

TRANSFORM DESIGN WITH SCALED AND NON-SCALED INTERFACES

[0001] The present application claims priority to provisional U.S. Application Serial No. 60/787,562, entitled "CONVERGENCE OF SCALED AND NON-SCALED IDCT ARCHITECTURES," filed March 29, 2006, assigned to the assignee hereof and incorporated herein by reference.

BACKGROUND

Field

[0002] The present disclosure relates generally to processing, and more specifically to techniques for performing transform on data.

Background

[0003] Transforms are commonly used to convert data from one domain to another domain. For example, discrete cosine transform (DCT) is commonly used to transform data from spatial domain to frequency domain, and inverse discrete cosine transform (IDCT) is commonly used to transform data from frequency domain to spatial domain. DCT is widely used for image/video compression to spatially decorrelate blocks of picture elements (pixels) in images or video frames. The resulting transform coefficients are typically much less dependent on each other, which makes these coefficients more suitable for quantization and encoding. DCT also exhibits energy compaction property, which is the ability to map most of the energy of a block of pixels to only few (typically low order) transform coefficients. This energy compaction property can simplify the design of encoding algorithms.

[0004] Transforms such as DCT and IDCT may be used for various applications that may support different image and video coding standards. Hence, it is desirable to provide interfaces that can receive and provide data in formats suitable for these applications. Furthermore, since transforms may be performed on large quantity of data, it is desirable to perform transforms as efficiently as possible.

SUMMARY

[0005] Techniques for efficiently performing full and scaled transforms on data received via full and scaled interfaces, respectively, are described herein. A full transform is a transform that implements the complete mathematical description of the transform. A full transform operates on or provides full transform coefficients (or simply, transform coefficients). A full transform may also be referred to as a non-scaled transform, a complete transform, etc. A scaled transform is a transform that operates on or provides scaled transform coefficients, which are scaled versions of the full transform coefficients. The scaled transform may have lower computational complexity and may be used by applications that can accept scaled transform coefficients. The full transform may be used by applications that desire to exchange full transform coefficients. The full and scaled transforms may be for a two-dimensional (2D) IDCT, which may be implemented in a separable manner with one-dimensional (1D) IDCTs. The full and scaled transforms may also be for a 2D DCT, which may be implemented in a separable manner with 1D DCTs. The 1D IDCTs and 1D DCTs may be implemented in a computationally efficient manner, as described below.

[0006] Various aspects and features of the disclosure are described in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0007] FIG. 1A shows a separable full 2D IDCT with 2D scaling.
- [0008] FIG. 1B shows a separable full 2D IDCT with row-column scaling.
- [0009] FIG. 1B shows a separable full 2D IDCT with 1D scaling.
- [0010] FIG. 1D shows a separable scaled 2D IDCT.
- [0011] FIG. 2 shows a flow graph of a factorization of an 8-point 1D IDCT.
- [0012] FIG. 3A shows a separable full 2D DCT with 2D scaling.
- [0013] FIG. 3B shows a separable full 2D DCT with row-column scaling.
- [0014] FIG. 3B shows a separable full 2D DCT with 1D scaling.
- [0015] FIG. 3D shows a separable scaled 2D DCT.
- [0016] FIG. 4 shows a flow graph of a factorization of an 8-point 1D DCT.
- [0017] FIG. 5 shows an IDCT processor supporting full and scaled interfaces.
- [0018] FIG. 6 shows a DCT processor supporting full and scaled interfaces.
- [0019] FIG. 7 shows a process for performing transform.

- [0020] FIG. 8 shows an encoding system and a decoding system.
 [0021] FIG. 9 shows a block diagram of the encoding system.
 [0022] FIG. 10 shows a block diagram of the decoding system.

DETAILED DESCRIPTION

[0023] The techniques described herein may be used for various types of transforms such as DCT, IDCT, discrete Fourier transform (DFT), inverse DFT (IDFT), modulated lapped transform (MLT), inverse MLT, modulated complex lapped transform (MCLT), inverse MCLT, etc. The techniques may also be used for various applications such as image, video, and audio processing, communication, computing, data networking, data storage, graphics, etc. In general, the techniques may be used for any application that uses a transform. For clarity, the techniques are described below for DCT and IDCT, which are commonly used in image and video processing.

[0024] An N-point 1D DCT and an N-point 1D IDCT of type II may be defined as follows:

$$X[k] = \frac{c(k)}{2} \cdot \sum_{n=0}^{N-1} x[n] \cdot \cos \frac{(2n+1) \cdot k\pi}{2N}, \text{ and} \quad \text{Eq (1)}$$

$$x[n] = \sum_{k=0}^{N-1} \frac{c(k)}{2} \cdot X[k] \cdot \cos \frac{(2n+1) \cdot k\pi}{2N}, \quad \text{Eq (2)}$$

$$\text{where } c(k) = \begin{cases} 1/\sqrt{2} & \text{if } k = 0 \\ 1 & \text{otherwise} \end{cases},$$

$x[n]$ is a 1D spatial domain function, and

$X[k]$ is a 1D frequency domain function.

[0025] The 1D DCT in equation (1) operates on N input samples or spatial domain values $x[0]$ through $x[N-1]$ and generates N transform coefficients $X[0]$ through $X[N-1]$. The 1D IDCT in equation (2) operates on N transform coefficients and generates N output samples. Type II DCT is one type of transform and is commonly believed to be one of the most efficient transforms among various energy compacting transforms often proposed for image/video compression.

[0026] The 1D DCT and 1D IDCT may be implemented in their original forms shown in equations (1) and (2), respectively. However, substantial reduction in computational complexity may be realized by finding factorizations that result in as few multiplications and additions as possible, as described below.

[0027] The 1D DCT in equation (1) may be expressed in matrix form, as follows:

$$\underline{\mathbf{x}} = \underline{\mathbf{T}} \underline{\mathbf{y}}, \quad \text{Eq (3)}$$

where $\underline{\mathbf{y}}$ is an $N \times 1$ vector of input samples,

$\underline{\mathbf{T}}$ is an $N \times N$ matrix of a full 1D DCT, and

$\underline{\mathbf{x}}$ is an $N \times 1$ vector of transform coefficients.

[0028] $\underline{\mathbf{y}}$ contains input samples $x[0]$ through $x[N-1]$, and $\underline{\mathbf{x}}$ contains transform coefficients $X[0]$ through $X[N-1]$. The elements of $\underline{\mathbf{T}}$ may be obtained based on equation (1).

[0029] The 1D DCT may be factored into a product of matrices, as follows:

$$\underline{\mathbf{T}} = \underline{\mathbf{S}} \underline{\mathbf{T}}_s, \quad \text{Eq (4)}$$

where $\underline{\mathbf{S}} = \text{diag}(A_0, \dots, A_{N-1})$ is a diagonal matrix of scale factors, and

$\underline{\mathbf{T}}_s$ is an $N \times N$ matrix of a scaled 1D DCT.

Equations (3) and (4) indicate that a full 1D DCT may be performed on $\underline{\mathbf{y}}$ by first performing a scaled 1D DCT on $\underline{\mathbf{y}}$ and then scaling the results with $\underline{\mathbf{S}}$.

[0030] An advantage of decomposing a full transform into a scaled transform and a scaling operation, e.g., as shown in equation (4), is that by properly choosing the scale factors, the multiplicative complexity of the scaled transform may be reduced. For example, a well-known decomposition of Arai, Agui, and Nakajima (AAN) in "A Fast DCT-SQ Scheme for Images," Transactions of the IEICE, November 1988, produces a scaled 8-point DCT that may be implemented with only five multiplications by irrational factors. In contrast, the best-known full 8-point DCT requires 11 such multiplications.

