

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
2 April 2009 (02.04.2009)

PCT

(10) International Publication Number
WO 2009/042101 A2

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

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

(21) International Application Number:
PCT/US2008/010990

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

(22) International Filing Date:
22 September 2008 (22.09.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/862,125 26 September 2007 (26.09.2007) US

(71) Applicant (*for all designated States except US*):
HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **MAURER, Ron** [IL/IL]; HP Labs Israel, Ltd., Kiryat Technion Technion, 32000 Haifa (IL). **STAEIN, Carl** [IL/IL]; HP Labs Israel, Ltd., Kiryat Technion Technion, 32000 Haifa (IL).

Published:

— *without international search report and to be republished upon receipt of that report*

(54) Title: PROCESSING AN INPUT IMAGE TO REDUCE COMPRESSION-RELATED ARTIFACTS

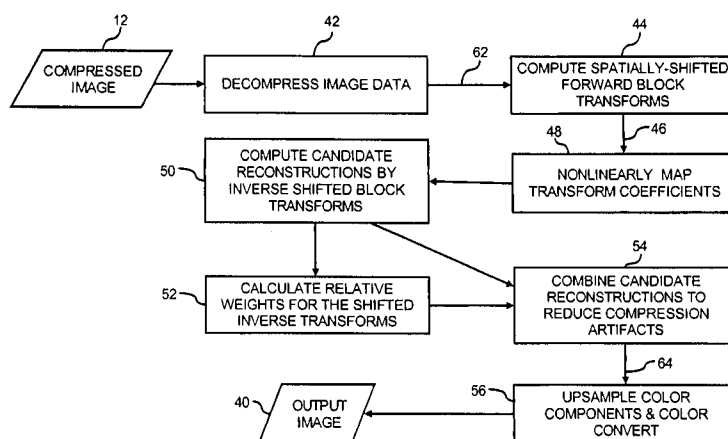


FIG. 3

(57) Abstract: An input image (62) is processed to reduce compression-related artifacts. The processing involves generating (44) block transforms of multiple versions of the input image, each version having a respective block-grid location that is shifted with respect to the block-grid locations of other versions. The transform coefficients are modified (46), and reverse transforms computed (50) to generate candidate reconstructions. Per-pixel relative weights are determined (52) for the candidate reconstructions where the relative weights for a pixel are related to its local context as set in each of the candidate reconstructions by the block in which that pixel is located. A weighted average of the candidate reconstructions is then computed (54) using the per-pixel relative weights to produce an artifact-reduced image.

PROCESSING AN INPUT IMAGE TO REDUCE COMPRESSION-RELATED ARTIFACTS

BACKGROUND

5 **[0001]** Digital images and video frames are typically compressed in order to reduce data storage and transmission requirements. In most image compression methods, certain image data is selectively discarded to reduce the amount of data needed to represent the image, while avoiding substantial degradation of the appearance of the image.

10 **[0002]** FIG. 1 shows a flow diagram of a conventional method of processing an original image 10 to produce a compressed image 12. As shown in FIG. 1, a forward transform 14 is applied to the original image 10 to produce a set of forward transform coefficients 16. The forward transform 14 may be any type of transform that is operable to convert the original image 10 into a set of coefficients in a
15 selected domain. A quantizer 18 is applied individually to the forward transform coefficients 16 to produce a set of quantized forward coefficients 20. The quantizer 18 discards some of the forward transform coefficient information, enabling the original image 10 to be compressed. An encoder 22 encodes the quantized forward transform coefficients using any type of lossless encoding
20 technique to produce the compressed image 12.

25 **[0003]** The original image 10 may be a binary image (for instance, a dark and bright dot pattern), a multilevel single-color image (for instance, a gray-level image), or a multilevel multi-color image. In general, the image compression process of FIG. 1 is applied individually to each color plane of the original image 10.

[0004] FIG. 2 shows a flow diagram of a conventional block-transform-based image method of compressing an original image 10. As shown in FIG. 2, the original image 10 is initially converted into a preselected luminance-based color space (for instance, the YCrCb color space), if the original image 10 is not already

specified in the preselected color space (block 24). Each color plane of the image in the preselected color space corresponds to a respective image (that is, an array of pixel values) that is processed individually as follows. The color components (for instance, the Cr and Cb color components) are downsampled (block 26). Each color plane is divided into blocks of pixels (for instance, 8x8 pixel blocks) (block 28). A block transform is applied to each pixel block individually (block 30). Any suitable type of block transform may be applied to the blocks. Exemplary types of block transforms include the cosine transform, Fourier transform, Hadamard transform, and Haar wavelet transform. The resulting transform coefficients are quantized (block 32). In addition, the quantized transform coefficients are encoded using a lossless coding technique to produce compressed image 12 (block 34).

[0005] To further illustrate the process depicted in FIG. 2, a particular example is provided in which the original image 10 is denoted as u and the compressed image 12 is denoted as f , which has been distorted during the compression process. The effect of the lossy part of a JPEG compression depends on the form of the block transform 30 used, which is characterized by a 64x64 2D-DCT transform matrix D and by an 8x8 quantization table Q . Both the transform matrix D and the quantization table Q are assumed to be known. In addition, the transform matrix D is typically the standard DCT-II transform, but it may be approximated in a form that may be implemented by fast integer-arithmetic as supported by the JPEG standard.

[0006] In this example, the following notations are used to mathematically describe the JPEG operation on image pixels of an image. The following notations are also used to describe the transform-domain filters employed later in the artifact reduction. The bold letters represent 2D indices, for instance, $\mathbf{n}=(n_x, n_y)$. JPEG divides the image having a size of (N_x, N_y) into 8x8 blocks. As such, each pixel-address $\mathbf{n}$ is split into a block index $\mathbf{b}$ and location within the block $\mathbf{m}$, such that $\mathbf{n} = 8\mathbf{b} + \mathbf{m}$. The DCT coefficient index will be usually be denoted as $\mathbf{k}=(k_x, k_y)$. The output of the standard JPEG decoder $f(\mathbf{b}, \mathbf{m})$ at block $\mathbf{b}$ and relative location $\mathbf{m}$ is expressed as:

$$\text{Equation (1)} \quad f(\mathbf{b}, \mathbf{m}) = \sum_{\mathbf{k}} D_{\mathbf{m}, \mathbf{k}}^{-1} Q_{\mathbf{k}} \left[\frac{1}{Q_{\mathbf{k}}} \sum_{\mathbf{j}} D_{\mathbf{k}, \mathbf{j}} u(\mathbf{b}, \mathbf{j}) \right] \quad \forall \mathbf{b}$$

where u is the un-compressed input image, the summation over $\mathbf{j}$ is run over all pixel positions within a block relative to the block corner, and the summation over $\mathbf{k}$ is run over all DCT coefficient indices, and the square brackets indicate that the values contained therein are to be rounded to the nearest integer.

[0007] The quantization of the DCT coefficients tends to generate mainly two types of artifacts, blocking and ringing artifacts. In blocks containing strong features (for instance, edges, lines, etc.), there is typically a large percentage of high frequency DCT components with substantial magnitudes. Some of these coefficients may fall under the relative quantization thresholds, thereby being zeroed out, while others survive. The unbalanced combination of post-quantization high-frequency components leaves patterns called “mosquito” or “ringing” artifacts, which resemble DCT-basis components. Ringing artifacts are more probable in pixels around a block center than in pixels near block-boundaries. Blocking artifacts generally originate from the fact that the quantization errors for pixels belonging to two neighboring blocks are not correlated. Blocking artifacts are known to originate mainly from quantization errors related to lower-frequency components, and to occur mainly near block boundaries.

[0008] Various image restoration algorithms have been proposed to overcome these image artifacts. One class of image restoration algorithms referred to as “overcomplete transform-domain filtering” is known to reduce blocking artifacts due to its ability to approximately restore the translation invariance of local-differences in the image statistics (the translation invariance was lost by the block-transform coding). The principle under these algorithms is to apply block-transform filters with block-grids aligned on different positions (shifts), and then to average the resulting outputs, which has been referred to as so-called “cycle-spinning” approach.

[0009] In a shifted block-grid with block size $M \times M$, every pixel address n can be expressed by the shift s to the first block boundary, the block index b and location within the block m in the following fashion: $n_s = Mb + m + s$. A block-transform filter is defined by the block-transform T employed and by a set of non-linear mappings Ψ_k that are each applied to their corresponding transform coefficient k . The mappings may correspond, for example, to soft or hard thresholding or quantization, and each of them is characterized by some scale parameter σ_k . Hence a block-transform filter with block boundaries at shift s is given by:

10 Equation (2): $\hat{u}_s(b, m) = \sum_k T_{m,k}^t \Psi_k(\sigma_k |\sum_j T_{k,j} f_s(b, j)|) \quad \forall b, s.$

The output of the cycle-spinning filter at each pixel is given by spatially averaging contributions from all shifts.

[0010] Unfortunately, the so-called "cycle-spinning" approach only partially reduces the ringing artifacts since in the shifts that contain a strong edge, the block-transform filter does not reduce ringing effectively. In addition, the surviving artifacts in these shifts are averaged into the final result, thus leaving residual ringing artifacts.

[0011] Guleryuz proposed a solution to this problem in the context of image denoising by introducing selectivity into the averaging process in order to reject "bad shifts". See, O.G. Guleryuz, "Weighted Overcomplete Denoising", "Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers," Nov. 9-12, 2003, Vol. 2, pp. 1992-1996, and also U.S. Patent Application Publication No. 2004/0240545(A1) to Guleryuz. More particularly, Guleryuz introduced per-pixel weighting of shifts, where the weights were determined by formulating the weighted combination as a linear estimation problem and solving it for the individual pixel-wise weight factors. Guleryuz's solution employs many statistical assumptions and approximations that are valid only for independent identically distributed (i.i.d.) noise. Guleryuz's solution, however, is not suitable for JPEG images since the statistics of the distortion

corresponding to JPEG compression artifacts due to transform coefficient quantization are both spatially varying and signal dependent, namely, non-i.i.d. An improved approach to eliminating compression artifacts in images, and more particularly, in JPEG images would therefore be beneficial. SUMMARY OF THE
5 INVENTION

[0012] According to one aspect of the present invention, there is provided a method, as set out in claim 1 hereinafter, for processing an input image to reduce compression-related artifacts. The present invention also provided a corresponding system as set out in claim 14.

