

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
10 January 2002 (10.01.2002)

PCT

(10) International Publication Number
WO 02/03710 A1

(51) International Patent Classification⁷: H04N 7/50, G06F 17/14

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(21) International Application Number: PCT/US01/40999

(22) International Filing Date: 15 June 2001 (15.06.2001)

(25) Filing Language: English

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

(26) Publication Language: English

(71) Applicant: MEDIA EXCEL, INC. [US/US]; 6800 Austin Center Blvd., Suite 1272, Austin, TX 78731 (US).

(71) Applicant and

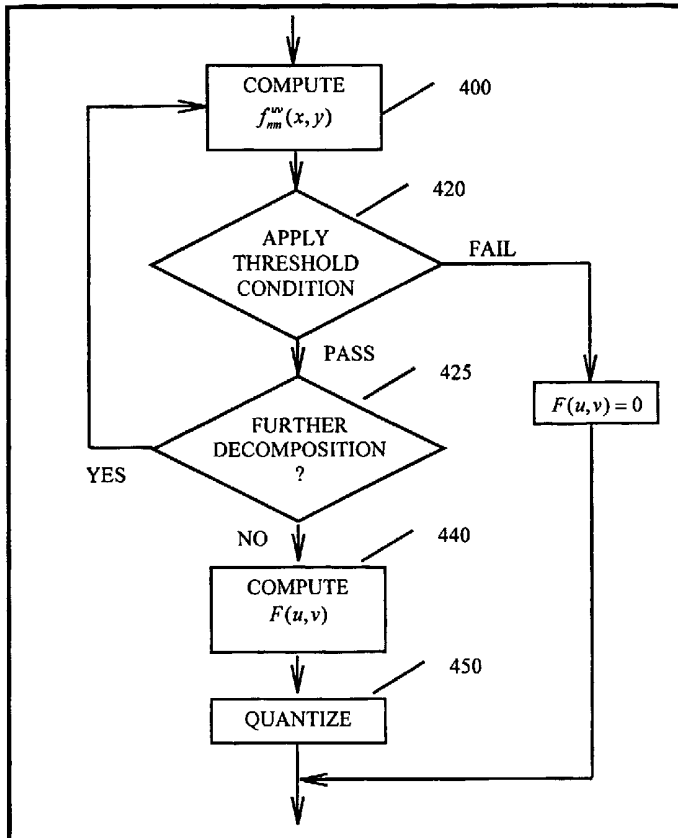
Published: — with international search report

(72) Inventor: JONGIL, Kim [US/US]; 6800 Austin Center Blvd. Suite 1272, Austin, TX 78731 (US).

(74) Agent: STELLITANO Patrick; 2803 Inridge Dr., Austin, TX 78745 (US).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: FAST EFFICIENT COMPUTATION OF THE DISCRETE COSINE TRANSFORM (DCT) OF DIGITAL VIDEO DATA



(57) Abstract: A method for selectively computing and quantizing the components of the Discrete Cosine Transform of image data is presented. The method employs decomposition of the image data to successively obtain functions possessing even or odd symmetry about axes of the data. Threshold conditions may be applied to these auxiliary functions to determine which transform components may be deemed to be zero without further computation. Considerable reduction in the number of computations and memory required to compress video data is achieved.



WO 02/03710 A1

**FAST EFFICIENT COMPUTATION OF THE DISCRETE COSINE
TRANSFORM (DCT) OF DIGITAL VIDEO DATA**

Technical Field of the Invention

5

This invention relates to compression of digitally sampled video data.

Background of the Invention

10 Compression of video data is frequently employed to increase the number of
video images that can be transmitted through a transmission channel per unit time and to
increase the number of video images that can be stored in a storage medium of a given
capacity. Several video compression standards, such as MPEG and H.263, have been
developed to enable interoperability between different video systems made by different
15 manufacturers. The major video compression standards require that the video encoder
that compresses the video data implement a Discrete Cosine Transform (DCT) of the
data. The standards also require that the video decoder that decompresses compressed
video data implement an Inverse Discrete Cosine Transform (IDCT) of the compressed
data.

20 A block diagram of a standard video encoder is shown in Figure 1. Each video
frame is divided into 16×16 pixel macro blocks. Motion Estimation 10 and Motion
Compensated Prediction 15 may be implemented for each macro-block as shown. After
the motion-compensated prediction process is performed for a macro-block, a Discrete
Cosine Transformation 100 is performed upon each of four 8×8 pixel blocks of
25 uncompressed video data comprising the macro-block.

The two dimensional Discrete Cosine Transform (DCT), $F(u, v)$, of an $N \times N$
digitally sampled video image, $f(x, y)$, is given by:

$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

for $u = 0, 1, \dots, N-1$, $v = 0, 1, \dots, N-1$ and where $C(u), C(v) = \sqrt{2}/2$ for $u, v = 0$ and $C(u), C(v) = 1$ otherwise. Thus, the 8×8 DCT, $F(u, v)$, of an 8×8 digitally sampled video image, $f(x, y)$, is given by:

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

for $u = 0, 1, \dots, 7$, $v = 0, 1, \dots, 7$ and where $C(u), C(v) = \sqrt{2}/2$ for $u, v = 0$ and $C(u), C(v) = 1$ otherwise. Here, the integer index, x , is the vertical coordinate of the image and the integer index, y , is the horizontal coordinate of the image. The integer index, u , is the vertical coordinate of the image transform and the integer index, v , is the horizontal coordinate of the image transform. Thus, the image, $f(x, y)$, is transformed into a function of the vertical and horizontal spatial frequency indices, u and v .

After computation of the DCT of the 8×8 video image, the transform components, $F(u, v)$, are quantized 150. Inverse quantization 250 is employed and an Inverse Discrete Cosine Transformation 200 is implemented for use in the motion compensated prediction process 15.

The two dimensional Inverse Discrete Cosine Transform (IDCT), $f(x, y)$, from the $N \times N$ DCT, $F(u, v)$, is given by:

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) F(u, v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

for $x = 0, 1, \dots, N-1$ and $y = 0, 1, \dots, N-1$. Thus, the 8×8 IDCT is given by:

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u) C(v) F(u, v) \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

for $x = 0, 1, \dots, 7$ and $y = 0, 1, \dots, 7$

The quantized transform components are zigzag scanned 20, as shown in Figure 2, to create a one-dimensional sequence of spatial frequency components in approximately descending order. Due to the “energy compaction property” of an orthogonal transform, the DCT of a video image generally produces a two-dimensional energy spectrum with most of the energy concentrated about the low frequency components, especially, the “DC” term, $F(0,0)$. Therefore, the one-dimensional sequence of quantized frequency components can be expected to approach zero after some number of terms, and the sequence will contain a long trail of zeros at the end. An End of Block (EOB) symbol is inserted after the last non-zero component to indicate where the non-trivial portion of the sequence ends.

Variable Length Coding 25 is employed to use shorter sequences to encode the more frequently occurring symbols and longer sequences to encode the less frequently occurring symbols to achieve a higher compression ratio. Finally, a buffer 30 is employed to enable adaptive adjustment of the quantization parameter to control the bit rate of the compressed video output by the encoder. By increasing the quantization step size, more DCT components will be quantized to zero. By decreasing the quantization step size less DCT components will be quantized to zero.

In the configuration of Figure 1, all of the DCT components are computed even though many will be quantized to zero. In particular, much of the energy in the transformed image can be expected to be concentrated in the low frequency components of $F(u,v)$ with the probability of a transform component quantizing to zero generally increasing with higher frequency. Therefore, computation of many of the transform components, $F(u,v)$, will prove to be a waste of computational resources since they will ultimately quantize to zero. A need exists for methods and systems to eliminate the computation of DCT components that would probably quantize to zero.

Summary of the Invention

The present invention increases the speed and efficiency of video image compression by providing efficient methods and systems for computing the Discrete Cosine Transform (DCT) and for selective application of thresholds to preclude computation and quantization of DCT components that have a high probability of quantizing to zero or that otherwise may be set to zero while maintaining image quality above a desired level.

The present invention provides methods for selectively determining transform components of the discrete cosine transform and omitting computation and quantization of transform components that may be deemed to be zero without substantial reduction in the quality of the image when reconstructed at an encoder. The present invention also provides methods for selectively determining transform components while minimizing the number of multiplications needed to compute transform components that are deemed necessary for adequate representation of the image. The present invention further provides for the efficient use of memory by reusing memory storage during the process of selectively computing transform components. The present invention further provides for the merger of quantization arithmetic with the multiplication of coefficients of the discrete cosine transform to obtain yet a further reduction in computations needed to compute the discrete cosine transform.

According to the present invention a selective computational approach is employed that exploits the symmetry and energy compaction properties of the Discrete Cosine Transform. This is accomplished by expressing the transform components in terms of a first set of symmetrical functions of the pixel coordinates of the image. The first set of symmetrical functions is computed without multiplication to achieve a considerable reduction in the complexity of computing the components of the DCT. The values of the first set of symmetrical functions can replace in memory the pixel values of the image to achieve a reduction in memory required to compute the DCT. Threshold conditions are then applied to one or more of the first set of symmetrical functions to determine whether further computation of the transform components corresponding to these functions is required. In many cases, further computation is not required and

additional reduction in computational burden is therefore achieved. In cases where further computation is required, the first set of symmetrical functions are themselves expressed in terms of a second set of symmetrical functions of the pixel coordinates of the image. This second set of symmetrical functions can also be computed without multiplication to
5 achieve yet a further reduction in the complexity of computing the components of the DCT. The values of the second set of symmetrical functions can replace in memory the values of the first set of symmetrical functions to achieve yet a further reduction in memory required to compute the DCT. Threshold conditions are then applied to one or more of the second set of functions to determine whether further computation of the
10 transform components corresponding to these functions is required. In many cases, further computation is not required and a further reduction in the amount of computation to determine the DCT is thereby achieved. The process of decomposing the DCT into successive sets of symmetrical functions may continue until no further symmetry decomposition can be achieved. In these cases, successive application of threshold
15 conditions to the remaining DCT components can be employed to further reduce the amount of computation required.

Thus, according to the methods of the present invention considerable efficiency is achieved by computing only those transform components that would not quantize to zero for a given quantization parameter. This results in a considerable reduction in the number
20 of multiplications required to compute the discrete cosine transform of the image. This also results in reduction of the quantization computations that must be employed in the image encoder. Further, the inverse quantization and inverse discrete cosine transformation performed in a video encoder as part of the motion estimation process can be applied only to the transform components that are not deemed to be zero, resulting in
25 yet a further reduction in the computations to be performed by the encoder. Yet further, the entire array of transform components need not be scanned since the non-zero components have been identified in the process of determining which transform components may be deemed to be zero.

These and other features and aspects of the present invention may be better
30 understood by reference to the attached drawings and the following written description of preferred embodiments.

Brief Description of the Drawings

1. Figure 1 is a block diagram of a standard video encoder.
2. Figure 2 is a block diagram of a matrix of transform components illustrating an order
5 of zigzag scanning.
3. Figure 3 is a digital computation structure for implementing computations according to the methods of the present invention.
4. Figure 4 is a flow chart of a threshold process applied according to the methods of the present invention.
- 10 5. Figure 5 is a block diagram of a system for implementing the methods of the present invention.
6. Figure 6 is an illustration of a two dimensional array of image pixels.
7. Figure 7 is a digital computation structure for implementing computations according to the methods of the present invention.
- 15 8. Figure 8 is an illustration of a two dimensional array of symmetrical function values.
9. Figure 9 is a digital computation structure for implementing computations according to the methods of the present invention.
10. Figure 10 is an illustration of a two dimensional array of symmetrical function values.
11. Figure 11 is a digital computation structure for implementing computations according
20 to the methods of the present invention.
12. Figure 12 is an illustration of a two dimensional array of symmetrical function values.
13. Figure 13 is a digital computation structure for implementing computations according to the methods of the present invention.
14. Figure 14 is a digital computation structure for implementing computations according
25 to the methods of the present invention.
15. Figure 15 is a digital computation structure for implementing computations according to the methods of the present invention.
16. Figure 16 is a digital computation structure for implementing computations according
30 to the methods of the present invention.

Description of Preferred Embodiments

The $N \times N$ DCT, $F(u, v)$, of an $N \times N$ digitally sampled video image, $f(x, y)$, is given by:

$$5 \quad F(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]$$

for $u = 0, 1, \dots, N-1$, $v = 0, 1, \dots, N-1$ and where $C(u), C(v) = \sqrt{2}/2$ for $u, v = 0$ and $C(u), C(v) = 1$ otherwise.

A significant reduction in the number of additions and multiplications to compute the transform components, $F(u, v)$, can be achieved by exploiting the symmetry properties of the cosine functions. The symmetry properties of the two dimensional DCT cosine functions can be described by reference to the one dimensional N-point DCT, $F(u)$, given by:

$$F(u) = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x) \cos \left[\frac{(2x+1)u\pi}{2N} \right]$$

for $u = 0, 1, \dots, N-1$ and where $C(u) = \sqrt{2}/2$ for $u = 0$ and $C(u) = 1$ otherwise. The cosine function possesses even symmetry about the line $x = (N-1)/2$ when u is an even integer and possesses odd symmetry about the line $x = (N-1)/2$ when u is an odd integer. If we define $C(u, x)$ as:

$$C(u, x) = \sqrt{\frac{2}{N}} C(u) \cos \left[\frac{(2x+1)u\pi}{2N} \right]$$

for $x = 0, 1, \dots, N-1$, and $u = 0, 1, \dots, N-1$, then $C(u, x') = (-1)^u C(u, x)$, where $x' = (N-1) - x$ for $x = 0, 1, \dots, (N/2-1)$. Using this notation, the equation for $F(u)$ may be rewritten as follows:

$$\begin{aligned}
 F(u) &= \sum_{x=0}^{N-1} f(x) C(u, x) \\
 &= \sum_{x=0}^{\frac{N-1}{2}} [f(x) C(u, x) + f(x') C(u, x')] \\
 &= \sum_{x=0}^{\frac{N-1}{2}} C(u, x) [f(x) + (-1)^u f(x')] \\
 &= \sum_{x=0}^{\frac{N-1}{2}} C(u, x) f_0^u(x)
 \end{aligned}$$

for $u = 0, 1, \dots, N-1$. The function, $f_0^u(x) = f(x) + (-1)^u f(x')$, possesses even symmetry about the axis, $(N-1)/2$, if u is an even integer and possesses odd symmetry about the axis, $(N-1)/2$, if u is an odd integer.

5 Thus, using the symmetry properties of the cosine function, the N-point DCT of $f(x)$ can be decomposed into an $(N/2)$ -point DCT of $f_0^u(x)$. This process can be repeated for $N \geq 4$ and u an even integer as follows:

$$\begin{aligned}
 F(u) &= \sum_{x=0}^{\frac{N-1}{2}} f_0^u(x) C(u, x) \\
 &= \sum_{x=0}^{\frac{N-1}{4}} [f_0^u(x) C(u, x) + f_0^u(x') C(u, x')] \\
 &= \sum_{x=0}^{\frac{N-1}{4}} C(u, x) [f_0^u(x) + (-1)^{u/2} f_0^u(x')] \\
 &= \sum_{x=0}^{\frac{N-1}{4}} C(u, x) f_1^u(x)
 \end{aligned}$$

10 for u equal to an even integer, and $x' = [(N/2) - 1] - x$ for $x = 0, 1, \dots, (N/4 - 1)$. The function, $f_1^u(x) = f_0^u(x) + (-1)^{u/2} f_0^u(x')$, possesses even symmetry about the axis, $[(N/2) - 1]/2$, if $u/2$ is an even integer and possesses odd symmetry about the axis, $[(N/2) - 1]/2$, if $u/2$ is an odd integer. Thus, the N-point DCT of $f(x)$ can be decomposed into an $(N/4)$ -point DCT of $f_1^u(x)$ for u an even integer. This process of decomposition can be continued as long as $u/2^n$ is an integer for $n = 0, 1, \dots, \log_2(N) - 1$.

The decomposition process just described can be applied to the two dimensional DCT as follows:

$$\begin{aligned}
 F(u, v) &= \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(u, v, x, y) f(x, y) \\
 &= \sum_{x=0}^{N-1} \sum_{y=0}^{N/2-1} C(u, v, x, y) [f(x, y) + (-1)^v f(x, y')] \\
 &= \sum_{x=0}^{N/2-1} \sum_{y=0}^{N/2-1} C(u, v, x, y) [f(x, y) + (-1)^v f(x, y') + (-1)^u f(x', y) + (-1)^{u+v} f(x', y')] \\
 &= \sum_{x=0}^{N/2-1} \sum_{y=0}^{N/2-1} C(u, v, x, y) f_{00}^{uv}(x, y)
 \end{aligned}$$

where $x' = (N-1) - x$ for $x = 0, 1, 2, 3, \dots, (N/2-1)$, $y' = (N-1) - y$ for $y = 0, 1, 2, 3, \dots, (N/2-1)$, and

$$C(u, v, x, y) = \frac{2}{N} C(u) C(v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

for $u, v = 0, 1, 2, 3, \dots, N-1$. The function, $f_{00}^{uv}(x, y)$, possesses even symmetry about the axis $x = (N-1)/2$ if u is an even integer and possesses odd symmetry about the axis $x = (N-1)/2$ if u is an odd integer. Similarly, the function, $f_{00}^{uv}(x, y)$, possesses even symmetry about the axis $y = (N-1)/2$ if v is an even integer and possesses odd symmetry about the axis $y = (N-1)/2$ if v is an odd integer.