[0031] An $N \times N$ 2D DCT may be defined as follows:

$$\underline{\mathbf{T}} \otimes \underline{\mathbf{T}} = (\underline{\mathbf{S}} \underline{\mathbf{T}}_s) \otimes (\underline{\mathbf{S}} \underline{\mathbf{T}}_s) = (\underline{\mathbf{S}} \otimes \underline{\mathbf{S}}) (\underline{\mathbf{T}}_s \otimes \underline{\mathbf{T}}_s) , \quad \text{Eq (5)}$$

where $\underline{\mathbf{T}} \otimes \underline{\mathbf{T}}$ is a Kroneker product of $\underline{\mathbf{T}}$ with itself and is a matrix of a full 2D DCT,

$\underline{\mathbf{T}}_s \otimes \underline{\mathbf{T}}_s$ is a matrix of a scaled 2D DCT, and

$\underline{\mathbf{S}} \otimes \underline{\mathbf{S}}$ is a matrix of scale factors for the scaled 2D DCT.

The result of the operation in equation (5) is a matrix of an N×N 2D DCT.

[0032] A 2D DCT may be performed on an N×N matrix of input samples $\underline{\mathbf{Y}}$ in a separable manner, for one dimension at a time. For example, a 1D DCT may be performed on each row of $\underline{\mathbf{Y}}$ to obtain an intermediate matrix, and a 1D DCT may then be performed on each column of the intermediate matrix. Alternatively, a 1D DCT may be performed on each column of $\underline{\mathbf{Y}}$ followed by a 1D DCT on each row of the intermediate matrix.

[0033] Equation (5) indicates that a 2D DCT may be performed on $\underline{\mathbf{Y}}$ by first performing a scaled 2D DCT on $\underline{\mathbf{Y}}$ and then scaling the results. Equation (5) also indicates that the scaling for the row and column 1D DCTs may be merged into a single step (which is matrix product by $\underline{\mathbf{S}} \otimes \underline{\mathbf{S}}$) applied to the results of the scaled 2D DCT.

[0034] The 1D IDCT in equation (2) may be expressed in matrix form, as follows:

$$\underline{\mathbf{T}}^{-1} = \underline{\mathbf{T}}^t = \underline{\mathbf{T}}_s^t \underline{\mathbf{S}} , \quad \text{Eq (6)}$$

where $\underline{\mathbf{T}}^t$ is an N×N matrix of a full 1D IDCT and “ t ” denotes the transpose.

[0035] A 2D IDCT may be expressed as:

$$(\underline{\mathbf{T}} \otimes \underline{\mathbf{T}})^{-1} = \underline{\mathbf{T}}^{-1} \otimes \underline{\mathbf{T}}^{-1} = (\underline{\mathbf{T}}_s^t \underline{\mathbf{S}}) \otimes (\underline{\mathbf{T}}_s^t \underline{\mathbf{S}}) = (\underline{\mathbf{T}}_s^t \otimes \underline{\mathbf{T}}_s^t) (\underline{\mathbf{S}} \otimes \underline{\mathbf{S}}) . \quad \text{Eq (7)}$$

[0036] Equation (7) indicates that a 2D IDCT may be performed on an N×N matrix of transform coefficients $\underline{\mathbf{X}}$ by first scaling the transform coefficients and then performing a scaled 2D IDCT on the scaled transform coefficients. Equation (7) also indicates that the scaling for the row and column 1D IDCTs may be merged into a single step preceding the scaled 2D IDCT.

[0037] A scaled architecture is a design that uses a scaled transform, and a full architecture is a design that uses a full transform. A scaled architecture may have lower

multiplicative complexity than a full architecture. For example, a scaled architecture may perform a scaled 2D IDCT ($\underline{\mathbf{T}}_s^t \otimes \underline{\mathbf{T}}_s^t$) in a separable row-column fashion and may use an 8-point scaled ID IDCT $\underline{\mathbf{T}}_s$ from the AAN decomposition for each row and each column. The multiplicative complexity of this scaled architecture may be $8*8 + 16*5 = 64 + 80 = 144$, or 64 multiplications for scaling and 5 multiplications for each of the 8 rows and 8 columns. In certain situations, the scaling may be combined with quantization, in which case the multiplicative complexity of the scaled architecture may be reduced to about 80 multiplications. A full architecture may perform a 2D IDCT ($\underline{\mathbf{T}} \otimes \underline{\mathbf{T}}$) in a row-column fashion and may use the best-known full 8-point ID IDCT $\underline{\mathbf{T}}$ for each row and each column. The multiplicative complexity of this full architecture may be $16*11 = 176$, or 11 multiplications for each of the 8 rows and 8 columns. For the separable implementation, the scaled architecture may have lower multiplicative complexity than the full architecture.

[0038] A scaled architecture may be used in designs that favor low complexity. The scaled architecture may be advantageous when there are only few non-zero transform coefficients to scale, which may often be the case for 2D IDCT in an image/ video decoder. The scaled architecture may also be advantageous in designs that allow the scaling of transform coefficients to be combined with quantization and/or inverse quantization in an image/video encoder/decoder (codec), e.g., as shown in FIG. 8.

[0039] A full architecture may be desirable in designs that favor ease of use. For example, many computing environments and applications may support multiple image and video coding standards. In these instances, it may be more convenient to have a transform engine that implements full transform and to provide a flexible input/output interface to allow the transform engine to be used with various quantizers and codecs. The full architecture may provide simple interface and may be more suitable in such environments.

[0040] A transform design that can flexibly support different applications via scaled and full interfaces are described herein. The transform design may receive full input values via the full interface, perform a full transform on these input values, and provide full output values, similar to the full architecture. The transform design may also receive scaled input values via the scaled interface, perform a scaled transform on these input values, and provide scaled output values, similar to the scaled architecture. The

transform design may internally implement a separable scaled transform to potentially achieve lower complexity and/or improved precision. The transform design may thus be able to achieve lower complexity for certain applications, provide ease of use for other applications, or provide both lower complexity and ease of use in certain instances. The transform design may be used for both forward transforms (e.g., DCT) and inverse transforms (e.g., IDCT). For clarity, the transform design is specifically described below for IDCT.

[0041] The scaled and non-scaled/full 2D IDCT architectures may be expressed as follows.

$$\text{scaled 2D IDCT:} \quad \underline{\mathbf{T}}^{-1} \otimes \underline{\mathbf{T}}^{-1} = (\underline{\mathbf{T}}_s^t \otimes \underline{\mathbf{T}}_s^t) (\underline{\mathbf{S}} \otimes \underline{\mathbf{S}}) , \quad \text{and} \quad \text{Eq (8)}$$

$$\text{non-scaled 2D IDCT:} \quad \underline{\mathbf{T}}^{-1} \otimes \underline{\mathbf{T}}^{-1} = (\underline{\mathbf{T}}_s^t \underline{\mathbf{S}}) \otimes (\underline{\mathbf{T}}_s^t \underline{\mathbf{S}}) . \quad \text{Eq (9)}$$

[0042] A full/non-scaled interface may receive transform coefficients. A full 2D IDCT may be performed on these transform coefficients, as follows:

$$\underline{\mathbf{Y}} = \Theta(\underline{\mathbf{X}}) \approx \underline{\mathbf{T}}^t \underline{\mathbf{X}} \underline{\mathbf{T}} , \quad \text{Eq (10)}$$

where $\underline{\mathbf{X}}$ is a matrix of transform coefficients,

$\Theta(.)$ is an approximation of the full 2D IDCT, and

$\underline{\mathbf{Y}}$ is a matrix of output samples.

The operator notation $(.)$ in equation (10) is used to indicate that fixed-point approximations may not be based solely on linear operations.

[0043] The full 2D IDCT may be achieved by performing a full 1D IDCT on each row and each column of $\underline{\mathbf{X}}$, as follows:

$$\theta(\underline{\mathbf{x}}_i) \approx \underline{\mathbf{T}}^t \underline{\mathbf{x}}_i , \quad \text{Eq (11)}$$

where $\underline{\mathbf{x}}_i$ is the i -th row or column of $\underline{\mathbf{X}}$, and

$\theta(.)$ is an approximation of the full 1D IDCT.

$\theta(.)$ may be used for row-column implementation of 2D operator $\Theta(.)$.

