

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
6 April 2006 (06.04.2006)

PCT

(10) International Publication Number
WO 2006/036796 A1

(51) International Patent Classification:
H04N 7/26 (2006.01)

(21) International Application Number:
PCT/US2005/034164

(22) International Filing Date:
22 September 2005 (22.09.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/946,940 22 September 2004 (22.09.2004) US

(71) Applicant (for all designated States except US):
HEWLETT-PACKARD DEVELOPMENT COMPANY, L. P. [US/US]; 20555 S. H. 249, Houston, TX 77070 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **STAE LIN, Carl** [US/IL]; Technion City, 32000 Haifa (IL). **FISCHER, Mani** [IL/IL]; Technion City, 32000 Haifa (IL). **NACHLIL, Hila** [IL/IL]; Technion City, 32000 Haifa (IL).

(74) Agent: **HEMINGER, Susan, E.**; Hewlett-Packard Company, Intellectual Property Administration, P O Box 272400, M/S 35, Fort Collins, CO 80527-2400 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

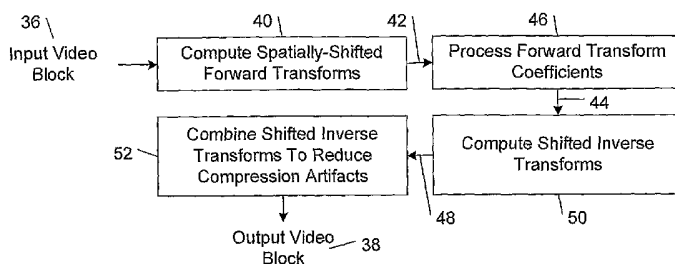
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

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: PROCESSING VIDEO FRAMES



(57) Abstract: Methods, machines, and computer-readable media storing machine-readable instructions for processing video frames are described. In one aspect, a repetitive set of three-dimensional forward transform coefficients (42) is computed for each of multiple positions of a three-dimensional blocking grid relative to an input video block (36) comprising a selected set of video frames. The sets of three-dimensional forward transform coefficients (42) are processed. A respective three-dimensional inverse transform (48) is computed from each set of processed forward transform coefficients (42). An output video block (38) is generated based on the computed three-dimensional inverse transforms (48).

-1-

PROCESSING VIDEO FRAMES

BACKGROUND

Digital images and video frames are compressed in order to reduce data storage and transmission requirements. In most image compression methods, certain image data is discarded selectively to reduce the amount of data needed to represent the image while avoiding substantial degradation of the appearance of the image.

Transform coding is a common image compression method that involves representing an image by a set of transform coefficients. The transform coefficients are quantized individually to reduce the amount of data that is needed to represent the image. A representation of the original image is generated by applying an inverse transform to the transform coefficients. Block transform coding is a common type of transform coding method. In a typical block transform coding process, an image is divided into small rectangular regions (or "blocks"), which are subjected to forward transform, quantization and coding operations. Many different kinds of block transforms may be used to encode the blocks. Among the common types of block transforms are the cosine transform (which is the most common), the Fourier transform, the Hadamard transform, and the Haar wavelet transform. These transforms produce an $M \times N$ array of transform coefficients from an $M \times N$ block of image data, where M and N have integer values of at least 1.

The quality of images and video frames often are degraded by the presence of noise. A block transform coding process is a common source of noise in compressed image and video frames. For example, discontinuities often are introduced at the block boundaries in the reconstructed images and video frames, and ringing artifacts often are introduced near image boundaries.

SUMMARY

The invention features methods, machines, and computer-readable media storing machine-readable instructions for processing video frames.

In one aspect, the invention features a method of processing a sequence of video frames. In accordance with this inventive method, a respective set of three-

-2-

dimensional forward transform coefficients is computed for each of multiple positions of a three-dimensional blocking grid relative to an input video block comprising a selected set of video frames. The sets of three-dimensional forward transform coefficients are processed. A respective three-dimensional inverse transform is computed from each set of processed forward transform coefficients. An output video block is generated based on the computed three-dimensional inverse transforms.

The invention also features a machine and a computer-readable medium storing machine-readable instructions for implementing the above-described video sequence processing method.

Other features and advantages of the invention will become apparent from the following description, including the drawings and the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram of a prior art system for compressing a video sequence and decompressing the compressed video sequence.

FIG. 2 is a diagrammatic view of an exemplary video block composed of a set of video frames selected from an input video sequence.