Therefore, by exploiting the symmetry properties of the cosine functions, the $N \times N$ DCT of $f(x, y)$ has been decomposed into an $(N/2) \times (N/2)$ DCT of $f_{00}^{uv}(x, y)$.

This process of decomposition can be continued as long as $u/2^n$ or $v/2^m$ is an integer for

$n, m = 0, 1, \dots, \log_2(N) - 1$. For example, if $N \geq 4$, the DCT can be further decomposed for v equal to an even integer as follows:

$$\begin{aligned}
 F(u, v) &= \sum_{x=0}^{N/2-1} \sum_{y=0}^{N/4-1} C(u, v, x, y) [f_{00}^{uv}(x, y) + (-1)^v f_{00}^{uv}(x, y')] \\
 &= \sum_{x=0}^{N/2-1} \sum_{y=0}^{N/4-1} C(u, v, x, y) f_{01}^{uv}(x, y)
 \end{aligned}$$

for $x' = [(N-1)] - x$ for $x = 0, 1, 2, \dots, (N/2)-1$, $y' = [(N/2)-1] - y$ for $y = 0, 1, 2, \dots, (N/4)-1$, $u = 0, 1, 2, 3 \dots N-1$, and $v = 2, 4, 6, \dots N-2$. The function $f_{01}^{uv}(x, y)$ possesses even symmetry about the axis $y = [(N/2)-1]/2$ for $v/2$ an even integer and possesses odd symmetry about the axis $y = [(N/2)-1]/2$ for $v/2$ an odd integer. For u equal to an even integer, the DCT

5 can be decomposed yet further as follows:

$$\begin{aligned} F(u, v) &= \sum_{x=0}^{N/4-1} \sum_{y=0}^{N/4-1} C(u, v, x, y) [f_{01}^{uv}(x, y) + (-1)^u f_{01}^{uv}(x', y)] \\ &= \sum_{x=0}^{N/4-1} \sum_{y=0}^{N/4-1} C(u, v, x, y) [f_{00}^{uv}(x, y) + (-1)^u f_{00}^{uv}(x', y) + (-1)^v f_{00}^{uv}(x, y') + (-1)^{u+v} f_{00}^{uv}(x', y')] \\ &= \sum_{x=0}^{N/4-1} \sum_{y=0}^{N/4-1} C(u, v, x, y) f_{11}^{uv}(x, y) \end{aligned}$$

for $x' = [(N/2)-1] - x$ for $x = 0, 1, 2, \dots, (N/4)-1$, $y' = [(N/2)-1] - y$ for $y = 0, 1, 2, \dots, (N/4)-1$, $u = 0, 2, 4, 6 \dots N-2$, and $v = 0, 2, 4, 6, \dots N-2$. The function $f_{11}^{uv}(x, y)$ possesses even symmetry about the axis $x = [(N/2)-1]/2$ for $u/2$ an even integer and possesses odd symmetry about the axis $x = [(N/2)-1]/2$ for $u/2$ an odd integer.

In general, therefore, the Discrete Cosine Transform may be successively decomposed and expressed in terms of a set of functions, $f_{nm}^{uv}(x, y)$ that possess even or odd symmetry about the x or y axis according to whether $u/2^n$ or $v/2^m$ is an even or odd integer, respectively, for $n, m = 0, 1, \dots, \log_2(N) - 1$. In terms of the functions, $f_{nm}^{uv}(x, y)$, the DCT can be expressed as:

$$F(u, v) = \sum_{x=0}^{L_n} \sum_{y=0}^{L_m} C(u, v, x, y) f_{nm}^{uv}(x, y)$$

where $L_n = \left(\frac{N}{2^{n+1}}\right) - 1$, $L_m = \left(\frac{N}{2^{m+1}}\right) - 1$, $u = 2^n l_x$ for $l_x = 0, 1, \dots, (N/2^n - 1)$, and $v = 2^m l_y$ for $l_y = 0, 1, \dots, (N/2^m - 1)$. A function, $f_{nm}^{uv}(x, y)$, possesses even symmetry about

20 the axis $x = \frac{1}{2} \left[\left(\frac{N}{2^n}\right) - 1 \right]$ when $u/2^n$ is an even integer and possesses odd symmetry

about the axis $x = \frac{1}{2} \left[\left(\frac{N}{2^n} \right) - 1 \right]$ when $u/2^n$ is an odd integer. Similarly, a function, $f_{nm}^{uv}(x, y)$, possesses even symmetry about the axis $y = \frac{1}{2} \left[\left(\frac{N}{2^m} \right) - 1 \right]$ when $v/2^m$ is an even integer and possesses odd symmetry about the axis $y = \frac{1}{2} \left[\left(\frac{N}{2^m} \right) - 1 \right]$ when $v/2^m$ is an odd integer.

5 Given the initial function $f_{-1-1}^{uv}(x, y) = f(x, y)$, the functions, $f_{nm}^{uv}(x, y)$, can be derived successively from preceding functions as follows:

$$f_{nm}^{uv}(x, y) = f_{n(m-1)}^{uv}(x, y) + (-1)^{\frac{v}{2^m}} f_{n(m-1)}^{uv}(x, y'), \text{ when } v/2^m \text{ is an integer;}$$

$$f_{nm}^{uv}(x, y) = f_{(n-1)m}^{uv}(x, y) + (-1)^{\frac{u}{2^n}} f_{(n-1)m}^{uv}(x', y), \text{ when } u/2^n \text{ is an integer; and}$$

$$f_{nm}^{uv}(x, y) = f_{(n-1)(m-1)}^{uv}(x, y) + (-1)^{\frac{v}{2^m}} f_{(n-1)(m-1)}^{uv}(x, y') \\ + (-1)^{\frac{u}{2^n}} f_{(n-1)(m-1)}^{uv}(x', y) + (-1)^{\frac{u}{2^n} + \frac{v}{2^m}} f_{(n-1)(m-1)}^{uv}(x', y')$$

10 when $v/2^m$ is an integer and $u/2^n$ is an integer, for $x' = [(N/2^n) - 1] - x$ for $x = 0, 1, \dots, (N/2^{n+1}) - 1$, $y' = [(N/2^m) - 1] - y$ for $y = 0, 1, \dots, (N/2^{m+1}) - 1$, and where n or m takes on non-negative integer values less than or equal to $\log_2(N) - 1$, respectively. In particular, when $v/2^m$ is an integer and $u/2^n$ is an integer, then we have the following explicit expressions:

15 When $u/2^n$ and $v/2^m$ are both even we have:

$$f_{nm}^{uv}(x, y) = f_{nm}^{ee}(x, y) = f_{(n-1)(m-1)}^{uv}(x, y) + f_{(n-1)(m-1)}^{uv}(x, y') + f_{(n-1)(m-1)}^{uv}(x', y) + f_{(n-1)(m-1)}^{uv}(x', y')$$

When $u/2^n$ is odd and $v/2^m$ is even we have:

$$f_{nm}^{uv}(x, y) = f_{nm}^{oe}(x, y) = f_{(n-1)(m-1)}^{uv}(x, y) + f_{(n-1)(m-1)}^{uv}(x, y') - f_{(n-1)(m-1)}^{uv}(x', y) - f_{(n-1)(m-1)}^{uv}(x', y')$$

20

When $u/2^n$ is even and $v/2^m$ is odd we have:

$$f_{nm}^{uv}(x, y) = f_{nm}^{oo}(x, y) = f_{(n-1)(m-1)}^{uv}(x, y) - f_{(n-1)(m-1)}^{uv}(x, y') + f_{(n-1)(m-1)}^{uv}(x', y) - f_{(n-1)(m-1)}^{uv}(x', y')$$

When $u/2^n$ and $v/2^m$ are both odd we have:

$$f_{nm}^{uv}(x, y) = f_{nm}^{oo}(x, y) = f_{(n-1)(m-1)}^{uv}(x, y) - f_{(n-1)(m-1)}^{uv}(x, y') - f_{(n-1)(m-1)}^{uv}(x', y) + f_{(n-1)(m-1)}^{uv}(x', y')$$

5

These functions may be computed efficiently using the digital computation structure shown in Figure 3, where the arrows indicate multiplication by minus 1, and the dots imply two-input integer adders. The structure of Figure 3 shows that the functions, $f_{nm}^{uv}(x, y)$, require 8 additions and no multiplications for each coordinate pair (x, y) for a total of $8 \times (N/2^{n+1}) \times (N/2^{m+1})$ additions. As is customary, multiplication by minus one is not included when determining the computational burden of a digital implementation, since multiplication by minus one can be accomplished by negation of a sign bit.

According to the methods of the present invention, a threshold condition can be applied to functions, $f_{nm}^{uv}(x, y)$, that are obtained by successive decomposition of the DCT to selectively omit the computation of corresponding transform components, $F(u, v)$, that would be quantized to zero. This is illustrated in Figure 4. First, a function, $f_{nm}^{uv}(x, y)$, is computed 400 as described above. Then a threshold condition is applied to the function 420. If the function passes the threshold condition then further decomposition may be implemented 425. If not, then the transform components, $F(u, v)$, corresponding to the function $f_{nm}^{uv}(x, y)$ may be computed 440 and quantized 450. If the function fails the threshold condition then the corresponding transform components, $F(u, v)$, are deemed to be zero 430.

Thus, selective computation of transform components is achieved by successive application of threshold conditions to the functions, $f_{nm}^{uv}(x, y)$, that are obtained by successive decomposition of the DCT. Significant improvement in speed and efficiency is achieved by eliminating the computation of transform components that may be deemed to be zero and also eliminating the quantization arithmetic for these components. In conventional video encoding systems, quantization is applied to a transform component

25

after it is computed. Thus, after a transform component is computed it may be quantized as follows: $\frac{F(u, v)}{2qw^{uv}}$ is computed and truncated or rounded to a nearest integer, where q is

the quantization parameter, which is typically an integer between 1 and 31, and w^{uv} is a human visual weighting parameter that accounts for the spatial frequency dependence of human visual perception. The visual weighting factor, w^{uv} , is a non-decreasing function of frequency because the frequency sensitivity of human visual perception decreases as spatial frequency increases. Thus, elimination of the transform components corresponding to higher spatial frequencies is less likely to result in a degradation of the image that can be perceived by a human being. It will be understood that in some implementations, the visual weighting factor is assumed to be equal to one for all u and v .

In many cases, the transform components, $F(u, v)$, will be quantized to zero for all but the very lowest values of u and v , because the energy in the frequency spectrum of an image is typically concentrated about $F(0,0)$. Thus, computation and quantization of many of the transform components is a waste of time. According to the methods of the present invention, threshold conditions are applied prior to and during the process of computing the transform components to selectively eliminate computation and quantization of those transform components that are likely to be quantized to zero. This results in a substantial reduction in the number of computations that are performed to encode the image and substantially increases the speed of the encoder.

Therefore, after computing a set of symmetry functions, $f_{nm}^{uv}(x, y)$, we apply threshold conditions to one or more of these functions to make an initial determination whether the computation of any transform components may be eliminated. The threshold conditions may be applied to the functions as follows: First, we compute:

$$\overline{f_{nm}^{uv}} = \sum_{x=0}^{L_n} \sum_{y=0}^{L_m} |f_{nm}^{uv}(x, y)|$$

which requires $(L_n + 1) \times (L_m + 1)$ additions for each term, $\overline{f_{nm}^{uv}}$, and no multiplications. The magnitude operation involves the mere setting of a sign bit. Then, we apply a threshold test: If $\overline{f_{nm}^{uv}} < Q^{uv}$, then the corresponding transform components, $F(u, v)$, are

deemed to be zero. In other words, for a function, $f_{nm}^{uv}(x, y)$, we compute the sum of the absolute values of its data components. If the sum does not exceed a threshold level, Q^{uv} , we deem the corresponding transform components, $F(u, v)$, to be zero. If that sum does exceed the threshold, then we retain the data components for use in further
 5 computations. Thus, a transform component will be deemed zero if its corresponding symmetry function, $f_{nm}^{uv}(x, y)$, fails the threshold condition, and further computations will be performed upon $f_{nm}^{uv}(x, y)$ otherwise.

The threshold level, Q^{uv} , is preferably a composite of the following factors: q , the quantization parameter; w^{uv} , the visual weighting parameter for $F(u, v)$; and k^{uv} , as pre-
 10 determined from $C(u, v, x, y)$; and where

$$Q^{uv} = 2qw^{uv}k^{uv}$$

Clearly, if $\frac{F(u, v)}{Q^{uv}}$ quantizes to zero then the sum of absolute values of the coefficients, $f^{uv}(x, y)$, of the cosine functions in the equation for $F(u, v)$ will not exceed Q^{uv} . Therefore, deeming these values of $F(u, v)$ to be zero is justified. In practice, since the
 15 DC component is visually more important than high frequency components, w^{uv} can be set equal to $1/(2q)$ for u and v both zero, so that $F(0, 0)$ is finely quantized regardless of the value of the quantization parameter q .

The value of k^{uv} may be selected according to the desired degree of correlation between the transform components that are deemed to be zero and the transform
 20 components that would quantize to zero if actually computed and quantized. Thus, in one embodiment k^{uv} may be chosen as the maximum, over (x, y) , of the magnitude of $C^{-1}(u, v, x, y)$, where the superscript, -1, indicates the reciprocal. Alternatively, k^{uv} may be chosen as the average, over (x, y) , of the values of the magnitude of $C^{-1}(u, v, x, y)$. Yet another alternative is to choose k^{uv} as a weighted average, over (x, y) , of the values of the
 25 magnitude of $C^{-1}(u, v, x, y)$. The corresponding values of $Q^{uv} = 2qw^{uv}k^{uv}$ may be computed in advance, stored in Read-Only-Memory (ROM) and retrieved as needed.

Figure 5 illustrates a system for implementing the methods of the present invention described above. The system 500 comprises Random Access Memory (RAM) 510 for storing values of the image pixels and the values of the functions $f_{nm}^{uv}(x,y)$. The threshold levels Q^{uv} are stored in ROM 520. The computations and logical operations described above, including computation of the functions, $f_{nm}^{uv}(x,y)$, and $\overline{f_{nm}^{uv}}$, and implementation of the threshold conditions, are performed by processor 530. Processor 530 may be implemented as a microprocessor, digital signal processor or other processing device with suitable functionality operating under the directions of software. Alternatively, system 500 may be implemented in an Application Specific Integrated Circuit (ASIC) or in other suitably designed hardware. Configuration of system 500 is within the skill of persona of ordinary skill in the art given the disclosure provided herein.

Thus, a selective computation of an $N \times N$ image is accomplished by decomposing the image into a set of auxiliary functions that possess even or odd symmetry about axes of the image. Threshold conditions may then be applied to these functions to exclude computation and quantization of transform components that would quantize to zero for a given quantization parameter. For the transform components that are not deemed to be zero as a result of the application of these threshold conditions, the auxiliary functions possessing even symmetry may be further decomposed to obtain a second set of auxiliary functions that possess even or odd symmetry about an x or y axis. Threshold conditions may then be applied to this second set of auxiliary functions to further exclude computation and quantization for transform components that would quantize to zero. This process is repeated until no further decomposition can be obtained. Successive application of threshold conditions for the remaining DCT components may be applied to obtain further computational reductions. The method of the present invention described above for selectively computing transform components for an $N \times N$ image block may be specifically applied to the 8×8 DCT as will now be described. Figure 6 shows an 8×8 block of image pixels, obtained by digitally sampling a video image, which may be represented by a function, $f(x,y)$, for $x = 0,1,\dots,7$ and $y = 0,1,\dots,7$. As shown, x is the vertical coordinate and y is the horizontal coordinate. To compress a digitally sampled video image, the Discrete Cosine Transform is employed to obtain a set of 8×8

transform components, $F(u, v)$, where u is the vertical frequency index and v is the horizontal frequency index.

For the 8×8 DCT we have the following results: the function $\cos\left[\frac{(2x+1)u\pi}{16}\right]$

possesses even symmetry about the axis, $x_c = 7/2$, when u is an even integer, and

5 possesses odd symmetry about the axis, $x_c = 7/2$, when u is an odd integer. Therefore,

$$\cos\left[\frac{(2x+1)u\pi}{16}\right] = (-1)^u \cos\left[\frac{(2(7-x)+1)u\pi}{16}\right]$$

for $x = 0, 1, 2, 3$.