[0044] A scaled interface may receive scaled transform coefficients, which may be obtained as follows:

$$\underline{\mathbf{X}}_S = \underline{\Sigma}(\underline{\mathbf{X}}) \approx \underline{\mathbf{S}}^t \underline{\mathbf{X}} \underline{\mathbf{S}}, \quad \text{Eq (12)}$$

where $\underline{\Sigma}(\cdot)$ is an approximation of the 2D scaling operation, and

$\underline{\mathbf{X}}_S$ is a matrix of scaled transform coefficients.

[0045] A scaled 2D IDCT may be performed on the scaled transform coefficients, as follows:

$$\underline{\mathbf{Y}} = \underline{\Xi}(\underline{\mathbf{X}}_S) = \underline{\mathbf{T}}_S^t \underline{\mathbf{X}}_S \underline{\mathbf{T}}_S, \quad \text{Eq (13)}$$

where $\underline{\Xi}(\cdot)$ is an approximation of the scaled 2D IDCT.

[0046] The scaled 2D IDCT may be achieved by performing a scaled 1D IDCT on each row and each column of $\underline{\mathbf{X}}_S$, as follows:

$$\xi(\underline{\mathbf{x}}_{S,i}) \approx \underline{\mathbf{T}}_S^t \underline{\mathbf{x}}_{S,i}, \quad \text{Eq (14)}$$

where $\underline{\mathbf{x}}_{S,i}$ is the i -th row or column of $\underline{\mathbf{X}}_S$, and

$\xi(\cdot)$ is an approximation of the scaled 1D IDCT.

$\xi(\cdot)$ may be used for row-column implementation of 2D operator $\underline{\Xi}(\cdot)$.

[0047] As shown in equation (13), the scaled interface may be realized by implementing the 2D operator $\underline{\Xi}(\cdot)$. As shown in equations (12) and (13), the full interface may be realized by implementing the 2D scaling operator $\underline{\Sigma}(\cdot)$ in addition to the 2D operator $\underline{\Xi}(\cdot)$. The full 2D IDCT may then be performed as follows:

$$\underline{\mathbf{Y}} = \underline{\Xi}(\underline{\Sigma}(\underline{\mathbf{X}})) =: \underline{\Theta}(\underline{\mathbf{X}}). \quad \text{Eq (15)}$$

[0048] Equation (15) indicates that a full 2D IDCT may be performed on full transform coefficients $\underline{\mathbf{X}}$ by first scaling these transform coefficients with the 2D scaling operator $\underline{\Sigma}(\cdot)$ and then performing a scaled 2D IDCT on the scaled transformed coefficients with the 2D operator $\underline{\Xi}(\cdot)$. The 2D operator $\underline{\Xi}(\cdot)$ may, in turn, be implemented by a row-column cascade of 1D operators $\xi(\cdot)$.

[0049] The 2D operator $\Theta(.)$ for a separable full 2D IDCT may thus be implemented with the 2D operator $\Xi(.)$ for a separable scaled 2D IDCT and the 2D scaling operator $\Sigma(.)$. The 2D scaling may be implemented in various manners, as described below. The resulting complexity and performance of the separable full 2D IDCT implemented with separable scaled 2D IDCT and 2D scaling may be comparable with a natively implemented full 2D IDCT.

[0050] **FIG. 1A** shows a design of a separable full 2D IDCT 100 with 2D scaling. 2D IDCT 100 comprises a 2D scaling stage 112, followed by a scaled 1D IDCT stage 114 for rows (or columns), further followed by a scaled 1D IDCT stage 116 for columns (or rows), and concluding with an output formatting stage 118. 2D scaling stage 112 receives an $N \times N$ block of transform coefficients $\underline{X}$ and may multiply each transform coefficient X_{ij} by a scale factor A_{ij} and further shift each scaled transform coefficient by P bits to the left, where P denotes the number of reserved “mantissa” bits. After the scaling, a quantity of $C = 2^{P-1}$ may be added to the DC transform coefficient to achieve the proper rounding in the output samples. To improve precision of scaling, $S = P + R$ bits may be used in the conversion of the scale factors to integers, and right shifts by R bits may be performed after multiplications. S may be any suitable value that can facilitate implementations on hardware platforms, e.g., S may be 15 or 16 for platforms with signed/unsigned 16-bit multipliers.

[0051] IDCT stage 114 performs an N -point scaled 1D IDCT on each row of the block of scaled transform coefficients from 2D scaling stage 112. IDCT stage 116 performs an N -point scaled 1D IDCT on each column of an intermediate block generated by IDCT stage 114. The scaled 1D IDCTs for stages 114 and 116 may operate directly on their input data without doing any internal pre- or post scaling. After all of the rows and columns are processed, output formatting stage 118 may shift the resulting quantities from IDCT stage 116 by P bits to the right to generate an $N \times N$ block of output samples $\underline{Y}$ for the full 2D IDCT. The scale factors and the precision constant P may be chosen such that the entire 2D IDCT may be implemented using registers of the desired width.

[0052] The 2D scaling in stage 112 may be expressed as:

$$X_{s,ij} = (X_{ij} \cdot A_i \cdot A_j) >> R, \quad \text{Eq (16)}$$

where X_{ij} is a transform coefficient in the i -th row and j -th column of $\underline{\mathbf{X}}$,

A_i and A_j are the i -th and j -th diagonal elements, respectively, of $\underline{\mathbf{S}}$,

$X_{S,ij}$ is a scaled transform coefficient in the i -th row and j -th column of $\underline{\mathbf{X}}_S$, and

“ $>> R$ ” denotes a signed right shift operation by R bits.

R is a constant ensuring P bits of added fixed-point precision in scaled transform coefficients $X_{S,ij}$.

[0053] A table may store scale factors $A_{ij} = A_i \cdot A_j$, for $i = 0, \dots, N-1$ and $j = 0, \dots, N-1$. Each element of $\underline{\mathbf{X}}$ may be multiplied by a corresponding scale factor in the table. Up to $N \cdot N$ multiplications may be performed for the $N \cdot N$ elements of $\underline{\mathbf{X}}$.

[0054] **FIG. 1B** shows a design of a separable full 2D IDCT 102 with separable row-column scaling. 2D IDCT 102 comprises a separable row-column scaling stage 122, followed by a scaled 1D IDCT stage 124 for rows (or columns), further followed by a scaled 1D IDCT stage 126 for columns (or rows), and concluding with an output formatting stage 128. Scaling stage 122 receives an $N \times N$ block of transform coefficients $\underline{\mathbf{X}}$ and may multiply each transform coefficient X_{ij} in each row i by a scale factor A_i and then multiply each resultant coefficient in each column j by a scale factor A_j to obtain scaled transform coefficients, as follows:

$$X'_{ij} = X_{ij} \cdot A_i, \quad \text{Eq (17)}$$

$$X_{S,ij} = (X'_{ij} \cdot A_j) >> R.$$

[0055] Scaling stage 122 may thus perform 2D scaling in a separable manner on rows followed by columns (or on columns followed by rows). Separable row-column scaling may allow the same hardware to be used for scaling of rows and scaling of columns, which may reduce implementation complexity. Up to $2 \cdot N \cdot N$ multiplications may be performed for the $N \cdot N$ elements of $\underline{\mathbf{X}}$. However, the actual number of multiplications may be much fewer than $2 \cdot N \cdot N$ since some of the scale factors A_0 through A_{N-1} may have trivial values (e.g., 256), and multiplication with these trivial scale factors may be implemented with simple shift operations. Stages 124, 126, and 128 may operate in the same manner as stages 114, 116, and 118, respectively, in FIG. 1A.

[0056] **FIG. 1C** shows a design of a separable full 2D IDCT 104 with scaling prior to each scaled 1D IDCT. 2D IDCT 104 comprises a scaled 1D IDCT with scaling stage 134 for rows (or columns), followed by a scaled 1D IDCT with scaling stage 136 for columns (or rows), and concluding with an output formatting stage 138. IDCT stage 134 performs scaling prior to an N-point scaled 1D IDCT on each row of a block of transform coefficients. IDCT stage 136 performs scaling prior to an N-point scaled 1D IDCT on each column of an intermediate block generated by IDCT stage 134. Stages 134 and 136 essentially perform full 1D IDCTs using scaled 1D IDCTs. The scale factors A_0 through A_{N-1} and the constant factors within the scaled 1D IDCT may be selected to reduce complexity and/or improve precision for the full 1D IDCT, as described below. Stage 138 may operate in the same manner as stage 118 in FIG. 1A.

