Invertible integer DCT algorithms

Gerlind Plonka and Manfred Tasche

Affiliations:

Gerlind Plonka
Institute of Mathematics
University of Duisburg–Essen
D – 47048 Duisburg
Germany

Manfred Tasche
Department of Mathematics
University of Rostock
D – 18051 Rostock
Germany

E–*mail addresses*:
plonka@math.uni-duisburg.de
manfred.tasche@mathematik.uni-rostock.de

Proposed running head: Invertible integer DCT algorithms

Author for correspondence:

Gerlind Plonka
Institute of Mathematics
University of Duisburg–Essen
D – 47048 Duisburg
Germany
Email: plonka@math.uni-duisburg.de
Telephone: 49 203 379 2677
Fax: 49 203 379 2689

Abstract

Integer DCTs have important applications in lossless coding. In this paper, an integer DCT of radix–2 length n is understood to be a nonlinear, (left–)invertible mapping which acts on $\mathbb{Z}^n$ and approximates the classical discrete cosine transform (DCT) of length n. In image compression, the DCT of type II (DCT–II) is of special interest. In this paper we present a new approach to invertible integer DCT–II and integer DCT–IV. Our method is based on a factorization of the cosine matrices of type II and IV into products of sparse, orthogonal matrices. Up to some permutations, each matrix factor is a block–diagonal matrix with blocks being orthogonal matrices of order 2. Hence one has to construct only integer transforms of length 2. We factorize an orthogonal matrix of order 2 into three lifting matrices and work with lifting steps and rounding–off. This allows the construction of new integer DCT algorithms. We give uniform bounds for the worst case difference between the results of exact DCT and the corresponding integer DCT. Finally, we present some numerical experiments for the integer DCT–II of length 8 and for the 2–dimensional integer DCT–II of size 8×8.

Mathematics Subject Classification 2000. 65T50, 65G50, 15A23, 94A08.

Key words. Discrete cosine transform, lossless coding, data compression, factorization of cosine matrix, lifting matrix, rounding–off, integer DCT, invertible integer DCT, worst case error, error estimate.

1 Introduction

The discrete cosine transform of type II (DCT–II) has found a wide range of applications in signal and image processing (see [17, 19]), especially in image compression. It has become the heart of international standards in image compression such as JPEG and MPEG (see [1]). In some applications, the input data consists of integer vectors or integer matrices. But the output of DCT–II does not consist of integers. For lossless coding it would be of great interest to be able to characterize the output completely again with integers. In the JPEG–2000 proposal [13], the use of the integer DCT–II for lossless image coding is recommended. However, up to now, lossless coding schemes are hardly based on integer DCTs which have been studied in recent years (see [3, 4, 5, 6, 9, 10, 11, 14, 20, 22]). Especially, integer DCTs of length 8 and 16 (see [14, 20]) have been proposed. Note that in some papers the notion *integer* DCT just means that floating point operations are avoided while the resulting vector consists of dyadic rationals (see [6, 11, 14, 20, 22]). In contrast, we consider integer–to–integer transforms.
In this paper, an invertible integer DCT of length n is understood to be a nonlinear, (left–)invertible mapping which acts on $\mathbb{Z}^n$ and approximates the classical DCT of length n. Integer DCT possesses some features of the classical DCT, whereas its computational cost is not higher than in the classical case. Integer–to–integer transforms have also been considered in [4, 5, 9, 10, 16].

Usually, an integer DCT is based on a factorization of the transform matrix into products of so–called lifting matrices and simple matrices. 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 (see e.g. [4, 5, 9, 10, 11, 22]). In order to obtain an integer–to–integer transform, a rounding procedure is added after each lifting step (see e.g. [4]). The difference between the results of exact DCT and the corresponding integer DCT is caused on the one hand by the approximation of matrix entries in lifting matrices by dyadic rationals, and on the other hand by the rounding procedure after each lifting step. Explicit error estimates for these algorithms have not been considered.
In this paper, we present new invertible integer DCT algorithms. Note that we are not building integer DCTs 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. In software applications, this should not affect speed, as in many of today's microprocessors floating point and integer computations are virtually equally fast.
Our algorithms are based on new factorizations of cosine matrices C_n^{II} and C_n^{IV} into sparse orthogonal matrices of simple structure. By suitable permutations, each matrix factor can be transferred to a block–diagonal matrix, where each block is an orthogonal matrix of order 2. Now the idea for construction of integer DCTs of radix–2 length n is very simple. For each block R_2 of order 2 and for arbitrary $\mathbf{x} \in \mathbb{Z}^2$, find a suitable integer approximation of $R_2\mathbf{x}$ such that this process is invertible.
In particular, we are firstly able to give upper bounds for the worst case difference between the results of exact DCT and the related integer DCT in the Euclidian and maximum norm, respectively. Using the factorizations of the corresponding cosine matrices in [15],

the applied methods can easily be transferred to other discrete trigonometric transforms.

The paper is organized as follows. In Section 2 we introduce cosine matrices of type II and IV and we sketch some recent results of [15] on the recursive factorization of these matrices into products of sparse, orthogonal matrices. In Section 3, we apply the lifting technique and rounding–off (see [4, 7, 11]), in order to construct an integer approximation of $R_2\mathbf{x}$ for a given invertible matrix $R_2 \in \mathbb{R}^{2\times 2}$ and arbitrary $\mathbf{x} \in \mathbb{Z}^2$. In particular, we estimate the error (see Theorem 3.1). The results of Section 2 and Section 3 are applied to integer DCT–II and integer DCT–IV of radix–2 length in Section 4. We propose two algorithms for the integer DCT–II and the integer DCT–IV. Further, we estimate the worst case error between the resulting vectors of the exact DCT and the corresponding integer DCT. Finally, in Section 5 we investigate the numerical behavior of the integer DCT–II of length 8 and of the 2–dimensional integer DCT–II of size 8×8.

2 Factorization of cosine matrices

Let $n \geq 2$ be a given integer. In the following, we consider *cosine matrices* of type II and IV of order n which are defined by

$$\begin{aligned} C_n^{II} &:= \sqrt{\tfrac{2}{n}} \left(\epsilon_n(j) \cos \tfrac{j(2k+1)\pi}{2n} \right)_{j,k=0}^{n-1}, \\ C_n^{IV} &:= \sqrt{\tfrac{2}{n}} \left(\cos \tfrac{(2j+1)(2k+1)\pi}{4n} \right)_{j,k=0}^{n-1}, \end{aligned}$$

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, while a superscript gives the "type" of the matrix. Observe that these matrices are orthogonal (see e.g. [17], pp. 13 – 14, [18, 19]). The *discrete cosine transforms* of type II (DCT–II) and of type IV (DCT–IV) are linear mappings of $\mathbb{R}^n$ onto $\mathbb{R}^n$, which are generated by C_n^{II} and C_n^{IV}, respectively. In [15], simple split–radix algorithms are proposed for these transforms of radix–2 length n, which are based on factorizations of C_n^{II} and C_n^{IV} into products of sparse, orthogonal matrices. In this paper, we want to use these factorizations in order to derive invertible integer DCTs, which are very close to the original DCT and map integer vectors to integer vectors. Naturally, these integer DCTs are not longer linear mappings.

Let us recall the factorizations for C_n^{II} and C_n^{IV} from [15]. 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 or blocks of zeros. The direct sum of two matrices A, B is defined to be a block–diagonal matrix $A \oplus B := \operatorname{diag}(A, B)$. Let $\Sigma_n := \operatorname{diag}((-1)^k)_{k=0}^{n-1}$ be the diagonal sign matrix.