Similarly, the function $\cos\left[\frac{(2y+1)v\pi}{16}\right]$ possesses even symmetry about the axis,

$y_c = 7/2$, when v is an even integer, and possesses odd symmetry about the axis,

10 $y_c = 7/2$, when v is an odd integer. Therefore,

$$\cos\left[\frac{(2y+1)v\pi}{16}\right] = (-1)^v \cos\left[\frac{(2(7-y)+1)v\pi}{16}\right]$$

for $y = 0, 1, 2, 3$.

By application of these symmetry properties, the equation for the transform components, $F(u, v)$, can be written as follows:

$$15 \quad F(u, v) = \frac{1}{4} C(u)C(v) \sum_{x=0}^3 \sum_{y=0}^3 f_{00}^{uv}(x, y) \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

where, $f_{00}^{uv}(x, y)$ is given by:

$$f_{00}^{uv}(x, y) = f(x, y) + (-1)^v f(x, 7-y) + (-1)^u f(7-x, y) + (-1)^{u+v} f(7-x, 7-y)$$

for $x = 0, 1, 2, 3$ and $y = 0, 1, 2, 3$. Henceforth, f^{uv} denotes f_{00}^{uv} , and thus, $f^{uv}(x, y)$

indicates $f_{00}^{uv}(x, y)$.

20 When u and v are both even we have:

$$f^{uv}(x, y) = f_{00}^{ee}(x, y) = f^{ee}(x, y) = f(x, y) + f(x, 7-y) + f(7-x, y) + f(7-x, 7-y)$$

When u is odd and v is even we have:

$$f^{uv}(x, y) = f^{oe}(x, y) = f(x, y) + f(x, 7-y) - f(7-x, y) - f(7-x, 7-y)$$

When u is even and v is odd we have:

$$f^{uv}(x, y) = f^{eo}(x, y) = f(x, y) - f(x, 7-y) + f(7-x, y) - f(7-x, 7-y)$$

5 When u and v are both odd we have:

$$f^{uv}(x, y) = f^{oo}(x, y) = f(x, y) - f(x, 7-y) - f(7-x, y) + f(7-x, 7-y)$$

The values of $f^{uv}(x, y)$ are now the intermediate coefficients of the cosine functions in the equation for computing $F(u, v)$. Each of the functions, $f^{uv}(x, y)$, are the algebraic sum of image pixel values that have coordinates equidistant from an axis of symmetry. These functions possess even symmetry about a vertical axis of the image if v is even or odd symmetry about a vertical axis of the image if v is odd, and possess even symmetry about a horizontal axis of the image if u is even or odd symmetry about a horizontal axis of the image if u is odd.

The symmetry functions, $f^{uv}(x, y)$, can be computed efficiently using the digital computation structure shown in Figure 7, where the arrows indicate multiplication by minus 1 and dots imply adders. The structure in Figure 7 shows that the computation of the functions, $f^{uv}(x, y)$, require 8×16 additions and no multiplications.

The data of $f^{uv}(x, y)$ may replace in memory the data of $f(x, y)$ as the computations depicted in Figure 7 are performed, thereby allowing for a reduction in the memory required to compute the Discrete Cosine Transform (DCT). The organization of the data of $f^{uv}(x, y)$ in memory may be visualized in matrix form as shown in Figure 8, wherein each of the four quadrants of the 8×8 matrix contain a 4×4 sub-matrix of the data of one of the four functions f^{ee} , f^{oe} , f^{eo} , and f^{oo} .

We now apply threshold conditions to the functions, $f^{uv}(x, y)$, to selectively eliminate the computation and quantization of transform components that are likely to be quantized to zero.

As previously noted, in many cases, the transform components, $F(u, v)$, will be quantized to zero for all but the very lowest values of u and v , because the energy in the

frequency spectrum of an image is typically concentrated about $F(0,0)$. Thus, computation of many of the transform components is a waste of time. According to the methods of the present invention, threshold conditions are applied prior to and during the process of computing the transform components to selectively eliminate computation and
 5 quantization of those transform components that are likely to be quantized to zero. This results in a substantial reduction in the number of computations that are performed to encode the image and substantially increases the speed of the encoder. Also, as previously noted, the Inverse Discrete Cosine Transform implemented by the encoder in the motion estimation process is only applied to the transform components that are not
 10 deemed to be zero, thereby achieving further reduction in computations performed by the encoder.

Therefore, after computing each of the first set of symmetry functions, $f^{uv}(x,y)$, we apply threshold conditions to one or more of these functions to make an initial determination whether the computation of any transform components may be eliminated.
 15 The threshold conditions may be applied to the functions as follows: First, we compute:

$$\overline{f^{uv}} = \sum_{x=0}^3 \sum_{y=0}^3 |f^{uv}(x,y)|$$

which requires 16 additions for each term, $\overline{f^{uv}}$, and no multiplications. The magnitude operation involves the mere setting of a sign bit. Then, we apply a threshold test: If $\overline{f^{uv}} < Q^{uv}$, then the corresponding transform components, $F(u,v)$, are deemed to be
 20 zero. In other words, for a function, $f^{uv}(x,y)$, we compute the sum of the absolute values of its data components. If the sum does not exceed a threshold level, Q^{uv} , the function, $f^{uv}(x,y)$, fails the threshold test, and we deem the corresponding transform components, $F(u,v)$, to be zero. If that sum does exceed the threshold, the function, $f^{uv}(x,y)$, passes the threshold test, and we retain the data components for use in further
 25 computations. Thus, a transform component will be deemed zero if its corresponding symmetry function, $f^{uv}(x,y)$, fails to satisfy the threshold condition, and further computations will be performed upon $f^{uv}(x,y)$ otherwise. For a certain quantization

parameter q , the threshold function, Q^{uv} , is regarded as a non-decreasing function of frequency, because w^{uv} is an increasing function of frequency and k^{uv} does not vary as much over frequency as w^{uv} does. Therefore, the minimum threshold value, Q^{uv} , over the all frequencies is deemed to be Q^{00} .

5 Accordingly, we proceed to apply a threshold condition to the function, $f^{ee}(x, y)$, as follows. First, we compute:

$$\overline{f^{ee}} = \sum_{x=0}^3 \sum_{y=0}^3 |f^{ee}(x, y)|$$

Then, we apply a threshold test: If $\overline{f^{ee}} < Q^{ee}$, then the corresponding transform components, $F(u, v)$, are deemed to be zero. In other words, for the function
 10 $f^{ee}(x, y)$ we compute the sum of the absolute values of its data components. If the sum does not exceed the threshold level, Q^{ee} , we deem the corresponding transform components, $F(u, v)$, to be zero. If that sum does exceed the threshold, then we retain the data components for use in further computations. Thus, the transform components of all even frequencies of u and even frequencies of v will be deemed zero if its corresponding
 15 symmetry function, $f^{ee}(x, y)$, fails the threshold condition, and further computations will be performed upon $f^{ee}(x, y)$ otherwise.

In one embodiment of the invention, the value of Q^{ee} may be different for each frequency pair (u, v) according to the value of the visual weighting parameter, w^{ee} . However, in a preferred embodiment, Q^{ee} is chosen as $Q^{ee} = Q^{00} = 2qw^{00}k^{00}$. That
 20 is, Q^{ee} is chosen to be the threshold level corresponding to $(u, v) = (0, 0)$ because it is the minimum threshold value among all even-even frequencies. If $\overline{f^{ee}}$ fails the threshold condition for this choice of Q^{ee} , then $F(0, 0)$ would quantize to zero. And further, the inequality $\overline{f^{ee}} < Q^{ee}$ holds for all other even-even frequencies of u and v , since Q^{00} is the minimum value of Q^{ee} . Thus, if $\overline{f^{ee}} < 2qw^{00}k^{00}$, then all transform components of even-
 25 even frequencies may be deemed to be zero.

In most cases, however, $\overline{f^{ee}}$ can be expected to exceed the threshold level, Q^{00} , because a large portion of the energy in an image, $f(x, y)$, is concentrated in the DC component, $F(0,0)$. Therefore, in most cases, further computation using the symmetry functions, $f_{nm}^{uv}(x, y)$, are required. If $\overline{f^{ee}}$ exceeds the threshold level, Q^{00} , then further decomposition is performed for $f_{nm}^{uv}(x, y)$, as long as symmetry exists, and in turn, threshold conditions are applied. When further symmetry does not exist and $\overline{f_{nm}^{ee}}$ exceeds the threshold level, then the transform components are computed using:

$$F(u, v) = \sum_{x=0}^{L_n} \sum_{y=0}^{L_m} C(u, v, x, y) f_{nm}^{ee}(x, y)$$

10 A threshold condition is next applied to $f^{oe}(x, y)$, or alternatively, $f^{eo}(x, y)$, as follows. First, we compute:

$$\overline{f^{oe}} = \sum_{x=0}^3 \sum_{y=0}^3 |f^{oe}(x, y)|$$

Then, we apply a threshold test: If $\overline{f^{oe}} < Q^{oe}$, then the corresponding transform components, $F(u, v)$, are deemed to be zero. In other words, for the function $f^{oe}(x, y)$ we compute the sum of the absolute values of its data components. If the sum does not exceed the threshold level, Q^{oe} , we deem the corresponding odd-even transform components, $F(u, v)$, to be zero. If that sum does exceed the threshold, then we retain the data components for use in further computations. Thus, a transform component will be deemed zero if its corresponding symmetry function, $f^{oe}(x, y)$, fails the threshold condition, and further computations will be performed upon $f^{oe}(x, y)$ otherwise.

In one embodiment of the invention, the value of Q^{oe} may be different for each frequency pair (u, v) according to the value of the visual weighting parameter, w^{oe} . However, in a preferred embodiment, Q^{oe} is chosen as $Q^{oe} = Q^{10} = 2qw^{10}k^{10}$. That is, Q^{oe} is chosen to be the threshold level corresponding to $(u, v) = (1, 0)$ because it is the minimum threshold value among all odd-even frequencies. If $\overline{f^{oe}}$ fails the threshold condition for this choice of Q^{oe} , then $F(1, 0)$ would quantize to zero. And further, the

inequality $\overline{f^{oe}} < Q^{oe}$ holds for all other odd-even frequencies of u and v , since Q^{10} is the minimum value of Q^{oe} . Thus, if $\overline{f^{oe}} < 2q\omega^{10}k^{10}$, then all transform components, $F(u, v)$, for u odd and v even may be deemed to be zero.

Threshold conditions are also applied to the functions, $f^{eo}(x, y)$ and $f^{oo}(x, y)$.

5 Thus, if $\overline{f^{eo}} < Q^{01}$ then all transform components for (u -even, v -odd) would be deemed to be zero, and further computations would be performed using $f^{eo}(x, y)$ otherwise. Similarly, if $\overline{f^{oo}} < Q^{11}$, then all transform components for (u -odd, v -odd) would be deemed to be zero, and further computations would be performed using $f^{oo}(x, y)$ otherwise.

10 As yet another alternative, certain transform components may automatically be deemed to be zero, without regard to the threshold conditions discussed above. For example, an implementation that automatically deems to be zero values of $F(u, v)$ for u and v both odd would reduce the computational burden of video encoding and yet may yield, at least in some applications, acceptable image quality when the compressed video
15 is reconstructed after transmission

It will also be noted that deeming a transform component to be zero would entail setting to zero the memory location where that component is to be stored, unless the memory locations where the transform components are stored are initialized to zero. Thus, if the location in memory where $F(u, v)$ is to be stored is initialized to zero, then it
20 is not necessary to again set to zero those memory locations where a value $F(u, v)$ deemed to be zero is stored. Alternatively, values of $F(u, v)$ may occupy the same memory locations as were stored the original image pixels, and then the memory locations wherein values of $F(u, v)$ that are deemed to be zero are stored will be set to zero.

25 For those transform components that are not deemed to be zero at this stage of the implementation, further computations are performed, as will now be described.

Decomposition for Even u and Even v : Further analysis of the symmetry properties of the cosine functions yields the following results. The function $\cos\left[\frac{(2x+1)u\pi}{16}\right]$ possesses even symmetry about the horizontal axis, $x_c = 3/2$, when u is 0 or 4, and possesses odd symmetry about the horizontal axis, $x_c = 3/2$, when u is 2 or 6. Therefore,

$$5 \quad \cos\left[\frac{(2x+1)u\pi}{16}\right] = (-1)^{u/2} \cos\left[\frac{(2(3-x)+1)u\pi}{16}\right]$$

for $u = 0, 2, 4, 6$ and $x = 0, 1$.

Similarly, the function $\cos\left[\frac{(2y+1)v\pi}{16}\right]$ possesses even symmetry about the vertical axis, $y_c = 3/2$, when v is 0 or 4, and possesses odd symmetry about the vertical axis, $y_c = 3/2$, when v is 2 or 6. Therefore,

$$10 \quad \cos\left[\frac{(2y+1)v\pi}{16}\right] = (-1)^{v/2} \cos\left[\frac{(2(3-y)+1)v\pi}{16}\right]$$

for $v = 0, 2, 4, 6$ and $y = 0, 1$.

By application of these symmetry properties we can write the equation for $F(u, v)$ when both u and v are even as follows:

$$F(u, v) = \frac{1}{4}C(u)C(v) \sum_{x=0}^1 \sum_{y=0}^1 g_{ee}^{uv}(x, y) \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

15 for $u = 0, 2, 4, 6$, $x = 0, 1$, $v = 0, 2, 4, 6$, $y = 0, 1$, and where

$$g_{ee}^{uv}(x, y) = f^{ee}(x, y) + (-1)^{v/2} f^{ee}(x, 3-y) + (-1)^{u/2} f^{ee}(3-x, y) + (-1)^{(u+v)/2} f^{ee}(3-x, 3-y)$$

where $f^{ee}(x, y) = f(x, y) + f(x, 7-y) + f(7-x, y) + f(7-x, 7-y)$.

When u is 0 or 4 and v is 0 or 4 we have:

$$20 \quad g_{ee}^{uv}(x, y) = g_{ee}^{ee}(x, y) = f^{ee}(x, y) + f^{ee}(x, 3-y) + f^{ee}(3-x, y) + f^{ee}(3-x, 3-y)$$

When u is 2 or 6 and v is 0 or 4 we have:

$$g_{ee}^{uv}(x, y) = g_{ee}^{oe}(x, y) = f^{ee}(x, y) + f^{ee}(x, 3-y) - f^{ee}(3-x, y) - f^{ee}(3-x, 3-y)$$

When u is 0 or 4 and v is 2 or 6 we have:

$$g_{ee}^{uv}(x, y) = g_{ee}^{oo}(x, y) = f^{ee}(x, y) - f^{ee}(x, 3-y) + f^{ee}(3-x, y) - f^{ee}(3-x, 3-y)$$

When u is 2 or 6 and v is 2 or 6 we have:

$$g_{ee}^{uv}(x, y) = g_{ee}^{oo}(x, y) = f^{ee}(x, y) - f^{ee}(x, 3-y) - f^{ee}(3-x, y) + f^{ee}(3-x, 3-y)$$

Finally, we have:

$$5 \quad f^{ee}(x, y) = \frac{1}{4}[g_{ee}^{ee}(x, y) + g_{ee}^{oe}(x, y) + g_{ee}^{eo}(x, y) + g_{ee}^{oo}(x, y)]$$

The functions, $g_{ee}^{uv}(x, y)$, are the algebraic sum of values of $f^{ee}(x, y)$ that have coordinates equidistant from an axis of symmetry. The functions $g_{ee}^{uv}(x, y)$ can be computed efficiently using the digital computation structure of Figure 9. The structure in Figure 9 shows that the computation of the functions, $g_{ee}^{uv}(x, y)$, require 4×8 additions
10 and no multiplications.

The data of $g_{ee}^{uv}(x, y)$ may replace in memory the data of $f^{ee}(x, y)$ as the computations depicted in Figure 9 are performed, thereby allowing for a further reduction in the memory required to compute the Discrete Cosine Transform (DCT). The organization of the data of $g_{ee}^{uv}(x, y)$ in memory may be visualized in matrix form as
15 shown in Figure 10, wherein each of the four quadrants of the 4×4 matrix contain a 2×2 sub-matrix of the data of one of the four functions g_{ee}^{ee} , g_{ee}^{oe} , g_{ee}^{eo} , and g_{ee}^{oo} .

The values of $g_{ee}^{uv}(x, y)$, are now the intermediate coefficients of the cosine functions in the equation for computing $F(u, v)$ for u and v both even. These functions possess even symmetry about the vertical axis, $y = 3/2$, if $v/2$ is even or odd symmetry about the vertical axis, $y = 3/2$, if $v/2$ is odd, and possess even symmetry about the
20 horizontal axis, $x = 3/2$, if $u/2$ is even or odd symmetry about the horizontal axis, $x = 3/2$, if $u/2$ is odd.