[0057] **FIG. 1D** shows a design of a separable scaled 2D IDCT 106. 2D IDCT 106 comprises a scaled 1D IDCT stage 144 for rows (or columns), followed by a scaled 1D IDCT stage 146 for columns (or rows), and concluding with an output formatting stage 148. IDCT stage 144 performs an N-point scaled 1D IDCT on each row of an $N \times N$ block of scaled transform coefficients $\underline{X}_s$. IDCT stage 146 performs an N-point scaled 1D IDCT on each column of an intermediate block generated by IDCT stage 144. Stage 148 may operate in the same manner as stage 118 in FIG. 1A.

[0058] As shown in FIGS. 1A to 1C, the scaling for the full 2D IDCT may be achieved in various manners such as with 2D scaling prior to the row-column 1D IDCTs in FIG. 1A, with separable row-column scaling prior to the row-column 1D IDCTs in FIG. 1B, or with scaling prior to each 1D IDCT in FIG. 1C. The scaling may also be performed in other manners. As shown in FIG. 1D, the scaled 2D IDCT may be achieved by simply omitting the scaling and performing 1D IDCTs for rows and columns.

[0059] Various types of scaled 1D IDCT may be used for the row-column 1D IDCTs in FIGS. 1A to 1D. For example, a scaled 1D IDCT based on the AAN decomposition may be used. Another scaled IDCT with possibly lower complexity is described below.

[0060] **FIG. 2** shows a flow graph 200 of an example factorization of an 8-point 1D IDCT. In flow graph 200, each addition is represented by symbol “ $\oplus$ ” and each multiplication is represented by a box. Each addition sums or subtracts two input values

and provides an output value. Each multiplication multiplies an input value with a constant factor shown inside the box and provides an output value. The factorization in FIG. 2 has multiplications with the following constant factors:

$$\alpha_1 = \alpha_2 = 1, \quad \text{Eq (18)}$$

$$\beta_1 = \beta_2 = C_{\pi/4} = \cos(\pi/4) \approx 0.707106781$$

$$\gamma_1 = C_{3\pi/8} = \cos(3\pi/8) \approx 0.382683432, \quad \text{and}$$

$$\delta_1 = S_{3\pi/8} = \sin(3\pi/8) \approx 0.923879533.$$

[0061] Flow graph 200 may receive eight transform coefficients $X[0]$ through $X[7]$ and scale these transform coefficients with scale factors A_0 through A_7 to obtain eight scaled transform coefficients $A_0 \cdot X[0]$ through $A_7 \cdot X[7]$. Alternatively, flow graph 200 may receive the eight scaled transform coefficients directly. In either case, flow graph 200 performs an 8-point 1D IDCT on the eight scaled transform coefficients and generates eight output samples $x[0]$ through $x[7]$. The scale factors A_0 through A_7 are as follows:

$$\begin{aligned} A_0 &= \frac{1}{2\sqrt{2}} \approx 0.3535533906, & A_1 &= \frac{\cos(7\pi/16)}{2\sin(3\pi/8) - \sqrt{2}} \approx 0.4499881115, \\ A_2 &= \frac{\cos(\pi/8)}{\sqrt{2}} \approx 0.6532814824, & A_3 &= \frac{\cos(5\pi/16)}{\sqrt{2} + 2\cos(3\pi/8)} \approx 0.2548977895, \\ A_4 &= \frac{1}{2\sqrt{2}} \approx 0.3535533906, & A_5 &= \frac{\cos(3\pi/16)}{\sqrt{2} - 2\cos(3\pi/8)} \approx 1.2814577239, \\ A_6 &= \frac{\cos(3\pi/8)}{\sqrt{2}} \approx 0.2705980501, & A_7 &= \frac{\cos(\pi/16)}{\sqrt{2} + 2\sin(3\pi/8)} \approx 0.3006724435. \end{aligned}$$

[0062] Flow graph 200 includes a number of butterfly operations. A butterfly operation receives two input values and generates two output values, where one output value is the sum of the two input values and the other output value is the difference of the two input values. For example, the butterfly operation on input values $A_0 \cdot X[0]$ and

$A_4 \cdot X[4]$ generates an output value $A_0 \cdot X[0] + A_4 \cdot X[4]$ for the top branch and an output value $A_0 \cdot X[0] - A_4 \cdot X[4]$ for the bottom branch.

[0063] The factorization shown in FIG. 2 results in a total of 6 multiplications and 28 additions, which are substantially fewer than the number of multiplications and additions required for direct computation of equation (2). The multiplications are with irrational constants representing the sine and cosine of different angles, which are multiples of $\pi/8$ for the 8-point 1D IDCT. An irrational constant is a constant that is not a ratio of two integers. The multiplications with irrational constants may be efficiently performed in fixed-point integer arithmetic when each irrational constant is approximated by a rational dyadic constant. A rational dyadic constant is a rational constant with a dyadic denominator and has the form $c/2^b$, where b and c are integers and $b > 0$.

[0064] Multiplication of an integer variable x with an irrational constant μ in fixed-point integer arithmetic may be achieved by approximating the irrational constant with a rational dyadic constant, as follows:

$$\mu \approx c/2^b, \quad \text{Eq (19)}$$

where μ is the irrational constant to be approximated, and

$c/2^b$ is the rational dyadic constant.

[0065] Given integer variable x and rational dyadic constant $u = c/2^b$, an integer-valued product

$$y = (x \cdot c)/2^b \quad \text{Eq (20)}$$

may be approximated using a series of intermediate values

$$x_0, x_1, x_2, \dots, x_t, \quad \text{Eq (21)}$$

where $x_0 = 0$, $x_1 = x$, and for all $2 \leq i \leq t$ values, x_i is obtained as follows:

$$x_i = \pm x_j \pm x_k \cdot 2^{s_i}, \quad \text{with } j, k < i, \quad \text{Eq (22)}$$

where $x_k \cdot 2^{s_i}$ implies either left or right shift (depending on the sign of constant s_i) of intermediate value x_k by $|s_i|$ bits.

[0066] In equation (22), x_i may be equal to $x_j + x_k \cdot 2^{s_i}$, $x_j - x_k \cdot 2^{s_i}$, or $-x_j + x_k \cdot 2^{s_i}$. Each intermediate value x_i in the series may be derived based on two prior intermediate values x_j and x_k in the series, where either x_j or x_k may be equal to zero. Each intermediate value x_i may be obtained with one shift and/or one addition. The shift is not needed if s_i is equal to zero. The addition is not needed if $x_j = x_0 = 0$. The total number of additions and shifts for the multiplication is determined by the number of intermediate values in the series, which is t , as well as the expression used for each intermediate value. The multiplication by rational dyadic constant u is essentially unrolled into a series of shift and add operations. The series is defined such that the final value in the series becomes the desired integer-valued product, or

$$x_t \approx y. \quad \text{Eq (23)}$$

[0067] Multiplication of an integer variable x with two irrational constants μ and η in fixed-point integer arithmetic may be achieved by approximating the irrational constants with rational dyadic constants, as follows:

$$\mu \approx c/2^b \quad \text{and} \quad \eta \approx e/2^d, \quad \text{Eq (24)}$$

where $c/2^b$ and $e/2^d$ are two rational dyadic constants, and

b, c, d and e are integers, with $b > 0$ and $d > 0$.

[0068] Given integer variable x and rational dyadic constants $u = c/2^b$ and $v = e/2^d$, two integer-valued products

$$y = (x \cdot c)/2^b \quad \text{and} \quad z = (x \cdot e)/2^d \quad \text{Eq (25)}$$

may be approximated using a series of intermediate values

$$x_0, x_1, x_2, \dots, x_t, \quad \text{Eq (26)}$$

where $x_0 = 0$, $x_1 = x$, and for all $2 \leq i \leq t$ values, x_i is obtained as follows:

$$x_i = \pm x_j \pm x_k \cdot 2^{s_i}, \quad \text{with } j, k < i. \quad \text{Eq (27)}$$

The series is defined such that the desired integer-valued products are obtained at steps m and n , as follows:

$$x_m \approx y \quad \text{and} \quad x_n \approx z, \quad \text{Eq (28)}$$

where $m, n \leq t$ and either m or n is equal to t .

