $\mathbf{x} \in \mathbf{Z}^2$ we want to find a suitable integer approximation of $R_2(\omega)\mathbf{x}$ such that this process is left–invertible. Afterwards, we construct an integer DCT–II of length 8 from these partial algorithms.

We use the following notations. For $a \in \mathbb{R}$ let $\lfloor a \rfloor := \max\{x \le a;\, x \in \mathbf{Z}\}$. Further let $\mathrm{rd}\,(a) := \lfloor a + 1/2 \rfloor$ be the integer next to a.

Let $s \in \mathbb{R}$ with $s \neq 0$ be given. Then matrices of the form

$$\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 \\ s & 1 \end{pmatrix}$$

are called *lifting matrices of order* 2 (see e.g. [3,4,5,6,9,14,19]). Note that the inverse of a lifting matrix is again a lifting matrix:

$$\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & -s \\ 0 & 1 \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 \\ s & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 \\ -s & 1 \end{pmatrix}.$$

Every rotation matrix $R_2(\omega)$ of order 2 can be represented as a product of three lifting matrices (see [6]):

$$R_2(\omega) = \begin{pmatrix} 1 & \tan\frac{\omega}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\sin\omega & 1 \end{pmatrix} \begin{pmatrix} 1 & \tan\frac{\omega}{2} \\ 0 & 1 \end{pmatrix}. \tag{3.1}$$

Note that the above factorization of $R_2(\omega)$ consists of *nonorthogonal* matrix factors. This factorization is used as follows.

A *lifting step* of the form

$$\hat{\mathbf{y}} := \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \mathbf{x}$$

with $\mathbf{x} = (x_0,\, x_1)^T \in \mathbf{Z}^2$ can be approximated by $\mathbf{y} = (y_0,\, y_1)^T \in \mathbf{Z}^2$ with

$$y_0 = x_0 + \lfloor s\,x_1 + \tfrac{1}{2} \rfloor = x_0 + \mathrm{rd}\,(sx_1), \qquad y_1 = x_1.$$

This transform is left–invertible and a left–inverse reads as follows

$$x_0 = y_0 - \lfloor s\,y_1 + \tfrac{1}{2} \rfloor = y_0 - \mathrm{rd}\,(sy_1), \qquad x_1 = y_1.$$

Indeed, we have that

$$y_0 - \lfloor s\,y_1 + \tfrac{1}{2} \rfloor = x_0 + \lfloor s\,x_1 + \tfrac{1}{2} \rfloor - \lfloor s\,x_1 + \tfrac{1}{2} \rfloor = x_0.$$

We obtain

Theorem 3.1. *Let $R_2(\omega)$ with $\omega \in (0, \frac{\pi}{4}]$ be a rotation matrix. Then for arbitrary $\mathbf{x} = (x_0, x_1)^T \in \mathbf{Z}^2$, a suitable integer approximation $\mathbf{y} = (y_0, y_1)^T \in \mathbf{Z}^2$ of $\hat{\mathbf{y}} := R_2(\omega)\mathbf{x}$ is given by $y_0 = z_2$, $y_1 = z_1$, where*

$$\begin{aligned} z_0 &:= x_0 + \mathrm{rd}\,(x_1 \tan\tfrac{\omega}{2}), \\ z_1 &:= x_1 + \mathrm{rd}\,(-z_0 \sin\omega), \\ z_2 &:= z_0 + \mathrm{rd}\,(z_1 \tan\tfrac{\omega}{2}). \end{aligned} \tag{3.2}$$

This integer transform is left–invertible and a left–inverse reads $x_0 = w_2$, $x_1 = w_1$, where

$$\begin{aligned} w_0 &:= y_0 - \mathrm{rd}\,(y_1 \tan \tfrac{\omega}{2}), \\ w_1 &:= y_1 - \mathrm{rd}\,(-w_0 \sin \omega), \\ w_2 &:= w_0 - \mathrm{rd}\,(w_1 \tan \tfrac{\omega}{2}). \end{aligned}$$

Further, the error estimates

$$(\hat{y}_0 - y_0)^2 + (\hat{y}_1 - y_1)^2 \le \tfrac{1}{4}(3 + 4\sin\omega + 2\cos\omega + (\tan\tfrac{\omega}{2})^2) \tag{3.3}$$

and

$$\begin{aligned} |\hat{y}_0 - y_0| &\le \tfrac{1}{2}(1 + \tan\tfrac{\omega}{2} + \cos\omega), \\ |\hat{y}_1 - y_1| &\le \tfrac{1}{2}(1 + \sin\omega) \end{aligned} \tag{3.4}$$

hold.

Proof. The formulas for y_0, y_1 directly follow by applying the lifting steps to the three matrices in (3.1). Using the inverse of (3.1), we obtain the formulas of x_0, x_1 analogously. For a proof of the error estimates see [14]. The estimates for the componentwise errors in (3.4) directly follow from the proof of Theorem 3.8 in [14]. □

Remark 3.2. (1) Let $\hat{\mathbf{y}} := R_2(\omega)\mathbf{x}$ with arbitrary $\mathbf{x} \in \mathbf{Z}^2$ and $\mathbf{y}$ its integer approximation constructed via (3.2). The special values for the error $\|\hat{\mathbf{y}} - \mathbf{y}\|_2$ and the componentwise absolute errors $\|\hat{\mathbf{y}} - \mathbf{y}\|_\infty$ via the lifting procedure for $\omega \in \{\frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{16}, \frac{3\pi}{16}\}$ follow by inserting into formulas (3.3) – (3.4). In particular, we obtain