For even $n \geq 4$, P_n denotes the *even–odd permutation matrix* (or 2–*stride permutation matrix*) defined by

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

Note that $P_n^{-1} = P_n^T$ is the n_1–stride permutation matrix with $n_1 := n/2$. Further, let $Q_n := (I_{n_1} \oplus J_{n_1})\, P_n$ be a *modified even–odd permutation matrix* with

$$Q_n\mathbf{x} := (x_0, x_2, \ldots, x_{n-2}, x_{n-1}, x_{n-3}, \ldots, x_1)^T.$$

Theorem 2.1 *Let $n \geq 4$ be an even integer.*
(i) *The matrix C_n^{II} can be factorized in the form*

$$C_n^{II} = P_n^T \, (C_{n_1}^{II} \oplus C_{n_1}^{IV}) \, T_n(0) \tag{2.1}$$

with the orthogonal matrix

$$T_n(0) := \tfrac{1}{\sqrt{2}} \begin{pmatrix} I_{n_1} & J_{n_1} \\ I_{n_1} & -J_{n_1} \end{pmatrix}.$$

(ii) *The matrix C_n^{IV} can be factorized in the form*

$$C_n^{IV} = P_n^T \, A_n(1) \, (C_{n_1}^{II} \oplus C_{n_1}^{II}) \, T_n(1), \tag{2.2}$$

where

$$A_n(1) := \tfrac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2} & & & \\ & I_{n_1-1} & I_{n_1-1} & \\ & I_{n_1-1} & -I_{n_1-1} & \\ & & & -\sqrt{2} \end{pmatrix} (I_{n_1} \oplus \Sigma_{n_1} J_{n_1})$$

is a modified addition matrix and where $T_n(1) := (I_{n_1} \oplus \Sigma_{n_1}) \, T_n'(1)$ with the cross–shaped twiddle matrix

$$T_n'(1) = \begin{pmatrix} \cos(\frac{\pi}{4n}) & & & & & & \sin(\frac{\pi}{4n}) \\ & \cos(\frac{3\pi}{4n}) & & & & \sin(\frac{3\pi}{4n}) & \\ & & \ddots & & \iddots & & \\ & & & \cos(\frac{(n-1)\pi}{4n}) & \sin(\frac{(n-1)\pi}{4n}) & & \\ & & & -\sin(\frac{(n-1)\pi}{4n}) & \cos(\frac{(n-1)\pi}{4n}) & & \\ & & \iddots & & \ddots & & \\ & -\sin(\frac{3\pi}{4n}) & & & & \cos(\frac{3\pi}{4n}) & \\ -\sin(\frac{\pi}{4n}) & & & & & & \cos(\frac{\pi}{4n}) \end{pmatrix}.$$

The two matrices $A_n(1)$ and $T_n(1)$ are orthogonal.

For a proof of these factorizations we refer to [15]. Note that

$$\begin{aligned} T_n(0) &= (I_{n_1} \oplus (-J_{n_1})) \, Q_n \left(\bigoplus_{k=0}^{n_1-1} R_2(\tfrac{\pi}{4}) \right) Q_n^T, \\ T_n(1) &= (I_{n_1} \oplus \Sigma_{n_1}) \, Q_n \left(\bigoplus_{k=0}^{n_1-1} R_2(\tfrac{(2k+1)\pi}{4n}) \right) Q_n^T, \\ A_n(1) &= \widetilde{Q}_n \, \Sigma_n \left(I_2 \oplus \bigoplus_{k=1}^{n_1-1} R_2(\tfrac{\pi}{4}) \right) \widetilde{Q}_n^T \, (I_{n_1} \oplus \Sigma_{n_1} J_{n_1}) \end{aligned}$$

with the *rotation matrix*

$$R_2(\omega) := \begin{pmatrix} \cos\omega & \sin\omega \\ -\sin\omega & \cos\omega \end{pmatrix},$$

and with $\widetilde{Q}_n := (I_{n_1} \oplus V_{n_1}) P_n$, where V_n is the shift matrix determined by

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

Hence, up to convenient permutations and changes of sign, the matrices $T_n(0)$, $T_n(1)$, and $A_n(1)$ can be represented as block–diagonal matrices, where each block is a rotation matrix $R_2(\omega)$ of order 2. The following constructions of integer DCT–II are based on this essential fact.

Let now $n = 2^t$ and $n_j := 2^{t-j}$, $j = 0, \ldots, t-1$. If the formulas (2.1) and (2.2) are applied recursively, then we obtain factorizations of the cosine matrices C_n^{II} and C_n^{IV}, where all matrix factors are orthogonal block matrices with blocks being permutation matrices or matrices of the form I_{n_j}, $A_{n_j}(1)$, $T_{n_j}(0)$, and $T_{n_j}(1)$ (see [15]). In particular, all matrix factors are sparse, i.e., they possess two nonzero entries at most in each row and each column. Hence, by suitable permutations, the matrix factors can be transferred to block–diagonal matrices, where each block is an orthogonal matrix of order 2.

3 Integer transforms of length 2

The main idea to obtain an integer DCT is now as follows. For a given invertible matrix $R_2 \in \mathbb{R}^{2\times 2}$ and for arbitrary $\mathbf{x} \in \mathbb{Z}^2$, find a suitable integer approximation of $R_2\mathbf{x}$ such that this process is invertible. The simple structure of the matrix factors of C_n^{II} implies that we need to find a suitable solution only for orthogonal matrices $R_2(\omega)$ with angles $\omega \in (0, \frac{\pi}{4}]$.

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 [7, 11]). 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,

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

The above factorization of $R_2(\omega)$ consists of *nonorthogonal* matrix factors. This factorization (see [7]) can be used for construction of integer DCT as follows.

For $a \in \mathbb{R}$ let $\lfloor a \rfloor := \max\{x \leq a : x \in \mathbb{Z}\}$ and $\{a\} := a - \lfloor a \rfloor \in [0, 1)$. Then $\{a\}$ is the noninteger part of a. Further let $\operatorname{rd} a := \lfloor a + \frac{1}{2} \rfloor$ be the integer next to a.

Now, a *lifting step* of the form

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

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

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

This transform is invertible with

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

Indeed, we have

$$y_0 - \mathrm{rd}\,(s\, y_1) = x_0 + \mathrm{rd}\,(s\, x_1) - \mathrm{rd}\,(s\, x_1) = x_0, \qquad y_1 = x_1,$$

and conversely

$$x_0 + \mathrm{rd}\,(s\, x_1) = y_0 - \mathrm{rd}\,(s\, y_1) + \mathrm{rd}\,(s\, y_1) = y_0, \qquad x_1 = y_1.$$

Using the factorization (3.1), we obtain

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

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

The procedure is invertible and its inverse reads $x_0 = v_2$, $x_1 = v_1$, where

$$\begin{aligned} v_0 &:= y_0 - \mathrm{rd}\,(y_1 \tan\tfrac{\omega}{2}), \\ v_1 &:= y_1 - \mathrm{rd}\,(-v_0 \sin\omega), \\ v_2 &:= v_0 - \mathrm{rd}\,(v_1 \tan\tfrac{\omega}{2}). \end{aligned}$$

Further, the error can be estimated by

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_2 \le (h(\omega))^{1/2}, \qquad \|\hat{\mathbf{y}} - \mathbf{y}\|_\infty \le g(\omega) \tag{3.2}$$

with

$$h(\omega) := \tfrac{3}{4} + \sin\omega + \tfrac{1}{2}\cos\omega + \tfrac{1}{4}(\tan\tfrac{\omega}{2})^2, \qquad g(\omega) := \tfrac{1}{2}(1 + \tan\tfrac{\omega}{2} + \cos\omega).$$