FIG. 3 is a flow diagram of an embodiment of a method of processing a compressed video sequence to produce an output video sequence characterized by reduced compression artifacts.

FIG. 4 is a block diagram of an embodiment of a video sequence processing system for implementing the method of FIG. 3.

FIG. 5 is a graph of the output of a denoising filter plotted as a function of input transform coefficient values.

FIG. 6 is a block diagram of an implementation of the output video generator module in shown in FIG. 4.

DETAILED DESCRIPTION

In the following description, like reference numbers are used to identify like elements. Furthermore, the drawings are intended to illustrate major features of exemplary embodiments in a diagrammatic manner. The drawings are not

-3-

intended to depict every feature of actual embodiments nor relative dimensions of the depicted elements, and are not drawn to scale.

FIG. 1 shows a prior art method of processing an original video sequence 10 to produce a compressed video sequence 12. In accordance with the illustrated method, an encoding module 13 applies a forward three-dimensional (3D) discrete cosine transform (DCT) to the original video sequence 10 to produce a set of forward transform coefficients 16 (block 14). Typically, each color plane of each video frame is divided into blocks of pixels (e.g., 8×8 pixel blocks), so-called video blocks from a sequence of frames are generated (e.g., $8 \times 8 \times 8$ pixel blocks), and the 3D DCT is applied to each video block. The encoding module 13 quantizes the forward transform coefficients 16 based on quantization tables 19 to produce a set of quantized forward coefficients 20 (block 18). During the quantization process 18, some of forward transform coefficient information is discarded, which enables the original video sequence 10 to be compressed. The encoding module 13 encodes the quantized forward transform coefficients using, for example, a variable length encoding technique based on Huffman tables 24 to produce the compressed video sequence 12 (block 22).

A decoding module 26 produces a decompressed video sequence 28 from the compressed video sequence 12 as follows. The decoding module 26 performs variable length decoding of the compressed video sequence 12 based on Huffman tables 24 (block 30). The decoding module 26 de-quantizes the decoded video data based on the same quantization tables 19 that were used to produce the compressed video sequence 12 (block 31). The decoding module 26 computes an inverse three-dimension DCT from the de-quantized video data to produce the decompressed video sequence 28 (block 32).

As explained above, the quality the resulting decompressed frames of the video sequence 28 often are degraded by noise and artifacts introduced by the 3D-DCT block transform coding process. For example, discontinuities often are introduced at the block boundaries in the reconstructed video frames, and ringing artifacts often are introduced near image boundaries.

The embodiments described below are configured to denoise video sequences. For example, these embodiments readily may be used to denoise home movies from sources like digital cameras, digital video cameras, and cell

-4-

phones. These embodiments also may be used to reduce artifacts inherently introduced by processes that are used to create compressed video sequences, including JPEG/MPEG artifacts in compressed video streams, such as VCD/DVD/broadcast video streams. In many instances, these embodiments

5 denoise and reduce video sequence compression artifacts without degrading video frame quality, such as by blurring features in the video frames. As described in detail below, some implementations of these embodiments are particularly well-suited to substantially reduce blocking compression artifacts that are introduced by block-transform-based compression techniques, such as block discrete cosine

10 transform (DCT) compression techniques.

Referring to FIG. 2, the video frame processing embodiments described in detail below operate with respect to input video blocks 36 that are composed of respective sets of L video frames 34 that are selected from a video frame sequence 35, where L is a positive integer. Each input video block 36 is defined with

15 respect to two spatial dimensions (x, y) and one temporal dimension (t) that corresponds to the temporal order of the frames 34 in the sequence 35.

FIG. 3 shows an embodiment of a method of processing an input video block 36 to produce a denoised output video block 38. The video block 36 is composed of a selected set of the video frames in a video sequence that is

20 generated by a block-transform-based image compression method, such as the method shown in FIG. 1. In the method of FIG. 3, the color planes of the frames in the video sequence are arranged into respective input video blocks 36 that are processed separately. If originally encoded (e.g., in accordance with a lossless encoding process), the frames of the input video block 36 initially are decoded

25 before being processed as follows.