We may now apply a threshold condition to each of the functions, $g_{ee}^{uv}(x, y)$, as follows. First we compute:

$$\overline{g_{ee}^{uv}} = \sum_{x=0}^1 \sum_{y=0}^1 |g_{ee}^{uv}(x, y)|$$

which requires 4×4 additions for all integer values of $u/2, v/2$. Then we apply a threshold test: If $\overline{g_{ee}^{uv}} < Q^{uv}$, then the corresponding transform components, $F(u, v)$, are deemed to be zero. In other words, for a function, $g_{ee}^{uv}(x, y)$, we compute the sum of the absolute values of its data components. If the sum does not exceed a threshold level, Q^{uv} , we deem the corresponding transform components, $F(u, v)$, for all integer values of $u/2, v/2$, to be zeroes. If the sum does exceed the threshold, then we retain the data components for use in further computations. Thus, a transform component will be deemed zero if its corresponding symmetry function, $g_{ee}^{uv}(x, y)$, fails the threshold condition, and further computations will be performed upon $g_{ee}^{uv}(x, y)$ otherwise.

Accordingly, for $u/2$ and $v/2$ both even integers, we proceed to apply a threshold condition to the function, $g_{ee}^{ee}(x, y)$ as follows. First we compute:

$$\overline{g_{ee}^{ee}} = \sum_{x=0}^1 \sum_{y=0}^1 |g_{ee}^{ee}(x, y)|$$

Then we apply a threshold test: If $\overline{g_{ee}^{ee}} < Q^{ee}$, then the corresponding transform components, $F(u, v)$, are deemed to be zero. Otherwise, the values of $g_{ee}^{ee}(x, y)$ are used in further computations. In one embodiment of the invention, the value of Q^{ee} may be different for each frequency pair (u, v) . However, in a preferred embodiment, Q^{ee} is chosen as $Q^{ee} = Q^{00} = 2q\omega^{00}k^{00}$. That is, Q^{ee} is chosen as the threshold level corresponding to $(u, v) = (0, 0)$. If $\overline{g_{ee}^{ee}}$ fails the threshold test for this choice of Q^{ee} , then for all even integer values of $u/2$ and $v/2$, $F(u, v)$ would quantize to zero.

In most cases, however, $\overline{g_{ee}^{ee}}$ can be expected to exceed the threshold level, Q^{00} , because a large portion of the energy in an image is concentrated in the DC term, $F(0, 0)$.

Therefore, in most cases, further computation using the symmetry functions, $g_{ee}^{uv}(x, y)$, are required.

A threshold condition is next applied to $g_{ee}^{oe}(x, y)$, or alternatively, $g_{ee}^{eo}(x, y)$, as follows. First, we compute:

$$5 \quad \overline{g_{ee}^{oe}} = \sum_{x=0}^1 \sum_{y=0}^1 |g_{ee}^{oe}(x, y)|$$

Then, we apply a threshold test: If $\overline{g_{ee}^{oe}} < Q^{oe}$, then the corresponding transform components, for all $u/2$ -odd and $v/2$ -even, $F(u, v)$, are deemed to be zero. Otherwise, the values of $g_{ee}^{oe}(x, y)$ are used in further computations. In one embodiment of the invention, the value of Q^{oe} may be different for each frequency pair (u, v) . However, in a preferred embodiment, Q^{oe} is chosen as $Q^{oe} = Q^{20} = 2qw^{20}k^{20}$. That is, Q^{oe} is chosen as the threshold level corresponding to $(u, v) = (2, 0)$. If $\overline{g_{ee}^{oe}}$ fails the threshold test for this choice of Q^{oe} , then for all $u/2$ -odd and $v/2$ -even, $F(u, v)$ would quantize to zero.

Threshold conditions are also applied to the functions $g_{ee}^{eo}(x, y)$ and $g_{ee}^{oo}(x, y)$. Thus, if $\overline{g_{ee}^{eo}} < Q^{oe}$, then all transform components for $(u/2$ -even, $v/2$ -odd), would be deemed to be zero, and further computations would be performed using $g_{ee}^{eo}(x, y)$ otherwise. Similarly, if $\overline{g_{ee}^{oo}} < Q^{oe}$, then all transform components for $(u/2$ -odd, $v/2$ -odd), would be deemed to be zero, and further computations would be performed using $g_{ee}^{oo}(x, y)$ otherwise.

As previously noted, in most cases we may anticipate that $\overline{g_{ee}^{ee}} > Q^{oo}$. To continue the decomposition process, we note that the components of $g_{ee}^{ee}(x, y)$ are intermediate coefficients for the computation of $F(u, v)$ for $u = 0, 4$ and $v = 0, 4$ only, which are the transform components for $u/2$ -even and $v/2$ -even. Further analysis of the symmetry properties of the cosine functions yields the following results. The function $\cos\left[\frac{(2x+1)u\pi}{16}\right]$ possesses even symmetry about the horizontal axis, $x_c = 1/2$, when u is

0, and possesses odd symmetry about the horizontal axis, $x_c = 1/2$, when u is 4.

Therefore,

$$\cos\left[\frac{(2x+1)u\pi}{16}\right] = (-1)^{u/4} \cos\left[\frac{(2(1-x)+1)u\pi}{16}\right]$$

for $u = 0, 4$ and $x = 0$.

5

Similarly, the function $\cos\left[\frac{(2y+1)v\pi}{16}\right]$ possesses even symmetry about the

vertical axis, $y_c = 1/2$, when v is 0, and possesses odd symmetry about the vertical axis,

$y_c = 1/2$, when v is 4. Therefore:

$$\cos\left[\frac{(2y+1)v\pi}{16}\right] = (-1)^{v/4} \cos\left[\frac{(2(1-y)+1)v\pi}{16}\right]$$

10

for $v = 0, 4$ and $y = 0$.

By application of these symmetry properties we can write the equation for $F(u, v)$ when $u = 0, 4$ and $v = 0, 4$ as follows:

$$15 \quad F(u, v) = \frac{1}{4} C(u) C(v) h_{ee}^{uv} \cos\left[\frac{u\pi}{16}\right] \cos\left[\frac{v\pi}{16}\right]$$

where

$$h_{ee}^{uv} = g_{ee}^{ee}(0,0) + (-1)^{v/4} g_{ee}^{ee}(0,1) + (-1)^{u/4} g_{ee}^{ee}(1,0) + (-1)^{(u+v)/4} g_{ee}^{ee}(1,1)$$

When u and v are both zero we have:

$$h_{ee}^{00} = h_{ee}^{ee} = g_{ee}^{ee}(0,0) + g_{ee}^{ee}(0,1) + g_{ee}^{ee}(1,0) + g_{ee}^{ee}(1,1)$$

20 When u is 4 and v is 0 we have:

$$h_{ee}^{40} = h_{ee}^{oe} = g_{ee}^{ee}(0,0) + g_{ee}^{ee}(0,1) - g_{ee}^{ee}(1,0) - g_{ee}^{ee}(1,1)$$

When u is 0 and v is 4 we have:

$$h_{ee}^{04} = h_{ee}^{eo} = g_{ee}^{ee}(0,0) - g_{ee}^{ee}(0,1) + g_{ee}^{ee}(1,0) - g_{ee}^{ee}(1,1)$$

When both u and v are 4 we have:

$$25 \quad h_{ee}^{44} = h_{ee}^{oo} = g_{ee}^{ee}(0,0) - g_{ee}^{ee}(0,1) - g_{ee}^{ee}(1,0) + g_{ee}^{ee}(1,1)$$

The values of h_{ee}^{uv} are now the intermediate coefficients of the cosine functions in the equation for computing $F(u, v)$ for $u/2$ and $v/2$ both even. The functions, h_{ee}^{uv} , are the algebraic sum of the values of $g_{ee}^{ee}(x, y)$ that have coordinates equidistant from an axis of symmetry. We may utilize the digital computation structure depicted in Figure 11 to
 5 compute the functions h_{ee}^{uv} . Computation of h_{ee}^{uv} requires 8 additions and no multiplications. The data of h_{ee}^{uv} may replace in memory the data of $g_{ee}^{ee}(x, y)$ as the computations depicted in Figure 11 are performed, thereby allowing for yet a further reduction in the memory required to compute the Discrete Cosine Transform (DCT). The organization of the data of h_{ee}^{uv} in memory may be visualized in matrix form as shown in
 10 Figure 12, wherein each of the four quadrants of the 2×2 matrix contain one of the four functions h_{ee}^{00} , h_{ee}^{40} , h_{ee}^{04} , and h_{ee}^{44} .

We may now apply a threshold condition to each of the terms h_{ee}^{uv} as follows. First we compute:

$$\overline{h_{ee}^{uv}} = |h_{ee}^{uv}|$$

15 Then, we apply a threshold test: if $\overline{h_{ee}^{uv}} < Q^{uv}$, the corresponding transform component, $F(u, v)$ is deemed to be zero. If the absolute value of a term, h_{ee}^{uv} , exceeds the predetermined threshold, Q^{uv} , then we use the term to compute $F(u, v)$ as follows:

$$\begin{aligned} F(0,0) &= \frac{1}{8} h_{ee}^{00} \\ F(4,0) &= \frac{1}{8} h_{ee}^{40} \\ 20 \quad F(0,4) &= \frac{1}{8} h_{ee}^{04} \\ F(4,4) &= \frac{1}{8} h_{ee}^{44} \end{aligned}$$

where it will be recognized that division by eight can be implemented digitally by a binary shift operation.

We now continue the description by addressing the computation of $F(u, v)$ for the
 25 remaining values of u and v . When u is 2 or 6 and v is 0 or 4, we have:

$$g_{ee}^{uv}(x, y) = g_{ee}^{oe}(x, y) = f^{ee}(x, y) + f^{ee}(x, 3-y) - f^{ee}(3-x, y) - f^{ee}(3-x, 3-y)$$

where, $f^{ee}(x, y) = f(x, y) + f(x, 7 - y) + f(7 - x, y) + f(7 - x, 7 - y)$,

and application of the symmetry properties of the cosine function yields the following equation:

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^1 h_{oe}^{uv}(x) \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{v\pi}{16}\right]$$

5 for $u = 2, 6$ and $v = 0, 4$ and where:

$$h_{oe}^{uv}(x) = g_{ee}^{oe}(x, 0) + g_{ee}^{oe}(x, 1) \text{ for } u = 2 \text{ or } 6 \text{ and } v = 0;$$

$$h_{oe}^{uv}(x) = g_{ee}^{oe}(x, 0) - g_{ee}^{oe}(x, 1) \text{ for } u = 2 \text{ or } 6 \text{ and } v = 4.$$

The functions, h_{oe}^{uv} , are the algebraic sum of values of the symmetry function $g_{ee}^{oe}(x, y)$ that have coordinates equidistant from an axis of symmetry. The functions h_{oe}^{uv} can be computed efficiently using the digital computation structure depicted in Figure 13, which requires 2×4 additions and no multiplications. The data of h_{oe}^{uv} can replace the data of $g_{ee}^{oe}(x, y)$ in memory to conserve memory resources.

10 The values of $h_{oe}^{uv}(x)$ are now the intermediate coefficients of the cosine functions for computing $F(u, v)$. These functions possess even symmetry about the vertical axis, $y = \frac{1}{2}$, if $v = 0$ and odd symmetry about the vertical axis, $y = \frac{1}{2}$, if $v = 4$.

We may now apply a threshold condition to each of the functions h_{oe}^{uv} as follows.

First we compute:

$$\overline{h_{oe}^{uv}} = \sum_{x=0}^1 |h_{oe}^{uv}(x)|$$

for $u = 2, 6$ and $v = 0, 4$, which requires 2×4 additions and no multiplications. We then apply a threshold test: If $\overline{h_{oe}^{uv}} < Q^{uv}$, then the corresponding transform component is deemed to be zero. In other words, for each function, h_{oe}^{uv} , we determine if the sum of the absolute values of its terms exceeds the predetermined threshold, Q^{uv} . If the absolute value of a term, h_{oe}^{uv} , exceeds the predetermined threshold, Q^{uv} , then we use the term to compute $F(u, v)$ as follows:

$$F(2,0) = \frac{\sqrt{2}}{8} \sum_{x=0}^1 h_{oe}^{20}(x) \cos \left[\frac{(2x+1)2\pi}{16} \right]$$

$$F(6,0) = \frac{\sqrt{2}}{8} \sum_{x=0}^1 h_{oe}^{60}(x) \cos \left[\frac{(2x+1)6\pi}{16} \right]$$

$$F(2,4) = \frac{\sqrt{2}}{8} \sum_{x=0}^1 h_{oe}^{24}(x) \cos \left[\frac{(2x+1)2\pi}{16} \right]$$

$$F(6,4) = \frac{\sqrt{2}}{8} \sum_{x=0}^1 h_{oe}^{64}(x) \cos \left[\frac{(2x+1)6\pi}{16} \right]$$

5

We now continue the description by addressing the computation of $F(u, v)$ for the remaining values of u and v . When u is 0 or 4 and v is 2 or 6, we have:

$$g_{ee}^{uv}(x, y) = g_{ee}^{eo}(x, y) = f^{ee}(x, y) - f^{ee}(x, 3-y) + f^{ee}(3-x, y) - f^{ee}(3-x, 3-y)$$

$$\text{where, } f^{ee}(x, y) = f(x, y) + f(x, 7-y) + f(7-x, y) + f(7-x, 7-y),$$

10 and application of the symmetry properties of the cosine function yields the following equation:

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{y=0}^1 h_{eo}^{uv}(y) \cos \left[\frac{u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right]$$

for $u = 0, 4$ and $v = 2, 6$ and where:

$$h_{eo}^{uv}(y) = g_{ee}^{eo}(0, y) + g_{ee}^{eo}(1, y) \text{ for } u = 0 \text{ and } v = 2 \text{ or } 6;$$

15
$$h_{eo}^{uv}(y) = g_{ee}^{eo}(0, y) - g_{ee}^{eo}(1, y) \text{ for } u = 4 \text{ and } v = 2 \text{ or } 6.$$

The functions, h_{eo}^{uv} , are the algebraic sum of values of the symmetry function $g_{ee}^{eo}(x, y)$ that have coordinates equidistant from an axis of symmetry. The functions h_{eo}^{uv} can be computed efficiently using the digital computation structure depicted in Figure 14 as shown, which requires 2×4 additions and no multiplications. The data of h_{eo}^{uv} can
20 replace the data of $g_{ee}^{eo}(x, y)$ in memory to conserve memory resources.

The values of $h_{eo}^{uv}(y)$ are now the coefficients of the cosine functions for computing $F(u, v)$. These functions possess even symmetry about the horizontal axis, $x = \frac{1}{2}$, if $u = 0$ and odd symmetry about the horizontal axis, $x = \frac{1}{2}$, if $u = 4$.

We may now apply a threshold condition to each of the functions h_{eo}^{uv} as follows.

First we compute:

$$\overline{h_{eo}^{uv}} = \sum_{y=0}^1 |h_{eo}^{uv}(y)|$$

for $u = 0, 4$ and $v = 2, 6$, which requires 2×4 additions and no multiplications. We then
 5 apply a threshold test: If $\overline{h_{eo}^{uv}} < Q^{uv}$, then the corresponding transform component is
 deemed to be zero. In other words, for each function, h_{eo}^{uv} , we determine if the sum of the
 absolute values of its terms exceeds the predetermined threshold, Q^{uv} .

If the absolute value of a term, h_{eo}^{uv} , exceeds the predetermined threshold, Q^{uv} ,
 then we use the term to compute $F(u, v)$ as follows:

$$\begin{aligned} 10 \quad F(0,2) &= \frac{\sqrt{2}}{8} \sum_{y=0}^1 h_{eo}^{02}(y) \cos \left[\frac{(2y+1)2\pi}{16} \right] \\ F(0,6) &= \frac{\sqrt{2}}{8} \sum_{y=0}^1 h_{eo}^{06}(y) \cos \left[\frac{(2y+1)6\pi}{16} \right] \\ F(4,2) &= \frac{\sqrt{2}}{8} \sum_{y=0}^1 h_{eo}^{42}(y) \cos \left[\frac{(2y+1)2\pi}{16} \right] \\ F(4,6) &= \frac{\sqrt{2}}{8} \sum_{y=0}^1 h_{eo}^{46}(y) \cos \left[\frac{(2y+1)6\pi}{16} \right] \end{aligned}$$

The remaining values of $F(u, v)$ for both u and v even are $F(2,2)$, $F(2,6)$,
 15 $F(6,2)$, and $F(6,6)$. For these values of u and v $F(u, v)$ is determined according to:

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^1 \sum_{y=0}^1 g_{ee}^{oo}(x, y) \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right]$$

for $u = 2, 6$, $x = 0, 1$, $v = 2, 6$, $y = 0, 1$, and where

$$g_{ee}^{oo}(x, y) = f^{ee}(x, y) - f^{ee}(x, 3-y) - f^{ee}(3-x, y) + f^{ee}(3-x, 3-y)$$

Further reduction in computations may be achieved as follows. First, a scale
 20 of $F(2,2)$ is computed and its magnitude is subtracted from $\overline{g_{ee}^{oo}}$ as computed above:

$$\overline{g_{ee}^{oo}} = \overline{g_{ee}^{oo}} - |F(2,2)| \times \alpha$$

where α is a scale factor that may be adjusted to achieve a desired level of correlation between the transform components that are deemed to be zero and the transform components that would actually quantized to zero if computed and quantized. Decreasing α increases that level of correlation, whereas increasing α decreases that level of correlation. As α is decreased to zero, the correlation approaches 100%. For α equal to 8, the correlation is about 75%. Setting α equal to 4 yields about 95% correlation. It will be recognized that division and multiplication by a power of two may be implemented as a binary shift operation rather than as an actual division or multiplication operation. Therefore, α may preferably be set equal to 4.