Proof. The formulas for y_0, y_1 and x_0, x_1 (after inverse transform) directly follow by applying the lifting steps to the three matrices in (3.1). Now we prove the error estimates (3.2).
1. First we represent the components $\hat{y}_0 - y_0$ and $\hat{y}_1 - y_1$ of $\hat{\mathbf{y}} - \mathbf{y}$ in a convenient way. Let

$$\epsilon_0 := \{x_0 \sin\omega\}$$

denote the noninteger part of $x_0 \sin\omega$, and similarly

$$\epsilon_1 := \{x_1 \tan\tfrac{\omega}{2} + \tfrac{1}{2}\}, \quad \delta_0 := \{\hat{y}_0\} = \{x_0 \cos\omega + x_1 \sin\omega\}, \quad \delta_1 := \{x_1 \cos\omega\}.$$

Hence

$$\{\hat{y}_1\} = \{x_1 \cos\omega - x_0 \sin\omega\} = \{\delta_1 - \epsilon_0\}.$$

Using $(\tan\frac{\omega}{2})\,\sin\omega = 1-\cos\omega$, it follows that

$$x_1\,(1-\cos\omega)+\tfrac{1}{2}\sin\omega = (x_1\tan\tfrac{\omega}{2}+\tfrac{1}{2})\sin\omega = (\lfloor x_1\,\tan\tfrac{\omega}{2}+\tfrac{1}{2}\rfloor+\epsilon_1)\,\sin\omega$$

such that

$$\begin{aligned} y_1 &= z_1 = \lfloor(-\sin\omega)\,(\lfloor x_1\,\tan\tfrac{\omega}{2}+\tfrac{1}{2}\rfloor+x_0)+\tfrac{1}{2}\rfloor+x_1 \\ &= \lfloor -x_1\,(1-\cos\omega)+(\epsilon_1-\tfrac{1}{2})\sin\omega - x_0\,\sin\omega+\tfrac{1}{2}\rfloor+x_1 \\ &= \lfloor x_1\,\cos\omega - x_0\,\sin\omega+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor = \lfloor\hat{y}_1+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor. \end{aligned}$$

Thus we obtain

$$\hat{y}_1-y_1 = \hat{y}_1-\lfloor\hat{y}_1+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor = (\delta_1-\epsilon_0)-\lfloor\delta_1-\epsilon_0+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor. \quad (3.3)$$

Observing that

$$\begin{aligned} y_1 &= \lfloor\lfloor x_1\cos\omega\rfloor+\delta_1-\lfloor x_0\sin\omega\rfloor-\epsilon_0+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor \\ &= \lfloor x_1\,\cos\omega\rfloor-\lfloor x_0\,\sin\omega\rfloor+\lfloor\delta_1-\epsilon_0+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor \end{aligned}$$

and that by $(\cos\omega)\,\tan\frac{\omega}{2} = \sin\omega-\tan\frac{\omega}{2}$,

$$\begin{aligned} &(\lfloor x_1\,\cos\omega\rfloor-\lfloor x_0\,\sin\omega\rfloor)\,\tan\tfrac{\omega}{2} = (x_1\,\cos\omega-x_0\,\sin\omega-\delta_1+\epsilon_0)\,\tan\tfrac{\omega}{2} \\ &= x_1\,(\sin\omega-\tan\tfrac{\omega}{2})-x_0\,(1-\cos\omega)+(\epsilon_0-\delta_1)\,\tan\tfrac{\omega}{2}, \end{aligned}$$

we find

$$\begin{aligned} y_0 &= z_2 = \lfloor y_1\,\tan\tfrac{\omega}{2}+\tfrac{1}{2}\rfloor+\lfloor x_1\,\tan\tfrac{\omega}{2}+\tfrac{1}{2}\rfloor+x_0 \\ &= \lfloor x_1\,(\sin\omega-\tan\tfrac{\omega}{2})-x_0\,(1-\cos\omega) \\ &\quad +(\epsilon_0-\delta_1+\lfloor\delta_1-\epsilon_0+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor)\,\tan\tfrac{\omega}{2}+\tfrac{1}{2}\rfloor+\ \lfloor x_1\,\tan\tfrac{\omega}{2}+\tfrac{1}{2}\rfloor+x_0 \\ &= \lfloor x_1\,\sin\omega+x_0\,\cos\omega+1-\epsilon_1+(\epsilon_0-\delta_1+\lfloor\delta_1-\epsilon_0+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor)\,\tan\tfrac{\omega}{2}\rfloor \\ &= \lfloor\hat{y}_0+1-\epsilon_1+(\epsilon_0-\delta_1+\lfloor\delta_1-\epsilon_0+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor)\,\tan\tfrac{\omega}{2}\rfloor. \end{aligned}$$

Hence we get

$$\begin{aligned} \hat{y}_0-y_0 &= \hat{y}_0-\lfloor\hat{y}_0+1-\epsilon_1+(\epsilon_0-\delta_1+\lfloor\delta_1-\epsilon_0+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor)\,\tan\tfrac{\omega}{2}\rfloor \\ &= \delta_0-\lfloor\delta_0+1-\epsilon_1+(\epsilon_0-\delta_1+\lfloor\delta_1-\epsilon_0+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega\rfloor)\,\tan\tfrac{\omega}{2}\rfloor\,. \end{aligned} \quad (3.4)$$

2. Now we can estimate the truncation error in the following way. Putting

$$\mu_0 := \delta_1-\epsilon_0+\tfrac{1}{2}+(\epsilon_1-\tfrac{1}{2})\sin\omega, \quad \mu_1 := \delta_0+1-\epsilon_1+(\epsilon_0-\delta_1+\lfloor\mu_0\rfloor)\,\tan\tfrac{\omega}{2},$$

the formulas (3.3) and (3.4) imply

$$\|\hat{\mathbf{y}}-\mathbf{y}\|_2^2 = (\delta_1-\epsilon_0-\lfloor\mu_0\rfloor)^2+(\delta_0-\lfloor\mu_1\rfloor)^2 \quad (3.5)$$

and

$$\|\hat{\mathbf{y}}-\mathbf{y}\|_\infty = \max\{|\delta_1-\epsilon_0-\lfloor\mu_0\rfloor|,\,|\delta_0-\lfloor\mu_1\rfloor|\}. \quad (3.6)$$

Since ϵ_0, ϵ_1, and δ_1 are contained in $[0,1)$, it follows that $\mu_0\in(-1,2)$, i.e. $\lfloor\mu_0\rfloor\in\{-1,\,0,1\}$. Then from

$$\max\{\lfloor\mu_0\rfloor-\tfrac{1}{2}-(\epsilon_1-\tfrac{1}{2})\,\sin\omega,\,-1\}\le\delta_1-\epsilon_0<(\lfloor\mu_0\rfloor+1)-\tfrac{1}{2}-(\epsilon_1-\tfrac{1}{2})\,\sin\omega$$

it follows that

$$-\tfrac{1}{2} - (\epsilon_1 - \tfrac{1}{2})\,\sin\omega \le \delta_1 - \epsilon_0 - \lfloor \mu_0 \rfloor < \tfrac{1}{2} - (\epsilon_1 - \tfrac{1}{2})\sin\omega \tag{3.7}$$

and especially for $\lfloor \mu_0 \rfloor = -1$ even

$$0 \le \delta_1 - \epsilon_0 - \lfloor \mu_0 \rfloor < \tfrac{1}{2} - (\epsilon_1 - \tfrac{1}{2})\sin\omega. \tag{3.8}$$