Spatiotemporally-shifted, three-dimensional forward transforms are computed from the input video block 36 (block 40). In this process, a forward transform operation is applied to each of multiple positions of a three-dimensional blocking grid relative to the input video block 36 to produce multiple respective

30 sets of three-dimensional forward transform coefficients 42. In an implementation in which the input video block 36 was originally compressed based on blocks of L video frame patches of $M \times N$ pixels, the forward transform operation is applied to a subset of the input image data containing K shifts from the $L \times M \times N$

-5-

independent shifts possible in an $L \times M \times N$ transform to produce K sets of forward transform coefficients, where K , L , M , and N have integer values of at least 1. In one exemplary implementation, both M and N have a value of 8.

The three-dimensional forward transform coefficients 42 of each set are
5 processed as explained in detail below to produce respective sets of processed forward transform coefficients 44 (block 46). In general, the forward transform coefficients 42 may be processed in any of a wide variety of different ways. In some implementations, a filter (e.g., a denoising filter, a sharpening filter, a bilateral filter, or a bi-selective filter) is applied to the forward transform
10 coefficients 42. In other implementations, a transform (e.g., JPEG or MPEG) artifact reduction process may be applied to the forward transform coefficients 42.

An inverse transform operation is applied to each of the sets of processed forward transform coefficients 44 to produce respective shifted, three-dimensional inverse transforms 48 (block 50). In particular, the inverse of the forward
15 transform operation that is applied during the forward transform process 40 is computed from the sets of processed forward transform coefficients 44 to generate the shifted inverse transforms 48.

As explained in detail below, the shifted inverse transforms 48 are combined to reduce noise and compression artifacts in the color planes of at least
20 a subset of video frames in the input video block 36 (block 52). In some implementations, the resulting color component video planes (e.g., Cr and Cb) are converted back to the original color space (e.g., the Red-Green-Blue color space) of the input video block 36. The video planes then are combined to produce the output video block 38.

25 FIG. 4 shows an embodiment of a system 58 for processing the input video block 36 to produce a compression-artifact-reduced output video sequence 60. Processing system 58 includes a forward transform module 66, a transform coefficient processor module 68, an inverse transform module 70, and an output video generator module 72. In general, the modules 66-72 of system 58 are not
30 limited to any particular hardware or software configuration, but rather they may be implemented in any computing or processing environment, including in digital electronic circuitry or in computer hardware, firmware, device driver, or software. For example, in some implementations, these modules 66-72 may be embedded in

-6-

the hardware of any one of a wide variety of digital and analog electronic devices, including desktop and workstation computers, digital still image cameras, digital video cameras, printers, scanners, and portable electronic devices (e.g., mobile phones, laptop and notebook computers, and personal digital assistants).

5 A. Forward Transform Module

The forward transform module 66 computes from the input video block 36 K sets ($C_1, C_2, \dots, C_K$) of shifted forward transforms, corresponding to K unique positions of a three-dimensional blocking grid relative to the input video block 36. The shifting of the blocking grid near the boundaries of the video data may be
 10 accommodated using any one of a variety of difference methods, including symmetric or anti-symmetric extension, row, column and temporal replication, and zero-shift replacement. In some implementations, an anti-symmetric extension is performed in each of the spatial and temporal dimensions. In one exemplary approach, the temporal dimension is divided into blocks and the video
 15 frame data is taken as the extension in the temporal dimension.

In one example, each three-dimensional block of the forward transform is computed based on a unitary frequency-domain transform D. Each block of the spatiotemporally-shifted forward transforms C_l ($l = 1, 2, \dots, K$) may be computed based on the separable application of the transform D in three dimensions as
 20 follows:

$$B = D X D^T \quad (4)$$

where X corresponds to the input video block 36, D^T corresponds to the transpose of transform D, and B corresponds to the transform coefficients of the input video block X.

25 In some implementations, D is a block-based linear transform, such as a discrete cosine transform (DCT). In one dimension, the DCT transform is given to four decimal places by the following 8 by 8 matrix:

-7-

$$\begin{aligned}
 & \begin{matrix} & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ D = & 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \end{matrix} \quad (5) \\
 5 \quad & \begin{matrix} & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{matrix} \\
 & \begin{matrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{matrix}
 \end{aligned}$$

10 In some implementations, the blocks of the spatiotemporally-shifted forward transforms ($C_1, C_2, \dots, C_K$) are computed based on a factorization of the transform D , as described in U.S. Patent No. 6,473,534, for example.

In some other implementations, D is a wavelet-based decomposition transform. In one of these implementations, for example, D may be a forward
 15 discrete wavelet transform (DWT) that decomposes a one-dimensional (1-D) sequence into two sequences (called sub-bands), each with half the number of samples. In this implementation, the 1-D sequence may be decomposed according to the following procedure: the 1-D sequence is separately low-pass and high-pass filtered by an analysis filter bank; and the filtered signals are
 20 downsampled by a factor of two to form the low-pass and high-pass sub-bands.