10 Now, if the remaining value of $\overline{g_{ee}^{oo}}$ is less than the threshold Q_{ee}^{oo} , then $F(2,6)$, $F(6,2)$, and $F(6,6)$ are deemed to be zero. Otherwise, a scale of $F(2,6)$ is computed and its magnitude is subtracted from the current value of $\overline{g_{ee}^{oo}}$:

$$\overline{g_{ee}^{oo}} = \overline{g_{ee}^{oo}} - |F(2,6)| \times \alpha$$

15 Then if the remaining value of $\overline{g_{ee}^{oo}}$ is less than the threshold Q_{ee}^{oo} then $F(6,2)$, and $F(6,6)$ are deemed to be zero. Otherwise, a scale of $F(6,2)$ is computed and its magnitude is subtracted from the current value of $\overline{g_{ee}^{oo}}$:

$$\overline{g_{ee}^{oo}} = \overline{g_{ee}^{oo}} - |F(6,2)| \times \alpha$$

Then if the remaining value of $\overline{g_{ee}^{oo}}$ is less than the threshold Q_{ee}^{oo} then $F(6,6)$ is deemed to be zero. Otherwise, $F(6,6)$ is computed.

20 Thus, selective computation of the transform components, $F(u,v)$, when $u/2$ and $v/2$ are both odd, is accomplished by computing the scaled magnitude of each value of $F(u,v)$ when $u/2$ and $v/2$ are both odd, in a sequential order until the scaled magnitude of $F(u,v)$ in the sequence falls below a threshold level to which it is compared; wherein for each scaled magnitude of $F(u,v)$ computed, the threshold level to which it is compared is computed as the threshold level to which the previous value of the scaled magnitude of $F(u,v)$ was compared minus the scaled magnitude of the previous value of $F(u,v)$ computed; and deeming a value of $F(u,v)$ to be zero if its scaled magnitude

falls below the threshold to which it is compared and deemed to be zero all subsequent values of $F(u, v)$ in the sequence.

This completes the process for computing all values of $F(u, v)$ for both u and v even.

5 **Decomposition for Odd u and Even v :** When u is odd and v is even, we have:

$$F(u, v) = \sum_{x=0}^3 \sum_{y=0}^3 f^{oe} C(u, v, x, y)$$

where:

$$C(u, v, x, y) = \frac{1}{4} C(u) C(v) \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right]$$

with: $f^{uv}(x, y) = f^{oe}(x, y) = f(x, y) + f(x, 7-y) - f(7-x, y) - f(7-x, 7-y)$

10 for $u = 1, 3, 5, 7$ and $v = 0, 2, 4, 6$.

To facilitate selective computation of $F(u, v)$ for these frequency pairs we compute the following functions:

$$g_{oe}^{ov}(x, 0) = g_{oe}^{o0}(x) = f^{oe}(x, 0) + f^{oe}(x, 1) + f^{oe}(x, 2) + f^{oe}(x, 3);$$

$$g_{oe}^{ov}(x, 1) = g_{oe}^{o2}(x) = f^{oe}(x, 0) + f^{oe}(x, 1) - f^{oe}(x, 2) - f^{oe}(x, 3);$$

15 $g_{oe}^{ov}(x, 2) = g_{oe}^{o4}(x) = f^{oe}(x, 0) - f^{oe}(x, 1) - f^{oe}(x, 2) + f^{oe}(x, 3);$

$$g_{oe}^{ov}(x, 3) = g_{oe}^{o6}(x) = f^{oe}(x, 0) - f^{oe}(x, 1) + f^{oe}(x, 2) - f^{oe}(x, 3).$$

for $u = 1, 3, 5, 7$, $x = 0, 1, 2, 3$ and $v = 0, 2, 4, 6$.

The functions, $g_{oe}^{ov}(x, y)$, are the algebraic sum of values of $f^{oe}(x, y)$ that have coordinates equidistant from an axis of symmetry. The functions g_{oe}^{ov} can be computed efficiently using the digital computation structure of Figure 15. The structure of Figure 15 shows that the computation of the functions g_{oe}^{ov} requires 4×8 additions and no multiplications. The data of g_{oe}^{ov} may replace in memory the data of $f^{oe}(x, y)$ as the computations depicted in Figure 15 are performed, thereby allowing for yet a further reduction in the amount of memory required to compute the Discrete Cosine Transform.

25 To compute the transform components for u odd and $v = 0$ we proceed as follows. We first apply a threshold condition by computing:

$$\overline{g_{oe}^{o0}} = \sum_{x=0}^3 |g_{oe}^{o0}(x)|$$

This requires 4 additions and no multiplications. If $\overline{g_{oe}^{o0}} < Q_{oe}^{10}$, then the transform components for u odd and v even are deemed to be zero. Otherwise, we compute as follows:

5
$$F(1,0) = \sum_{x=0}^3 g_{oe}^{o0}(x)C(1,0,x,0)$$

If $\overline{g_{oe}^{o0}} < Q_{oe}^{30}$, then the transform components for $u = 3,5,7$ and $v = 0$ are deemed to be zero. Otherwise, we compute as follows:

$$F(3,0) = \sum_{x=0}^3 g_{oe}^{o0}(x)C(3,0,x,0)$$

If $\overline{g_{oe}^{o0}} < Q_{oe}^{50}$, then the transform components for $u = 5,7$ and $v = 0$ are deemed to be zero.

10 Otherwise, we compute as follows:

$$F(5,0) = \sum_{x=0}^3 g_{oe}^{o0}(x)C(5,0,x,0)$$

If $\overline{g_{oe}^{o0}} < Q_{oe}^{70}$, then the transform component for $u = 7$ and $v = 0$ is deemed to be zero.

Otherwise, we compute as follows:

$$F(7,0) = \sum_{x=0}^3 g_{oe}^{o0}(x)C(7,0,x,0)$$

15 This completes the computations for the transform components for u odd and $v = 0$.

To compute the transform components for u odd and $v = 4$ we proceed as follows.

We first apply a threshold condition by computing:

$$\overline{g_{oe}^{o4}} = \sum_{x=0}^3 |g_{oe}^{o4}(x)|$$

This requires 4 additions and no multiplications. If $\overline{g_{oe}^{o4}} < Q_{oe}^{14}$, then the transform

20 components for $u = 1,3,5,7$ and $v = 4$ are deemed to be zero. Otherwise, we compute as follows:

$$F(1,4) = \sum_{x=0}^3 g_{oe}^{o4}(x)C(1,4,x,0)$$

If $\overline{g_{oe}^{o4}} < Q_{oe}^{34}$, then the transform components for $u = 3,5,7$ and $v = 4$ are deemed to be zero. Otherwise, we compute as follows:

$$F(3,4) = \sum_{x=0}^3 g_{oe}^{o4}(x)C(3,4,x,0)$$

If $\overline{g_{oe}^{o4}} < Q_{oe}^{54}$, then the transform components for $u = 5,7$ and $v = 4$ are deemed to be zero.

5 Otherwise, we compute as follows:

$$F(5,4) = \sum_{x=0}^3 g_{oe}^{o4}(x)C(5,4,x,0)$$

If $\overline{g_{oe}^{o4}} < Q_{oe}^{74}$, then the transform component for $u = 7$ and $v = 4$ is deemed to be zero.

Otherwise, we compute as follows:

$$F(7,4) = \sum_{x=0}^3 g_{oe}^{o4}(x)C(7,4,x,0)$$

10 This completes the computations for the transform components for u odd and $v = 4$.

To compute the transform components for u odd and $v = 2$ we proceed as follows.

We first apply a threshold condition by computing:

$$\overline{g_{oe}^{o2}} = \sum_{x=0}^3 |g_{oe}^{o2}(x)|$$

This requires 4 additions and no multiplications. If $\overline{g_{oe}^{o2}} < Q_{oe}^{12}$, then $F(1,2)$,

15 $F(3,2)$ $F(5,2)$ and $F(7,2)$ are deemed to be zero. Otherwise, we compute as follows:

$$h_{oe}(x) = [f^{oe}(x,0) - f^{oe}(x,3)] + k_1 [f^{oe}(x,1) - f^{oe}(x,2)]$$

for $x=0,1,2,3$, and where $k_1 = 0.41421$ is the ratio of $C(u,2,x,1)$ to $C(u,2,x,0)$. This requires 4×4 additions and 4 multiplications. Note, that the quantities in the square brackets are obtained from nodes A and B of the digital computation structure of Figure

20 15. Then we apply the following threshold condition:

$$\text{If } |h_{oe}(0) + h_{oe}(1) + h_{oe}(2) + h_{oe}(3)| < Q_{oe}^{12}, \text{ then, } F(1,2), F(1,2), F(3,2), F(5,2) \text{ and}$$

$F(7,2)$ are deemed to be zero. Otherwise, we compute:

$$F(1,2) = \sum_{x=0}^3 h_{oe}(x)C(1,2,x,0)$$

Then we apply the following threshold condition:

If $\left| h_{oe}(0) - h_{oe}(1) - h_{oe}(2) - h_{oe}(3) \right| < Q_{oe}^{32}$, then $F(3,2)$, $F(5,2)$ and $F(7,2)$ are deemed to be zero. Otherwise, we compute:

$$F(3,2) = \sum_{x=0}^3 h_{oe}(x) C(3,2, x, 0)$$

Then we apply the following threshold condition:

5 If $\left| h_{oe}(0) - h_{oe}(1) + h_{oe}(2) + h_{oe}(3) \right| < Q_{oe}^{52}$, then, $F(5,2)$ and $F(7,2)$ are deemed to be zero. Otherwise, we compute:

$$F(5,2) = \sum_{x=0}^3 h_{oe}(x) C(5,2, x, 0)$$

Then we apply the following threshold condition:

If $\left| h_{oe}(0) - h_{oe}(1) + h_{oe}(2) - h_{oe}(3) \right| < Q_{oe}^{72}$, then, $F(7,2)$ is deemed to be zero.

10 Otherwise, we compute:

$$F(7,2) = \sum_{x=0}^3 h_{oe}(x) C(7,2, x, 0)$$

This completes the decomposition for when u is odd and v equal to 2. We note that additional reduction in computations can be achieved by computing, storing and re-using the partial sums, $h_{oe}(2) + h_{oe}(3)$, $h_{oe}(0) - h_{oe}(1)$, $h_{oe}(2) - h_{oe}(3)$ as required for the application of the threshold conditions just given.

To compute the transform components for u odd and $v = 6$ we proceed as follows. We first apply a threshold condition by computing:

$$\overline{g_{oe}^{o6}} = \sum_{x=0}^3 \left| g_{oe}^{o6}(x) \right|$$

This requires 4 additions and no multiplications. If $\overline{g_{oe}^{o6}} < Q_{oe}^{o6}$, then $F(1,6)$, $F(3,6)$, $F(5,6)$ and $F(7,6)$ are deemed to be zero. Otherwise, we compute as follows:

$$h_{oe}(x) = k_1 [f^{oe}(x,0) - f^{oe}(x,3)] - [f^{oe}(x,1) - f^{oe}(x,2)]$$

for $x=0,1,2,3$, and where $k_1 = 0.41421$ is the ratio of $C(u,6,x,0)$ to $C(u,6,x,1)$. This requires 4×4 additions and 4 multiplications. Then we apply the following threshold condition:

If $|h_{oe}(0) + h_{oe}(1) + h_{oe}(2) + h_{oe}(3)| < Q_{oe}^{16}$, then $F(1,6)$, $F(3,6)$, $F(5,6)$ and $F(7,6)$ are deemed to be zero. Otherwise, we compute:

$$F(1,6) = \sum_{x=0}^3 h_{oe}(x)C(1,6,x,1)$$

Then we apply the following threshold condition:

5 If $|h_{oe}(0) - h_{oe}(1) - h_{oe}(2) - h_{oe}(3)| < Q_{oe}^{36}$, then $F(3,6)$, $F(5,6)$ and $F(7,6)$ are deemed to be zero. Otherwise, we compute:

$$F(3,6) = \sum_{x=0}^3 h_{oe}(x)C(3,6,x,1)$$

Then we apply the following threshold condition:

10 If $|h_{oe}(0) - h_{oe}(1) + h_{oe}(2) + h_{oe}(3)| < Q_{oe}^{56}$, then $F(5,6)$ and $F(7,6)$ are deemed to be zero. Otherwise, we compute:

$$F(5,6) = \sum_{x=0}^3 h_{oe}(x)C(5,6,x,1)$$

Then we apply the following threshold condition:

If $|h_{oe}(0) - h_{oe}(1) + h_{oe}(2) - h_{oe}(3)| < Q_{oe}^{76}$, then, $F(7,6)$ is deemed to be zero.

Otherwise, we compute:

15
$$F(7,6) = \sum_{x=0}^3 h_{oe}(x)C(7,6,x,1)$$

This completes the decomposition for when u is odd and v equal to 6. We note that additional reduction in computations can be achieved by computing, storing and re-using the partial sums, $h_{oe}(2) + h_{oe}(3)$, $h_{oe}(0) - h_{oe}(1)$, $h_{oe}(2) - h_{oe}(3)$ as required for the application of the threshold conditions just given.

20 This completes the process of determining $F(u, v)$ for u odd and v even. Next, the decomposition of the Discrete Cosine Transform for u even and v odd will be described.

Decomposition for Even u and Odd v : When u is even and v is odd, we have:

$$F(u, v) = \sum_{x=0}^3 \sum_{y=0}^3 f^{eo} C(u, v, x, y)$$

where:

$$C(u, v, x, y) = \frac{1}{4} C(u) C(v) \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

with: $f^{uv}(x, y) = f^{eo}(x, y) = f(x, y) - f(x, 7-y) + f(7-x, y) - f(7-x, 7-y)$

for $v = 1, 3, 5, 7$ and $u = 0, 2, 4, 6$.

5 To facilitate selective computation of $F(u, v)$ for these frequency pairs we compute the following functions:

$$g_{eo}^{uo}(0, y) = g_{eo}^{0o}(y) = f^{eo}(0, y) + f^{eo}(1, y) + f^{eo}(2, y) + f^{eo}(3, y);$$

$$g_{eo}^{uo}(1, y) = g_{eo}^{2o}(y) = f^{eo}(0, y) + f^{eo}(1, y) - f^{eo}(2, y) - f^{eo}(3, y);$$

$$g_{eo}^{uo}(2, y) = g_{eo}^{4o}(y) = f^{eo}(0, y) - f^{eo}(1, y) - f^{eo}(2, y) + f^{eo}(3, y);$$

$$10 \quad g_{eo}^{uo}(3, y) = g_{eo}^{6o}(y) = f^{eo}(0, y) - f^{eo}(1, y) + f^{eo}(2, y) - f^{eo}(3, y).$$

for $v = 1, 3, 5, 7$, $y = 0, 1, 2, 3$ and $u = 0, 2, 4, 6$.

The functions, $g_{eo}^{uo}(x, y)$, are the algebraic sum of values of $f^{eo}(x, y)$ that have coordinates equidistant from an axis of symmetry. The functions g_{eo}^{uo} can be computed efficiently using the digital computation structure of Figure 16. The structure of Figure 16 shows that the computation of the functions g_{eo}^{uo} requires 4×8 additions and no multiplications. The data of g_{eo}^{uo} may replace in memory the data of $f^{eo}(x, y)$ as the computations depicted in Figure 16 are performed, thereby allowing for yet a further reduction in the amount of memory required to compute the Discrete Cosine Transform.