Using (3.7), we obtain the estimate for μ_1,

$$\delta_0 + 1 - \epsilon_1 + (-\tfrac{1}{2} + (\epsilon_1 - \tfrac{1}{2})\sin\omega)\,\tan\tfrac{\omega}{2} < \mu_1 \le \delta_0 + 1 - \epsilon_1 + (\tfrac{1}{2} + (\epsilon_1 - \tfrac{1}{2})\,\sin\omega)\,\tan\tfrac{\omega}{2}$$

and equivalently

$$\delta_0 + \tfrac{1}{2} - \tfrac{1}{2}\tan\tfrac{\omega}{2} + (\tfrac{1}{2} - \epsilon_1)\cos\omega < \mu_1 \le \delta_0 + \tfrac{1}{2} + \tfrac{1}{2}\tan\tfrac{\omega}{2} + (\tfrac{1}{2} - \epsilon_1)\cos\omega. \tag{3.9}$$

Since δ_0, $\epsilon_1 \in [0,1)$, we only need to consider the cases $\lfloor \mu_1 \rfloor \in \{-1,0,1,2\}$.
3. We estimate the truncation errors (3.5) and (3.6). For $\lfloor \mu_1 \rfloor = 0$ we find with (3.5) and (3.7) the error estimates

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 \le \max\{(-\tfrac{1}{2} - (\epsilon_1 - \tfrac{1}{2})\sin\omega)^2, (\tfrac{1}{2} - (\epsilon_1 - \tfrac{1}{2})\sin\omega)^2\} + \delta_0^2 < \tfrac{1}{4}(1+\sin\omega)^2 + 1$$

and $\|\hat{\mathbf{y}} - \mathbf{y}\|_\infty < 1$.
For $\lfloor \mu_1 \rfloor = 1$ we find similarly

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 < \tfrac{1}{4}(1+\sin\omega)^2 + (\delta_0 - 1)^2 \le \tfrac{1}{4}(1+\sin\omega)^2 + 1$$

and $\|\hat{\mathbf{y}} - \mathbf{y}\|_\infty < 1$.
For $\lfloor \mu_1 \rfloor = 2$ it follows by (3.9) that $\delta_0 > 2 - \frac{1}{2} - \frac{1}{2}\tan\frac{\omega}{2} - (\frac{1}{2} - \epsilon_1)\cos\omega$ and we find

$$\begin{aligned}
\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 &< \tfrac{1}{4}(1+\sin\omega)^2 + (\delta_0 - 2)^2 \\
&< \tfrac{1}{4}(1+\sin\omega)^2 + (\tfrac{1}{2} + \tfrac{1}{2}\tan\tfrac{\omega}{2} + (\tfrac{1}{2} - \epsilon_1)\cos\omega)^2 \\
&\le \tfrac{1}{4}((1+\sin\omega)^2 + (1 + \tan\tfrac{\omega}{2} + \cos\omega)^2) \\
&= \tfrac{3}{4} + \sin\omega + \tfrac{1}{2}\cos\omega + \tfrac{1}{4}(\tan\tfrac{\omega}{2})^2
\end{aligned}$$

and

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_\infty < \tfrac{1}{2}(1 + \tan\tfrac{\omega}{2} + \cos\omega).$$

Finally, for $\lfloor \mu_1 \rfloor = -1$ it follows by (3.9) that $\delta_0 < -\frac{1}{2} + \frac{1}{2}\tan\frac{\omega}{2} - (\frac{1}{2} - \epsilon_1)\cos\omega$ and hence

$$\begin{aligned}
\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 &< \tfrac{1}{4}(1+\sin\omega)^2 + (\delta_0 + 1)^2 \\
&< \tfrac{1}{4}(1+\sin\omega)^2 + (\tfrac{1}{2} + \tfrac{1}{2}\tan\tfrac{\omega}{2} - (\tfrac{1}{2} - \epsilon_1)\cos\omega)^2 \\
&\le \tfrac{3}{4} + \sin\omega + \tfrac{1}{2}\cos\omega + \tfrac{1}{4}(\tan\tfrac{\omega}{2})^2
\end{aligned}$$

as before and again

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_\infty < \tfrac{1}{2}(1 + \tan\tfrac{\omega}{2} + \cos\omega).$$

For $\omega \in (0, \frac{\pi}{4}]$ we have

$$1 + \tfrac{1}{4}(1+\sin\omega)^2 < \tfrac{3}{4} + \sin\omega + \tfrac{1}{2}\cos\omega + \tfrac{1}{4}(\tan\tfrac{\omega}{2})^2$$

such that the assertions (3.2) are proved. *q.e.d.*

Remark 3.2 1. *Note that the procedure of Theorem* 3.1 *can also be obtained for reflected rotation matrices* $R_2 := \Sigma_2\, R_2(\omega)$. *In this case, the integer approximation* $\mathbf{y} = (y_0,\, y_1)^T \in \mathbb{Z}^2$ *of* $R_2\,\mathbf{x}$ *is of the form* $y_0 := z_2$, $y_1 := -z_1$ *with* z_0, z_1, z_2 *as in Theorem* 3.1, *and the error estimates hold as before.*
2. *Let* $\hat{\mathbf{y}} := R_2(\omega)\mathbf{x}$ *with arbitrary* $\mathbf{x} \in \mathbb{Z}^2$ *be given and let* $\mathbf{y}$ *its integer approximation. The special values for the errors* $\|\hat{\mathbf{y}} - \mathbf{y}\|_2$ *and* $\|\hat{\mathbf{y}} - \mathbf{y}\|_\infty$ *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* (3.2). *In particular, we obtain*

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_2 \le \begin{cases} 1.361453 & \text{for } \omega = \frac{\pi}{4}, \\ 1.266694 & \text{for } \omega = \frac{\pi}{8}, \\ 1.199128 & \text{for } \omega = \frac{\pi}{16}, \\ 1.320723 & \text{for } \omega = \frac{3\pi}{16}, \end{cases} \qquad \|\hat{\mathbf{y}} - \mathbf{y}\|_\infty \le \begin{cases} 1.060660 & \text{for } \omega = \frac{\pi}{4}, \\ 1.061396 & \text{for } \omega = \frac{\pi}{8}, \\ 1.039638 & \text{for } \omega = \frac{\pi}{16}, \\ 1.067408 & \text{for } \omega = \frac{3\pi}{16}. \end{cases}$$

Further, we have

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_\infty \le \max\{g(\omega) : \omega \in [0, \tfrac{\pi}{4}]\} \approx 1.067442$$

with $g(\omega)$ *in Theorem* 3.1.

4 Integer DCT of radix–2 length

Using the method of Theorem 3.1, we want to derive algorithms for the integer DCT–II and integer DCT–IV of length $n = 2^t$. We want to propose two algorithms using the factorizations of matrices C_n^{II} and C_n^{IV} in Section 2 together with lifting steps and rounding–off procedures of Theorem 3.1.

Algorithm A. The first idea is to apply the lifting steps and rounding–off procedures to all (reflected) rotation matrices in the orthogonal matrix factors of C_n^{II} (and C_n^{IV}, respectively). In this way we are able to give a direct integer approximation of $C_n^{II}\mathbf{x}$ (and $C_n^{IV}\mathbf{x}$, respectively). The inverse of an integer DCT computed by Algorithm A follows simply by going backward and taking inverse lifting procedures of Theorem 3.1. Note that integer DCTs realized by Algorithm A are invertible on $\mathbb{Z}^n$.