10

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] Features of the present invention will become apparent to those skilled in the art from the following description with reference to the figures, in which:

15 **[0013]** FIG. 1 depicts a flow diagram of a conventional method of processing an original image to produce a compressed image;

[0014] FIG. 2 shows a flow diagram of a conventional block-transform-based method of compressing an original image;

20 **[0015]** FIG. 3 shows a flow diagram of a method for processing an input image to reduce compression-related artifacts, according to an embodiment of the invention;

[0016] FIG. 4 shows a system for processing the decompressed image data generated by the method of FIG. 3 to produce a compression-artifact-reduced output image, according to an embodiment of the invention;

25 **[0017]** FIGS. 5A and 5B, respectively, show graphs of nonlinear mappings used in an implementation of the nonlinear mapping module depicted in FIG. 4 to

map input transform coefficient values to output transform coefficient values, according to embodiments of the invention;

[0018] FIGS. 6A and 6B show respective block diagrams of a system for processing an input image to reduce compression-related artifacts, according to
5 embodiments of the invention;

[0019] FIG. 7 shows a diagrammatic view of a set of candidate reconstructions and a ringing corrected image generated from the set of candidate reconstructions, according to embodiments of the invention; and

[0020] FIG. 8 illustrates a computer system, which may be employed to
10 perform various functions of systems depicted in FIGS. 4 and 6, according to an embodiment of the invention.

DETAILED DESCRIPTION

[0021] For simplicity and illustrative purposes, the present invention is
15 described by referring mainly to an exemplary embodiment thereof. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent however, to one of ordinary skill in the art, that the present invention may be practiced without limitation to these specific details. In other instances, well known methods and
20 structures have not been described in detail so as not to unnecessarily obscure the present invention.

[0022] Disclosed herein are systems and methods designed to reduce compression artifacts that are inherently introduced by processes used to create lossily compressed images. In one respect, the systems and methods disclosed
25 herein are designed to reduce image compression artifacts in a computationally efficient manner without substantially degrading image quality. More particularly, for instance, the systems and methods disclosed herein are designed to substantially reduce ringing and blocking compression artifacts that are typically

introduced by block-transform-based image compression techniques, such as block discrete cosine transform (DCT) image compression techniques. It should, however, be understood that the artifact reduction techniques disclosed herein may readily be used to reduce artifacts in images compressed by other non-block-transform-based lossy image compression techniques.

[0023] The present disclosure includes many of the same figures and corresponding descriptions as originally presented in commonly assigned U.S. Patent Application Serial No. 10/696,140, filed by Ramin Samadani on October 29, 2003 (hereinafter the "'140 Application"). The '140 Application was published as U.S. Patent Application Publication No. 2005/0094893 on May 5, 2005. Some of the disclosure from the '140 Application is reproduced herein to provide a detailed understanding of how parts of the systems and methods disclosed herein operate. The systems and methods disclosed herein, however, differ from the disclosure contained in the '140 Application in many respects.

[0024] In the '140 Application, the pixel contexts in which local activity measures are calculated for each pixel are the same across all the intermediate images (called 'candidate reconstructions' below) independently of the block-grid shifts concerned. In particular, the activity measure contexts are centered around the pixel of interest and thus, the activity measures in all of the shifts refer to the same spatial context, that is, the same sets of pixels.

[0025] In contrast, in the presently disclosed systems and methods the local-activity context for a given pixel varies in each of a plurality of block-grid shifts. In particular, the activity context per each block-grid shift corresponds to a different context of pixels that includes the pixel of interest but is not centered around it, hence the activity measures in all of the block-grid shifts refer to different spatial contexts, that is, to different sets of pixels (see FIG. 7).

[0026] The activities are measured in a block-wise manner and are not centered around a pixel of interest and thus, the pixel of interest has a different context, for instance, a different $M \times M$ block, for each block-grid shift. As such, for a pixel located on a first side of an edge, some of the shifted contexts may include

pixels on a second side of the edge, which may lead to relatively large activity measures, while some other shifted contexts may include only pixels on the first side of the edge, which are more likely to have relatively similar intensities as the pixel of interest.

5 **[0027]** In these instances, the candidate reconstructions (intermediate images) that do not include the edge are preferred (or more heavily weighted, as discussed below) over those shifts that include the edge to thus reduce the possibility that the pixels on the second side of the edge cause the reconstructed edge to be blurred. As such, the block-wise activity measures implemented in the
10 systems and methods disclosed herein substantially eliminate the undesired coupling between ringing artifact reduction and edge-blurring.

[0028] In addition, in instances where a partial artifact reduction is needed, the systems and methods disclosed herein may use a part of the total number of possible block-grid shifts (64 shifts for 8x8 blocks) while still providing a relatively
15 high level of artifact reduction. In one respect, the average computational efficiency per batch of images may therefore be improved without a high level of sacrifice in artifact reduction. Accordingly, the block-wise activity measurements need not be determined separately for each of the pixels contained in a block.

[0029] With reference first to FIG. 3, there is shown a flow diagram of a
20 method for processing an image 12, to produce an output image 40 with reduced compression-related artifacts, according to an example embodiment. It should be apparent to those of ordinary skill in the art that the method depicted in FIG. 3 represents a generalized illustration and that other steps may be added or existing steps may be removed, modified or rearranged without departing from a scope of
25 the method.

[0030] According to Figure 3 method, the image 12 may comprise a compressed image 12 that has been generated by the block-transform-based image compression method of FIG. 2. In any regard, each color-component plane of the compressed image 12 is processed individually. The compressed image
30 data initially is decompressed (block 42). Spatially-shifted forward transforms are

computed from the decompressed image data 62 (block 44). More particularly, a forward transform operation is applied to each of multiple versions of the decompressed image data 62 having shifted block-grids to produce multiple respective sets of forward transform coefficients 46. For example, in an implementation in which the image 12 is compressed based on blocks of $M \times M$ pixels, the forward transform operation is applied to the decompressed image data 62 on a subset containing K shifts from the $M \times M$ independent shifts possible in an $M \times M$ transform to produce K sets of forward transform coefficients, where K and M have integer values of at least 1. According to an example, M has a value of 8, similar to the block size used in the JPEG compression standard.

[0031] The forward transform coefficients 46 of each set are nonlinearly mapped (block 48) to generate a plurality of respective intermediate transform domain filtered images having modified transform coefficients of the decompressed input image data 62. In one embodiment, an inverse block-transform operation (block 50) is then applied to the intermediate transform domain filtered images to produce respective candidate reconstructions. Relative per-pixel weights ($w_s(\mathbf{b}, \mathbf{m})$) for the candidate reconstructions that correspond to the respective confidence levels are calculated (block 52). As explained in detail below, the candidate reconstructions are combined by weighted averaging using the per-pixel relative weights ($w_s(\mathbf{b}, \mathbf{m})$) to reduce compression artifacts in each color plane 64 (block 54). The color component image planes (for instance, Cr and Cb) may be upsampled to the original resolution, if needed, and the resulting image planes are converted back to the color space (for instance, the Red-Green-Blue color space) of the original image 10 (block 56). The image planes are then combined to produce the output image 40.

[0032] The per-pixel relative weights ($w_s(\mathbf{b}, \mathbf{m})$) have been described with respect to FIG. 3 as being calculated based upon the plurality of candidate reconstructions generated from the inverse transforms of the intermediate transform domain filtered images of the decompressed input image data 62. It should, however, be understood that the per-pixel relative weights ($w_s(\mathbf{b}, \mathbf{m})$) may be calculated based upon the intermediate transform domain filtered versions of

the decompressed input image data 62 and that the inverse transforms of the intermediate transform domain filtered images of the decompressed input image data 62 may be computed following the calculation of the relative-per pixel weights without departing from a scope of the method depicted in FIG. 3.

5 **[0033]** Furthermore, for some embodiments where the per-pixel relative weights $w_s(b, m)$ are uniform across each block for a given block-grid shift, the weighting of the shifted candidate reconstructions may be applied in an interleaved fashion with the inverse transforms implemented in a separable fashion without changing the final output, thereby allowing computational savings as demonstrated
10 in co-pending and commonly assigned U.S. Patent Application Publication No. 2006/0153467, entitled "Enhancement of Digital Images," filed on January 11, 2005, by Carl Staelin, the disclosure of which is hereby incorporated by reference in its entirety. Hence, interleaving part of the computational steps of the inverse transform and the application of per-pixel relative weighting does not depart from
15 the scope of the invention.

[0034] Turning now to FIG. 4, there is shown an example system 60 for processing the decompressed image data 62 generated by the method of FIG. 2 to produce the compression-artifact-reduced output image 40. It should be understood that the system 60 may include additional components and that some
20 of the components described herein may be removed and/or modified without departing from a scope of the system 60.

[0035] As shown in FIG. 4, the processing system 60 includes a forward transform module 66, a nonlinear mapping module 68, an inverse transform module 70, and an output image generator module 72. In general, the modules
25 66-72 of system 60 are not 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 the hardware of any one of a wide variety of
30 electronic devices, including digital cameras, printers, and portable electronic

devices (for instance, mobile phones and personal digital assistants). In addition, although full (or complete) input images are processed in the illustrated examples, other examples may be configured to sequentially process a series of sub-image portions (for instance, swaths) of an input image.