To compute the transform components for v odd and $u = 0$ we proceed as follows.
20 We first apply a threshold condition by computing:

$$\overline{g_{eo}^{0o}} = \sum_{y=0}^3 |g_{eo}^{0o}(y)|$$

This requires 4 additions and no multiplications. If $\overline{g_{eo}^{0o}} < Q_{eo}^{01}$, then the transform components for v odd and u even are deemed to be zero. Otherwise, we compute as follows:

$$25 \quad F(0, 1) = \sum_{y=0}^3 g_{eo}^{0o}(y) C(0, 1, 0, y)$$

If $\overline{g_{eo}^{0o}} < Q_{eo}^{03}$, then the transform components for $\nu = 3, 5, 7$ and $u = 0$ are deemed to be zero. Otherwise, we compute as follows:

$$F(0,3) = \sum_{y=0}^3 g_{eo}^{0o}(y)C(0,3,0,y)$$

If $\overline{g_{eo}^{0o}} < Q_{eo}^{05}$, then the transform components for $\nu = 5, 7$ and $u = 0$ are deemed to be zero. Otherwise, we compute as follows:

$$F(0,5) = \sum_{y=0}^3 g_{eo}^{0o}(y)C(0,5,0,y)$$

If $\overline{g_{eo}^{0o}} < Q_{eo}^{07}$, then the transform component for $\nu = 7$ and $u = 0$ is deemed to be zero. Otherwise, we compute as follows:

$$F(0,7) = \sum_{y=0}^3 g_{eo}^{0o}(y)C(0,7,0,y)$$

10 This completes the computations for the transform components for ν odd and $u = 0$. To compute the transform components for ν odd and $u = 4$ we proceed as follows. We first apply a threshold condition by computing:

$$\overline{g_{eo}^{4o}} = \sum_{y=0}^3 |g_{eo}^{4o}(y)|$$

15 This requires 4 additions and no multiplications. If $\overline{g_{eo}^{4o}} < Q_{eo}^{41}$, then the transform components for $\nu = 1, 3, 5, 7$ and $u = 4$ are deemed to be zero. Otherwise, we compute as follows:

$$F(4,1) = \sum_{y=0}^3 g_{eo}^{4o}(y)C(4,1,0,y)$$

If $\overline{g_{eo}^{4o}} < Q_{eo}^{43}$, then the transform components for $\nu = 3, 5, 7$ and $u = 4$ are deemed to be zero. Otherwise, we compute as follows:

20
$$F(4,3) = \sum_{y=0}^3 g_{eo}^{4o}(y)C(4,3,0,y)$$

If $\overline{g_{eo}^{4o}} < Q_{eo}^{45}$, then the transform components for $\nu = 5, 7$ and $u = 4$ are deemed to be zero. Otherwise, we compute as follows:

$$F(4,5) = \sum_{y=0}^3 g_{eo}^{4o}(y)C(4,5,0,y)$$

If $\overline{g_{eo}^{4o}} < Q_{eo}^{47}$, then the transform component for $v = 7$ and $u = 4$ is deemed to be zero. Otherwise, we compute as follows:

$$F(4,7) = \sum_{y=0}^3 g_{eo}^{4o}(y)C(4,7,0,y)$$

5 This completes the computations for the transform components for u odd and $v = 4$.

To compute the transform components for v odd and $u = 2$ we proceed as follows. We first apply a threshold condition by computing:

$$\overline{g_{eo}^{2o}} = \sum_{y=0}^3 |g_{eo}^{2o}(y)|$$

10 This requires 4 additions and no multiplications. If $\overline{g_{eo}^{2o}} < Q_{eo}^{21}$, then $F(2,1)$, $F(2,3)$, $F(2,5)$ and $F(2,7)$ are deemed to be zero. Otherwise, we compute as follows:

$$h_{eo}(y) = [f^{eo}(0,y) - f^{eo}(3,y)] + k_1[f^{eo}(1,y) - f^{eo}(2,y)]$$

for $y=0,1,2,3$, and where $k_1 = 0.41421$ is the ratio of $C(2,v,1,y)$ to $C(2,v,0,y)$. This requires 4x4 additions and 4 multiplications. Note that the quantities in the square
15 brackets can be obtained from nodes A and B of the digital computation structure of Figure 16. Then we apply the following threshold condition:

If $|h_{eo}(0) + h_{eo}(1) + h_{eo}(2) + h_{eo}(3)| < Q_{eo}^{21}$, then $F(2,1)$, $F(2,3)$, $F(2,5)$ and $F(2,7)$ are deemed to be zero. Otherwise, we compute:

$$F(2,1) = \sum_{y=0}^3 h_{eo}(y)C(2,1,0,y)$$

20 Then we apply the following threshold condition:

If $|h_{eo}(0) - h_{eo}(1) - h_{eo}(2) - h_{eo}(3)| < Q_{eo}^{23}$, then $F(2,3)$, $F(2,5)$ and $F(2,7)$ are deemed to be zero. Otherwise, we compute:

$$F(2,3) = \sum_{y=0}^3 h_{eo}(y)C(2,3,0,y)$$

Then we apply the following threshold condition:

If $|h_{eo}(0) - h_{eo}(1) + h_{eo}(2) + h_{eo}(3)| < Q_{eo}^{25}$, then, $F(2,5)$ and $F(2,7)$ are deemed to be zero. Otherwise, we compute:

$$F(2,5) = \sum_{y=0}^3 h_{eo}(y)C(2,5,0,y)$$

Then we apply the following threshold condition:

5 If $|h_{eo}(0) - h_{eo}(1) + h_{eo}(2) - h_{eo}(3)| < Q_{eo}^{27}$, then, $F(2,7)$ is deemed to be zero.

Otherwise, we compute:

$$F(2,7) = \sum_{y=0}^3 h_{eo}(y)C(2,7,0,y)$$

This completes the decomposition for when v is odd and u equal to 2. We note that additional reduction in computations can be achieved by computing, storing and re-
10 using the partial sums, $h_{eo}(2) + h_{eo}(3)$, $h_{eo}(0) - h_{eo}(1)$, $h_{eo}(2) - h_{eo}(3)$ as required for the application of the threshold conditions just given.

To compute the transform components for u odd and $v = 6$ we proceed as follows. We first apply a threshold condition by computing:

$$\overline{g_{eo}^{6o}} = \sum_{y=0}^3 |g_{eo}^{6o}(y)|$$

15 This requires 4 additions and no multiplications. If $\overline{g_{eo}^{6o}} < Q_{eo}^{6o}$, then $F(6,1)$, $F(6,3)$, $F(6,5)$ and $F(6,7)$ are deemed to be zero. Otherwise, we compute as follows:

$$h_{eo}(y) = k_1 [f^{eo}(0,y) - f^{eo}(3,y)] - [f^{eo}(1,y) - f^{eo}(2,y)]$$

for $y=0,1,2,3$, and where $k_1 = 0.41421$ is the ratio of $C(6,v,0,y)$ to $C(6,v,1,y)$. This requires 4×4 additions and 4 multiplications. Then we apply the following threshold
20 condition:

If $|h_{eo}(0) + h_{eo}(1) + h_{eo}(2) + h_{eo}(3)| < Q_{eo}^{61}$, then $F(6,1)$, $F(6,3)$, $F(6,5)$ and $F(6,7)$ are deemed to be zero. Otherwise, we compute:

$$F(6,1) = \sum_{y=0}^3 h_{eo}(y)C(6,1,1,y)$$

Then we apply the following threshold condition:

If $|h_{eo}(0) - h_{eo}(1) - h_{eo}(2) - h_{eo}(3)| < Q_{eo}^{63}$, then $F(6,3)$, $F(6,5)$ and $F(6,7)$ are deemed to be zero. Otherwise, we compute:

$$F(6,3) = \sum_{y=0}^3 h_{eo}(y)C(6,3,1,y)$$

Then we apply the following threshold condition:

5 If $|h_{eo}(0) - h_{eo}(1) + h_{eo}(2) + h_{eo}(3)| < Q_{eo}^{65}$, then, $F(6,5)$ and $F(6,7)$ are deemed to be zero. Otherwise, we compute:

$$F(6,5) = \sum_{y=0}^3 h_{eo}(y)C(6,5,1,y)$$

Then we apply the following threshold condition:

If $|h_{eo}(0) - h_{eo}(1) + h_{eo}(2) - h_{eo}(3)| < Q_{eo}^{67}$, then, $F(6,7)$ is deemed to be zero.

10 Otherwise, we compute:

$$F(6,7) = \sum_{y=0}^3 h_{eo}(y)C(6,7,1,y)$$

This completes the decomposition for when v is odd and u equal to 6. We note that additional reduction in computations can be achieved by computing, storing and re-using the partial sums, $h_{eo}(2) + h_{eo}(3)$, $h_{eo}(0) - h_{eo}(1)$, $h_{eo}(2) - h_{eo}(3)$ as required for the application of the threshold conditions just given.

This completes the process of determining $F(u, v)$ for v odd and u even. Next, the decomposition of the Discrete Cosine Transform for u odd and v odd will be described.

Decomposition for Odd u and Odd v : When u and v are both odd, $F(u, v)$ is determined according to:

$$F(u, v) = \sum_{x=0}^3 \sum_{y=0}^3 f^{oo}(x, y)C(u, v, x, y)$$

where:

$$C(u, v, x, y) = \frac{1}{4}C(u)C(v) \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

for u and v both odd and:

$$f^{oo}(x, y) = f(x, y) - f(x, 7 - y) - f(7 - x, y) + f(7 - x, 7 - y)$$

Further reduction in computations can be achieved as follows. First, the order of computing $F(u, v)$ is as shown in Figure 2, so that successive values of $F(u, v)$ can be expected to be less in magnitude than a preceding value. For each transform component, $F(u, v)$, computed, its magnitude is subtracted from $\overline{f^{oo}}$ as computed above so that $\overline{f^{oo}}$ is reduced according to:

$$\overline{f^{oo}} = \overline{f^{oo}} - |F(u, v)| \times \alpha$$

where α is a scale factor that may be adjusted to achieve a desired level of correlation between the transform components that are deemed to be zero and the transform components that would actually quantized to zero if computed and quantized. Decreasing α increases that level of correlation, whereas increasing α decreases that level of correlation. As α is decreased to zero, the correlation approaches 100%. For α equal to 8, the correlation is about 75%. Setting α equal to 4 yields about 95% correlation.

Then, if the remaining value of $\overline{f^{oo}}$ is less than the given threshold Q^{oo} , then the remaining values of $F(u, v)$ are deemed to be zero. Otherwise, the next descending value of $F(u, v)$ is computed, $\overline{f^{oo}}$ is again reduced according to:

$$\overline{f^{oo}} = \overline{f^{oo}} - |F(u, v)| \times \alpha$$

and $\overline{f^{oo}}$ is again compared to the threshold. This process is continued until $\overline{f^{oo}} < Q^{oo}$.

Thus, selective computation of the transform components, $F(u, v)$, when u and v are both odd, is accomplished by computing the scaled magnitude of each value of $F(u, v)$ when u and v are both odd, in a sequential order until the scaled magnitude of a value of $F(u, v)$ in the sequence falls below a threshold level to which it is compared; wherein for each scaled magnitude of $F(u, v)$ computed, the threshold level to which it is compared is computed as the threshold level to which the previous value of the scaled magnitude of $F(u, v)$ was compared minus the scaled magnitude of the previous value of $F(u, v)$ computed; and deeming a value of $F(u, v)$ to be zero if its scaled magnitude falls

below the threshold to which it is compared and deeming to be zero all subsequent values of $F(u, v)$ in the sequence.

This completes the process of selectively determining the transform components, $F(u, v)$, of the discrete cosine transform. To summarize, according to the present invention, the DCT of $N \times N$ pixel image is successively decomposed into sets of intermediate functions which possess even or odd symmetry according to whether a scale of the frequency variables, (u, v) , are even or odd integers. Specifically, the DCT can be decomposed in terms of functions that possess even or odd symmetry according to whether $u/2^n$ and $v/2^n$, $n=0, 1, 2, \dots, \log_2(N) - 1$, are even or odd integers. A threshold condition is then applied to one or more of these functions. If the function fails the threshold condition, the corresponding one or more transform components are deemed to be zero. Otherwise, the transform component is computed and quantized. All quantization factors, as well as the cosine functions, can be computed in advance, stored in memory tables, and retrieved as needed. Finally, when further decomposition of the DCT is not possible, successive threshold conditions are applied to the remaining transform components to further reduce the computations required to compute the DCT.

The methods of the invention described above may be implemented in a microprocessor, digital signal processor or other processing device operating under the directions of software in conjunction with sufficient memory. Alternatively, the invention may be implemented in an applications specific integrated circuit or other suitably designed hardware.

The foregoing written description of preferred embodiments illustrate that the present invention is well suited to achieve a substantial increase in the speed and efficiency of video compression while minimizing computational complexity and memory usage. Persons of ordinary skill in the art will readily recognize variations of these embodiments, given the disclosure herein, and, therefore, the invention is not limited by the foregoing written description. Rather, the invention is defined by the appended claims.

What is claimed is:

Claims

1. A method for selectively determining components, $F(u, v)$, of a discrete cosine transform of a two-dimensional image, $f(x, y)$, comprising the steps of:

computing a set of functions, $f^{uv}(x, y)$, possessing even or odd symmetry

5 about an axis of the image; and

deeming a transform component, $F(u, v)$, to be zero if a function, $f^{uv}(x, y)$, fails to satisfy a threshold condition applied to the function, $f^{uv}(x, y)$.

2. The method of claim 1, wherein the step of computing a function, $f^{uv}(x, y)$, possessing symmetry about an axis comprises the step of determining an algebraic sum of image pixels that have coordinates equidistant from the axis of symmetry.

3. The method of claim 1, wherein values of the image are replaced in memory by values of a function $f^{uv}(x, y)$.

4. The method of claim 1, wherein the step of applying a threshold condition to a function, $f^{uv}(x, y)$, comprises determining if the sum of the magnitudes of the values of the function exceeds a threshold value.

5. The method of claim 1, further comprising the steps of:

determining a sequential order of transform components;

deeming a transform component to be zero if a scaled magnitude of the transform component fails a threshold condition; and

deeming to be zero all transform components in the sequence that succeed a transform component deemed to be zero.

6. The method of claim 1, further comprising the steps of:

computing a set of functions, $g^{uv}(x, y)$, derived from a function, $f^{uv}(x, y)$,

possessing even or odd symmetry about an axis of the image; and

deeming a transform component, $F(u, v)$, to be zero if a function, $g^{uv}(x, y)$, fails to satisfy a threshold condition applied to the function, $f^{uv}(x, y)$.

7. The method of claim 6, wherein the step of computing a function, $g^{uv}(x, y)$, possessing symmetry about an axis comprises the step of determining an algebraic sum of values of a function, $f^{uv}(x, y)$, that have coordinates that are equidistant from the axis of symmetry.

5 8. The method of claim 6, wherein values of a function, $f^{uv}(x, y)$, are replaced in memory by values of a function $g^{uv}(x, y)$.

9. The method of claim 6, wherein the step of applying a threshold condition to a function, $g^{uv}(x, y)$, comprises determining if the sum of the magnitudes of the values of the function exceeds a threshold value.

10 10. The method of claim 1, further comprising the steps of:
determining a sequential order of transform components;
deeming a transform component to be zero if a scaled magnitude of the transform component fails a threshold condition; and
deeming to be zero all transform components in the sequence that succeed
15 a transform component deemed to be zero.

11. A video compression system for selectively determining components, $F(u, v)$, of a discrete cosine transform of a two-dimensional image, $f(x, y)$, comprising:
a processor for:

20 computing a set of one or more functions, $f^{uv}(x, y)$, possessing even or odd symmetry about an axis of the image; and
deeming a transform component, $F(u, v)$, to be zero if a function, $f^{uv}(x, y)$, fails to satisfy a threshold condition applied to the function, $f^{uv}(x, y)$; and

memory for:
25 storing values of $F(u, v)$ and $f^{uv}(x, y)$.

12. The system of claim 11, wherein the process of computing a function, $f^{uv}(x, y)$, possessing symmetry about an axis comprises the step of determining an

algebraic sum of image pixels that have coordinates equidistant from the axis of symmetry.

13. The system of claim 11, wherein values of the image are replaced in memory by values of a function $f^{uv}(x, y)$.

5 14. The system of claim 11, wherein the process of applying a threshold condition to a function, $f^{uv}(x, y)$, comprises determining if the sum of the magnitudes of the values of the function exceeds a threshold value.

15. The system of claim 11, further comprising the process of:
determining a sequential order of transform components;

10 deeming a transform component to be zero if a scaled magnitude of the transform component fails a threshold condition; and

deeming to be zero all transform components in the sequence that succeed a transform component deemed to be zero.

16. The system of claim 11, further comprising the process of:

15 computing a set of functions, $g^{uv}(x, y)$, derived from a function, $f^{uv}(x, y)$, possessing even or odd symmetry about an axis of the image: and

deeming a transform component, $F(u, v)$, to be zero if a function, $g^{uv}(x, y)$, fails to satisfy a threshold condition applied to the function, $f^{uv}(x, y)$.

17. The system of claim 16, wherein the process of computing a function,
20 $g^{uv}(x, y)$, possessing symmetry about an axis comprises the step of determining an algebraic sum of values of a function, $f^{uv}(x, y)$, that have coordinates that are equidistant from the axis of symmetry.