Example 4.1 Let $n = 8$. The orthogonal factorization of the cosine matrix C_8^{II} looks by (2.1) and (2.2) as follows (see [15]):

$$\begin{aligned} C_8^{II} &= P_8^T\,(C_4^{II} \oplus C_4^{IV})\,T_8(0) \\ &= B_8\,(I_4 \oplus A_4(1))\,(C_2^{II} \oplus C_2^{IV} \oplus C_2^{II} \oplus C_2^{II})\,(T_4(0) \oplus T_4(1))\,T_8(0) \end{aligned}$$

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

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

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

and with

$$C_2^{II} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad C_2^{IV} = \begin{pmatrix} \cos\frac{\pi}{8} & \sin\frac{\pi}{8} \\ \sin\frac{\pi}{8} & -\cos\frac{\pi}{8} \end{pmatrix} = \Sigma_2 \begin{pmatrix} \cos\frac{\pi}{8} & \sin\frac{\pi}{8} \\ -\sin\frac{\pi}{8} & \cos\frac{\pi}{8} \end{pmatrix}.$$

Note that this factorization of C_8^{II} implies a fast algorithm of the DCT–II of length 8 with 11 multiplications and 29 additions. This algorithm is very similar to that of C. Loeffler et al. [12]. We apply lifting steps and rounding–off procedures to the four reflected rotation matrices of the form $\Sigma_2\, R_2(\frac{\pi}{4})$ in $T_8(0)$, to the four (reflected) rotation matrices of the form $\Sigma_2\, R_2(\frac{\pi}{4})$, $\Sigma_2\, R_2(\frac{\pi}{16})$, $R_2(\frac{3\pi}{16})$ in $(T_4(0) \oplus T_4(1))$, to the four reflected rotation matrices of the form $\Sigma_2\, R_2(\frac{\pi}{4})$, $\Sigma_2\, R_2(\frac{\pi}{8})$ in $C_2^{II} \oplus C_2^{IV} \oplus C_2^{II} \oplus C_2^{II}$ and to the reflected rotation matrix $\Sigma_2\, R_2(\frac{\pi}{4})$ in $I_4 \oplus A_4(1)$. Hence we need 39 multiplications, 39 additions, and 39 rounding–off operations for this integer DCT–II algorithm, i.e., the arithmetical complexity of such an algorithm is relatively high.

The high arithmetical complexity of Algorithm A is due to the fact that matrix factors containing the rotation matrices $R_2(\frac{\pi}{4})$ (as e.g. $T_8(0)$) are computed by expensive lifting steps. An alternative integer DCT with smaller arithmetical complexity is obtained, if we admit a scaling factor.

Algorithm B. We propose for C_n^{II} and C_n^{IV} the scaling factor $\sqrt{n_1}$ with $n_1 = \frac{n}{2}$ and $n_2 = \frac{n}{4}$. Then we use for $n \geq 8$ the factorizations

$$\sqrt{n_1}\, C_n^{II} = P_n^T \left(\sqrt{n_2}\, (C_{n_1}^{II} \oplus C_{n_1}^{IV})\right) \left(\sqrt{2}\, T_n(0)\right), \tag{4.1}$$

$$\sqrt{n_1}\, C_n^{IV} = P_n^T \left(\sqrt{2}\, A_n(1)\right) \left(\sqrt{n_2}\, (C_{n_1}^{II} \oplus C_{n_1}^{II})\right) T_n(1). \tag{4.2}$$

We start with

$$\begin{aligned} \sqrt{2}\, C_4^{II} &= P_4^T\, (C_2^{II} \oplus C_2^{IV})\, (\sqrt{2}\, T_4(0)), \\ \sqrt{2}\, C_4^{IV} &= P_4^T\, A_4(1) \left(\sqrt{2}\, (C_2^{II} \oplus C_2^{II})\right) T_4(1), \end{aligned}$$

where in the factorization of $\sqrt{2}\, C_4^{IV}$ the scaling factor is used for the matrix factor $C_2^{II} \oplus C_2^{II}$ differing from the rule for $n \geq 8$. Note that $\sqrt{2}\, C_2^{II}$ generates a left–invertible mapping on $\mathbb{Z}^2$. Therefore, integer DCTs of Algorithm B are only left–invertible on $\mathbb{Z}^n$.

Example 4.2 Let $n = 8$. We consider now the following factorization of $2\, C_8^{II}$,

$$2\, C_8^{II} = B_8\, (I_4 \oplus A_4(1)) \left((C_2^{II} \oplus C_2^{IV}) \oplus \sqrt{2}\, (C_2^{II} \oplus C_2^{II})\right) (\sqrt{2}\, T_4(0) \oplus T_4(1))\, \sqrt{2}\, T_8(0).$$

We apply lifting steps and rounding–off procedures only to the two (reflected) rotation matrices $\Sigma_2\, R_2(\frac{\pi}{16})$ and $R_2(\frac{3\pi}{16})$ in the submatrix $T_4(1)$, to the two reflected rotation matrices $\Sigma_2\, R_2(\frac{\pi}{4})$, $\Sigma_2\, R_2(\frac{\pi}{8})$ in the submatrix $C_2^{II} \oplus C_2^{IV}$ and to the reflected rotation matrix $\Sigma_2\, R_2(\frac{\pi}{4})$ in $A_4(1)$. Since the matrices $\sqrt{2}\, T_8(0)$, $\sqrt{2}\, T_4(0)$, and $\sqrt{2}\, (C_2^{II} \oplus C_2^{II})$ contain only integers, rounding–off procedures are not necessary after multiplication with these matrices, and rounding errors do not occur. Hence, the Algorithm B for the scaled integer DCT–II of length 8 needs only 15 multiplications, 31 additions and 15 rounding operations. Now, its arithmetical complexity is nearly optimal, keeping in mind that the best algorithm of DCT–II with length 8 requires 11 multiplications and 29 additions without counting the scaling by $2\sqrt{2}$ (see [12]). An explicit algorithm for this example can be found in [16].

We want to estimate the worst case difference between the results of the exact (scaled) DCT and the corresponding integer DCT. First we consider Algorithm A, where all multiplications with (reflected) rotation matrices in the factorizations (2.1) and (2.2) are replaced by the lifting and rounding procedure of Theorem 3.1.
Let $n = 2^t$, $t \geq 1$, and let $\mathbf{x} \in \mathbb{Z}^n$ be an arbitrary vector. Further let $\mathbf{y} = \mathbf{y}(\mathbf{x}) \in \mathbb{Z}^n$ be the resulting integer approximation of $\hat{\mathbf{y}} := C_n^{II}\,\mathbf{x}$ applying Algorithm A. With $e_{n,2}^{II}$ and $e_{n,\infty}^{II}$, we denote the *worst case error* in the Euclidean norm and maximum norm, respectively, i.e.,

$$e_{n,2}^{II} := \sup\{\|C_n^{II}\mathbf{x} - \mathbf{y}\|_2 : \mathbf{x} \in \mathbb{Z}^n\}, \qquad e_{n,\infty}^{II} := \sup\{\|C_n^{II}\mathbf{x} - \mathbf{y}\|_\infty : \mathbf{x} \in \mathbb{Z}^n\}.$$

Analogously for the integer approximation $\mathbf{w} \in \mathbb{Z}^n$ of $C_n^{IV}\,\mathbf{x}$ via Algorithm A, we denote the worst case errors by $e_{n,2}^{IV}$ and $e_{n,\infty}^{IV}$, i.e.,

$$e_{n,2}^{IV} := \sup\{\|C_n^{IV}\mathbf{x} - \mathbf{w}\|_2 : \mathbf{x} \in \mathbb{Z}^n\}, \qquad e_{n,\infty}^{IV} := \sup\{\|C_n^{IV}\mathbf{x} - \mathbf{w}\|_\infty : \mathbf{x} \in \mathbb{Z}^n\}.$$