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_2 \le \begin{cases} 1.361453 & \text{for } \omega = \frac{\pi}{4}, \\ 1.266694 & \text{for } \omega = \frac{\pi}{8}, \\ 1.199128 & \text{for } \omega = \frac{\pi}{16}, \\ 1.320723 & \text{for } \omega = \frac{3\pi}{16}, \end{cases}$$

$$|\hat{y}_0 - y_0| \le \begin{cases} 1.060660 & \text{for } \omega = \frac{\pi}{4}, \\ 1.061396 & \text{for } \omega = \frac{\pi}{8}, \\ 1.039638 & \text{for } \omega = \frac{\pi}{16}, \\ 1.067408 & \text{for } \omega = \frac{3\pi}{16}. \end{cases} \qquad |\hat{y}_1 - y_1| \le \begin{cases} 0.853553 & \text{for } \omega = \frac{\pi}{4}, \\ 0.691342 & \text{for } \omega = \frac{\pi}{8}, \\ 0.597545 & \text{for } \omega = \frac{\pi}{16}, \\ 0.777785 & \text{for } \omega = \frac{3\pi}{16}. \end{cases}$$

(2) In particular, for $\omega = \frac{\pi}{4}$, we have

$$R_2(\tfrac{\pi}{4}) = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

such that $\mathbf{y} := \sqrt{2} R_2(\frac{\pi}{4})\mathbf{x}$ remains to be an integer vector for arbitrary $\mathbf{x} \in \mathbf{Z}^2$. The multiplication of $\sqrt{2} R_2(\frac{\pi}{4})$ with $\mathbf{x} \in \mathbf{Z}^2$ is left–invertible and moreover, it does not produce any rounding error. Further, this procedure to handle rotation matrices with angle $\frac{\pi}{4}$ requires 2 additions only instead of 3 multiplications and 3 additions for 3 lifting steps. Thus we shall use the multiplication of $\sqrt{2} R_2(\frac{\pi}{4})$ with $\mathbf{x} \in \mathbf{Z}^2$ instead of the lifting method of Theorem 3.1 for the integer DCT–II as often as possible, when the rotation matrix with angle $\frac{\pi}{4}$ occurs in the factorization (2.3).

Let us now consider the nonlinear mapping $f_\omega : \mathbf{Z}^2 \to \mathbf{Z}^2$ with $f_\omega((x_0, x_1)^T) = (y_0, y_1)^T$ where $y_0 := z_2$, $y_1 := z_1$ and z_0, z_1, z_2 are given in (3.2). Note that for arbitrary $\mathbf{x} \in \mathbf{Z}^2$, $f_\omega(\mathbf{x})$ is an integer approximation of $R_2(\omega)\mathbf{x}$. Then we observe the following properties of f_ω.

Lemma 3.3. *For all* $\omega \in (0, \frac{\pi}{4}]$ *we have*

$$f_\omega((0,0)^T) = (0,0)^T.$$

Further, for $\omega = \frac{\pi}{4}$ *and* $x_0 \in \mathbf{Z}$ *we have*

$$\begin{aligned} f_{\pi/4}((x_0, x_0)^T) &= (\mathrm{rd}(\sqrt{2}\, x_0),\, 0)^T, \\ f_{\pi/4}((x_0, -x_0)^T) &= (0,\, -\mathrm{rd}(\sqrt{2}\, x_0) + \delta)^T, \end{aligned}$$

where

$$\delta := \begin{cases} 1 & if \quad \sqrt{2}x_0 - \mathrm{rd}\,(\sqrt{2}x_0) \in (-\frac{1}{2}, -1 + \frac{\sqrt{2}}{2}], \\ 0 & if \quad \sqrt{2}x_0 - \mathrm{rd}(\sqrt{2}x_0) \in (-1 + \frac{\sqrt{2}}{2}, 1 - \frac{\sqrt{2}}{2}], \\ -1 & if \quad \sqrt{2}x_0 - \mathrm{rd}\,(\sqrt{2}x_0) \in (1 - \frac{\sqrt{2}}{2}, \frac{1}{2}). \end{cases}$$

Proof. The first assertion follows immediately from the definition of z_0, z_1, z_2, since for $x_0 = x_1 = 0$ we obviously have $z_0 = z_1 = z_2 = 0$.

We consider the second assertion. Let $\epsilon \in (-\frac{1}{2}, \frac{1}{2})$ be defined by

$$\sqrt{2}\, x_0 = \mathrm{rd}(\sqrt{2}\, x_0) + \epsilon.$$

Then we obtain for $\omega = \frac{\pi}{4}$ and $x_0 = x_1$ by $\tan \frac{\pi}{8} = \sqrt{2} - 1$ that

$$z_0 = x_0 + \mathrm{rd}(x_0(\sqrt{2} - 1)) = \mathrm{rd}(\sqrt{2}x_0) = \sqrt{2}x_0 - \epsilon$$

and

$$z_1 = x_0 + \mathrm{rd}(-z_0\tfrac{\sqrt{2}}{2}) = x_0 + \mathrm{rd}((-\sqrt{2}x_0 + \epsilon)\tfrac{\sqrt{2}}{2}) = x_0 - x_0 + \mathrm{rd}(\tfrac{\sqrt{2}}{2}\epsilon) = 0$$

as well as $z_2 = z_0 = \mathrm{rd}(\sqrt{2}x_0)$.

For $x_1 = -x_0$ we observe that

$$z_0 = x_0 + \mathrm{rd}(-x_0(\sqrt{2}-1)) = 2x_0 + \mathrm{rd}(-\sqrt{2}x_0) = 2x_0 - \mathrm{rd}(\sqrt{2}x_0),$$

since for $s \in \mathbb{R} \setminus (\frac{1}{2} + \mathbf{Z})$ we simply find that $\mathrm{rd}(-s) = -\mathrm{rd}(s)$. Further,

$$\begin{aligned} z_1 &= -x_0 + \mathrm{rd}((-2x_0 + \mathrm{rd}(\sqrt{2}x_0))\tfrac{\sqrt{2}}{2}) \\ &= -x_0 + \mathrm{rd}(-\sqrt{2}x_0 + (\sqrt{2}x_0 - \epsilon)\tfrac{\sqrt{2}}{2}) \\ &= \mathrm{rd}(-\sqrt{2}x_0 - \tfrac{\sqrt{2}}{2}\epsilon) = \mathrm{rd}(-\mathrm{rd}(\sqrt{2}x_0) - \epsilon - \tfrac{\sqrt{2}}{2}\epsilon) \\ &= -\mathrm{rd}(\sqrt{2}x_0) + \mathrm{rd}(-(1+\tfrac{\sqrt{2}}{2})\epsilon). \end{aligned}$$