18. The system of claim 16, wherein values of a function, $f^{uv}(x, y)$, are replaced in memory by values of a function $g^{uv}(x, y)$.

25 19. The system of claim 11, further comprising the steps of:

determining a sequential order of transform components;

deeming a transform component to be zero if a scaled magnitude of the transform component fails a threshold condition; and

deeming to be zero all transform components in the sequence that succeed a transform component deemed to be zero.

20. A method for selectively determining components, $F(u, v)$, of a discrete cosine transform of a two-dimensional image, $f(x, y)$, comprising the steps of:

5 (a) decomposing the discrete cosine transform to obtain a set of functions, $f^{uv}(x, y)$, possessing even or odd symmetry about an axis of the image; and

(b) deeming a transform component, $F(u, v)$, to be zero if a function, $f^{uv}(x, y)$, fails to satisfy a threshold condition applied to the function, $f^{uv}(x, y)$; and

(c) repeating steps (a) and (b) until no further decomposition can be
10 achieved.

PRIOR ART

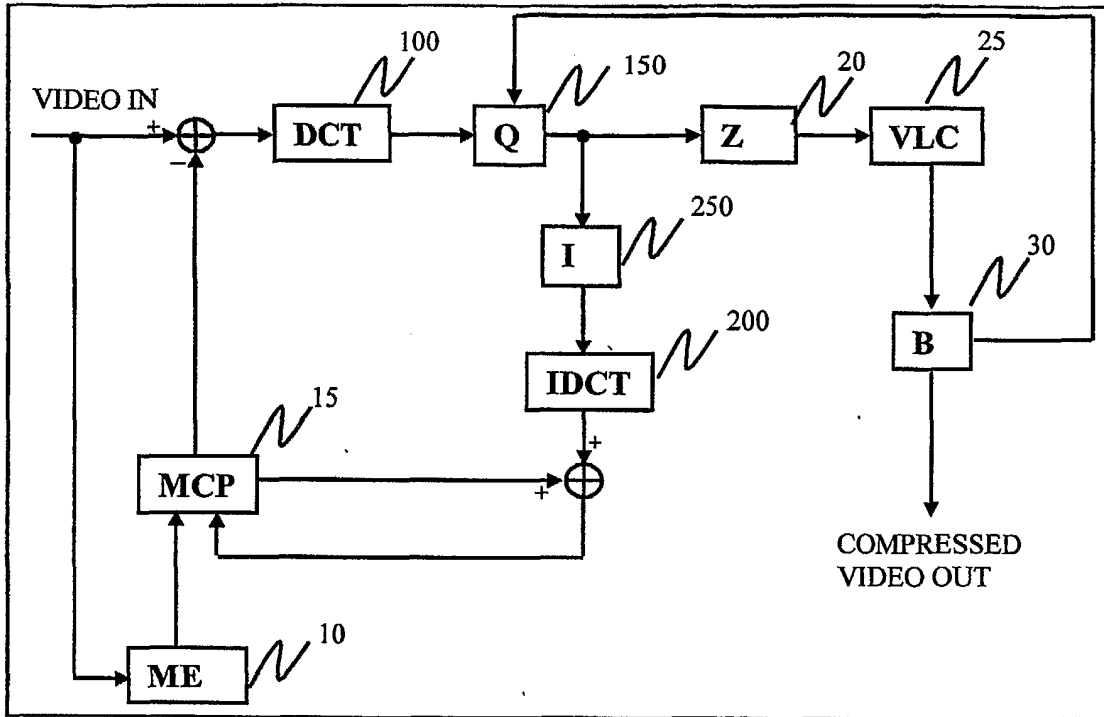


FIG. 1

PRIOR ART

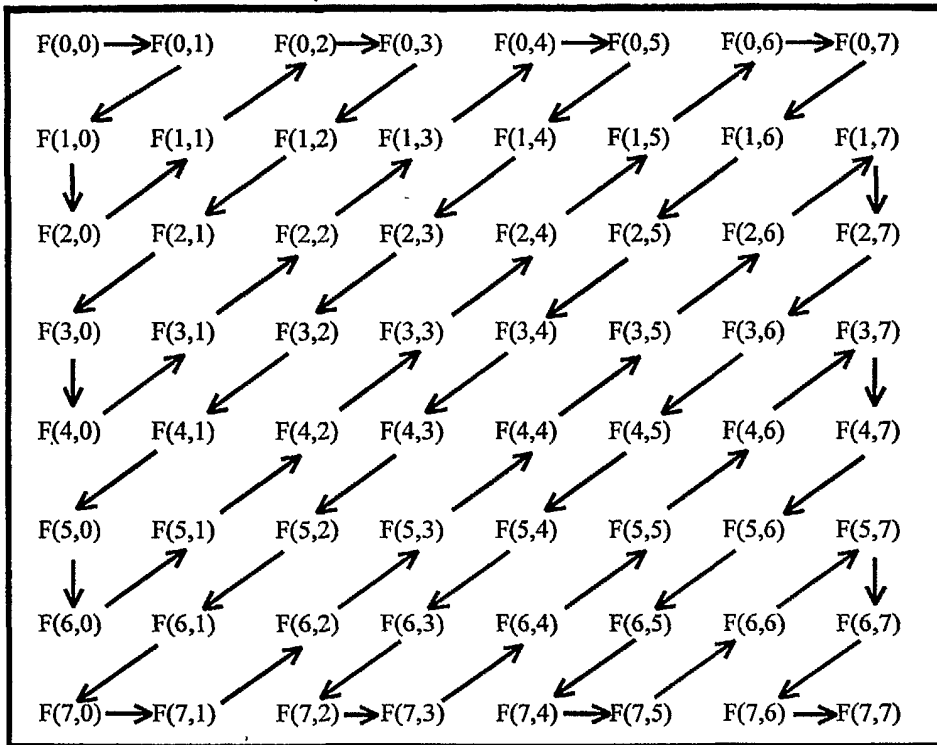


FIG. 2

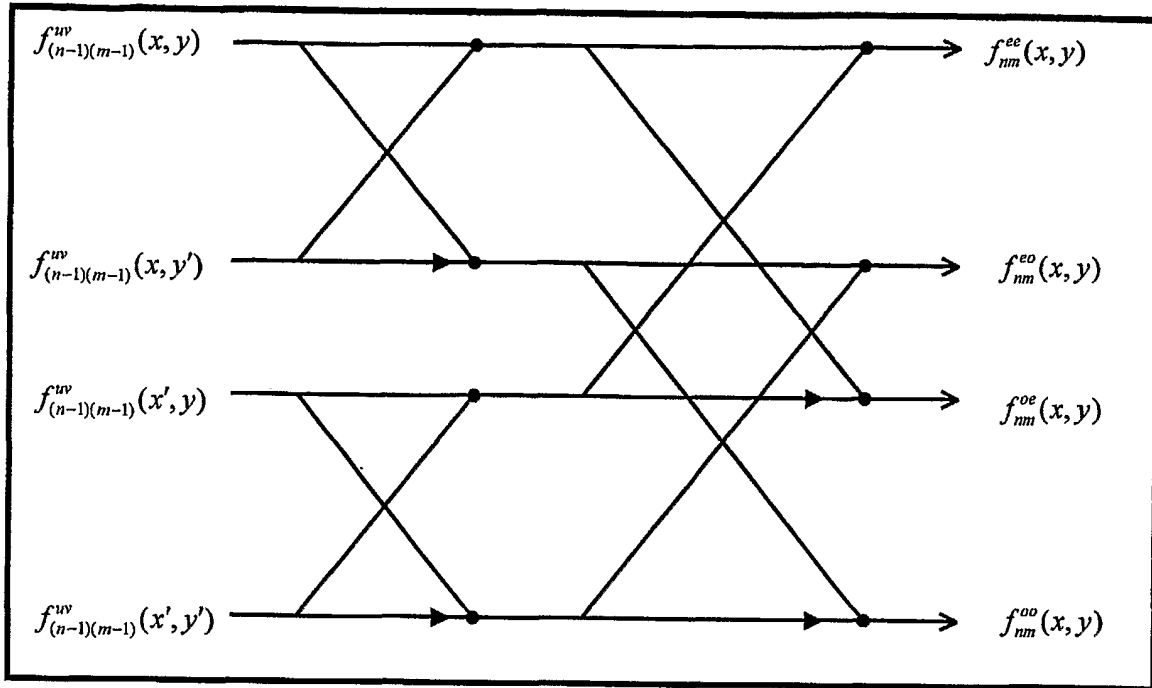


FIG. 3

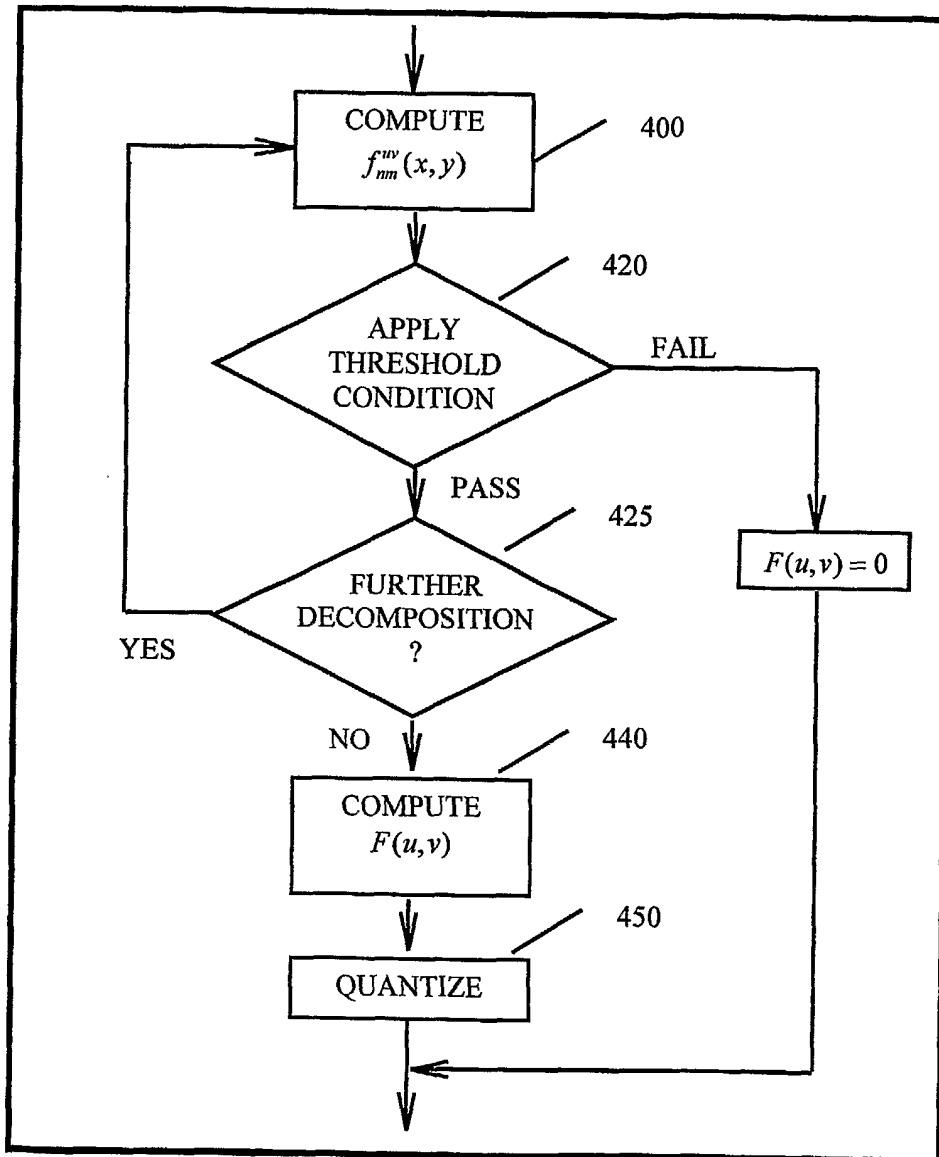


FIG. 4

5/12

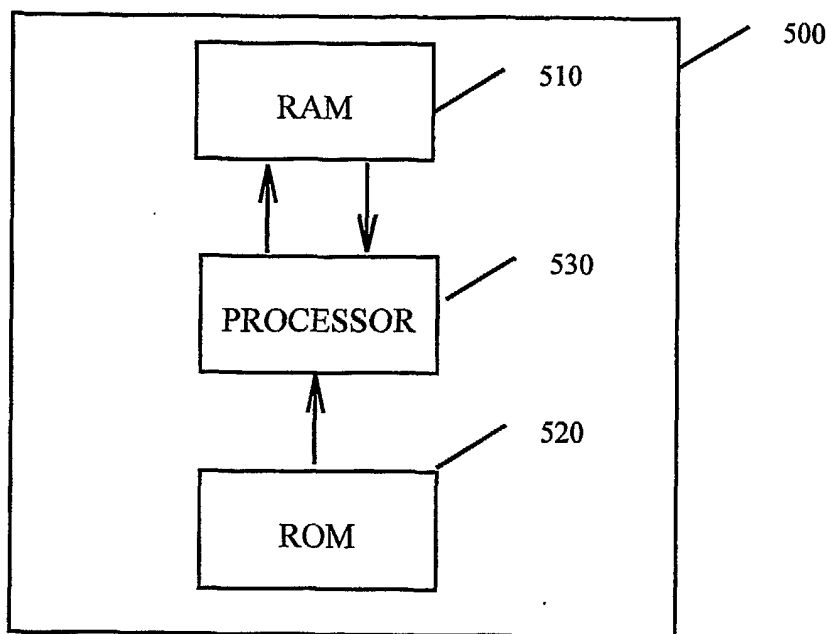


FIG. 5

6/12

PRIOR ART

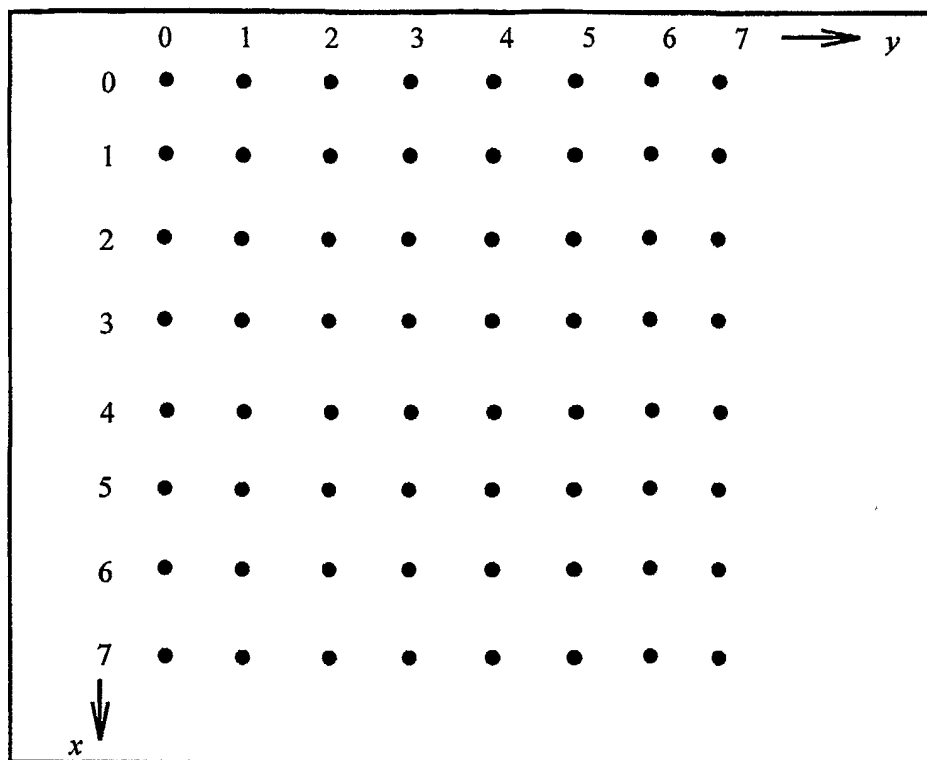


FIG. 6

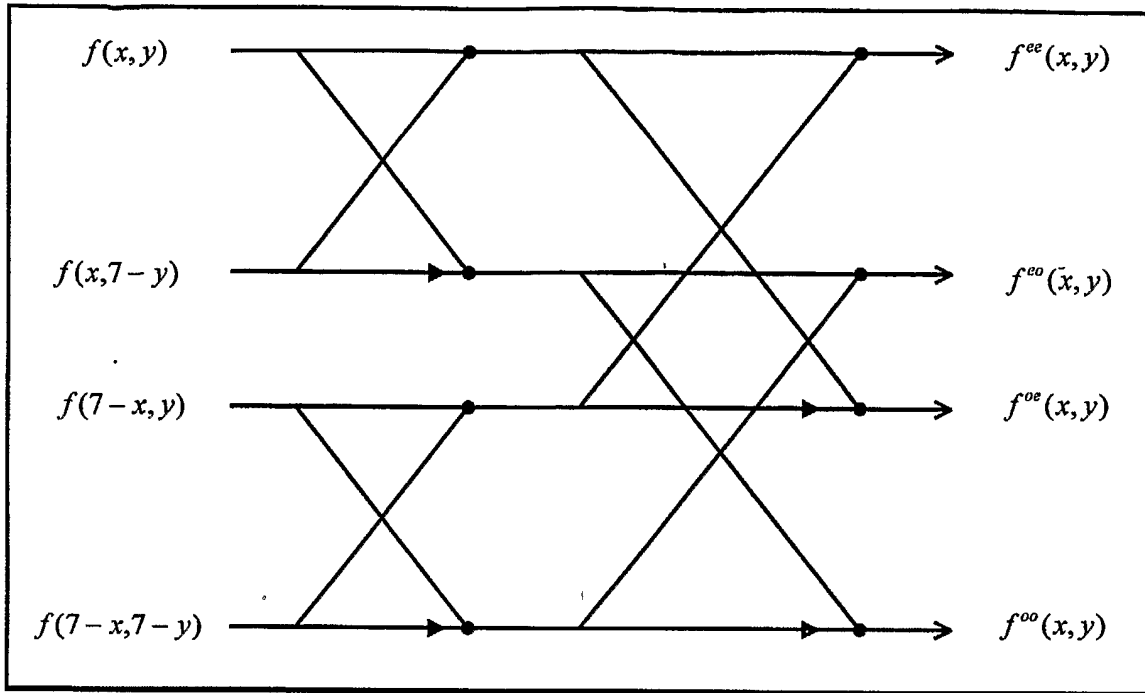


FIG. 7

$f^{ee}(0,0)$	$f^{ee}(0,1)$	$f^{ee}(0,2)$	$f^{ee}(0,3)$	$f^{eo}(0,3)$	$f^{eo}(0,2)$	$f^{eo}(0,1)$	$f^{eo}(0,0)$
$f^{ee}(1,0)$	$f^{ee}(1,1)$	$f^{ee}(1,2)$	$f^{ee}(1,3)$	$f^{eo}(1,3)$	$f^{eo}(1,2)$	$f^{eo}(1,1)$	$f^{eo}(1,0)$
$f^{ee}(2,0)$	$f^{ee}(2,1)$	$f^{ee}(2,2)$	$f^{ee}(2,3)$	$f^{eo}(2,3)$	$f^{eo}(2,2)$	$f^{eo}(2,1)$	$f^{eo}(2,0)$
$f^{ee}(3,0)$	$f^{ee}(3,1)$	$f^{ee}(3,2)$	$f^{ee}(3,3)$	$f^{eo}(3,3)$	$f^{eo}(3,2)$	$f^{eo}(3,1)$	$f^{eo}(3,0)$
$f^{oe}(3,0)$	$f^{oe}(3,1)$	$f^{oe}(3,2)$	$f^{oe}(3,3)$	$f^{oo}(3,3)$	$f^{oo}(3,2)$	$f^{oo}(3,1)$	$f^{oo}(3,0)$
$f^{oe}(2,0)$	$f^{oe}(2,1)$	$f^{oe}(2,2)$	$f^{oe}(2,3)$	$f^{oo}(2,3)$	$f^{oo}(2,2)$	$f^{oo}(2,1)$	$f^{oo}(2,0)$