Theorem 4.3 *Let $n = 2^t$, $t \geq 1$, and let $e_{n,2}^{II}$, $e_{n,\infty}^{II}$, $e_{n,2}^{IV}$, and $e_{n,\infty}^{IV}$ be the worst case errors occurring, if exact* DCT *output vectors are compared with the corresponding integer* DCT *results of Algorithm* A*. Then upper bounds of the worst case errors in the Euclidean norm can be recursively computed by*

$$\begin{aligned}
e_{2,2}^{II} &\leq \left(h(\tfrac{\pi}{4})\right)^{1/2}, \qquad e_{2,2}^{IV} \leq \left(h(\tfrac{\pi}{8})\right)^{1/2}, \\
e_{n,2}^{II} &\leq \left((e_{n_1,2}^{II})^2 + (e_{n_1,2}^{IV})^2\right)^{1/2} + \left(n_1\, h(\tfrac{\pi}{4})\right)^{1/2}, && t \geq 2, \\
e_{n,2}^{IV} &\leq \left(\sum_{k=0}^{n_1-1} h(\tfrac{(2k+1)\pi}{4n})\right)^{1/2} + \sqrt{2}\, e_{n_1,2}^{II} + \left((n_1-1)\, h(\tfrac{\pi}{4})\right)^{1/2}, && t \geq 2.
\end{aligned}$$

Upper bounds of the worst case errors in the maximum norm are

$$\begin{aligned}
e_{2,\infty}^{II} &\leq g(\tfrac{\pi}{4}), \qquad e_{2,\infty}^{IV} \leq g(\tfrac{\pi}{8}), \\
e_{n,\infty}^{II} &\leq \max\{e_{n_1,\infty}^{II},\, e_{n_1,\infty}^{IV}\} + \sqrt{n_1}\, g(\tfrac{\pi}{4}), && t \geq 2, \\
e_{n,\infty}^{IV} &\leq \sqrt{n}\, \max\{g(\omega) : \omega \in [0,\, \tfrac{\pi}{4}]\} + \sqrt{2}\, e_{n_1,\infty}^{II} + g(\tfrac{\pi}{4}), && t \geq 2.
\end{aligned}$$

Proof. We show the above estimates using the relations (2.1) and (2.2).
We consider the worst case errors in the Euclidian norm. The estimates for $e_{2,2}^{II}$ and $e_{2,2}^{IV}$ directly follow from Theorem 3.1 and Remark 3.2. By (2.1) we have

$$C_n^{II} = P_n^T\,(C_{n_1}^{II} \oplus C_{n_1}^{IV})\,T_n(0).$$

Let $\mathbf{x} \in \mathbb{Z}^n$ be an arbitrary input vector. We set $\hat{\mathbf{x}}^{(1)} := T_n(0)\,\mathbf{x}$, $\hat{\mathbf{x}}^{(2)} := (C_{n_1}^{II} \oplus C_{n_1}^{IV})\,\hat{\mathbf{x}}^{(1)}$, and $\hat{\mathbf{y}} := P_n^T\,\hat{\mathbf{x}}^{(2)} = C_n^{II}\,\mathbf{x}$. Further, let $\mathbf{x}^{(1)}$ be the integer approximation of $\hat{\mathbf{x}}^{(1)}$ using the lifting steps and rounding–off procedures of Theorem 3.1 for all n_1 (reflected) rotation matrices of $T_n(0)$. Let $\mathbf{x}^{(2)}$ be the integer approximation of $\hat{\mathbf{x}}^{(2)}$, which is obtained by applying the integer algorithm to $(C_{n_1}^{II} \oplus C_{n_1}^{IV})\,\mathbf{x}^{(1)}$. Finally, the integer approximation $\mathbf{y}$ of $\hat{\mathbf{y}}$ is only a permutation of $\mathbf{x}^{(2)}$. For $t \geq 2$ we find that

$$\begin{aligned}
\|\hat{\mathbf{y}} - \mathbf{y}\|_2 &= \|\hat{\mathbf{x}}^{(2)} - \mathbf{x}^{(2)}\|_2 \\
&\leq \|\hat{\mathbf{x}}^{(2)} - (C_{n_1}^{II} \oplus C_{n_1}^{IV})\,\mathbf{x}^{(1)}\|_2 + \|(C_{n_1}^{II} \oplus C_{n_1}^{IV})\,\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_2 \\
&\leq \|\hat{\mathbf{x}}^{(1)} - \mathbf{x}^{(1)}\|_2 + \left((e_{n_1,2}^{II})^2 + (e_{n_1,2}^{IV})^2\right)^{1/2} \\
&\leq \left(n_1\, h(\tfrac{\pi}{4})\right)^{1/2} + \left((e_{n_1,2}^{II})^2 + (e_{n_1,2}^{IV})^2\right)^{1/2}.
\end{aligned}$$

For $e_{n,2}^{IV}$, $e_{n,\infty}^{II}$, and $e_{n,\infty}^{IV}$, the proofs work analogously. *q.e.d.*

Remark 4.4 *Note that $h(\omega)$ is concave on $[0,\,\frac{\pi}{4}]$ such that for all $\omega \in [0,\,\frac{\pi}{4}]$*

$$h(\omega) \le \tilde{h}(\omega) := h(0) + h'(0)\,\omega = \tfrac{5}{4} + \omega.$$

Consequently, we can estimate

$$\sum_{k=0}^{n_1-1} h(\tfrac{(2k+1)\pi}{4n}) \le \sum_{k=0}^{n_1-1} \tilde{h}(\tfrac{(2k+1)\pi}{4n}) = \tfrac{2n}{\pi} \int_0^{\pi/4} \tilde{h}(\omega)\,\mathrm{d}\omega = \tfrac{10+\pi}{16}\,n,$$

since the midpoint rule is exact for linear functions.
Further, one can show that $\frac{3}{2}\sqrt{n}\,\log_2 n$ is an upper bound of $e_{n,2}^{II}$ and $e_{n,2}^{IV}$, respectively, and that $\frac{13}{2}\sqrt{n}$ is an upper bound of $e_{n,\infty}^{II}$ and $e_{n,\infty}^{IV}$, respectively.

Example 4.5 Theorem 4.3 yields the following upper bounds of the worst case errors for $n = 2^t$, $t = 1, \ldots, 6$:

n	$e_{n,2}^{II}$	$e_{n,\infty}^{II}$	$e_{n,2}^{IV}$	$e_{n,\infty}^{IV}$
2	1.361453	1.060660	1.266694	1.061396
4	3.784973	2.561396	5.070715	4.695545
8	9.050477	6.816865	10.23107	7.702204
16	17.51041	10.702203	19.96457	14.97093
32	32.00139	19.21357	35.07515	22.23433
64	55.18159	28.23423	59.96287	36.77229

The fast growing worst case error between the exact DCT and the corresponding integer DCT is obviously due to the large number of lifting steps and rounding–off procedures of Theorem 3.1.

Now let us consider Algorithm B based on the factorizations (4.1) and (4.2), where all multiplications with integer matrix factors are just evaluated without further change. Lifting steps and rounding–off procedures of Theorem 3.1 are only applied to the remaining (reflected) rotation matrices in the matrix factors.
Let again $n = 2^t$, $t \ge 1$, and let $\mathbf{x} \in \mathbb{Z}^n$ be an arbitrary input vector. Further let $\mathbf{y} \in \mathbb{Z}^n$ be the resulting integer approximation of $\hat{\mathbf{y}} := \sqrt{n_1}\,C_n^{II}\,\mathbf{x}$ applying Algorithm B. With $e_{n,2}^{II}$ and $e_{n,\infty}^{II}$, we again denote the worst case error in Euclidean norm and maximum norm, respectively. For the integer approximation $\mathbf{w} \in \mathbb{Z}^n$ of $\sqrt{n_1}\,C_n^{IV}\,\mathbf{x}$ via Algorithm B, we denote the worst case errors by $e_{n,2}^{IV}$ and $e_{n,\infty}^{IV}$.