B. Transform Coefficient Processor Module

The transform coefficient processor module 68 processes the sets of forward transform coefficients 42 corresponding to the spatiotemporally-shifted forward transforms ($C_1, C_2, \dots, C_K$) that are computed by the forward transform
 25 module 66. In one exemplary implementation, the transform coefficient processor module 68 denoises the sets of forward transform coefficients 42 by nonlinearly transforming the forward transform coefficients ($C_1, C_2, \dots, C_K$) that are computed by the forward transform module 66.

In some implementations, the transform coefficient processor module
 30 denoises the sets of three-dimensional forward transform coefficients by applying

-8-

at least one of the following to the sets of forward transform coefficients: a soft threshold; a hard threshold; a bilateral filter; or a bi-selective filter

Referring to FIG. 5, in some implementations, the sets of forward transform coefficients are transformed in accordance with respective nonlinear thresholding transformations ($T_1, T_2, \dots, T_K$). In the illustrated implementation, the forward transform coefficients are nonlinearly transformed in accordance with a soft threshold by setting to zero each coefficient with an absolute value below a respective threshold (t_{ij} , where i, j refer to the indices of the quantization element, with i having values in the range of 0 to $M-1$ and j having values in the range of 0 to $N-1$) and leaving unchanged each coefficient with an absolute value equal to or above a respective threshold (t_{ij}). Quantization matrices 76 (or "Q Matrices") can be used to set the parameters t_{ij} for the nonlinear thresholding transformations ($T_1, T_2, \dots, T_K$). In some of these implementations, the quantization matrices contain the same quantization parameters q_{ij} that were originally used to compress video sequence 12. These quantization parameters may be stored in the compressed image 12 in accordance with a standard video compression scheme (e.g., MPEG). In some implementations, the threshold parameters are set in block 77 by a function M that maps the quantization parameters q_{ij} of the Q matrices to the corresponding threshold parameters. In other implementations, the thresholds are determined by the parameters used to describe the marginal distribution of the coefficients.

In some implementations, the parameters of the nonlinear thresholding transformations ($T_1, T_2, \dots, T_K$) are the same for the entire input video block 36. In other implementations, the parameters of the nonlinear thresholding transformations ($T_1, T_2, \dots, T_K$) may vary for different regions of the input video block 36. In some implementations, the threshold parameters vary according to video frame content (e.g., face region or textured region). In other implementations, threshold parameters vary based on transform component.

In some implementations, the transform coefficient processor module 68 processes the sets of three-dimensional forward transform coefficients 42 by applying a transform artifact reduction process to the sets of forward transform coefficients 42. In some exemplary implementations, the transform artifact

-9-

reduction process is applied instead of or in addition to (e.g., after) the process of denoising the sets of forward transform coefficients.

C. Inverse Transform Module

The inverse transform module 70 computes sets of inverse transforms (C_1^{-1} , C_2^{-1} , ..., C_K^{-1}) from the sets of processed forward transform coefficients 44. The inverse transform module 70 applies the inverse of the forward transform operation that is applied by forward transform module 66. The outputs of the inverse transform module 70 are intermediate video blocks ($V_1, V_2, \dots, V_K$) representing the video data in the spatial and temporal domains. The terms inverse transforms ($C_1^{-1}, C_2^{-1}, \dots, C_K^{-1}$) and intermediate video blocks ($V_1, V_2, \dots, V_K$) are used synonymously herein. The blocks of the spatiotemporally-shifted inverse transforms ($C_1^{-1}, C_2^{-1}, \dots, C_K^{-1}$) may be computed from equation (6):

$$C^{-1} = D^{-1} F (D^T)^{-1} \quad (6)$$

where F corresponds to output of the transform domain filter module 68, D is the forward transform, D^{-1} is the inverse transform, and D^T is the transpose of the transform D.