Finally it follows that

$$\begin{aligned} z_2 &= z_0 + \mathrm{rd}((\sqrt{2}-1)\, z_1) \\ &= 2x_0 - \mathrm{rd}(\sqrt{2}x_0) + \mathrm{rd}\left((\sqrt{2}-1)\left(-\mathrm{rd}(\sqrt{2}x_0) + \mathrm{rd}(-(1+\tfrac{\sqrt{2}}{2})\epsilon)\right)\right) \\ &= 2x_0 + \mathrm{rd}\left(-\sqrt{2}\,\mathrm{rd}(\sqrt{2}x_0) + \sqrt{2}\,\mathrm{rd}(-(1+\tfrac{\sqrt{2}}{2})\epsilon) - \mathrm{rd}(-(1+\tfrac{\sqrt{2}}{2})\epsilon)\right) \\ &= 2x_0 + \mathrm{rd}\left(-2x_0 + \sqrt{2}\epsilon + \sqrt{2}\,\mathrm{rd}(-(1+\tfrac{\sqrt{2}}{2})\epsilon)\right) - \mathrm{rd}(-(1+\tfrac{\sqrt{2}}{2})\epsilon) \\ &= \mathrm{rd}\left(\sqrt{2}\epsilon + \sqrt{2}\,\mathrm{rd}(-(1+\tfrac{\sqrt{2}}{2})\epsilon)\right) - \mathrm{rd}\left(-(1+\tfrac{\sqrt{2}}{2})\epsilon\right). \end{aligned}$$

Now for $\epsilon \in (-1+\frac{\sqrt{2}}{2}, 1-\frac{\sqrt{2}}{2}]$ we have $-(1+\frac{\sqrt{2}}{2})\epsilon \in [-\frac{1}{2}, \frac{1}{2})$, i.e., $\mathrm{rd}(-(1+\frac{\sqrt{2}}{2})\epsilon) = 0$, and we find $z_2 = \mathrm{rd}(\sqrt{2}\epsilon) = 0$, since

$$|\sqrt{2}\,\epsilon| \le \sqrt{2}\,(1-\tfrac{\sqrt{2}}{2}) = \sqrt{2}-1 < \tfrac{1}{2}.$$

For $\epsilon \in (1-\frac{\sqrt{2}}{2}, \frac{1}{2})$ we have $-\frac{1}{2}-\frac{\sqrt{2}}{4} < -(1+\frac{\sqrt{2}}{2})\epsilon < -\frac{1}{2}$, i.e., $\mathrm{rd}(-(1+\frac{\sqrt{2}}{2})\epsilon) = -1$. Hence we obtain

$$z_2 = \mathrm{rd}(\sqrt{2}\epsilon - \sqrt{2}) + 1 = 0$$

by $\sqrt{2}(\epsilon - 1) \in (-1, -\frac{\sqrt{2}}{2})$.

For $\epsilon \in (-\frac{1}{2}, -1+\frac{\sqrt{2}}{2}]$ we have $\mathrm{rd}(-(1+\frac{\sqrt{2}}{2})\epsilon) = 1$ and $z_2 = \mathrm{rd}(\sqrt{2}\epsilon + \sqrt{2}) - 1 = 0$, since $\sqrt{2}(\epsilon+1) \in (\frac{\sqrt{2}}{2}, 1]$. □

Remark 3.4. Lemma 3.3 shows that for the special cases $\mathbf{x} = (x_0, x_0)^T$ and $\mathbf{x} = (x_0, -x_0)^T$, respectively, the error $R_2(\frac{\pi}{4})\mathbf{x} - f_{\pi/4}(\mathbf{x})$ vanishes in the second (resp. first) component and is smaller than $\frac{1}{2}$ in the other component. This property of f_ω implies the conjecture that the integer DCT–II Algorithm 4.1, presented in the next section, preserves vanishing components, i.e., if a component $\hat{y}_j$ $(j \in \{0, \ldots, 7\})$ of $\hat{\mathbf{y}} = 2C_8^{II}\mathbf{x}$ vanishes, then the j–th component of the integer DCT–II equals zero too. Recently it has been shown in [15], that this conjecture is true indeed.

4 Integer DCT–II of Length 8

Now we present an integer DCT–II algorithm, where the lifting method of Theorem 3.1 is used. Further, we estimate the truncation errors in the worst case.

Based on the factorization

$$2C_8^{II} = B_8 A_8(0,1)(I_4 \oplus \sqrt{2}I_4)\, T_8(0,1,0,0)(\sqrt{2}I_4 \oplus I_4)T_8(0,1)\sqrt{2}T_8(0) \quad (4.1)$$

we apply lifting to the submatrix $T_4(1)$ of $T_8(0,1)$, to the submatrix $C_2^{II} \oplus C_2^{IV}$ of $T_8(0,1,0,0)$ and to the submatrix $A_4(1)$ of $A_8(0,1)$. The other matrix–vector products are computed directly. The above factorization (4.1) ensures that the most rotation matrices with angle $\frac{\pi}{4}$ can be computed without lifting using the fact that $\sqrt{2}R_2(\frac{\pi}{4})$ has only ±1 as entries (see Remark 3.2, (2)).

Algorithm 4.1.
Input: $\mathbf{x} \in \mathbf{Z}^8$.