5 **[0036]** The forward transform module 66 computes from the decoded image data 62 into K sets ($T_1, T_2, \dots, T_K$) of shifted forward block-transforms, corresponding to K unique positions of a blocking grid relative to the decompressed image 62. The forward block-transforms typically belong to the class of orthogonal transforms, more particularly, discrete cosine transform (DCT)
10 or an approximation of the DCT that may be implemented efficiently using integer arithmetic. Yet other transforms, however, may be used to trade-off filtering quality against computational efficiency.

[0037] The nonlinear mapping module 68 nonlinearly maps the output transform coefficients that are computed from the sets ($T_1, T_2, \dots, T_K$) of shifted
15 forward transforms that are computed by the forward transform module 66.

[0038] Referring to FIG. 5A, there is shown a graph 80 of hard-threshold output in an implementation of the nonlinear mapping module depicted in FIG. 4 plotted as a function of input transform coefficient values, according to a example. In some examples, the sets of forward transform coefficients are mapped in
20 accordance with respective nonlinear thresholding mappings ($\psi_1, \psi_2, \dots, \psi_K$). In particular, the forward transform coefficients are nonlinearly mapped 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
25 each coefficient with an absolute value equal to or above a respective threshold (t_{ij}). Quantization matrices 76 (or "Q Matrices") are used to set the parameters t_{ij} for the nonlinear hard threshold mappings ($\psi_1, \psi_2, \dots, \psi_K$). In some implementations, the quantization matrices contain the same quantization parameters q_{ij} that were originally used to compress the compressed image 12. In
30 other cases, they may be scaled by some gain factors p_{ij} (different for each DCT),

for instance, to accentuate or de-accentuate the filtering of certain frequencies. These quantization parameters q_{ij} may be stored in the compressed image 12 in accordance with a standard image compression scheme (for instance, JPEG).

[0039] In various instances, the nonlinear mapping module 68 also
 5 incorporates sharpness enhancements by modifying the nonlinearities, ψ_n . In particular, as shown in the graph 90 in Fig. 5B, the nonlinear mapping module 68 may replace the thresholding mapping between 0 and t_{ij} by a power-law relation $\psi_n(x) = \text{sign}(x) \times t_{ij} \times |x/t_{ij}|^{P_{ij}}$ where $P > 1$ (for P much larger than 1, the power-law relation approaches the Hard-threshold formula). In addition, the slope S_{ij} from t_{ij}
 10 to $t_{ij} + \Delta t_{ij}$ is made larger than 1 to induce sharpening by different amount to different transform coefficients. Then from $t_{ij} + \Delta t_{ij}$ onwards the slope remains in unity, so that there is a fixed additive gain of the output relative to input values above $t_{ij} + \Delta t_{ij}$. For instance, values of $2 \leq P_{ij} \leq 3$, and $1.2 \leq S_{ij} \leq 2$ provide adequate sharpness to compensate for the slight blurring that might be caused by the output image
 15 generator methods.

[0040] The "identity mapping" depicts a reference line. On the positive (negative) side of the axes, the area below (above) the reference line indicates a weakening of the coefficient magnitude that corresponds to local smoothing and the area above (below) the reference line indicates an increasing of the coefficient
 20 magnitude that corresponds to local sharpening.

[0041] The inverse transform module 70 computes sets $(T^{-1}_1, T^{-1}_2, \dots, T^{-1}_k)$ of inverse block transforms from the sets of nonlinearly mapped forward transform coefficients. The inverse transform module 70 applies the inverse of the forward block transform operation that is applied by forward transform module 66. The
 25 outputs of the inverse transform module 70 are candidate reconstructions $(\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k)$ representing differently filtered versions of the image data in the spatial domain.

[0043] The forward transform module 66, nonlinear mapping module 68 and inverse transform module 70 can be considered as together forming block

transform filters for the shifted versions of the input image, the filters outputting filtered blocks constituting the candidate reconstructions ($\hat{u}_i$).

[0044] The output image generator module 72 combines the candidate reconstructions ($\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$) to form the image planes 64 of the final output image 40. In general, the output image generator module 72 computes output image 40 based on a per-pixel weighted average of candidate reconstructions ($\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$). An example embodiment of the output image generator module 72 is shown in Figure 6A and comprises an arrangement 100 including a local weights generator 110 and a weighted average generator 120, both arranged to receive the candidate reconstructions 108 from the block transform filters. The arrangement 100 may include additional components and that some of the components described herein may be removed and/or modified. Various manners in which the arrangement 100 operates are described in greater detail below.

[0042] Generally speaking, each candidate reconstruction $\hat{u}_s(\mathbf{b}, \mathbf{m})$ obtained with block-grid shifted by $\mathbf{s}$ is attributed a pixel-wise weight $w_s(\mathbf{b}, \mathbf{m})$ that corresponds to the confidence of that candidate reconstruction $\hat{u}_s(\mathbf{b}, \mathbf{m})$ relative to other candidate reconstructions ($\hat{u}_{s_1}(\mathbf{b}, \mathbf{m}), \hat{u}_{s_2}(\mathbf{b}, \mathbf{m}), \dots, \hat{u}_{s_k}(\mathbf{b}, \mathbf{m})$) at different block-grid shifts s_1 - s_k . As discussed above, $\mathbf{b}$ is the block-index of the block in which a pixel of interest is located, $\mathbf{m}$ is the location of the pixel of interest in its block, such that, for blocks sized M by M , $\mathbf{n} = M\mathbf{b} + \mathbf{m} + \mathbf{s}$, where $\mathbf{n}$ is a pixel address, and $\mathbf{s}$ denotes the shift of the block-transform grid, for instance, characterized by vertical and horizontal shifts (such as, between 0 to 7 pixels for the preferred block size of $M=8$). The weighted average of the candidate reconstructions computed by the output image generator module 72 is effected by:

- multiplying values of the candidate reconstructions with their corresponding per-pixel relative weights to obtain respective products and accumulating the products to obtain a summed product per-pixel,
- accumulating the relative weights to obtain a normalization sum per-pixel,
- and

- dividing the summed product per-pixel by the normalization sum per-pixel.

The general expression for the artifact-reduced image $\hat{u}_{AR}(\mathbf{n})$ is:

$$\text{Equation (3): } \hat{u}_{AR}(\mathbf{n}) = \frac{\sum_s \sum_{b,m} \delta(\mathbf{n}, 8\mathbf{b} + \mathbf{m} + \mathbf{s}) \cdot w_s(\mathbf{b}, \mathbf{m}) \cdot \hat{u}_s(\mathbf{b}, \mathbf{m})}{\sum_s \sum_{b,m} \delta(\mathbf{n}, 8\mathbf{b} + \mathbf{m} + \mathbf{s}) \cdot w_s(\mathbf{b}, \mathbf{m})}$$

where $\delta(\mathbf{n}_1, \mathbf{n}_2)$ is 1 if $\mathbf{n}_1 = \mathbf{n}_2$ and 0 otherwise.

5

[0043] In order to suppress ringing artifacts, or other types of compression artifacts near edges, for instance, the local weights generator 110 assigns relatively greater weights $w_s(\mathbf{b}, \mathbf{m})$ to those candidate reconstructions $\hat{u}_s(\mathbf{b}, \mathbf{m})$ that are less likely to be corrupted by ringing artifacts or other types of distortions relative to the other candidate reconstructions $\hat{u}_s(\mathbf{b}, \mathbf{m})$. The ringing likelihood is estimated based on a local activity measure (A) computed from a neighborhood of each pixel in the candidate reconstruction $\hat{u}_s(\mathbf{b}, \mathbf{m})$, that is, after operation of the block-transform filter 106 (FIG. 6B). Various examples of local activity measures (A) are discussed herein below.

[0044] The weights generated by the local weights generator 110 and corresponding to each of the candidate reconstructions $(\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k)$ correspond to different block-grid shifts are inputted into a weighted average generator 120. The weighted average generator 120 is configured to receive both the candidate reconstructions $\hat{u}_s(\mathbf{b}, \mathbf{m})$ and their associated weights $w_s(\mathbf{b}, \mathbf{m})$ and to combine the candidate reconstructions based upon the values of these values (FIG. 3, block 54). More particularly, for instance, the weighted average generator 120 may calculate the artifact-reduced image $\hat{u}_{AR}(\mathbf{n})$ through implementation of Equation (3) discussed above. In addition, the weighted average generator 120 may combine the candidate reconstructions $(\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k)$ to reduce compression artifacts in a plurality of color planes 40.

25

[0045] Figure 6B shows an embodiment of the processing system 60 in which the arrangement 100 depicted in FIG. 6A is depicted in a greater level of detail. In addition, FIG. 6B illustrates an example of how an input image is processed on a per-pixel basis for a pixel of interest 302 in accordance with an embodiment of the invention.

[0046] The input to the system 60 of Figure 6B, is an input image here formed by decompressed image data resulting from decompression of an image which may have been encoded, for instance, through the compression technique discussed above with respect to FIG. 2. The system 100 depicts a local neighborhood 150 of the input image. As shown therein, one of the pixels contained in the local neighborhood 150 is labeled as the pixel of interest 302, which may be defined as the current pixel being processed for artifact-reduction. In addition, therefore, the remaining pixels contained in the input image 12 may be processed either concurrently or during future iterations.

[0047] In any regard, spatially-shifted forward transforms are computed from the decompressed image data 62, following which the forward transform coefficients are nonlinearly mapped, and inverse transforms computed thereby to generate candidate reconstructions 108.