[0069] As shown in equations (24) through (28), the multiplication of integer variable x with irrational constants μ and η may be approximated with a common series of intermediate values generated by shift and add operations and using intermediate results to reduce the total number of operations.

[0070] In the computation described above, trivial operations such as additions and subtractions of zeros and shifts by zero bits may be omitted. The following simplifications may be made:

$$x_i = \pm x_0 \pm x_k \cdot 2^{s_i} \quad \Rightarrow \quad x_i = \pm x_k \cdot 2^{s_i}, \quad \text{Eq (29)}$$

$$x_i = \pm x_j \pm x_k \cdot 2^0 \quad \Rightarrow \quad x_i = \pm x_j \pm x_k. \quad \text{Eq (30)}$$

[0071] In equation (29), the expression to the left of “ $\Rightarrow$ ” involves an addition or subtraction of zero (denoted by x_0) and may be performed with one shift, as shown by the expression to the right of “ $\Rightarrow$ ”. In equation (30), the expression to the left of “ $\Rightarrow$ ” involves a shift by zero bits (denoted by 2^0) and may be performed with one addition, as shown by the expression to the right of “ $\Rightarrow$ ”. Equations (29) and (30) may be applied to equations (22) and (27) in the computation of x_i .

[0072] To reduce computation, a first common factor F_1 may be applied to constant factors α_1 and β_1 in flow graph 200, and second common factor F_2 may be applied to constant factors α_2 , β_2 , δ_1 and γ_1 , as follows:

$$\alpha'_1 = \alpha_1 \cdot F_1, \quad \beta'_1 = \beta_1 \cdot F_1, \quad \text{and} \quad \text{Eq (31)}$$

$$\alpha'_2 = \alpha_2 \cdot F_2, \quad \beta'_2 = \beta_2 \cdot F_2, \quad \gamma'_1 = \gamma_1 \cdot F_2, \quad \delta'_1 = \delta_1 \cdot F_2.$$

[0073] The scale factors A_0 through A_7 may also be scaled to account for the common factors F_1 and F_2 , as follows:

$$A'_0 = A_0, \quad A'_4 = A_4, \quad \text{Eq (32)}$$

$$A'_2 = A_2 / F_1, \quad A'_6 = A_6 / F_1, \quad \text{and}$$

$$A'_1 = A_1 / F_2, \quad A'_5 = A_5 / F_2, \quad A'_3 = A_3 / F_2, \quad A'_7 = A_7 / F_2.$$

[0074] Different combinations of values for common factors F_1 and F_2 may be evaluated. For each combination of F_1 and F_2 values, the total number of logical and arithmetic operations for the 1D IDCT and the precision of the output samples may be determined.

[0075] Table 1 shows an example fixed-point approximation for the 1D IDCT in FIG. 2 with $F_1 = 2523/2048$ and $F_2 = 2607/2048$. In Table 1, the scale factors A'_0 through A'_7 and the scaled common factors are given in the first column. Each factor may be approximated with a rational dyadic constant given in the second column. The series of intermediate values for multiplication of variable x with one or two rational dyadic constants is given in the third column. The numbers of add and shift operations for each multiplication are given in the fourth and fifth columns, respectively. The number of times each multiplication is used for the 1D IDCT is given in the sixth column. The total number of add operations for the 1D IDCT is equal to the sum of the number of add operations in the fourth column times the number given in the sixth column. The total number of shift operations for the 1D IDCT is equal to the sum of the number shift operations in the fifth column times the number given in the sixth column.

Table 1

Factor	Rational Dyadic Constant	Multiplication of variable x with one or two rational dyadic constants	Num of Adds	Num of Shifts	Times Used
A'_0	256	$y=x<<8;$	0	1	1
A'_1	256	$y=x<<8;$	0	1	1
A'_2	384	$y=(x+(x<<1))<<7;$	1	2	1
A'_3	145	$x_2=x+(x<<3); y=x+(x_2<<4);$	2	2	1

A'_4	256	$y=x<<8$;	0	1	1
A'_5	729	$x_2=x+(x<<2)$; $x_3=x+(x_2<<4)$; $y=x_3+(x_3<<3)$;	3	3	1
A'_6	159	$x_2=x+(x<<2)$; $y=(x_2<<5)-x$;	2	2	1
A'_7	171	$x_2=(x<<1)+(x<<3)$; $x_3=x+x_2$; $y=x_3+(x_2<<4)$;	3	3	1
α'_1	2523/2048	$x_2=x-(x>>9)$; $x_3=x_2-(x>>4)$; $y=x_2+(x_3>>2)$;	3	3	1
β'_1	223/256	$x_2=x+(x>>5)$; $y=x-(x_2>>3)$;	2	2	1
α'_2	2607/2048	$x_2=x-(x>>6)$; $x_3=x_2+(x>>5)$; $y=x_3+(x_3>>5)$;	3	3	1
β'_2	3687/4096	$x_2=x+(x>>9)$; $x_3=x_2-(x>>2)$; $x_4=x_3+(x>>4)$; $y=x_2-(x_4>>3)$;	4	4	1
γ'_1	1995/4096	$x_2=x+(x>>5)$; $x_3=x_2+(x>>7)$; $x_4=x_3+(x_2>>3)$;	5	6	2
δ'_1	4817/4096	$y=x_4+(x_2>>7)$; $z=(x_3-(x_3>>4))>>1$;			
Complexity of scaled 1D IDCT:			50	24	
Complexity of full 1D IDCT:			61	39	

[0076] In Table 1, right shift by 8 cancels out multiplication by a factor of 256 in scale factors A'_0 , A'_1 , A'_2 , and A'_4 , which enables additional complexity reduction.

[0077] In the design shown in Table 1, an 8-point scaled 1D IDCT may be performed with 50 add operations and 24 shift operations. An 8-point full 1D IDCT may be performed with 61 add operations and 39 shift operations. The scaling with scale factors A'_0 through A'_7 may be performed prior to the scaled 1D IDCT (as shown in FIG. 1C) or may be performed prior to the row-column 1D IDCTs (as shown in FIGS. 1A and 1B). An 8×8 full 2D IDCT may be performed with 977 add operations and 688 shift operations. The 977 add operations include 61 adds for each of 16 1D IDCTs for the 8 rows and 8 columns plus 1 add to sum 2^{P-1} with the DC transform coefficient after scaling. The 688 shift operations include 39 shifts for each of 16 IDCTs for the 8 rows and 8 columns plus 64 shifts to shift the 64 values from the second IDCT stage by P bits. The design shown in Table 1 meets or exceeds precision metrics defined in IEEE Standard 1180-1190 and its pending replacement.

[0078] Table 1 shows an example fixed-point approximation for the 1D IDCT in FIG. 2. The scale factors A_0 through A_7 and the constant factors α_1 , β_1 , α_2 , β_2 , δ_1 and γ_1 may also be approximated with other rational dyadic constants, which may have

different complexity and/or precision. Other types of scaled IDCT may also be used to implement separable scaled and full 2D IDCTs.

[0079] **FIG. 3A** shows a design of a separable full 2D DCT 300 with 2D scaling. 2D DCT 300 comprises an input formatting stage 312, followed by a scaled 1D DCT stage 314 for rows (or columns), further followed by a scaled 1D DCT stage 316 for columns (or rows), and concluding with a 2D scaling stage 318. Input formatting stage 312 may pre-multiply an $N \times N$ block of input samples. DCT stage 314 performs an N -point scaled 1D DCT on each row of a block of pre-multiplied samples from stage 312 and generates a first intermediate block. DCT stage 316 performs an N -point scaled 1D DCT on each column of the first intermediate block and generates a second intermediate block. Scaling stage 318 scales each element of the second intermediate matrix and provides a block of transformed coefficients for the full 2D DCT.