1. Compute $\mathbf{x}^{(1)} := \sqrt{2}T_8(0)\,\mathbf{x}$.

2. Put $\mathbf{w}^{(0)} := (x_0^{(1)}, x_1^{(1)}, x_2^{(1)}, x_3^{(1)})^T$, $\mathbf{w}^{(1)} := (x_4^{(1)}, x_5^{(1)})^T$, $\mathbf{w}^{(2)} := (x_7^{(1)}, x_6^{(1)})^T$. Compute $\mathbf{z} := \sqrt{2}T_4(0)\mathbf{w}^{(0)}$ and

$$\begin{aligned}
\mathbf{z}^{(0)} &:= \mathrm{rd}\,((\tan\tfrac{\pi}{32} \oplus \tan\tfrac{3\pi}{32})\mathbf{w}^{(2)}) + \mathbf{w}^{(1)},\\
\mathbf{z}^{(1)} &:= \mathrm{rd}\,(((-\sin\tfrac{\pi}{16}) \oplus (-\sin\tfrac{3\pi}{16}))\mathbf{z}^{(0)}) + \mathbf{w}^{(2)},\\
\mathbf{z}^{(2)} &:= \mathrm{rd}((\tan\tfrac{\pi}{32} \oplus \tan\tfrac{3\pi}{32})\mathbf{z}^{(1)}) + \mathbf{z}^{(0)}.
\end{aligned}$$

Put $\mathbf{x}^{(2)} := (\mathbf{z}^T, z_0^{(2)}, z_1^{(2)}, z_1^{(1)}, -z_0^{(1)})^T$.

3. Put $\mathbf{w}^{(0)} := (x_0^{(2)}, x_2^{(2)})^T$, $\mathbf{w}^{(1)} := (x_1^{(2)}, x_3^{(2)})^T$, $\mathbf{w}^{(2)} := (x_4^{(2)}, x_5^{(2)}, x_6^{(2)}, x_7^{(2)})^T$.
   Compute $\mathbf{z} := (\sqrt{2}\, C_2^{II} \oplus \sqrt{2}\, C_2^{II})\mathbf{w}$ and
   $$\begin{aligned} \mathbf{z}^{(0)} &:= \mathrm{rd}\,((\tan\tfrac{\pi}{8} \oplus \tan\tfrac{\pi}{16})\mathbf{w}^{(1)}) + \mathbf{w}^{(0)}, \\ \mathbf{z}^{(1)} &:= \mathrm{rd}\,(((-\sin\tfrac{\pi}{4}) \oplus (-\sin\tfrac{\pi}{8}))\mathbf{z}^{(0)}) + \mathbf{w}^{(1)}, \\ \mathbf{z}^{(2)} &:= \mathrm{rd}\,((\tan\tfrac{\pi}{8} \oplus \tan\tfrac{\pi}{16})\mathbf{z}^{(1)}) + \mathbf{z}^{(0)}. \end{aligned}$$
   Put $\mathbf{x}^{(3)} := (z_0^{(2)}, -z_0^{(1)}, z_1^{(2)}, -z_1^{(1)}, \mathbf{z}^T)^T$.
4. For $j = 0, \ldots, 4$ put $x_j^{(4)} := x_j^{(3)}$ and $x_7^{(4)} := x_6^{(3)}$. Compute
   $$\begin{aligned} z_0 &:= \mathrm{rd}\,(x_7^{(3)} \tan\tfrac{\pi}{8}) + x_5^{(3)}, \quad x_6^{(4)} := -\mathrm{rd}\,(-z_0 \sin\tfrac{\pi}{4}) - x_7^{(3)}, \\ x_5^{(4)} &:= \mathrm{rd}\,(-x_6^{(4)} \tan\tfrac{\pi}{8}) + z_0. \end{aligned}$$
5. Compute $\mathbf{y} := B_8\, \mathbf{x}^{(4)}$.

Output: $\mathbf{y} \in \mathbf{Z}^8$ integer approximation of $\hat{\mathbf{y}} = 2\, C_8^{II}\mathbf{x}$.

Algorithm 4.1 needs only 15 multiplications, 31 additions and 15 rounding operations. Hence, its arithmetical complexity is nearly optimal, keeping in mind that best algorithms for DCT–II of length 8 need 11 multiplications and 29 additions without counting the final scaling by $2\sqrt{2}$ (see [10,13,16]).

The left–inverse integer DCT–II algorithm for Algorithm 4.1 simply follows by going backward and taking the left–inverse lifting procedure of Theorem 3.1. Now we analyze the error caused by Algorithm 4.1 comparing the resulting integer vector $\mathbf{y}$ with the exact result $\hat{\mathbf{y}} = 2\, C_8^{II}\mathbf{x}$ of the DCT–II of length 8. A detailed consideration of the componentwise error will lead us to better error estimates than given in [14].

Theorem 4.2. *Let $\mathbf{x} \in \mathbf{Z}^8$ be an arbitrary vector of integers. Using Algorithm 4.1, the resulting integer approximation $\mathbf{y}$ of $\hat{\mathbf{y}} = 2C_8^{II}\mathbf{x}$ satisfies the error estimate*
$$\|\hat{\mathbf{y}} - \mathbf{y}\|_2 \le 5.743824. \tag{4.2}$$
Further an analysis of the componentwise error gives
$$\begin{aligned} |\hat{y}_0 - y_0| &\le 1.060660, & |\hat{y}_1 - y_1| &\le 2.107046, \\ |\hat{y}_2 - y_2| &\le 1.061396, & |\hat{y}_3 - y_3| &\le 3.523072, \\ |\hat{y}_4 - y_4| &\le 0.853553, & |\hat{y}_5 - y_5| &\le 3.315965, \\ |\hat{y}_6 - y_6| &\le 0.691342, & |\hat{y}_7 - y_7| &\le 1.375330, \end{aligned} \tag{4.3}$$

and in particular,

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_\infty \le 3.523072.$$

Proof. Let us denote the preliminary results of the exact DCT–II using the factorization (4.1) by

$$\begin{aligned}
\hat{\mathbf{x}}^{(1)} &:= \sqrt{2}\,T_8(0)\,\mathbf{x},\\
\hat{\mathbf{x}}^{(2)} &:= (\sqrt{2}I_4 \oplus I_4)\,T_8(0,\,1)\hat{\mathbf{x}}^{(1)},\\
\hat{\mathbf{x}}^{(3)} &:= (I_4 \oplus \sqrt{2}I_4)\,T_8(0,\,1,\,0,\,0)\,\hat{\mathbf{x}}^{(2)},\\
\hat{\mathbf{x}}^{(4)} &:= A_8(0,\,1)\,\hat{\mathbf{x}}^{(3)}.
\end{aligned}$$