[0048] More particularly, the decompressed image data is divided into multiple sets of a plurality of raw blocks (f_s) or neighborhoods 104, where the pixel of interest 302 has different contexts in each of the block-grid shifted neighborhoods 104. The boundaries of the plurality of raw blocks (f_s) 104 contained in one local neighborhood 104 differs or is block-grid shifted with respect to the plurality of raw blocks (f_s) 104 contained in the remaining local neighborhoods. As such, the local neighborhoods 104 represent different partitions of the raw image into non-overlapping blocks. By way of example, each of the local neighborhoods 104 may contain a plurality of non-overlapping 8x8 pixel blocks and the blocks may be shifted by one or more pixels in each of the different local neighborhoods 104.

[0049] The raw block sets of each of the shifted local neighborhoods 104 are inputted into a block-transform filter 106, where forward transforms are computed, transform coefficients mapped, and inverse transforms computed, for instance, as described with respect to the modules 66-70 in FIG. 4 above. The output of the block-transform filters 106 are candidate reconstructions ($\hat{u}_s$) 108 of each of the shifted local neighborhoods 104 containing the pixel of interest 302 to be filtered.

[0050] As further shown in FIG. 6B, each of the candidate reconstructions ($\hat{u}_s$) 108 are inputted into the local weights generator 110. The local weights generator 110 is configured to calculate per-pixel relative weights 112 of the candidate reconstructions ($\hat{u}_s$) 108 (FIG. 3, block 52). More particularly, the system 100 obtains multiple candidate reconstructions ($\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$) 108 in which the block-grid location in which a pixel of interest 302 is located differs in each of the candidate reconstructions ($\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$) 108 (as shown in the block diagram 300 in FIG. 7). As such, the pixels contained in the local neighborhood of the pixel of interest 302 for each of the candidate reconstructions ($\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$) 108 differs in each of the candidate reconstructions ($\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$) 108.

[0051] In calculating the per-pixel relative weights 112 of the candidate reconstructions ($\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$) 108 at block 52, the local weights generator 110 determines activity measures (A_s) for each of the blocks in the different candidate reconstructions ($\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$) 108 to obtain relative self-confidences (C_s) of each of the candidate reconstructions ($\hat{u}_s$) or the value of reconstruction at the pixel of interest 302. As such, for each of the candidate reconstructions ($\hat{u}_s$), each of the block-wise activity measures (A_s) has a different context (that is, pixels contained in the 8x8 blocks) from which the activity measure is determined. Consequently, for instance, the activity measure (A_{s1}) of a block 304a in one candidate reconstruction ($\hat{u}_{s1}$) may have a different value as compared with the activity measure (A_{s2}) of the block 304b in a second candidate reconstructions ($\hat{u}_{s2}$) because the activity measures are determined based upon differing sets of pixel values. Each of the

block-wise activity measures (A_s) and therefore the derived relative self-confidences (C_s) have a different context and the per-pixel relative weights (w_s) are derived from the relative self-confidences (C_s).

[0052] The relative preference between two possible candidate reconstructions $\hat{u}_s(\mathbf{b}, \mathbf{m})$ at block-grid shifts $s1$ and $s2$ are defined as the ratio between the respective activity measures (A) raised to some negative power $-p$, for instance, $p \sim 0.5-1.0$. This inverse power law is regularized by a small positive constant ε to avoid divergence or instability when the activity measure is small, for instance, in flat blocks. The relative preference $R_{s1, s2}(\mathbf{b}, \mathbf{m})$ may thus be defined according to the following equation:

$$\text{Equation (4): } R_{s1, s2}(\mathbf{b}, \mathbf{m}) = \left(\frac{\varepsilon + A[\hat{u}_{s1}(\mathbf{b}, \mathbf{m})]}{\varepsilon + A[\hat{u}_{s2}(\mathbf{b}, \mathbf{m})]} \right)^{-p}.$$

[0053] Because Equation (4) is monotonically decreasing, the candidate reconstruction $\hat{u}_s(\mathbf{b}, \mathbf{m})$ with the higher activity measure (A) has a lower relative preference. In addition, a property of Equation (4) is that it is homogeneous, in that, it depends only on ratios between a function of each activity measure (A). As such, the relative preference may be expressed as the ratio of two self-confidence attributes C_s that are computed for each block-grid shift separately.

$$\text{Equation (5): } R_{s1, s2}(\mathbf{b}, \mathbf{m}) = \frac{C_{s1}(\mathbf{b}, \mathbf{m})}{C_{s2}(\mathbf{b}, \mathbf{m})} \text{ where } C_s(\mathbf{b}, \mathbf{m}) \equiv (\varepsilon + A[\hat{u}_s(\mathbf{b}, \mathbf{m})])^{-p}.$$

[0054] Consequently, by setting the per-pixel relative weight to be the self confidence expression $w_s(\mathbf{b}, \mathbf{m}) = C_s(\mathbf{b}, \mathbf{m})$, the expression in the numerator and the denominator in Equation (5) are sums of quantities that may be computed a shift at a time, without the need to cache intermediate results or access them for later computations. Accumulating the products of the per-pixel relative weights (w_s) with the plurality of candidate reconstructions ($\hat{u}_s$) and accumulating a sum of per-pixel weights (w_s) can therefore be effected in a single pass without having to store either candidate reconstructions or intermediate per-pixel weights. This property

enables an efficient application of embodiments of the invention in hardware systems that have limited cache memory, such as ASIC, or excessive memory access latency such as GPUs (Graphical Processing Units).

[0055] Generally speaking, the local weights generator 110 selects an activity measure (A) that provides relatively good de-ringing, while retaining relatively good de-blocking capabilities, and preserving genuine edges. In addition, the activity measure (A) is computed based on some quantity that is related to the magnitude of differences between the pixels contained in a local neighborhood, for instance, the pixels contained in an 8x8 block of pixels. The activity measure (A) may comprise, for instance, the variance of the collection of pixels in the local neighborhood, the difference between the minimum and maximum values, the mean of absolute differences between each pair of pixels, etc.

[0056] Thus, in one regard, a block may be considered to be more active (have a higher activity measure) if there is a relatively large variability among the pixels contained in that block, as may occur when the block contains an edge, a strong texture, etc. More particularly, if there is a relatively strong edge anywhere in a block, it induces large amplitudes of high frequency components which overlap part of the high-frequencies related to ringing artifacts in that block. Therefore, the mere presence of a relatively strong edge within a block severely limits the ability of the block-transform filter 106 to reduce the ringing artifacts all across that block, since it does not suppress strong components in order to preserve edges.

[0057] However, because the system 100 measures the block-wise activity after the operation of the block transform filter 106, a relatively high activity level indicates the presence of an edge, which indirectly indicates that if there were artifacts anywhere in that block, the block-transform filter 106 most likely did not properly clean those artifacts. As such, the self-confidence measure (C) that is derived from the activity measure (A) computed over the entire block may be applied to each pixel included in that block, since the plausibility of remaining artifacts is not local but across the block. That is, the computed self-confidence measure (C) is distributed to each of the pixels included in that block.

[0058] For each block, the block-activity measure A is computed from the same pixel-set as the filtered reconstruction for that block $A[\hat{u}_s(\mathbf{b}, \mathbf{m})]$. As such, the activity measure A may be defined in the transform domain and frequency weighting may relatively easily be applied to enable the activity measure to be tuned to better respond to artifacts or edges. By way of example, the block-variance corresponds to the sum of squares of all AC coefficients in the block (assuming a unitary transform like DCT). The block-variance measure may be generalized by frequency weighting matrix α_k where the k^{th} entry corresponds to the k^{th} 2D-transform coefficient as shown in the following equation:

Equation (6):
$$A_s(\mathbf{b}) = \sum_{k \neq 0} \alpha_k \left[\sum_m T_{k,m} \hat{u}_s(\mathbf{b}, \mathbf{m}) \right]^2 = \sum_{k \neq 0} \alpha_k \cdot \left[\psi_k \left(Q_k \left| \sum_j T_{k,j} f_s(\mathbf{b}, j) \right| \right) \right]^2$$

 $\forall \mathbf{b}, \mathbf{s}.$

[0059] In Equation (6), $A_s(\mathbf{b})$ denotes a block-activity for block $\mathbf{b}$ at block-grid shift $\mathbf{s}$, T denotes the block transform employed, and ψ_k denotes the set of non-linear mappings that are each applied to their corresponding transform coefficient k , with respective characteristics scale Q_k taken from the JPEG quantization matrix.

[0060] According to an example, the local weights generator 110 selects a frequency weighting matrix α that is more responsive to edges and lines than to textures, since ringing artifacts are much more likely and much more visually disturbing near edges than on textures, and it is relatively difficult to differentiate between ringing artifacts and textures. For instance, the local weights generator 110 may select the frequency weighting matrix to be $\alpha_k \sim [(1 + k_x)(1 + k_y)]^{-1}$, which generally accentuates the contribution of edge features to the activity measure relative to less ordered features.

[0061] The weights $(w_1, w_2, \dots, w_n)$ 112 generated by the local weights generator 110 and corresponding to each of the candidate reconstructions $(\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k)$ 108 are inputted into the weighted average generator 120. The weighted average generator 120 is configured to determine the output pixel 140 value from

the per-pixel relative weights ($w_1, w_2, \dots, w_n$) 112 and the candidate reconstructions ($\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k$) 108. The output pixel 140 may then be written into an appropriate output buffer and the data contained in the buffer may be used to reproduce an artifact-reduced version of the input image 62.