[0080] **FIG. 3B** shows a design of a separable full 2D DCT 302 with separable row-column scaling. 2D DCT 302 comprises an input formatting stage 322, followed by a scaled 1D DCT stage 324 for rows (or columns), further followed by a scaled 1D DCT stage 326 for columns (or rows), and concluding with a separable row-column scaling stage 328. Stages 322, 324 and 326 may operate in the same manner as stages 312, 314 and 316, respectively, in FIG. 3A. Scaling stage 328 may scale the second intermediate block from DCT stage 326 row-wise and then column-wise to generate the block of transformed coefficients for the full 2D DCT.

[0081] **FIG. 3C** shows a design of a separable full 2D DCT 304 with scaling after each scaled 1D DCT. 2D DCT 304 comprises an input formatting stage 332, followed by a scaled 1D DCT with scaling stage 334 for rows (or columns), and further followed by a scaled 1D DCT with scaling stage 336 for columns (or rows). Stage 332 may operate in the same manner as stage 312 in FIG. 3A. DCT stage 334 performs an N -point scaled 1D DCT followed by scaling on each row of a block of pre-multiplied samples from stage 332. DCT stage 336 performs an N -point scaled 1D DCT followed by scaling on each column of an intermediate block generated by DCT stage 334.

[0082] **FIG. 3D** shows a design of a separable scaled 2D DCT 306. 2D DCT 306 comprises an input formatting stage 342, followed by a scaled 1D DCT stage 344 for rows (or columns), and further followed by a scaled 1D DCT stage 346 for columns (or rows). Stage 342 may operate in the same manner as stage 312 in FIG. 3A. DCT stage 344 performs an N -point scaled 1D DCT on each row of a block of pre-multiplied

samples from stage 342. DCT stage 346 performs an N-point scaled 1D DCT on each column of an intermediate block generated by DCT stage 344.

[0083] As shown in FIGS. 3A to 3C, the scaling for the full 2D DCT may be achieved in various manners such as with 2D scaling after the row-column 1D DCTs in FIG. 3A, with separable row-column scaling after the row-column 1D DCTs in FIG. 3B, or with scaling after each 1D DCT in FIG. 3C. The scaling may also be performed in other manners. As shown in FIG. 3D, the scaled 2D DCT may be achieved by simply omitting the scaling and performing 1D DCTs for rows and columns.

[0084] Various types of scaled 1D DCT may be used for the scaled 1D DCTs in FIGS. 3A to 3D. An example scaled DCT with low complexity is described below.

[0085] FIG. 4 shows a flow graph 400 of an example factorization of an 8-point 1D DCT. Flow graph 400 receives eight input samples $x[0]$ through $x[7]$, performs an 8-point scaled 1D DCT on these input samples, and generates eight scaled transform coefficients $8A_0 \cdot X[0]$ through $8A_7 \cdot X[7]$. The scale factors A_0 through A_7 are given above. The factorization in FIG. 4 has multiplications with the following constant factors:

$$\alpha_a = \alpha_b = 1 , \quad \text{Eq (33)}$$

$$\beta_a = \beta_b = 1 / C_{\pi/4} = 1 / \cos(\pi/4)$$

$$\gamma_a = 2C_{3\pi/8} = 2 \cos(3\pi/8) , \quad \text{and}$$

$$\delta_a = 2S_{3\pi/8} = 2 \sin(3\pi/8) .$$

[0086] The flow graphs for the IDCT and DCT in FIGS. 2 and 4 are similar and involve multiplications by essentially the same constant factors (with the difference in 1/2). Such similarity may be advantageous for implementation of the DCT and IDCT on an integrated circuit. In particular, the similarity may enable savings of silicon or die area to implement the butterflies and the multiplications by transform constants, which are used in both the forward and inverse transforms.

[0087] FIG. 5 shows a block diagram of a design of an IDCT processor 500 that supports full and scaled interfaces. Within IDCT processor 500, a scaling unit 510 receives a block of transform coefficients via the full interface, performs scaling (e.g., element-wise 2D scaling or separable row-column scaling) on the block of transform

coefficients, and provides a block of scaled transform coefficients. A multiplexer (Mux) 512 receives and provides the scaled transform coefficients from scaling unit 510 when the full interface is selected. Multiplexer 512 also receives scaled transform coefficients via the scaled interface and provides these scaled transform coefficients when the scaled interface is selected. A multiplexer 514 provides the output of multiplexer 512 or the output of a buffer 518 to an IDCT engine 516.

[0088] IDCT engine 516 may perform scaled 1D IDCTs on rows of the block of scaled transform coefficients from multiplexer 512 and provide rows of a block of intermediate results to buffer 518. IDCT engine 516 may then perform scaled 1D IDCTs on columns of the block of intermediate results from buffer 518 and provide a block of final results to buffer 518. A unit 520 performs output formatting on the final results in buffer 518 and provides output samples.

[0089] **FIG. 6** shows a block diagram of a design of a DCT processor 600 that supports full and scaled interfaces. Within DCT processor 600, a unit 610 performs input formatting on a block of input samples. A DCT engine 614 receives the block of input values from unit 610 via a multiplexer 612, performs scaled 1D DCTs on rows of the block of input values, and provides a block of intermediate results to a buffer 616. DCT engine 614 may then perform scaled 1D DCTs on columns of the block of intermediate results from buffer 616 and provide a block of scaled transform coefficients to buffer 616. Buffer 616 provides the block of scaled transform coefficients for the scaled interface. A scaling unit 618 receives and scales the block of scaled transform coefficients and provides full transform coefficients for the full interface.

[0090] For clarity, much of the description above is for DCT and IDCT. In general, the techniques described herein may be used for any type of transform such as DCT, IDCT, DFT, IDFT, MLT, inverse MLT, MCLT, inverse MCLT, etc. The techniques may also be used for any factorization of a transform, with several example factorizations being given in FIGS. 2 and 4. The techniques may also be used for transforms of any size, with 8-point transforms being given in FIGS. 2 and 4. 8×8 DCT and 8×8 IDCT are commonly used for image and video processing. The techniques may also be used for various image and video coding standards such as JPEG, MPEG-1, MPEG-2, MPEG-4 (P.2), H.261, H.263, etc.

[0091] FIG. 7 shows a design of a process 700 for performing transform. First input values may be received via a first interface (a full or non-scaled interface) (block 712). A full transform may be performed on the first input values to obtain first output values (block 714). Second input values may be received via a second interface (a scaled interface) (block 716). A scaled transform may be performed on the second input values to obtain second output values (block 718).

[0092] The full transform may be a full 2D inverse transform. In this case, a block of first input values may be received via the first interface and scaled to obtain a block of scaled input values. A scaled 1D transform may be performed on each row of the block of scaled input values to obtain an intermediate block. A scaled 1D transform may be performed on each column of the intermediate block to obtain a block of first output values. The scaled 1D transforms may also be performed on columns followed by rows.

[0093] The full transform may also be a full 2D forward transform. In this case, a block of first input values may be received via the first interface. A scaled 1D transform may be performed on each row of the block of input values to obtain a first intermediate block. A scaled 1D transform may be performed on each column of the first intermediate block to obtain a second intermediate block. The second intermediate block may be scaled to obtain a block of first output values.

[0094] The full transform may be a full 2D IDCT. In this case, a block of transform coefficients may be received via the first interface and scaled to obtain a block of scaled transform coefficients. Each transform coefficient may be scaled with a respective scale factor to obtain a corresponding scaled transform coefficient, e.g., as shown in FIG. 1A. Alternatively, the block of transform coefficients may be scaled row-wise and column-wise to obtain the block of scaled transform coefficients, e.g., as shown in FIG. 1B. A scaled 1D IDCT may be performed on each row of the block of scaled transform coefficients to obtain an intermediate block. A scaled 1D IDCT may be performed on each column of the intermediate block to obtain a block of output samples. Alternatively, scaling and scaled one 1D IDCT may be performed on each row of the block of transform coefficients to obtain an intermediate block, and scaling and scaled 1D IDCT may be performed on each column of the intermediate block, e.g., as shown in FIG. 1C.