D. Output Image Generator Module

The output video generator module 72 combines the intermediate video blocks ($V_1, V_2, \dots, V_K$) to form the video planes of the output video sequence 60. In general, the output image generator module 72 computes the output video sequence 60 based on a function of some or all of the intermediate video blocks ($V_1, V_2, \dots, V_K$). For example, in some implementations, the video sequence 60 is computed from a weighted combination of the intermediate video blocks ($V_1, V_2, \dots, V_K$). In general, the weights may be constant for a given output video sequence 60 being constructed or they may vary for different regions of the given output video sequence 60. For example, in one of these implementations, the output video sequence 60 corresponds to a weighted average of the intermediate video blocks ($V_1, V_2, \dots, V_K$). In other implementations, the weights may be a function of the transform coefficient magnitude, or measures of video frame content (e.g., texture or detected faces). In some of these implementations, the weights of the intermediate video blocks (V_j) that correspond to blocks with too

-10-

many coefficients above a given threshold (which indicates edge or texture in the original video data) are set to zero, and only the intermediate video blocks that are obtained from blocks with more coefficients below the threshold are used to compute the output video sequence 60. In other of these implementations, the
5 output video sequence 60 corresponds to the median of the intermediate video blocks ($V_1, V_2, \dots, V_K$).

FIG. 6 shows an embodiment of the output video generator module 72 that includes a weighted combination generator module 80 that computes a base video block (V_{AVE}) from a combination of the intermediate video blocks ($V_1, V_2, \dots, V_K$).
10 The base video block corresponds to an estimate of the original uncompressed version of the input video block 36. In the illustrated embodiment, weighted combination generator module 80 computes a base video block (V_{AVE}) that has pixel values corresponding to averages of corresponding pixels in the intermediate video blocks ($V_1, V_2, \dots, V_K$).

15 Other embodiments are within the scope of the claims.

For example, although the above denoising and compression artifact reduction embodiments are described in connection with an input video block 36 that is compressed by a block-transform-based video compression method, these embodiments readily may be used to denoise and/or reduce artifacts in video
20 sequences compressed by other non-block-transform-based video compression techniques.

WHAT IS CLAIMED IS:

- 1 1. A method of processing a sequence of video frames, comprising:
2 computing a respective set of three-dimensional forward transform
3 coefficients (42) for each of multiple positions of a three-dimensional blocking
4 grid relative to an input video block (36) comprising a selected set of video
5 frames;
6 processing the sets of three-dimensional forward transform coefficients
7 (42);
8 computing a respective three-dimensional inverse transform (48) from each
9 set of processed forward transform coefficients (44); and
10 generating an output video block (38) based on the computed three-
11 dimensional inverse transforms (48).
- 1 2. The method of claim 1, wherein processing the sets of three-
2 dimensional forward transform coefficients (42) comprises denoising the sets of
3 forward transform coefficients (42) based on nonlinear mappings of input
4 coefficient values to output coefficient values.
- 1 3. The method of claim 2, wherein denoising comprises applying at
2 least one of the following to the sets of three-dimensional forward transform
3 coefficients (42): a soft threshold; a hard threshold; a bilateral filter; or a bi-
4 selective filter.
- 1 4. The method of claim 1, wherein processing the sets of forward
2 transform coefficients (42) comprises applying an artifact reduction process to the
3 sets of forward transform coefficients (42).
- 1 5. The method of claim 1, wherein generating the output video block
2 (38) comprises computing a weighted combination of the three-dimensional
3 inverse transforms (48).
- 1 6. The method of claim 5, wherein the output video block (38)
2 corresponds to a weighted average of the three-dimensional inverse transforms
3 (48).

-12-

1 7. The method of claim 5, wherein the weighted combination is
2 computed based on weights that vary as a function of transform coefficient
3 magnitude.

1 8. The method of claim 5, wherein the weighted combination is
2 computed based on weights that vary as a function of video frame content.

1 9. A machine for processing a sequence of video frames, comprising:
2 a forward transform module (66) configured to compute a respective set of
3 three-dimensional forward transform coefficients (42) for each of multiple
4 positions of a three-dimensional blocking grid relative to an input video block
5 (36) comprising a selected set of video frames;
6 a transform coefficient processor module (68) configured to process the
7 sets of three-dimensional forward transform coefficients (42);
8 an inverse transform module (70) configured to compute a respective
9 three-dimensional inverse transform (48) from each set of processed forward
10 transform coefficients (44); and
11 an output image generator module (72) configured to generate an output
12 video block (36) based on the computed three-dimensional inverse transforms
13 (48).

1 10. The machine of claim 9, wherein the transform coefficient processor
2 module (66) processes the sets of three-dimensional forward transform
3 coefficients (42) by denoising the sets of forward transform coefficients (42) based
4 on nonlinear mappings of input coefficient values to output coefficient values,
5 and the output image generator module (72) generates the output video block
6 (38) by computing a weighted combination of the three-dimensional inverse
7 transforms (48).