- 5 **[0062]** According to another example, the local weights generator 110 may employ a modified weighting scheme that enhances control over feature preservation as compared with the relative weight $w_s(\mathbf{b}, \mathbf{m})$ based upon block-wise self-confidence $C_s(\mathbf{b}, \mathbf{m})$. As discussed above, each candidate reconstruction $\hat{u}_s(\mathbf{b}, \mathbf{m})$ is weighted according to a block-wise activity $w[A_s(\mathbf{b})]$. However, for pixels
- 10 at or near features, such as, edges and lines, the candidate reconstructions from different block-grid shifts ($\hat{u}_{s1}(\mathbf{b}, \mathbf{m}), \hat{u}_{s2}(\mathbf{b}, \mathbf{m}), \dots, \hat{u}_{sk}(\mathbf{b}, \mathbf{m})$) may have very different values, even if their corresponding activity measures are similar. In order to better preserve features in these situations, those candidate reconstructions ($\hat{u}_{s1}(\mathbf{b}, \mathbf{m}), \hat{u}_{s2}(\mathbf{b}, \mathbf{m}), \dots, \hat{u}_{sk}(\mathbf{b}, \mathbf{m})$) that are closer in value to the corresponding pixel value in
- 15 the unfiltered JPEG image are given preference. This type of weighting mechanism is generally known as "photometric weighting." One solution to enhancing control over feature preservation is to multiply the block-activity based weight by a photometric weight, which is a decreasing function $g(\cdot)$ of the difference between the reconstruction $\hat{u}_s(\mathbf{b}, \mathbf{m})$ and the original pixel value in that
- 20 location ($\mathbf{n}$), $g_{\sigma(n)}(|\hat{u}_s(\mathbf{n}) - f(\mathbf{n})|)$. By way of example, the system 100 may employ the following photometric function:

$$\text{Equation (7): } g_{\sigma}(t) = \frac{1}{1 + \left(\frac{t}{\sigma}\right)^2}.$$

- [0063]** The selectivity of the photometric function g depends on a local feature-contrast scale $\sigma(n)$ that characterizes what differences are considered to
- 25 correspond to a "feature". For instance, $g_{\sigma}(0)=1$, $g_{\sigma}(\sigma)\sim 0.5$, and $g_{\sigma}(t \gg \sigma) \ll 1$. Note that if a photometric weighting is applied with a uniform scale parameter across the image, not only are the edges preserved, but the artifacts may also be over-

protected from being filtered out, since local differences in regions with ringing artifacts may be as high as a genuine image features. The local weights generator 110 may avoid this situation by letting the feature contrast scale parameter depend locally on a feature detector that is insensitive to spatially ordered edge/line features, but is sensitive to non-ordered features such as ringing artifacts, for example a high-order mixed derivative operator (D_{xyy}). For pixels which have a relatively large response to that detector are identified as "artifact pixels" and the local weights generator 110 takes the feature-contrast scale to be high, which reduces the sensitivity of the photometric function g and effectively disables the feature-preservation mechanism so that the artifact may be filtered out. For pixels that have relatively low responses to that detector are identified as "non-artifact pixels" and the local weights generator 110 actively prefers reconstructions that are closer to the original unfiltered value because the photometric function becomes more sensitive and performs its feature preserving role, for instance.

Equation (8): $\sigma(n) = \max(\sigma_0, |D_{xyy} * f|)$; $D_{xyy} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$,

where the "*" denotes a linear convolution, D_{xyy} is the convolution mask corresponding to a feature detector that is sensitive to non-ordered features but not to ordered features like straight edges and lines, and where σ_0 is a positive regularizing parameter to avoid zero division.

[0064] The modified weighting scheme $w_s(b, m)$ may thus be written as:

Equation (9): $w_s(b, m) = C_s(b) \cdot g_{\sigma(n)}(|\hat{u}_s(n) - f_s(n)|)$.

[0065] In any regard, the weighted average generator 120 may also upsample the color component image planes (for instance, Cr and Cb) to the original resolution and may convert the resulting image planes back to the color space (for instance, the Red-Green-Blue color space) of the original image (FIG. 3, block 56). The weighted average generator 120 may then combine the image planes to produce the output image 40.

[0069] As previously indicated the per-pixel relative weights can be calculated from either the intermediate transform domain filtered images or the respective candidate reconstructions. The various ways described above for calculating the per-pixel relative weights are applicable both when calculating these weights from the intermediate transform domain filtered images and from the respective candidate reconstructions except that equation (6) is only applicable when calculating the per-pixel relative weights from the intermediate transform domain filtered images.

[0070] The operations set forth in the methods disclosed herein may be contained as one or more utilities, programs, or subprograms, in any desired computer accessible or readable medium. In addition, the methods disclosed herein may be embodied by a computer program, which can exist in a variety of forms both active and inactive. For example, it can exist as software program(s) comprised of program instructions in source code, object code, executable code or other formats. Any of the above can be embodied on a computer readable medium, which include storage devices and signals, in compressed or uncompressed form.

[0071] Exemplary computer readable storage devices include conventional computer system RAM, ROM, EPROM, EEPROM, and magnetic or optical disks or tapes. Exemplary computer readable signals, whether modulated using a carrier or not, are signals that a computer system hosting or running the computer program can be configured to access, including signals downloaded through the Internet or other networks. Concrete examples of the foregoing include distribution of the programs on a CD ROM or via Internet download. In a sense, the Internet itself, as an abstract entity, is a computer readable medium. The same is true of computer networks in general. It is therefore to be understood that any electronic device capable of executing the above-described functions may perform those functions enumerated above.

[0072] FIG. 8 illustrates a computer system 800, which may be employed to perform the various functions of the systems 60 and 100 described herein above with, according to an example. In this respect, the computer system 800 may be used as a platform for executing one or more of the functions described hereinabove with respect to the systems 60 and 100.

[0073] The computer system 800 includes a processor 802, which may be used to execute some or all of the steps described in the methods disclosed herein. Commands and data from the processor 802 are communicated over a communication bus 804. The computer system 800 also includes a main memory 806, such as a random access memory (RAM), where the program code may be executed during runtime, and a secondary memory 808. The secondary memory 808 includes, for example, one or more hard disk drives 810 and/or a removable storage drive 812, representing a floppy diskette drive, a magnetic tape drive, a compact disk drive, etc., where a copy of the program code for processing a compressed image may be stored.

[0074] The removable storage drive 810 reads from and/or writes to a removable storage unit 814 in a well-known manner. User input and output devices may include a keyboard 816, a mouse 818, and a display 820. A display adaptor 822 may interface with the communication bus 804 and the display 820 and may receive display data from the processor 802 and convert the display data into display commands for the display 820. In addition, the processor 802 may communicate over a network, for instance, the Internet, LAN, etc., through a network adaptor 824.

[0075] It will be apparent to one of ordinary skill in the art that other known electronic components may be added or substituted in the computer system 800. In addition, the computer system 800 may include a system board or blade used in a rack in a data center, a conventional "white box" server or computing device, etc. Also, one or more of the components in FIG. 8 may be optional (for instance, user input devices, secondary memory, etc.).

[0076] What has been described and illustrated herein is a preferred embodiment of the invention along with some of its variations. The terms, descriptions and figures used herein are set forth by way of illustration only and are not meant as limitations. Those skilled in the art will recognize that many variations are possible within the scope of the invention, which is intended to be defined by the following claims -- and their equivalents -- in which all terms are meant in their broadest reasonable sense unless otherwise indicated.

CLAIMS

1. A method for processing an input image (62) to reduce compression-related artifacts, said method comprising:

5 generating (44) block transforms (T) of multiple versions (f_s) of the input image, each version having a respective block-grid location that is shifted with respect to the block-grid locations of other versions;

generating (48) from the block transforms (T) a plurality of respective intermediate transform domain filtered images having modified transform
10 coefficients;

computing (50) inverse block transforms (T^{-1}) of the intermediate transform domain filtered images to generate a plurality of respective candidate reconstructions ($\hat{u}_s$);

determining (52) per-pixel relative weights (w_s) for the candidate
15 reconstructions ($\hat{u}_s$) wherein the relative weights for a pixel are related to its local context in each of the candidate reconstructions as set by the block in which that pixel is located;

generating an artifact-reduced image ($\hat{u}_{AR}$) by computing (54) a weighted average of the candidate reconstructions ($\hat{u}_s$) using the per-pixel relative weights

20 (w_s); and

outputting the artifact-reduced image ($\hat{u}_{AR}$).

2. The method according to claim 1, wherein determining (52) the per-pixel relative weights (w_s) comprises calculating the per-pixel relative weights from either
25 the intermediate transform domain filtered images or the respective candidate reconstructions ($\hat{u}_s$).

3. The method according to claim 2, wherein computing (54) the weighted average of the candidate reconstructions using the per-pixel relative
30 weights (w_s) comprises:

multiplying values of the candidate reconstructions ($\hat{u}_s$) with their

corresponding per-pixel relative weights (w_s) to obtain respective products and accumulating the products to obtain a summed product per-pixel,

accumulating the relative weights (w_s) to obtain a normalization sum per-pixel, and

dividing the summed product per-pixel by the normalization sum per-pixel.

4. The method according to claim 3, wherein accumulating the products of the per-pixel relative weights (w_s) with the plurality of candidate reconstructions ($\hat{u}_s$) and accumulating a sum of per-pixel weights (w_s) for producing the artifact-reduced image ($\hat{u}_{AR}$) is effected in a single pass without having to store either candidate reconstructions or intermediate per-pixel weights, thereby improving the computational efficiency in processing the input image and increasing a range of hardware platforms that accommodate the processing.

5. The method according to claim 3, wherein computing (54) the weighted average of the candidate reconstructions $\hat{u}_s(\mathbf{b}, \mathbf{m})$ using per-pixel relative weights $w_s(\mathbf{b}, \mathbf{m})$ is effected according to the following equation:

$$\hat{u}_{AR}(\mathbf{n}) = \frac{\sum_s \sum_{\mathbf{b}, \mathbf{m}} \delta(\mathbf{n}, M\mathbf{b} + \mathbf{m} + \mathbf{s}) \cdot w_s(\mathbf{b}, \mathbf{m}) \cdot \hat{u}_s(\mathbf{b}, \mathbf{m})}{\sum_s \sum_{\mathbf{b}, \mathbf{m}} \delta(\mathbf{n}, M\mathbf{b} + \mathbf{m} + \mathbf{s}) \cdot w_s(\mathbf{b}, \mathbf{m})}$$

where:

$\mathbf{n}$ is a pixel address of a pixel of interest,

$\mathbf{b}$ is the block index of the block in which the pixel of interest is located,

$\mathbf{m}$ is the location of the pixel of interest in its block,

the blocks are sized M by M ,

$\mathbf{s}$ denotes the shift of the block-grid, and

$\delta(\mathbf{n}_1, \mathbf{n}_2)$ is 1 if $\mathbf{n}_1 = \mathbf{n}_2$ and 0 otherwise.