Then we have $\hat{\mathbf{y}} = 2C_8^{II}\mathbf{x} = B_8\hat{\mathbf{x}}^{(4)}$. Further, let $\mathbf{e}^{(s)} := \mathbf{x}^{(s)} - \tilde{\mathbf{x}}^{(s)}$ ($s \in \{1,2,3,4\}$) denote the error of a single step in the algorithm, where $\mathbf{x}^{(s)}$ are defined in Algorithm 4.1, and where $\tilde{\mathbf{x}}^{(1)} := \sqrt{2}\,T_8(0)\mathbf{x}$, $\tilde{\mathbf{x}}^{(2)} := (\sqrt{2}\,I_4 \oplus I_4)\,T_8(0,1)\mathbf{x}^{(1)}$, $\tilde{\mathbf{x}}^{(3)} := (I_4 \oplus \sqrt{2}\,T_4)\,T_8(0,1,0,0)\mathbf{x}^{(2)}$, $\tilde{\mathbf{x}}^{(4)} := A_8(0,1)\,\mathbf{x}^{(3)}$.

Observe that $\mathbf{e}^{(1)} = \mathbf{x}^{(1)} - \tilde{\mathbf{x}}^{(1)} = \mathbf{0}$. Then for the error $\|\hat{\mathbf{y}} - \mathbf{y}\|_2$ we find

$$\begin{aligned}
\|\hat{\mathbf{y}} - \mathbf{y}\|_2 &= \|\hat{\mathbf{x}}^{(4)} - \mathbf{x}^{(4)}\|_2 = \|\hat{\mathbf{x}}^{(4)} - \tilde{\mathbf{x}}^{(4)} - \mathbf{e}^{(4)}\|_2\\
&\le \|A_8(0,\,1)\|_2\|\hat{\mathbf{x}}^{(3)} - \mathbf{x}^{(3)}\|_2 + \|\mathbf{e}^{(4)}\|_2\\
&\le \|(I_4 \oplus \sqrt{2}I_4)T_8(0,\,1,\,0,\,0)\|_2\|\hat{\mathbf{x}}^{(2)} - \mathbf{x}^{(2)}\|_2 + \|\mathbf{e}^{(3)}\|_2 + \|\mathbf{e}^{(4)}\|_2\\
&= \sqrt{2}\,\|\mathbf{e}^{(2)}\|_2 + \|\mathbf{e}^{(3)}\|_2 + \|\mathbf{e}^{(4)}\|_2.
\end{aligned}$$

With $h(t) := \frac{3}{4} + \sin t + \frac{1}{2}\cos t + \frac{1}{4}(\tan\frac{t}{2})^2$ we find by Theorem 3.1 that

$$\begin{aligned}
\|\mathbf{e}^{(2)}\|_2 &\le (h(\tfrac{\pi}{16}) + h(\tfrac{3\pi}{16}))^{1/2} \approx 1.783877,\\
\|\mathbf{e}^{(3)}\|_2 &\le (h(\tfrac{\pi}{4}) + h(\tfrac{\pi}{8}))^{1/2} \approx 1.859588,\\
\|\mathbf{e}^{(4)}\|_2 &\le h(\tfrac{\pi}{4})^{1/2} \approx 1.361453,
\end{aligned}$$

and

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_2 \le 5.743824.$$

Let us now estimate the componentwise error using the same notations as above. After the first step of Algorithm 4.1 we obtain $\hat{\mathbf{x}}^{(1)} = \mathbf{x}^{(1)}$. After the second step of Algorithm 4.1 we find $\hat{x}_j^{(2)} - x_j^{(2)} = 0$ for $j \in \{0,1,2,3\}$ and by Remark 3.2, (1)

$$\begin{aligned}
|\hat{x}_4^{(2)} - x_4^{(2)}| &\le 1.039638, \qquad |\hat{x}_5^{(2)} - x_5^{(2)}| \le 1.067408,\\
|\hat{x}_6^{(2)} - x_6^{(2)}| &\le 0.777785, \qquad |\hat{x}_7^{(2)} - x_7^{(2)}| \le 0.597545.
\end{aligned}$$

After the third step, we obtain by Remark 3.2, (1) that

$$\begin{aligned} |\hat{x}_0^{(3)} - x_0^{(3)}| &\le 1.060660, & |\hat{x}_1^{(3)} - x_1^{(3)}| &\le 0.853553, \\ |\hat{x}_2^{(3)} - x_2^{(3)}| &\le 1.061396, & |\hat{x}_3^{(3)} - x_3^{(3)}| &\le 0.691342, \\ |\hat{x}_4^{(3)} - x_4^{(3)}| &\le 2.107046, & |\hat{x}_5^{(3)} - x_5^{(3)}| &\le 2.107046, \\ |\hat{x}_6^{(3)} - x_6^{(3)}| &\le 1.375330, & |\hat{x}_7^{(3)} - x_7^{(3)}| &\le 1.375330. \end{aligned}$$

The multiplication of $A_8(0,1)$ in the fourth step leads to

$$\begin{aligned} |\hat{x}_j^{(4)} - x_j^{(4)}| &= |\hat{x}_j^{(3)} - x_j^{(3)}|, \qquad j \in \{0,1,2,3,4\}, \\ |\hat{x}_5^{(4)} - x_5^{(4)}| &\le \tfrac{1}{\sqrt{2}} \cdot 3.482376 + 1.060660 = 3.523072, \\ |\hat{x}_6^{(4)} - x_6^{(4)}| &\le \tfrac{1}{\sqrt{2}} \cdot 3.482376 + 0.853553 = 3.315965, \\ |\hat{x}_7^{(4)} - x_7^{(4)}| &= |\hat{x}_6^{(3)} - x_6^{(3)}| \le 1.375330. \end{aligned}$$

Finally, the permutation in the last step provides the componentwise worst case error as given in (4.3). In particular, we see that in two components, the error is smaller than 1 always, and in 3 further components, the error can exceed 1 only slightly. □

In fact, the numerical results imply that Algorithm 4.1 performs really well.