[0095] The scaled transform may be a scaled 2D IDCT. In this case, a block of scaled transform coefficients may be received via the second interface. A scaled 1D IDCT may then be performed on each row of the block of scaled transform coefficients to obtain an intermediate block. A scaled 1D IDCT may be performed on each column of the intermediate block to obtain a block of output samples.

[0096] The full transform may be a full 2D DCT, and the scaled transform may be a scaled 2D DCT. A full 2D DCT may be performed on a block of input samples as described above to obtain a block of full transform coefficients for the first interface. A scaled 2D DCT may also be performed on the block of input samples to obtain a block of scaled transform coefficients for the second interface.

[0097] **FIG. 8** shows a block diagram of a design of an encoding system 810 and a decoding system 850. At encoding system 810, an image/video encoder 820 may receive blocks of pixels and perform compression on each pixel block in accordance with a particular image or video compression algorithm. Encoder 820 may provide blocks of pixel difference values (or residuals) and blocks of pixels. A unit 822 may receive the residual blocks and pixel blocks as blocks of input samples, perform 2D DCT on each input sample block, and provide blocks of full or scaled transform coefficients. Unit 822 may support both full and scaled interfaces. A quantizer 824 may quantize the full or scaled transform coefficients and provide quantized coefficients. An entropy encoder 826 may perform entropy encoding on the quantized coefficients and provide compressed data in packets or a bit stream for transmission via a communication channel 840 and/or storage.

[0098] At decoding system 850, an entropy decoder 860 may perform entropy decoding on the compressed data in a manner complementary to the entropy encoding by encoder 826 and provide quantized coefficients. An inverse quantizer 862 may map the quantized coefficients to full or scaled transform coefficients. A unit 862 may perform full 2D IDCT on full transform coefficients or scaled 2D IDCT on scaled transform coefficients and provide blocks of output samples. Unit 862 may support both full and scaled interfaces. An image/video decoder 866 may perform decompression on the blocks of output samples and provide blocks of pixels.

[0099] At encoding system 810, unit 822 may perform scaled 2D DCT and provide scaled transform coefficients. Quantizer 824 may perform scaling (e.g., stage 318 in FIG. 3A or stage 328 in FIG. 3B) as well as quantization of the scaled transform

coefficients. At decoding system 850, unit 862 may perform inverse quantization as well as scaling (e.g., stage 112 in FIG. 1A or stage 122 in FIG. 1B) for the transform coefficients. Unit 864 may then perform scaled 2D IDCT on the scaled transform coefficients from unit 862.

[00100] FIG. 9 shows a block diagram of an encoding system 900, which may be used for encoding system 810 in FIG. 8. A capture device/memory 910 may receive a source signal, perform conversion to digital format, and provides input/raw data. Capture device 910 may be a video camera, a digitizer, or some other device. A processor 920 processes the raw data and generates compressed data. Within processor 920, the raw data may be transformed by a DCT unit 922, scanned by a zig-zag scan unit 924, quantized by a quantizer 926, encoded by an entropy encoder 928, and packetized by a packetizer 930. DCT unit 922 may perform 2D DCTs on the raw data in accordance with the techniques described herein and may support both full and scaled interfaces. Each of units 922 through 930 may be implemented a hardware, firmware and/or software. For example, DCT unit 922 may be implemented with dedicated hardware, a set of instructions for an arithmetic logic unit (ALU), etc.

[00101] A storage unit 940 may store the compressed data from processor 920. A transmitter 942 may transmit the compressed data. A controller/processor 950 controls the operation of various units in encoding system 900. A memory 952 stores data and program codes for encoding system 900. One or more buses 960 interconnect various units in encoding system 900.

[00102] FIG. 10 shows a block diagram of a decoding system 1000, which may be used for decoding system 850 in FIG. 8. A receiver 1010 may receive compressed data from an encoding system, and a storage unit 1012 may store the received compressed data. A processor 1020 processes the compressed data and generates output data. Within processor 1020, the compressed data may be de-packetized by a de-packetizer 1022, decoded by an entropy decoder 1024, inverse quantized by an inverse quantizer 1026, placed in the proper order by an inverse zig-zag scan unit 1028, and transformed by an IDCT unit 1030. IDCT unit 1030 may perform 2D IDCTs on the full or scaled transform coefficients in accordance with the techniques described herein and may support both full and scaled interfaces. Each of units 1022 through 1030 may be implemented a hardware, firmware and/or software. For example, IDCT unit 1030 may be implemented with dedicated hardware, a set of instructions for an ALU, etc.

[00103] A display unit 1040 displays reconstructed images and video from processor 1020. A controller/processor 1050 controls the operation of various units in decoding system 1000. A memory 1052 stores data and program codes for decoding system 1000. One or more buses 1060 interconnect various units in decoding system 1000.

[00104] Processors 920 and 1020 may each be implemented with one or more application specific integrated circuits (ASICs), digital signal processors (DSPs), and/or some other type of processors. Alternatively, processors 920 and 1020 may each be replaced with one or more random access memories (RAMs), read only memory (ROMs), electrical programmable ROMs (EPROMs), electrically erasable programmable ROMs (EEPROMs), magnetic disks, optical disks, and/or other types of volatile and nonvolatile memories known in the art.

[00105] The techniques described herein may be implemented in various types of apparatus. For example, the techniques may be implemented in different types of processors, different types of integrated circuits, different types of electronics devices, different types of electronics circuits, etc.

[00106] Those of skill in the art would understand that information and signals may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

[00107] Those of skill would further appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the disclosure may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present disclosure.

[00108] The various illustrative logical blocks, modules, and circuits described in connection with the disclosure may be implemented or performed with a general-purpose processor, a DSP, an ASIC, a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general-purpose processor may be a microprocessor, but in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[00109] The steps of a method or algorithm described in connection with the disclosure may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium is coupled to the processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a user terminal.

[00110] The previous description of the disclosure is provided to enable any person skilled in the art to make or use the present disclosure. Various modifications to the disclosure may be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other designs without departing from the spirit or scope of the disclosure. Thus, the present disclosure is not intended to be limited to the examples shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

[00111] **WHAT IS CLAIMED IS:**

CLAIMS

1. An apparatus comprising:
a processor configured to receive first input values via a first interface, to perform a full transform on the first input values to obtain first output values, to receive second input values via a second interface, and to perform a scaled transform on the second input values to obtain second output values; and
a memory coupled to the processor.
2. The apparatus of claim 1, wherein the full transform is a full two-dimensional (2D) transform, and wherein the processor is configured to receive a block of first input values via the first interface, to scale the block of first input values to obtain a block of scaled input values, to perform a scaled one-dimensional (1D) transform on each row of the block of scaled input values to obtain an intermediate block, and to perform a scaled 1D transform on each column of the intermediate block to obtain a block of first output values.
3. The apparatus of claim 1, wherein the full transform is a full two-dimensional (2D) transform, and wherein the processor is configured to receive a block of first input values via the first interface, to perform a scaled one-dimensional (1D) transform on each row of the block of input values to obtain a first intermediate block, to perform a scaled 1D transform on each column of the first intermediate block to obtain a second intermediate block, and to scale the second intermediate block to obtain a block of first output values.
4. The apparatus of claim 1, wherein the full transform is a full inverse discrete cosine transform (IDCT) and the scaled transform is a scaled IDCT.
5. The apparatus of claim 1, wherein the full transform is a full two-dimensional (2D) inverse discrete cosine transform (IDCT) and the scaled transform is a scaled 2D IDCT.

6. The apparatus of claim 5, wherein the processor is configured to receive a first block of transform coefficients as the first input values via the first interface, to scale the first block of transform coefficients to obtain a second block of scaled transform coefficients, to perform a scaled one-dimensional (1D) IDCT on each row of the second block to obtain an intermediate block, and to perform a scaled 1D IDCT on each column of the intermediate block.

7. The apparatus of claim 6, the processor is configured to scale each transform coefficient in the first block with a respective scale factor to obtain a corresponding scaled transform coefficient in the second block.