6. The method according to claim 2, wherein determining (52) the per-pixel relative weights (w_s) for the candidate reconstructions ($\hat{u}_s$) comprises

determining activity measures of each of the blocks of either the intermediate transform domain filtered images or the candidate reconstructions to obtain relative self-confidences of the candidate reconstructions ($\hat{u}_s$) with respect to a pixel of interest.

5

7. The method according to claim 6, wherein determining the activity measure per each block of either the intermediate transform domain filtered images or the candidate reconstructions ($\hat{u}_s$) further comprises computing the activity measures based upon a quantity that is related to a magnitude of differences between the pixels contained in the respective blocks of the shifted block grid corresponding to that intermediate transform domain filtered image or candidate reconstruction ($\hat{u}_s$).

10

8. The method according to claim 7, wherein the activity measure is block variance computed either directly from the pixel values or from transform coefficient values.

15

9. The method according to claim 8, wherein determining the activity measures further comprises determining the activity measures $A_s(\mathbf{b})$ for each of the blocks $\mathbf{b}$ in the plurality of candidate reconstructions $\hat{u}_s(\mathbf{b}, \mathbf{m})$, which are frequency-weighted variances, for a block-grid shift $\mathbf{s}$ based upon the following equation:

20

$$A_s(\mathbf{b}) = \sum_{k \neq 0} \alpha \left[\sum_m T_{k,m} \hat{u}_s(\mathbf{b}, \mathbf{m}) \right]^p = \sum_{k \neq 0} \alpha_k \cdot \left[\psi_k \left(\left| \sum_j T_{k,j} f_s(\mathbf{b}, j) \right| \right) \right]^p \quad \forall \mathbf{b}, \mathbf{s},$$

where:

25

α is a frequency weighting matrix,

T denotes the block transform employed, and

ψ_k denotes the set of non-linear mappings that are each applied to their corresponding transform coefficient k .

30

10. The method according to claim 6, wherein determining (52) per-pixel

relative weights for the plurality of candidate reconstructions further comprises assigning relatively lower self-confidences to the pixels in the intermediate transform domain filtered images or the candidate reconstructions ($\hat{u}_s$) having relatively higher activity measures and assigning relatively higher self-confidences to the pixels in the intermediate transform domain filtered images or the candidate reconstructions having relatively lower activity measures.

11. The method according to claim 10, wherein determining (52) per-pixel relative weights (w_s) further comprises determining a relative preference $R_{s1,s2}(\mathbf{b}, \mathbf{m})$ between two intermediate transform domain filtered images or two candidate reconstructions ($\hat{u}_s$) corresponding to two shifts of block-grids (s1) and (s2) based upon the following equation:

$$R_{s1,s2}(\mathbf{b}, \mathbf{m}) = \frac{C_{s1}(\mathbf{b}, \mathbf{m})}{C_{s2}(\mathbf{b}, \mathbf{m})} \text{ where } C_s(\mathbf{b}, \mathbf{m}) \equiv (\varepsilon + A[\hat{u}_s(\mathbf{b}, \mathbf{m})])^{-p}$$

where:

$\mathbf{b}$ is a block-index of the block in which the pixel of interest is located,
 $\mathbf{m}$ is the location of the pixel of interest in its block,
 $C_s(\mathbf{b}, \mathbf{m})$ is a per-pixel self-confidence attribute,
 ε is a small positive constant, and
 $A[\hat{u}_s(\mathbf{b}, \mathbf{m})]$ is a per-pixel activity measure determined from either an intermediate transform domain filtered image or a candidate reconstruction ($\hat{u}_s$).

12. The method according to claim 1, wherein determining (52) per-pixel relative weights $w_s(\mathbf{b}, \mathbf{m})$ for the plurality of candidate reconstructions $\hat{u}_s(\mathbf{b}, \mathbf{m})$ is effected in a manner designed to protect desired features from filtering while allowing artifacts to be filtered according to the following equation:

$$w_s(\mathbf{b}, \mathbf{m}) = C_s(\mathbf{b}) \cdot g_{\sigma(n)}(|\hat{u}_s(n) - f_s(n)|)$$

where:

$C_s(\mathbf{b})$ is a self-confidence attribute measure for block $\mathbf{b}$ and a shift $\mathbf{s}$,

$g_{\sigma}(t)$ is a decreasing function of the difference between $\hat{u}_s(\mathbf{b}, \mathbf{m})$ and the original pixel value in that location $f_x(\mathbf{n})$ divided by a positive local feature contrast scale $\sigma(\mathbf{n})$, such that $g_{\sigma}(0)=1$, $g_{\sigma}(\sigma)\sim 0.5$, $g_{\sigma}(t \gg \sigma) \ll 1$, and the local feature contrast scale $\sigma(\mathbf{n})$ is computed per pixel by a feature

5 detector that is insensitive to the features desired for protection, and sensitive to artifacts desired to be filtered.

13. The method according to claim 12, wherein the local feature contrast scale is computed according to the following equation:

$$10 \quad \sigma(\mathbf{n}) = \max(\sigma_0, |D_{xxyy} * f|); \quad D_{xxyy} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

where:

"*" denotes a linear convolution,

D_{xxyy} is the convolution mask corresponding to a feature detector that is sensitive to non-ordered features but not to ordered features like straight

15 edges and lines, and

σ_0 is a positive regularizing parameter to avoid zero division.

14. A system (60; 100) for processing an input image (62) to reduce compression-related artifacts, said system comprising:

20 a forward transform module (66) configured to generate block transforms (T) of multiple versions (f_s) of the input image, each version having a respective block-grid location that is shifted with respect to the block-grid locations of other versions;

a nonlinear mapping module (68) configured to generate from the block transforms a plurality of respective intermediate transform domain filtered images

25 having modified transform coefficients;

an inverse transform module (70) configured to compute inverse transforms (T^{-1}) of the intermediate transform domain filtered images to generate a plurality of respective candidate reconstructions ($\hat{u}_s$);

a local weights generator (110) configured to determine per-pixel relative

weights (w_s) for the candidate reconstructions ($\hat{u}_s$), wherein the relative weights for a pixel are related to its local context in each of the candidate reconstructions ($\hat{u}_s$) as set by the block in which that pixel is located;

a weighted average generator (120) configured to compute a weighted average of the candidate reconstructions ($\hat{u}_s$) using the per-pixel relative weights (w_s) to produce an artifact-reduced image ($\hat{u}_{AR}$); and

an output for outputting the artifact-reduced image ($\hat{u}_{AR}$).

15. The system according to claim 14, wherein the weighted average generator (110) is further configured to combine the plurality of candidate reconstructions $\hat{u}_s(\mathbf{b}, \mathbf{m})$ along with their corresponding per-pixel relative weights $w_s(\mathbf{b}, \mathbf{m})$ to produce the artifact-reduced image $\hat{u}_{AR}(\mathbf{n})$ according to the following equation:

$$\hat{u}_{AR}(\mathbf{n}) = \frac{\sum_s \sum_{\mathbf{b}, \mathbf{m}} \delta(\mathbf{n}, M\mathbf{b} + \mathbf{m} + \mathbf{s}) \cdot w_s(\mathbf{b}, \mathbf{m}) \cdot \hat{u}_s(\mathbf{b}, \mathbf{m})}{\sum_s \sum_{\mathbf{b}, \mathbf{m}} \delta(\mathbf{n}, M\mathbf{b} + \mathbf{m} + \mathbf{s}) \cdot w_s(\mathbf{b}, \mathbf{m})}$$

where:

$\mathbf{n}$ is a pixel address of a pixel of interest,

$\mathbf{b}$ is the block index of the block in which the pixel of interest is located,

$\mathbf{m}$ is the location of the pixel of interest in its block,

the blocks are sized M by M ,

$\mathbf{s}$ denotes the shift of the block-grid, and

$\delta(\mathbf{n}_1, \mathbf{n}_2)$ is 1 if $\mathbf{n}_1 = \mathbf{n}_2$ and 0 otherwise.

16. The system according to claim 15, wherein the local weights generator (110) is further configured to accumulate the product of the per-pixel relative weights (w_s) with the plurality of candidate reconstructions ($\hat{u}_s$) and to accumulate a sum of per-pixel weights (w_s) for producing the artifact-reduced image ($\hat{u}_{AR}$) without having to store intermediate per-pixel weights, thereby improving the computational efficiency in processing the input image and

increasing a range of hardware platforms that accommodate the processing.

17. The system according to claim 14, wherein the local weights generator (110) is further configured to determine activity measures $A_s(\mathbf{b})$ for each of the blocks $\mathbf{b}$ in the plurality of candidate reconstructions $\hat{u}_s(\mathbf{b}, \mathbf{m})$, which are frequency-weighted variances for a block-grid shift $\mathbf{s}$ based upon the following equation:

$$A_s(\mathbf{b}) = \sum_{\mathbf{k} \neq 0} \alpha \left[\sum_{\mathbf{m}} T_{\mathbf{k}, \mathbf{m}} \hat{u}_s(\mathbf{b}, \mathbf{m}) \right]^2 = \sum_{\mathbf{k} \neq 0} \alpha_{\mathbf{k}} \cdot \left[\psi_{\mathbf{k}} \left(\left| \sum_j T_{\mathbf{k}, j} f_s(\mathbf{b}, j) \right| \right) \right]^2 \quad \forall \mathbf{b}, \mathbf{s},$$

where:

10 T denotes the block transform employed, and

$\Psi_{\mathbf{k}}$ denotes the set of non-linear mappings that are each applied to their corresponding transform coefficient $\mathbf{k}$.