Example 4.3. For given $\mathbf{x} \in \mathbf{Z}^8$, $\mathbf{y}$ denotes the integer DCT–II of $\mathbf{x}$ computed by Algorithm 4.1 and $\hat{\mathbf{y}} = 2C_8^{II}\mathbf{x}$ is the exact DCT–II of $\mathbf{x}$ (scaled by 2 and rounded to 3 decimal places). In the following table we give some examples for the performance of the Algorithm 4.1.

$\mathbf{x}$	1	1	2	2	3	3	5	5
$\hat{\mathbf{y}}$	15.556	–8.000	1.307	–1.009	0	0.674	–0.541	1.591
$\mathbf{y}$	16	–8	1	–1	0	0	–1	1

$\mathbf{x}$	121	122	122	127	126	129	120	123
$\hat{\mathbf{y}}$	700.036	–3.992	–11.759	4.257	2.828	–3.607	4.871	–8.302
$\mathbf{y}$	700	–4	–12	5	3	–4	5	–8

$\mathbf{x}$	129	–13	12	45	–23	–69	–133	99
$\hat{\mathbf{y}}$	33.234	187.467	156.260	–115.689	320.319	–28.685	161.058	–60.160
$\mathbf{y}$	33	188	156	–115	320	–29	161	–61

$\mathbf{x}$	22	33	44	55	66	77	88	99
$\hat{\mathbf{y}}$	342.240	–141.731	0	–14.816	0	–4.420	0	–1.115
$\mathbf{y}$	342	–142	0	–15	0	–5	0	–1

For further numerical examples and a numerical consideration of the error distribution we refer to [14].

5 Two–Dimensional Integer DCT–II of Size 8×8

The two–dimensional (2–d) DCT–II has important applications in image compression (JPEG, MPEG). Therefore we extend our results of Section 4 to the 2–d integer DCT–II. Let $X \in \mathbf{Z}^{8\times 8}$ be given. Then the 2–d DCT–II *of size* 8×8 of X is defined by $C_8^{II}\, X\, (C_8^{II})^T$. Let $\hat{Y} = (2\, C_8^{II})\, X\, (2\, C_8^{II})^T$. The simple *row–column method* for computing of $\hat{Y}$ is based on the observation

$$\hat{Y} = (2\, C_8^{II}\, X)\, (2\, C_8^{II})^T = \hat{Z}\, (2\, C_8^{II})^T = (2\, C_8^{II}\, \hat{Z}^T)^T$$

with $\hat{Z} := 2\, C_8^{II}\, X$. Now we compute integer approximations of $\hat{Z}$ and $\hat{Y}$ by Algorithm 4.1.

Algorithm 5.1.
Input: $X = (\mathbf{x}_0, \ldots, \mathbf{x}_7) \in \mathbf{Z}^{8\times 8}$.

1. For $k = 0, \ldots, 7$ compute the integer approximation $\mathbf{z}_k$ of $\hat{\mathbf{z}}_k := 2\, C_8^{II} \mathbf{x}_k$ by Algorithm 4.1.

2. Set $Z := (\mathbf{z}_0, \ldots, \mathbf{z}_7)$ and $(\mathbf{u}_0, \ldots, \mathbf{u}_7) := Z^T$.

3. For $k = 0, \ldots, 7$ compute the integer approximation $\mathbf{v}_k$ of $\hat{\mathbf{v}}_k := 2\, C_8^{II} \mathbf{u}_k$ by Algorithm 4.1.

4. Form $Y := (\mathbf{v}_0, \ldots, \mathbf{v}_7)^T$.

Output: $Y \in \mathbf{Z}^{8\times 8}$ integer approximation of $\hat{Y} = 4\, C_8^{II} X\, (C_8^{II})^T$.

Let us consider the (worst case) errors of Algorithm 5.1 estimated in the Frobenius norm and in the maximum norm when the resulting integer matrix Y is compared with the exact (scaled) 2–d DCT–II $\hat{Y}$.

Theorem 5.2. *Let* $X \in \mathbf{Z}^{8\times 8}$ *be an arbitrary matrix. Using Algorithm* 5.1, *the resulting integer approximation* $Y \in \mathbf{Z}^{8\times 8}$ *of* $\hat{Y} = 4\, C_8^{II}\, X\, (C_8^{II})^T$ *satisfies the error estimate*

$$\|\hat{Y} - Y\|_F \le 48.737963.$$

Further, the componentwise errors can be estimated by the error matrix

$$(|\hat{y}_{jk} - y_{jk}|)_{j,k=0}^{7} \leq \begin{pmatrix} 6.718 & 6.187 & 6.287 & 6.186 & 6.718 & 6.186 & 6.287 & 6.186 \\ 7.764 & 7.233 & 7.333 & 7.233 & 7.764 & 7.233 & 7.333 & 7.233 \\ 6.718 & 6.187 & 6.288 & 6.187 & 6.718 & 6.187 & 6.288 & 6.187 \\ 9.180 & 8.649 & 8.749 & 8.649 & 9.180 & 8.649 & 8.749 & 8.649 \\ 6.510 & 5.979 & 6.080 & 5.979 & 6.510 & 5.979 & 6.080 & 5.979 \\ 8.973 & 8.442 & 8.542 & 8.442 & 8.973 & 8.442 & 8.542 & 8.442 \\ 6.348 & 5.817 & 5.918 & 5.817 & 6.348 & 5.817 & 5.918 & 5.817 \\ 7.032 & 6.501 & 6.602 & 6.501 & 7.032 & 6.501 & 6.602 & 6.501 \end{pmatrix},$$

where $\hat{Y} = (\hat{y}_{jk})_{j,k=0}^{7}$ *and* $Y = (y_{jk})_{j,k=0}^{7}$, *and in particular*

$$\|\hat{Y} - Y\|_{\infty} \leq 9.179926.$$

Proof. By (4.2), we know that the computed vectors $\mathbf{z}_k$ $(k = 0, \ldots, 7)$ in step 1 of Algorithm 5.1 satisfy the estimate