$f^{oe}(1,0)$	$f^{oe}(1,1)$	$f^{oe}(1,2)$	$f^{oe}(1,3)$	$f^{oo}(1,3)$	$f^{oo}(1,2)$	$f^{oo}(1,1)$	$f^{oo}(1,0)$
$f^{oe}(0,0)$	$f^{oe}(0,1)$	$f^{oe}(0,2)$	$f^{oe}(0,3)$	$f^{oo}(0,3)$	$f^{oo}(0,2)$	$f^{oo}(0,1)$	$f^{oo}(0,0)$

FIG. 8

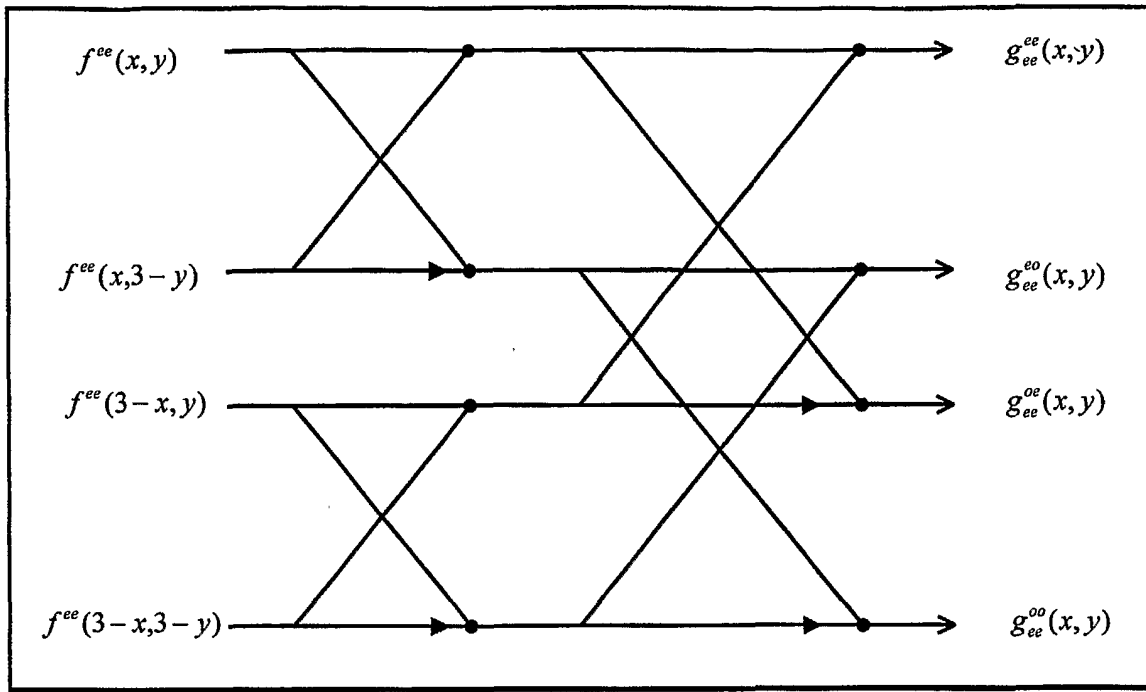


FIG. 9

$g_{ee}^{ee}(0,0)$	$g_{ee}^{ee}(0,1)$	$g_{ee}^{eo}(0,1)$	$g_{ee}^{eo}(0,0)$
$g_{ee}^{ee}(1,0)$	$g_{ee}^{ee}(1,1)$	$g_{ee}^{eo}(1,1)$	$g_{ee}^{eo}(1,0)$
$g_{ee}^{oe}(1,0)$	$g_{ee}^{oe}(1,1)$	$g_{ee}^{oo}(1,1)$	$g_{ee}^{oo}(1,0)$
$g_{ee}^{oe}(0,0)$	$g_{ee}^{oe}(0,1)$	$g_{ee}^{oo}(0,1)$	$g_{ee}^{oo}(0,0)$

FIG. 10

10/12

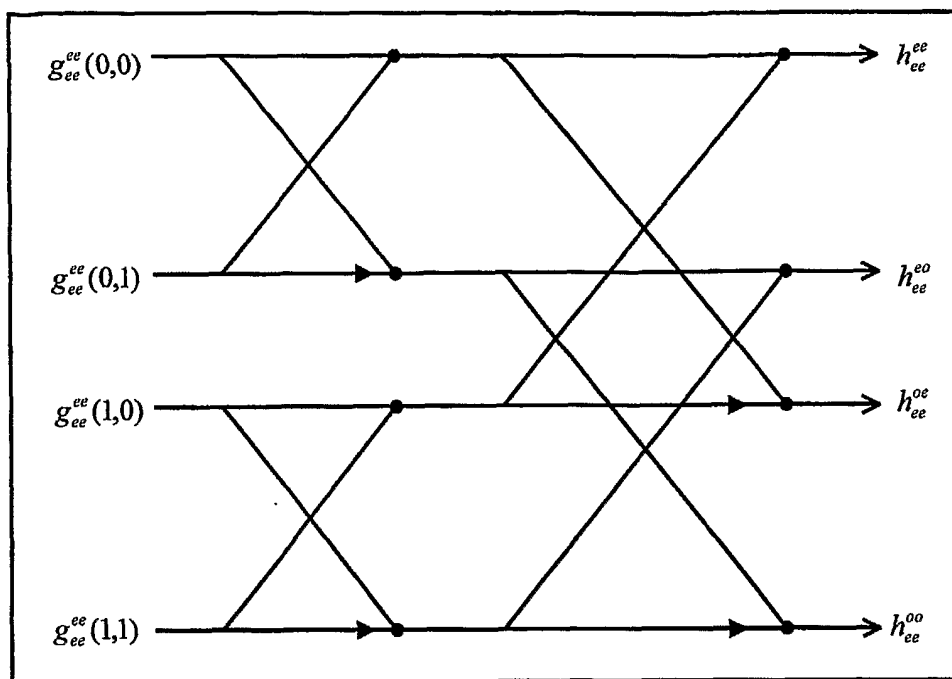


FIG. 11

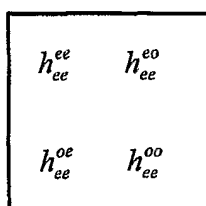


FIG. 12

11/12

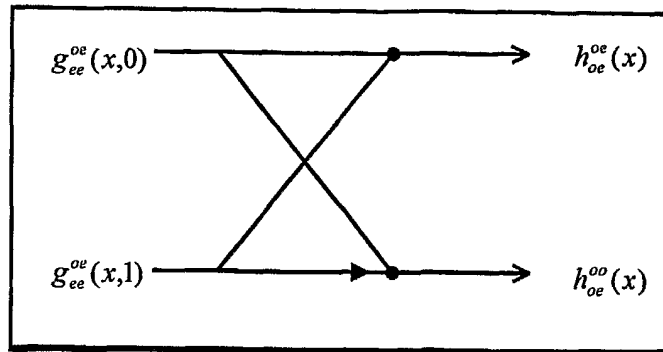


FIG. 13

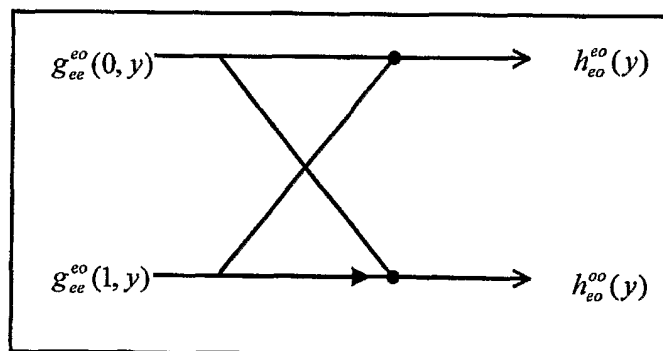


FIG. 14

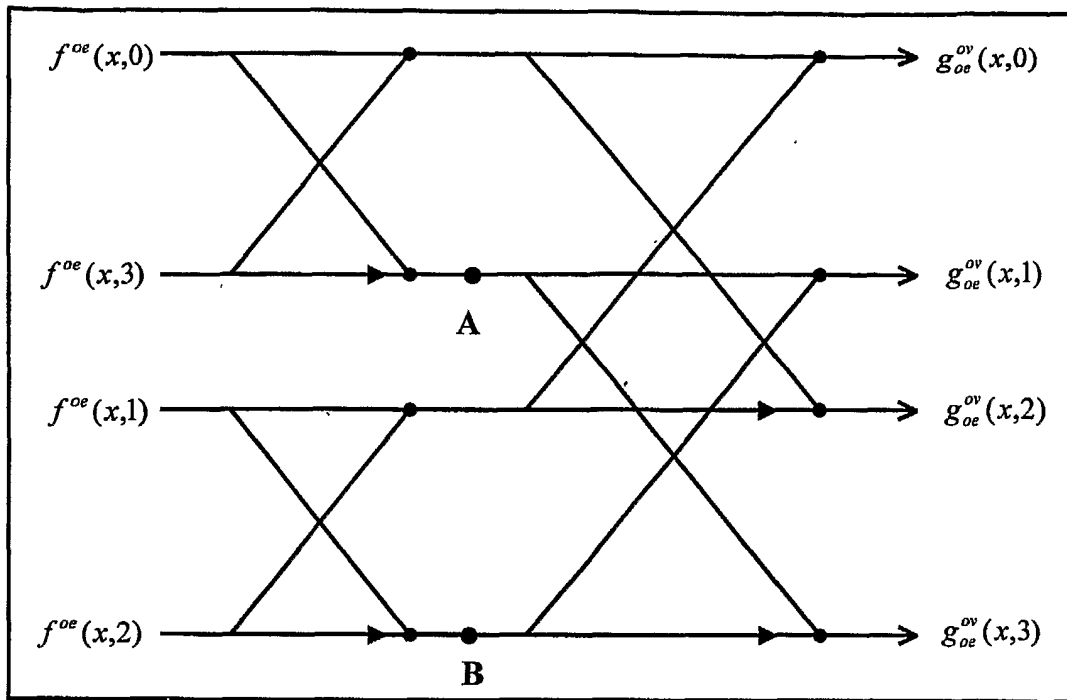


FIG. 15

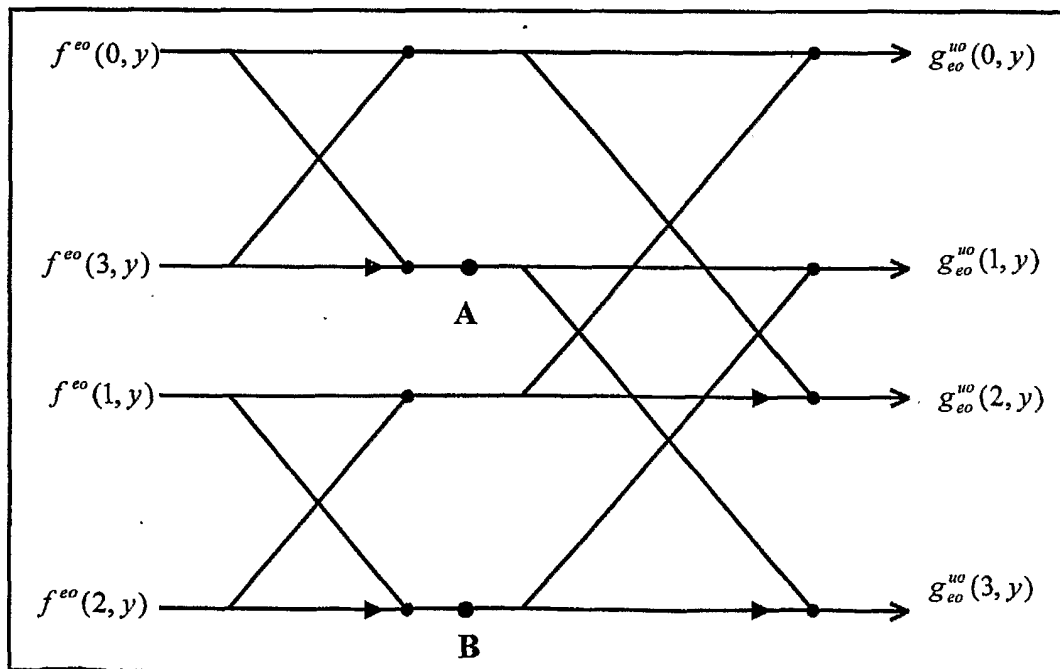


FIG. 16

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 01/40999

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N7/50 G06F17/14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

WPI Data, PAJ, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 301 136 A (MCMILLAN JR LEONARD ET AL) 5 April 1994 (1994-04-05) the whole document ----	1,11,20
A	US 4 841 464 A (CASSIMATIS ERIC ET AL) 20 June 1989 (1989-06-20) the whole document -----	1,11,20

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

° Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *&* document member of the same patent family

Date of the actual completion of the international search

9 October 2001

Date of mailing of the international search report

17/10/2001

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Gries, T

INTERNATIONAL SEARCH REPORT

Information on patent family members

In Application No
PCT/US 01/40999

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
US 5301136	A	05-04-1994	JP	6217146 A	05-08-1994
US 4841464	A	20-06-1989	FR	2582424 A1	28-11-1986
			CA	1252212 A1	04-04-1989
			DE	3687430 D1	18-02-1993
			DE	3687430 T2	03-06-1993
			EP	0204603 A1	10-12-1986
			ES	555225 D0	01-05-1989
			ES	8900240 A1	01-07-1989
			JP	2710928 B2	10-02-1998
			JP	61272870 A	03-12-1986