Theorem 4.6 *Let $n = 2^t$, $t \ge 1$, and let $e_{n,2}^{II}$, $e_{n,\infty}^{II}$, $e_{n,2}^{IV}$, and $e_{n,\infty}^{IV}$ be the worst case errors occurring, if exact* DCT *output vectors scaled by $\sqrt{n_1}$ are compared with corresponding integer* DCT *results of Algorithm* B*. Then upper bounds of worst case errors in the Euclidean norm can be recursively computed by*

$$\begin{aligned}
e_{2,2}^{II} &\le \left(h(\tfrac{\pi}{4})\right)^{1/2}, \qquad e_{n,2}^{II} \le \left((e_{n_1,2}^{II})^2 + (e_{n_1,2}^{IV})^2\right)^{1/2}, \qquad t \ge 2,\\
e_{2,2}^{IV} &\le \left(h(\tfrac{\pi}{8})\right)^{1/2}, \qquad e_{4,2}^{IV} \le \sqrt{2}\,\left(h(\tfrac{\pi}{16}) + h(\tfrac{3\pi}{16})\right)^{1/2} + \left(h(\tfrac{\pi}{4})\right)^{1/2},\\
e_{n,2}^{IV} &\le \sqrt{n_1}\left(\sum_{k=0}^{n_1-1} h(\tfrac{(2j+1)\pi}{4n})\right)^{1/2} + 2\,e_{n_1,2}^{II} + \tfrac{1}{2}\sqrt{2}, \qquad t \ge 3.
\end{aligned}$$

Upper bounds of the worst case errors in the maximum norm are

$$\begin{aligned} e^{II}_{2,\infty} &\leq g(\tfrac{\pi}{4}), \qquad e^{II}_{n,\infty} \leq \max\{e^{II}_{n_1,\infty},\, e^{IV}_{n_1,\infty}\}, \qquad t \geq 2, \\ e^{IV}_{2,\infty} &\leq g(\tfrac{\pi}{8}), \qquad e^{IV}_{4,\infty} \leq 2\, \max\{g(\tfrac{\pi}{16}),\, g(\tfrac{3\pi}{16})\} + g(\tfrac{\pi}{4}) \\ e^{IV}_{n,\infty} &\leq n_1\, \sqrt{2}\, \max\{g(\omega) : \omega \in [0,\, \tfrac{\pi}{4}]\} + 2\, e^{II}_{n_1,\infty} + \tfrac{1}{2}, \quad t \geq 3. \end{aligned}$$

Proof. Using the factorizations (4.1) and (4.2), the proof works similar to the proof of Theorem 4.3. *q.e.d.*

Example 4.7 Theorem 4.6 yields the following upper bounds of the worst case errors for $n = 2^t$, $t = 1, \ldots, 6$:

n	$e^{II}_{n,2}$	$e^{II}_{n,\infty}$	$e^{IV}_{n,2}$	$e^{IV}_{n,\infty}$
2	1.361453	1.060660	1.266694	1.061396
4	1.859588	1.061396	3.884236	3.195544
8	4.306432	3.195545	9.466687	8.661157
16	10.40017	8.661157	19.39821	18.96782
32	22.01032	18.96782	41.66265	41.92577
64	47.11932	41.92577	85.03752	86.74255

In particular, the upper bounds of the worst case errors for the scaled integer DCT–II of length 8 are reasonably small.

Remark 4.8 1. *For a special error estimate of the integer* DCT–II *of length* 8 *we refer to* [16]. *A componentwise investigation of the worst case error* $e^{II}_{8,\infty}$ *shows that* $e^{II}_{8,\infty}$ *can only occur in two components, while very small rounding errors appear in the other components.* 2. *The results can directly be extended to the* 2*–dimensional* (2–d) *integer* DCT–II. *Let* $X \in \mathbb{Z}^{n\times n}$ *be given. Then the* 2–d DCT–II *of size* $n \times n$ *of* X *is defined by* $C^{II}_n\, X\, (C^{II}_n)^T$. *Using the row–column method for computing of* $\hat{Y}$, *i.e.,*

$$\hat{Y} = (C^{II}_n\, X)\, (C^{II}_n)^T = \hat{Z}\, (C^{II}_n)^T = (C^{II}_n\, \hat{Z}^T)^T$$

with $\hat{Z} := C^{II}_n\, X$, *we can simply derive an algorithm of a* 2–d *integer* DCT–II *of size* $n \times n$ *by applying Algorithms* A *or* B *first to the columns of* X *and then to the rows of the resulting integer matrix. Moreover, worst case errors can be estimated using the results of Theorems* 4.3 *and* 4.6. *For* $n = 8$ *and Algorithm* B, *this has been done in* [16].

5 Numerical results

We want to apply the two algorithms proposed in Section 4 and compare them regarding their numerical errors. The following examples show the behavior of the Algorithms A and B in Section 4 for the integer DCT–II of length 8.
Let $\mathbf{x} \in \mathbb{Z}^8$ be a given integer vector. Let $\mathbf{y}_\circ$ denote the result of the integer DCT–II algorithm $\circ$ with $\circ \in \{A, B\}$. Further, let $\hat{\mathbf{y}}_A := C^{II}_8 \mathbf{x}$ and $\hat{\mathbf{y}}_B := 2\, C^{II}_8 \mathbf{x}$ be the exact vectors after applying (scaled) DCT–II of length 8. In the following tables we give the components of exact vectors $\hat{\mathbf{y}}_\circ$ (rounded to 3 decimal places) and the components of $\mathbf{y}_\circ$ for three examples of $\mathbf{x}$.
1. Let $\mathbf{x} := (100, 100, 100, 100, 0, 0, 0, 0)^T$.

$\hat{\mathbf{y}}_A$	141.421	128.146	0.000	−44.999	0.000	30.067	0.000	−25.490
$\mathbf{y}_A$	141	130	0	−45	0	30	0	−26
$\hat{\mathbf{y}}_B$	282.843	256.292	0.000	−89.998	0.000	60.134	0.000	−50.980
$\mathbf{y}_B$	283	256	0	−90	0	61	0	−51

For the errors in Euclidian norm, we obtain

$$\|\hat{\mathbf{y}}_A - \mathbf{y}_A\|_2 \approx 1.970, \qquad \|\hat{\mathbf{y}}_B - \mathbf{y}_B\|_2 \approx 0.927.$$

The absolute errors in the components can be seen from the table.
2. Let $\mathbf{x} := (1, 2, 3, 4, 5, 6, 7, 8)^T$.

$\hat{\mathbf{y}}_A$	12.728	−6.442	0.000	−0.673	0.000	−0.201	0.000	−0.051
$\mathbf{y}_A$	11	−7	0	−1	0	−1	0	0
$\hat{\mathbf{y}}_B$	25.456	−12.885	0.000	−1.347	0.000	−0.402	0.000	−0.101
$\mathbf{y}_B$	25	−13	0	−1	0	−1	0	0

For the errors in the Euclidian norm, we find

$$\|\hat{\mathbf{y}}_A - \mathbf{y}_A\|_2 \approx 2.011, \qquad \|\hat{\mathbf{y}}_B - \mathbf{y}_B\|_2 \approx 0.842.$$

3. Let $\mathbf{x} := (-30, -94, -112, 60, 26, -79, 27, 38)^T$.

$\hat{\mathbf{y}}_C$	−57.983	−89.501	−12.305	−9.729	124.451	51.364	−72.205	−3.414
$\mathbf{y}_C$	−58	−90	−13	−9	125	52	−72	−4
$\hat{\mathbf{y}}_D$	−115.966	−179.002	−24.610	−19.457	248.902	102.728	−144.410	−6.827
$\mathbf{y}_D$	−116	−179	−24	−20	249	103	−144	−7

For the errors in the Euclidian norm, we obtain

$$\|\hat{\mathbf{y}}_A - \mathbf{y}_A\|_2 \approx 1.535, \qquad \|\hat{\mathbf{y}}_B - \mathbf{y}_B\|_2 \approx 0.974.$$

We consider the distribution of the errors $\|\hat{\mathbf{y}}_\circ - \mathbf{y}_\circ\|_2$ and $\|\hat{\mathbf{y}}_\circ - \mathbf{y}_\circ\|_\infty$ generated by the two algorithms $\circ \in \{A, B\}$ in more detail. As input vectors we use 1000 random vectors in $\mathbb{Z}^8$ with entries in the range $[-1023, 1024]$, i.e., each component is computed by a random number generator in MAPLE which is supposed to return independent and uniformly distributed data in the given range. We compute the r–th quantiles for $r = \frac{j}{10}$, $j = 1, \ldots, 10$ for each algorithm. After sorting the errors of 1000 resulting vectors, 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. For $r = 1.0$ we obtain the maximal error occurring. In the following tables the r–th quantiles are rounded to three decimal places.