$$\|\mathbf{z}_k - \hat{\mathbf{z}}_k\|_2^2 \leq (5.743824)^2.$$

Summing up, this yields

$$\|Z - \hat{Z}\|_F^2 = \sum_{k=0}^{7} \|\mathbf{z}_k - \hat{\mathbf{z}}_k\|_2^2 < 8 \cdot (5.743824)^2$$

with the matrix $\hat{Z} := (\hat{\mathbf{z}}_0, \ldots, \hat{\mathbf{z}}_7)$. Hence,

$$\|Z - \hat{Z}\|_F \leq 2\sqrt{2} \cdot 5.743824.$$

Set $(\hat{\mathbf{u}}_0, \ldots, \hat{\mathbf{u}}_7)^T := \hat{Z}^T$ and $\tilde{\mathbf{v}}_k := 2\,C_8^{II}\hat{\mathbf{u}}_k$ $(k = 0, \ldots, 7)$. Further let $\mathbf{v}_k$ $(k = 0, \ldots, 7)$ be the computed vectors in step 3 of Algorithm 5.1. Applying again (4.2), we get

$$\|\mathbf{v}_k - \tilde{\mathbf{v}}_k\|_2^2 \leq (5.743824)^2$$

and hence

$$\|Y - \tilde{Y}\|_F \leq 2\sqrt{2} \cdot 5.743824$$

with $Y := (\mathbf{v}_0, \ldots, \mathbf{v}_7)^T$ and $\tilde{Y} := (\tilde{\mathbf{v}}_0, \ldots, \tilde{\mathbf{v}}_7)^T = Z\,(2\,C_8^{II})^T$. Since the Frobenius norm is unitarily invariant, we have $\|\tilde{Y} - \hat{Y}\|_F = 2\,\|Z - \hat{Z}\|_F$. The Frobenius norm is consistent such that we can estimate

$$\|Y - \hat{Y}\|_F \leq \|Y - \tilde{Y}\|_F + \|\tilde{Y} - \hat{Y}\|_F \leq 6\sqrt{2} \cdot 5.743824.$$

Now we consider the componentwise error. By (4.3) we obtain in step 1 of Algorithm 5.1 with $\mathbf{z}_k := (z_{jk})_{j=0}^7$ and $\hat{\mathbf{z}}_k := (\hat{z}_{jk})_{j=0}^7$ the error matrix

$$(|\hat{z}_{jk} - z_{jk}|)_{j,k=0}^7 \le (\mathbf{f}, \mathbf{f}, \dots, \mathbf{f}) \in \mathbb{R}^{8\times 8}$$

with

$$\begin{aligned} \mathbf{f} &:= (f_0, f_1, \dots, f_7)^T \\ &= (1.061, 2.107, 1.061, 3.523, 0.854, 3.316, 0.691, 1.375)^T. \end{aligned}$$

Here the errors in (4.3) are rounded to 3 decimal places. In the second step, the matrix is transposed only, i.e., with $\mathbf{1} := (1, 1, \dots, 1)^T \in \mathbb{R}^8$ we have

$$(|\hat{z}_{kj} - z_{kj}|)_{j,k=0}^7 \le (f_0\mathbf{1}, f_1\mathbf{1}, \dots, f_7\mathbf{1}).$$

In the third step we find with $\mathbf{v}_k := (v_{jk})_{j=0}^7$ and $\hat{\mathbf{v}}_k := (\hat{v}_{jk})_{j=0}^7$ that

$$\begin{aligned} (|\hat{v}_{jk} - v_{jk}|)_{j,k=0}^7 &\le (\mathbf{f}, \dots, \mathbf{f}) + 2\,|C_8^{II}|\,(|\hat{z}_{kj} - z_{kj}|)_{j,k=0}^7 \\ &\le \left(\mathbf{f} + 2f_0\,|C_8^{II}|\,\mathbf{1}, \dots, \mathbf{f} + 2f_7\,|C_8^{II}|\mathbf{1}\right), \end{aligned}$$

where

$$|C_8^{II}| := \tfrac{1}{2}\left(\epsilon_8(j)\,|\cos\tfrac{j(2k+1)\pi}{16}|\right)_{j,k=0}^7.$$

Here the error $2f_j|C_8^{II}|\mathbf{1}$ is caused by the error of the previous step and $\mathbf{f}$ by the integer DCT–II applied to $\mathbf{u}_k := (z_{kj})_{j=0}^7$. By

$$|C_8^{II}|\mathbf{1} \approx (2.828, 2.563, 2.613, 2.563, 2.828, 2.563, 2.613, 2.563)^T$$

and $(|\hat{y}_{jk} - y_{jk}|)_{j,k=0}^7 = (|\hat{v}_{jk} - v_{jk}|)_{j,k=0}^7$, we finally obtain the error matrix as given in the theorem. This completes the proof. □

We want to finish with some numerical results for the 2–d integer DCT–II applying Algorithm 5.1.

Example 5.3. Let $X \in \mathbf{Z}^{8\times 8}$ denote the input matrix, Y is the 2–d integer DCT–II of X computed by Algorithm 5.1, and $\hat{Y} = 4C_8^{II}\,X\,(C_8^{II})^T$ is the exact (scaled) 2–d DCT–II of X, where each entry is rounded to the next integer. For

$$X := \begin{pmatrix} 13 & 14 & 14 & 15 & 14 & 12 & 10 & 10 \\ 14 & 14 & 23 & 32 & 15 & 12 & 9 & 6 \\ 15 & 20 & 21 & 13 & 12 & 14 & 21 & 8 \\ 15 & 16 & 16 & 17 & 18 & 19 & 20 & 12 \\ 15 & 16 & 16 & 16 & 16 & 15 & 15 & 15 \\ 13 & 16 & 18 & 16 & 17 & 15 & 14 & 15 \\ 16 & 19 & 17 & 17 & 16 & 15 & 13 & 12 \\ 16 & 15 & 16 & 16 & 16 & 15 & 12 & 11 \end{pmatrix}$$