1/3

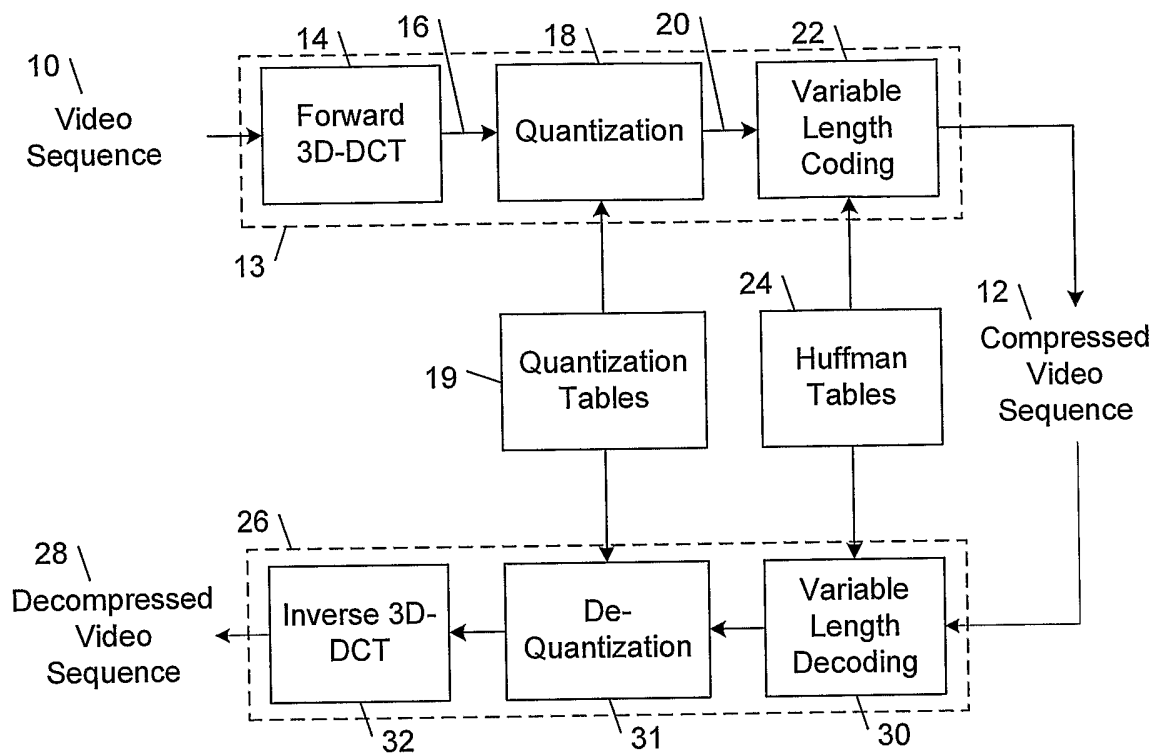


FIG. 1
(Prior Art)