15 18. The system according to claim 14, wherein the local weights generator (110) is further configured to calculate the per-pixel relative weights $w_s(\mathbf{b}, \mathbf{m})$ in a manner designed to protect desired features from filtering while allowing artifacts to be filtered according to the following equation:

$$w_s(\mathbf{b}, \mathbf{m}) = C_s(\mathbf{b}) \cdot g_{\sigma(n)}(|\hat{u}_s(n) - f_s(n)|),$$

20 where:

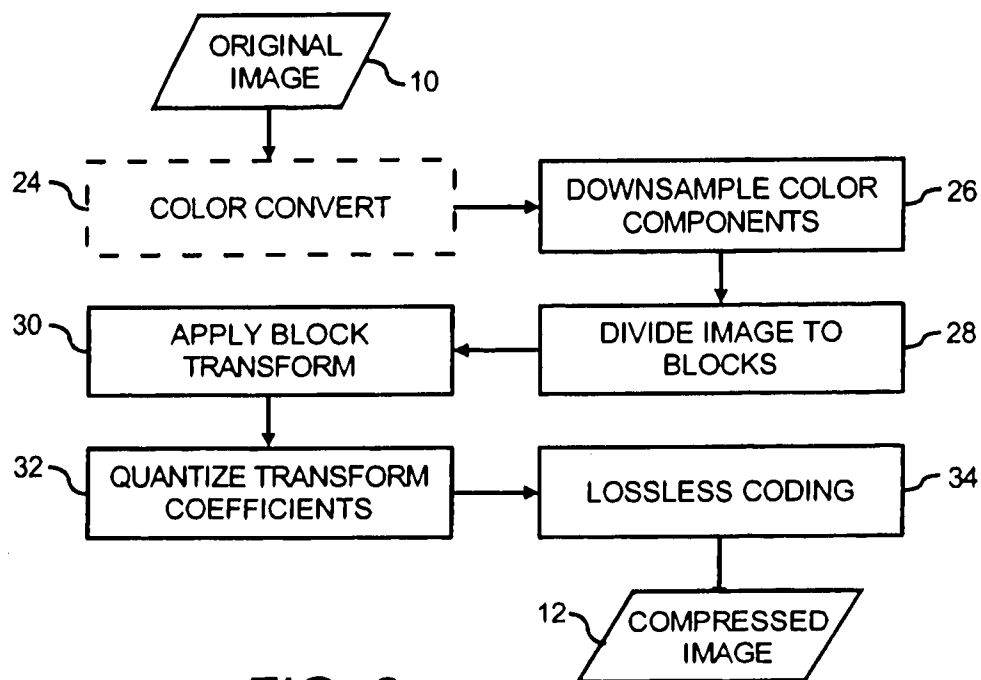
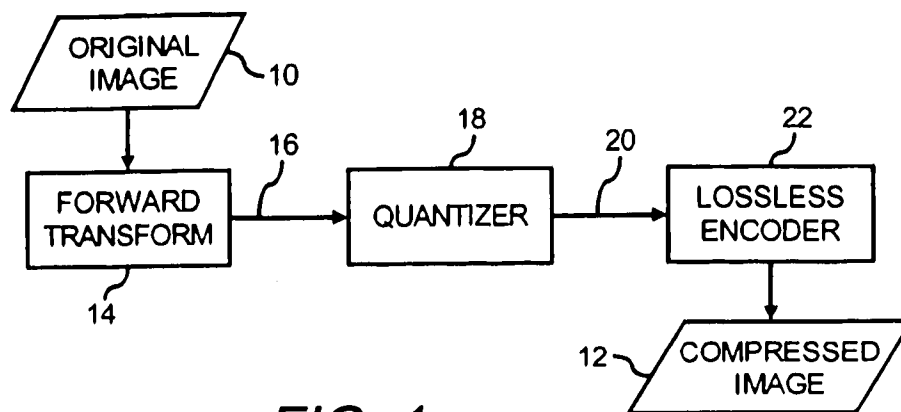
$C_s(\mathbf{b})$ is a self-confidence attribute measure for block $\mathbf{b}$ and a shift $\mathbf{s}$,

$g_{\sigma}(t)$ is a decreasing function of the difference between $\hat{u}_s(\mathbf{b}, \mathbf{m})$ and the original pixel value in that location $f_s(n)$ divided by a positive local feature

contrast scale $\sigma(n)$, such that $g_{\sigma}(0)=1$, $g_{\sigma}(\sigma) \sim 0.5$, $g_{\sigma}(t \gg \sigma) \ll 1$, and

25 the local feature contrast scale $\sigma(n)$ is computed per pixel by a feature detector that is insensitive to the features desired for protection, and sensitive to artifacts desired to be filtered.