Alg.	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
A	1.220	1.413	1.534	1.652	1.771	1.876	2.003	2.125	2.296	3.097
B	0.888	1.012	1.110	1.191	1.276	1.353	1.426	1.521	1.656	2.438

Table 1. r–th quantiles for the error $\|\hat{\mathbf{y}}_\circ - \mathbf{y}_\circ\|_2$ *with* $\circ \in \{A, B\}$

Alg.	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
A	0.744	0.883	0.958	1.039	1.108	1.194	1.278	1.397	1.576	2.345
B	0.535	0.631	0.697	0.759	0.822	0.894	0.966	1.070	1.245	2.270

Table 2. r–th quantiles for the error $\|\hat{\mathbf{y}}_\circ - \mathbf{y}_\circ\|_\infty$ *with* $\circ \in \{A, B\}$

The numerical results show that Algorithm B is most favorable. It possesses very small worst case errors and provides suitable integer approximations for the DCT–II of length 8, as seen in the numerical tests. The average error in Euclidian norm of this algorithm is less than 1.3 and the average error in maximum norm is even smaller than 1, i.e., in most cases Algorithm B provides one of the two nearest integers to the exact DCT component value in each component. Taking the arithmetical complexity into account, Algorithm B is most recommended. Otherwise, an integer DCT–II based on Algorithm B is only left–invertible.

Finally, let us look at the 2–d integer DCT–II. Now by A we denote the row–column algorithm based on Algorithm A. By B we denote the row–column algorithm applying Algorithm B. Let X be an input matrix of order 8, Y_A resp. Y_B are the 2–d integer DCT–II of X computed by method A resp. B, and $\hat{Y}_A := C_8^{II}\, X\, (C_8^{II})^T$, $\hat{Y}_B := 4\, C_8^{II}\, X\, (C_8^{II})^T$ are the corresponding exact (scaled) 2–d DCT–II of X, where each entry is rounded to the nearest integer. For the input matrix

$$X := \begin{pmatrix} 11 & 16 & 21 & 25 & 27 & 27 & 27 & 27 \\ 16 & 23 & 25 & 28 & 31 & 28 & 28 & 28 \\ 22 & 27 & 32 & 35 & 30 & 28 & 28 & 28 \\ 31 & 33 & 34 & 32 & 32 & 31 & 31 & 31 \\ 31 & 32 & 33 & 34 & 34 & 27 & 27 & 27 \\ 33 & 33 & 33 & 33 & 32 & 29 & 29 & 29 \\ 34 & 34 & 33 & 35 & 34 & 29 & 29 & 29 \\ 34 & 34 & 33 & 33 & 35 & 30 & 30 & 30 \end{pmatrix},$$

we obtain that

$$\hat{Y}_A = \begin{pmatrix} 236 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{pmatrix},$$

$$Y_A = \begin{pmatrix} 237 & 0 & -11 & -6 & 2 & -2 & -2 & 2 \\ -23 & -18 & -7 & -4 & -2 & 2 & 0 & -1 \\ -12 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -5 & -2 & 1 & 3 & 1 & 0 & 0 & 0 \\ -1 & 0 & 3 & 1 & 1 & 2 & 2 & 1 \\ 0 & 0 & 2 & -1 & 0 & 2 & 2 & -1 \\ -1 & -1 & 0 & -3 & -1 & 1 & 2 & 0 \\ -3 & 0 & -3 & -1 & 2 & 1 & 0 & -1 \end{pmatrix},$$

and

$$\hat{Y}_B = \begin{pmatrix} 943 & -4 & -48 & -21 & 9 & -7 & -11 & 5 \\ -90 & -70 & -25 & -13 & -11 & 0 & 2 & -5 \\ -44 & -37 & -6 & 6 & 1 & -4 & -2 & 0 \\ -28 & -8 & 1 & 6 & 4 & 0 & 0 & 1 \\ -2 & -3 & 6 & 6 & 0 & -3 & 2 & 5 \\ 7 & -1 & 6 & -1 & -3 & 6 & 4 & -4 \\ -5 & -1 & -1 & -6 & -2 & 7 & 4 & -3 \\ -10 & 6 & -15 & -7 & 7 & 5 & -2 & -2 \end{pmatrix},$$

$$Y_B := \begin{pmatrix} 942 & -5 & -49 & -19 & 9 & -8 & -12 & 6 \\ -92 & -70 & -26 & -13 & -10 & 2 & 2 & -5 \\ -43 & -37 & -8 & 5 & 1 & -3 & -2 & 0 \\ -32 & -4 & -1 & 5 & 5 & 0 & 0 & 2 \\ -2 & -6 & 5 & 6 & 0 & -2 & 3 & 6 \\ 4 & -3 & 6 & 0 & -2 & 4 & 3 & -3 \\ -3 & -1 & -2 & -5 & -2 & 7 & 5 & -3 \\ -12 & 6 & -14 & -7 & 8 & 4 & -1 & -2 \end{pmatrix}.$$

We get the errors in the Frobenius norm

$$\|\hat{Y}_A - Y_A\|_F \approx 7.153, \qquad \|\hat{Y}_B - Y_B\|_F \approx 10.240.$$

The above example is taken from [21].

Acknowledgement. The authors would like to thank S. Dekel for very instructive remarks improving the paper.

References

[1] V. Bhaskaran and K. Konstantinides, *Images and Video Compression Standards: Algorithms and Architectures*, Kluwer, Boston, 1997.

[2] 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.

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

[4] Y.–J. Chen, S. Oraintara, and T.Q. Nguyen, Integer discrete cosine transform IntDCT, Preprint, Univ. Boston, 2000.

[5] 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, Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Vol. **3**, 2002, 3136–3139.

[6] L.Z. Cheng, H. Xu, and Y. Luo, Integer discrete cosine transform and its fast algorithm, Electron. Lett. **37** (2001), 64–65.

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

[8] E. Feig and S. Winograd, Fast algorithms for the discrete cosine transform, IEEE Trans. Signal Process. **40** (1992), 2174 – 2193.

[9] K. Komatsu and K. Sezaki, Reversible discrete cosine transform, Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., 1998, 1769–1772.

[10] K. Komatsu and K. Sezaki, 2D lossless discrete cosine transform, Proc. IEEE Internat. Conf. Image Process., 2001, 466–469.

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

[12] 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.

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

[14] W. Philips, Lossless DCT for combined lossy/lossless image coding, Proc. IEEE Internat. Conf. Image Process., vol. 3, 1998, pp. 871 – 875.

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

[16] G. Plonka and M. Tasche, Integer DCT–II by lifting steps, in: *Advances in Multivariate Approximation*, (W. Haußmann, K. Jetter, M. Reimer, J. Stöckler (eds.)), Birkhäuser, Basel, 2003, to appear.

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

[18] U. Schreiber, Fast and numerically stable trigonometric transforms (in German), Thesis, Univ. Rostock, 1999.

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

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

[21] G.K. Wallace, The JPEG still picture compression standard, Comm. ACM **34** (1991), 32 – 44.

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