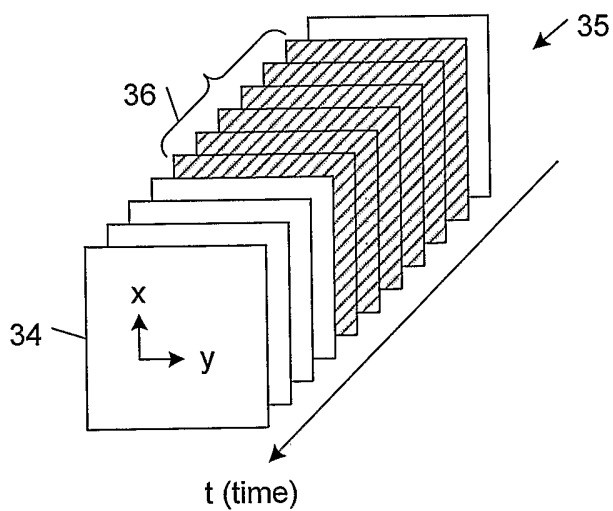


FIG. 2

2/3

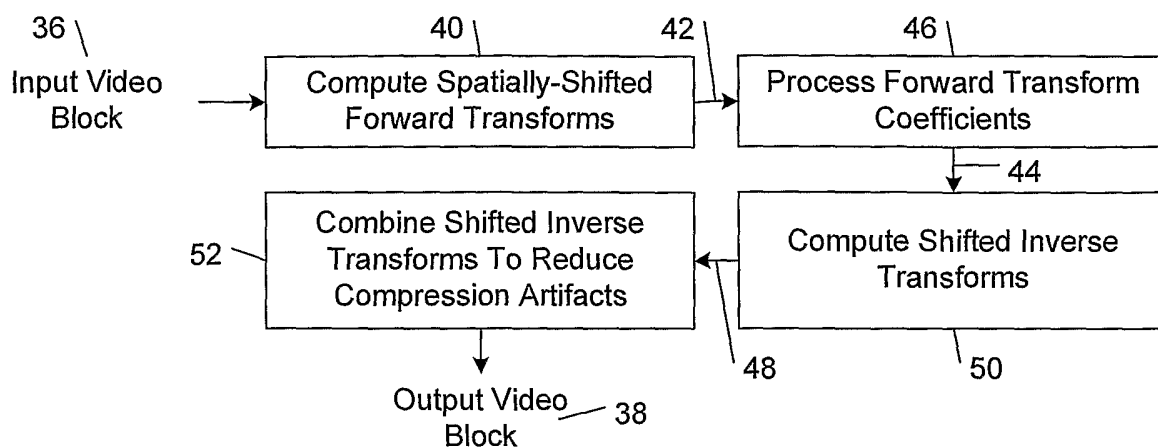


FIG. 3

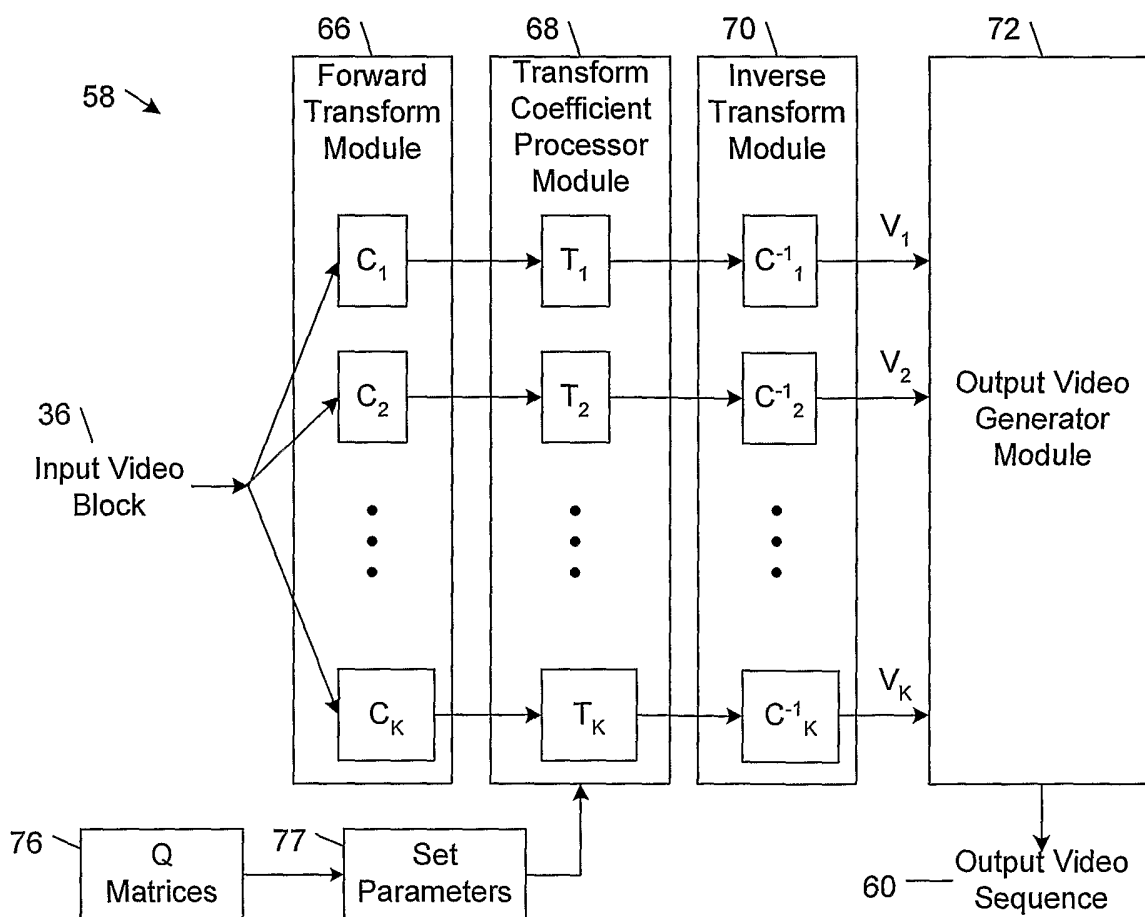


FIG. 4

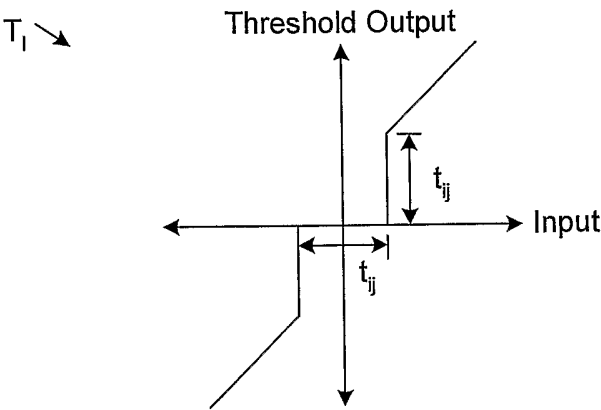


FIG. 5

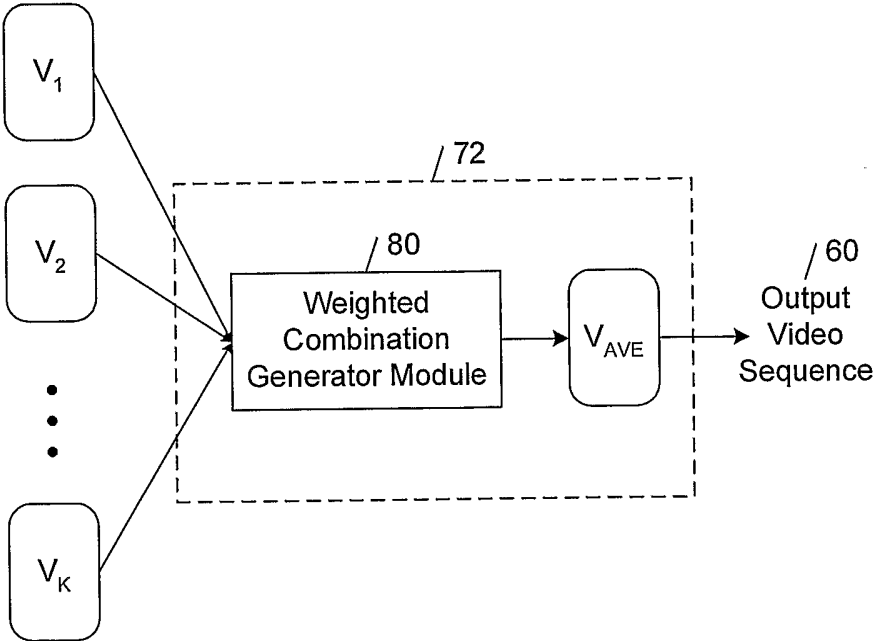


FIG. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/034164

A. CLASSIFICATION OF SUBJECT MATTER

H04N7/26

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)
H04N

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)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 616 465 A (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD) 21 September 1994 (1994-09-21)	1-3,9
Y	abstract; claim 1; figures 1,6 column 2, line 40 - column 3, line 4 column 3, line 37 - column 6, line 30	4-8,10
Y	NOSRATINIA A: "ENHANCEMENT OF JPEG-COMPRESSED IMAGES BY RE-APPLICATION OF JPEG" JOURNAL OF VLSI SIGNAL PROCESSING SYSTEMS FOR SIGNAL, IMAGE, AND VIDEO TECHNOLOGY, KLUWER ACADEMIC PUBLISHERS, DORDRECHT, NL, vol. 27, no. 1/2, February 2001 (2001-02), pages 69-79, XP001116260 ISSN: 0922-5773 abstract; figures 3,5 paragraphs '0003!, '0005!	4-8,10

☐ 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.
- *Z* document member of the same patent family

Date of the actual completion of the international search

26 January 2006

Date of mailing of the international search report

03/02/2006

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

Cakiroglu, S

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/034164

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
EP 0616465	A	21-09-1994	CA	2118978 A1	19-09-1994
			JP	6343170 A	13-12-1994
			US	5327242 A	05-07-1994