8. The apparatus of claim 6, the processor is configured to scale the first block of transform coefficients row-wise and column-wise to obtain the second block of scaled transform coefficients.

9. The apparatus of claim 5, wherein the processor is configured to receive a block of transform coefficients as the first input values via the first interface, to perform scaling and a scaled one-dimensional (1D) IDCT on each row of the block of transform coefficients to obtain an intermediate block, and to perform scaling and a scaled 1D IDCT on each column of the intermediate block.

10. The apparatus of claim 5, wherein the processor is configured to receive a block of scaled transform coefficients as the second input values via the second interface, to perform a scaled one-dimensional (1D) IDCT on each row of the block of scaled transform coefficients to obtain an intermediate block, and to perform a scaled 1D IDCT on each column of the intermediate block.

11. The apparatus of claim 1, wherein the full transform is a full discrete cosine transform (DCT) and the scaled transform is a scaled DCT.

12. The apparatus of claim 1, wherein the full transform is a full two-dimensional (2D) discrete cosine transform (DCT) and the scaled transform is a scaled 2D DCT.

13. The apparatus of claim 12, wherein the processor is configured to receive a block of input samples as the first input values via the first interface, to perform a scaled one-dimensional (1D) DCT on each row of the block of input samples to obtain a first intermediate block, to perform a scaled 1D DCT on each column of the first intermediate block to obtain a second intermediate block, to scale the second intermediate block to obtain a block of transform coefficients, and to provide the block of transform coefficients as the first output values via the first interface.

14. The apparatus of claim 13, wherein the processor is configured to scale each element in the second intermediate block with a respective scale factor to obtain a corresponding transform coefficient in the block of transform coefficients.

15. The apparatus of claim 13, wherein the processor is configured to scale the second intermediate block row-wise and column-wise to obtain the block of transform coefficients.

16. The apparatus of claim 12, wherein the processor is configured to receive a block of input samples as the first input values via the first interface, to perform a scaled one-dimensional (1D) DCT and scaling on each row of the block of input samples to obtain an intermediate block, to perform a scaled 1D DCT and scaling on each column of the intermediate block to obtain a block of transform coefficients, and to provide the block of transform coefficients as the first output values via the first interface.

17. The apparatus of claim 12, wherein the processor is configured to receive a block of input samples as the second input values via the second interface, to perform a scaled one-dimensional (1D) DCT on each row of the block of input samples to obtain an intermediate block, to perform a scaled 1D DCT on each column of the intermediate block to obtain a block of scaled transform coefficients, and to provide the block of scaled transform coefficients as the second output values via the second interface.

18. A method comprising:

receiving first input values via a first interface;
performing a full transform on the first input values to obtain first output values;
receiving second input values via a second interface; and
performing a scaled transform on the second input values to obtain second output values.

19. The method of claim 18, wherein the full transform is a full two-dimensional (2D) transform, and wherein the performing a full transform on the first input values comprises

scaling a block of first input values to obtain a block of scaled input values,
performing a scaled one-dimensional (1D) transform on each row of the block of scaled input values to obtain an intermediate block, and
performing a scaled 1D transform on each column of the intermediate block to obtain a block of first output values.

20. The method of claim 18, wherein the full transform is a full two-dimensional (2D) transform, and wherein the performing a full transform on the first input values comprises

performing a scaled one-dimensional (1D) transform on each row of a block of input values to obtain a first intermediate block,
performing a scaled 1D transform on each column of the first intermediate block to obtain a second intermediate block, and
scaling the second intermediate block to obtain a block of first output values.

21. The method of claim 18, wherein the full transform is a full two-dimensional (2D) inverse discrete cosine transform (IDCT), wherein the receiving the first input values via the first interface comprises receiving a first block of transform coefficients as the first input values via the first interface, and wherein the performing a full transform on the first input values comprises

scaling the first block of transform coefficients to obtain a second block of scaled transform coefficients,
performing a scaled one-dimensional (1D) IDCT on each row of the second block to obtain an intermediate block, and

performing a scaled 1D IDCT on each column of the intermediate block.

22. The method of claim 18, wherein the full transform is a full two-dimensional (2D) discrete cosine transform (DCT), wherein the receiving the first input values via the first interface comprises receiving a block of input samples as the first input values via the first interface, and wherein the performing a full transform on the first input values comprises

performing a scaled one-dimensional (1D) DCT on each row of the block of input samples to obtain a first intermediate block,

performing a scaled 1D DCT on each column of the first intermediate block to obtain a second intermediate block,

scaling the second intermediate block to obtain a block of transform coefficients, and

providing the block of transform coefficients as the first output values via the first interface.

23. An apparatus comprising:

means for receiving first input values via a first interface;

means for performing a full transform on the first input values to obtain first output values;

means for receiving second input values via a second interface; and

means for performing a scaled transform on the second input values to obtain second output values.

24. The apparatus of claim 23, wherein the full transform is a full two-dimensional (2D) transform, and wherein the means for performing a full transform on the first input values comprises

means for scaling a block of first input values to obtain a block of scaled input values,

means for performing a scaled one-dimensional (1D) transform on each row of the block of scaled input values to obtain an intermediate block, and

means for performing a scaled 1D transform on each column of the intermediate block to obtain a block of first output values.

25. The apparatus of claim 23, wherein the full transform is a full two-dimensional (2D) transform, and wherein the means for performing a full transform on the first input values comprises

means for performing a scaled one-dimensional (1D) transform on each row of a block of input values to obtain a first intermediate block,

means for performing a scaled 1D transform on each column of the first intermediate block to obtain a second intermediate block, and

means for scaling the second intermediate block to obtain a block of first output values.

26. A processor-readable media for storing instructions to:

receive first input values via a first interface;

perform a full transform on the first input values to obtain first output values;

receive second input values via a second interface; and

perform a scaled transform on the second input values to obtain second output values.

27. An apparatus comprising:

a processor configured to scale a first block of transform coefficients to obtain a second block of scaled transform coefficients, to perform a scaled one-dimensional (1D) transform on each row of the second block to obtain an intermediate block, to perform a scaled 1D transform on each column of the intermediate block to obtain a block of output samples, and to perform multiplication on a set of variables with a set of rational dyadic constant that approximates a set of irrational constants for each scaled 1D transform; and

a memory coupled to the processor and configured to store the intermediate block.

28. The apparatus of claim 27, wherein the scaled 1D transform is a scaled 1D inverse discrete cosine transform (IDCT).

29. A method comprising:

scaling a first block of transform coefficients to obtain a second block of scaled transform coefficients;

performing a scaled one-dimensional (1D) transform on each row of the second block to obtain an intermediate block;

performing a scaled 1D transform on each column of the intermediate block to obtain a block of output samples; and

performing multiplication on a set of variables with a set of rational dyadic constant that approximates a set of irrational constants for each scaled 1D transform.

30. The method of claim 29, wherein the scaled 1D transform is a scaled 1D inverse discrete cosine transform (IDCT).

31. An apparatus comprising:

a processor configured to perform a scaled one-dimensional (1D) transform on each row of a block of input samples obtain a first intermediate block, to perform a scaled 1D transform on each column of the first intermediate block to obtain a second intermediate block, to scale the second intermediate block to obtain a block of transform coefficients, and to perform multiplication on a set of variables with a set of rational dyadic constant that approximates a set of irrational constants for each scaled 1D transform; and

a memory coupled to the processor and configured to store the first and second intermediate blocks.

32. The apparatus of claim 31, wherein the scaled 1D transform is a scaled 1D discrete cosine transform (DCT).

33. A method comprising:

performing a scaled one-dimensional (1D) transform on each row of a block of input samples obtain a first intermediate block;

performing a scaled 1D transform on each column of the first intermediate block to obtain a second intermediate block;

scaling the second intermediate block to obtain a block of transform coefficients;
and

performing multiplication on a set of variables with a set of rational dyadic constant that approximates a set of irrational constants for each scaled 1D transform.

34. The method of claim 33, wherein the scaled 1D transform is a scaled 1D discrete cosine transform (DCT).