we obtain that

$$\hat{Y} = \begin{pmatrix} 487 & 41 & -43 & -9 & -15 & 14 & -7 & -1 \\ -9 & 13 & -18 & -12 & -4 & 21 & -5 & -4 \\ -24 & 23 & -14 & -7 & 18 & 0 & 7 & -10 \\ -12 & -9 & 2 & -7 & 20 & -17 & 6 & -5 \\ -11 & -20 & 13 & 20 & 9 & -10 & 2 & 2 \\ -1 & -12 & 21 & 19 & -17 & -12 & -8 & 8 \\ -12 & -13 & 25 & 7 & -20 & -11 & -2 & 17 \\ -8 & 5 & 24 & 2 & -14 & -7 & -5 & 7 \end{pmatrix}$$

and

$$Y := \begin{pmatrix} 488 & 40 & -43 & -8 & -16 & 14 & -7 & 0 \\ -10 & 15 & -17 & -13 & -2 & 20 & -3 & -4 \\ -24 & 22 & -14 & -8 & 18 & -1 & 7 & -11 \\ -13 & -9 & 2 & -12 & 22 & -18 & 6 & -4 \\ -11 & -19 & 14 & 20 & 9 & -10 & 1 & 3 \\ -1 & -13 & 23 & 19 & -16 & -15 & -8 & 8 \\ -13 & -11 & 25 & 8 & -20 & -9 & -1 & 16 \\ -8 & 4 & 23 & 2 & -13 & -7 & -6 & 8 \end{pmatrix}.$$

The greatest componentwise error occurs in the $(3,3)$ position of Y and is 4.555. □

Finally, we consider the behaviour of the errors $\|\hat{Y} - Y\|_\infty := \max_{j,k=0,\ldots,7} |\hat{y}_{jk} - y_{jk}|$ in more detail. As input matrices we use 1000 random matrices in $\mathbf{Z}^{8\times 8}$ with entries in the range $[-127, 128]$. We compute the r–th quantiles for $r = \frac{j}{10}$, $j = 1, \ldots, 10$ for Algorithm 5.1. After sorting the errors of 1000 resulting matrices, the r–th quantile is the smallest value that separates the errors into two parts; $1000\,r$ of the sorted errors are less than or equal to the quantile value, the other $1000\,(1-r)$ errors are greater than the quantile. The 1–th–quantile is the maximal error occurring. In the following table the r–th quantiles are rounded to 3 decimal places.

r=0.1	r=0.2	r=0.3	r=0.4	r=0.5	r=0.6	r=0.7	r=0.8	r=0.9	r=1.0
2.333	2.496	2.627	2.751	2.860	2.984	3.125	3.365	3.655	5.654

References

[1] A. R. Calderbank, I. Daubechies, W. Sweldens and B. L. Yeo: *Wavelet transforms that map integers to integers,* Appl. Comput. Harmon. Anal. **5** (1998), 332–369.

[2] W. K. Cham and P. C. Yip: *Integer sinusoidal transforms for image processing,* Internat. J. Electron. **70** (1991), 1015–1030.

[3] Y.-J. Chen, S. Oraintara and T. Q. Nguyen: *Integer discrete cosine transform (IntDCT),* preprint, 2000.

[4] Y.-J. Chen, S. Oraintara, T. D. Tran, K. Amaratunga and T. Q. Nguyen: *Multiplierless approximation of transforms using lifting scheme and coordinate descent with adder constraint,* IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002, Vol. **3**, 3136–3139.

[5] L. Z. Cheng, H. Xu and Y. Luo: *Integer discrete cosine transform and its fast algorithm,* Electronics Letters **37**, 2001, 64–65.

[6] I. Daubechies, W. Sweldens: *Factoring wavelet transforms into lifting steps,* J. Fourier Anal. Appl. **4** (1998), 247–269.

[7] K. Komatsu and K. Sezaki: *Reversible discrete cosine transform,* Proc. IEEE ICASSP98, 1998, 1769-1772.

[8] K. Komatsu and K. Sezaki: *2D Lossless Discrete Cosine Transform,* IEEE ICIP2001, 2001, 466-469.

[9] J. Liang and T. D. Tran: *Fast multiplierless approximations of the* DCT *with the lifting scheme,* IEEE Trans. Signal Process. **49** (2001), 3032–3044.

[10] C. Loeffler, A. Lightenberg and G. Moschytz: *Practical fast* 1*–d* DCT *algorithms with* 11 *multiplications,* Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Vol. **2** (1989), 988–991.

[11] M. W. Marcellin, M. J. Gormish, A. Bilgin and M. P. Boliek: *An overview of* JPEG–2000, Proc. Data Compression Conf., 2000, 523–541.

[12] W. Philps: *Lossless* DCT *for combined lossy/lossless image coding,* Proc. IEEE Internat. Conf. Image Process., Vol. **3**, 1998, 871–875.

[13] G. Plonka and M. Tasche: *Split–radix algorithms for discrete trigonometric transforms,* Preprint, Gerhard–Mercator–Univ. Duisburg, 2002.

[14] G. Plonka and M. Tasche: *Reversible integer* DCT *algorithms,* Preprint, Gerhard–Mercator–Univ. Duisburg, 2002.

[15] M Primbs: *Integer DCT Algorithmen*, Diplomarbeit, Gerhard-Mercator-Univ. Duisburg, 2002.

[16] K. R. Rao and P. Yip: *Discrete Cosine Transform*: *Algorithms, Advantages, Applications*, Academic Press, Boston 1990.

[17] G. Strang: *The discrete cosine transform,* SIAM Rev. **41** (1999), 135–147.

[18] T. D. Tran: *The Bin*DCT: *Fast multiplierless approximation of the* DCT, IEEE Signal Process. Lett. **7** (2000), 141–144.

[19] Y. Zeng, L. Cheng, G. Bi and A. C. Kot: *Integer* DCT*s and fast algorithms,* IEEE Trans. Signal Process. **49** (2001), 2774–2782.

Addresses:

Gerlind Plonka
Gerhard–Mercator–University Duisburg
Institute of Mathematics
47048 Duisburg, Germany

Manfred Tasche
University of Rostock
Department of Mathematics
18051 Rostock, Germany