1 / 8



2 / 8

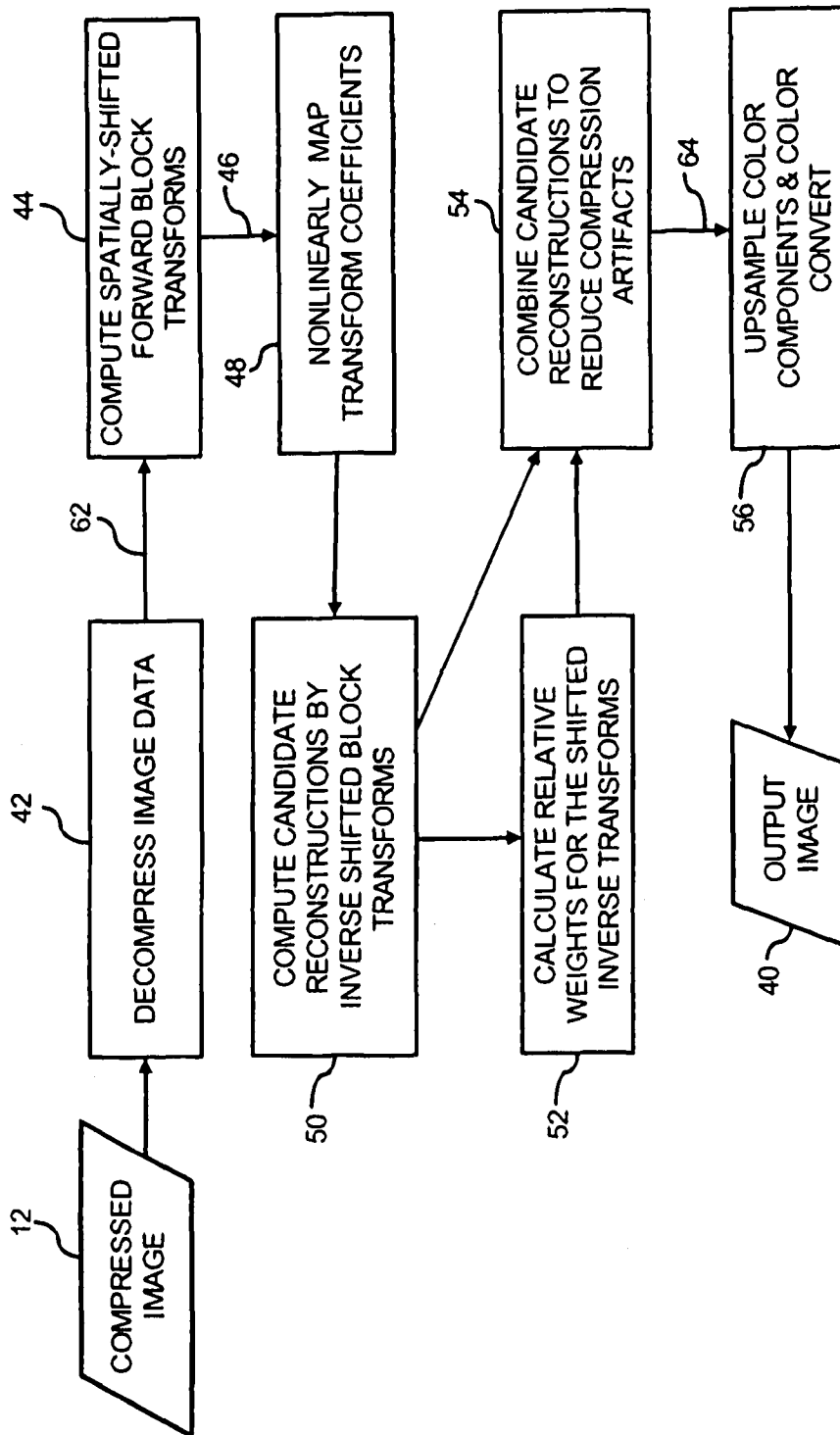


FIG. 3

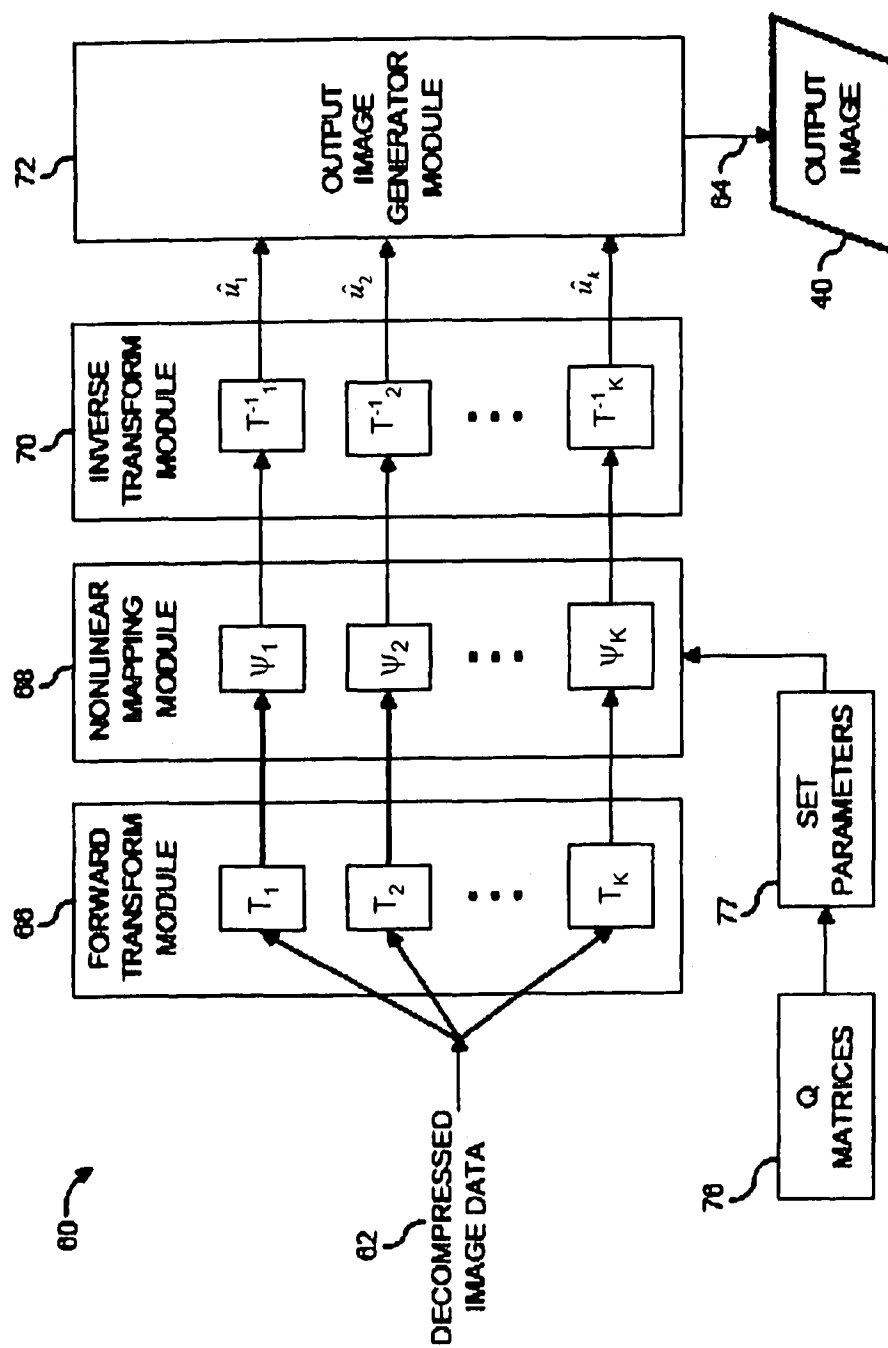
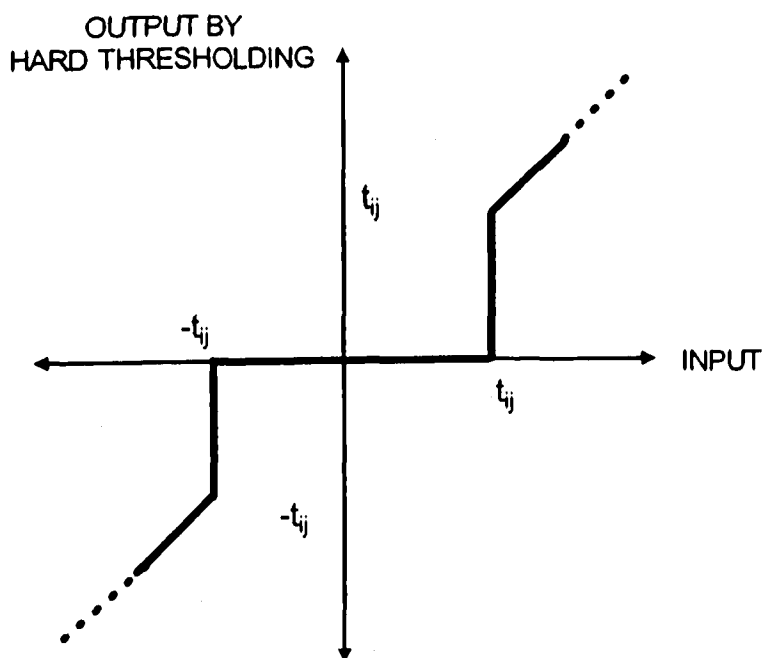


FIG. 4

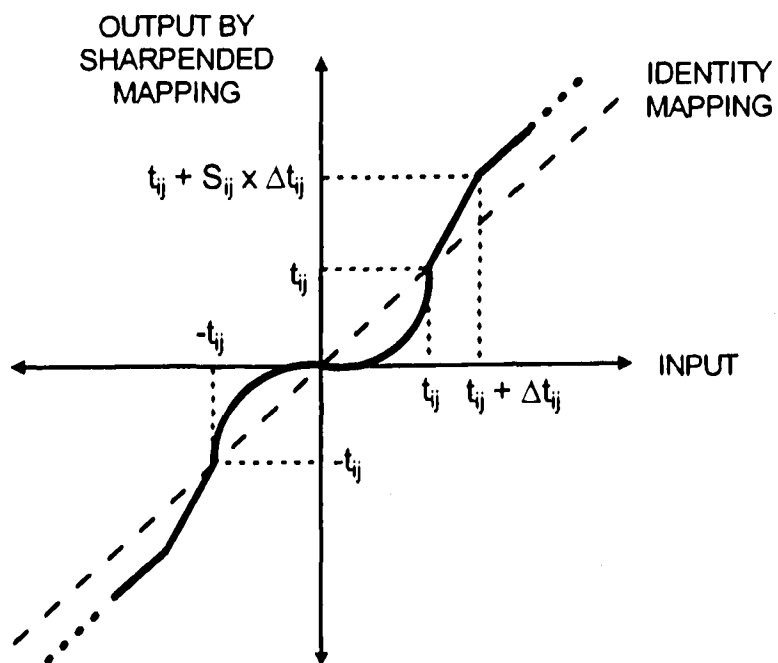
80

FIG. 5A



90

FIG. 5B



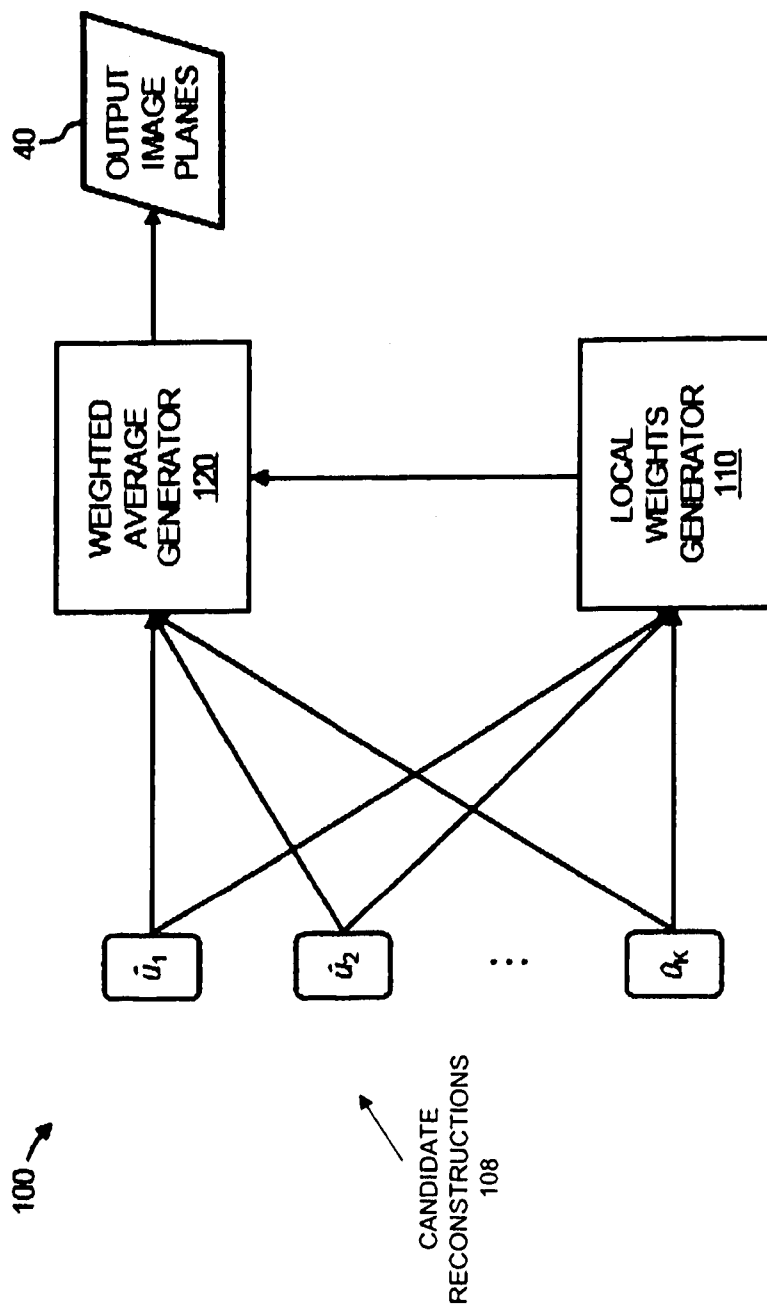


FIG. 6A

6 / 8

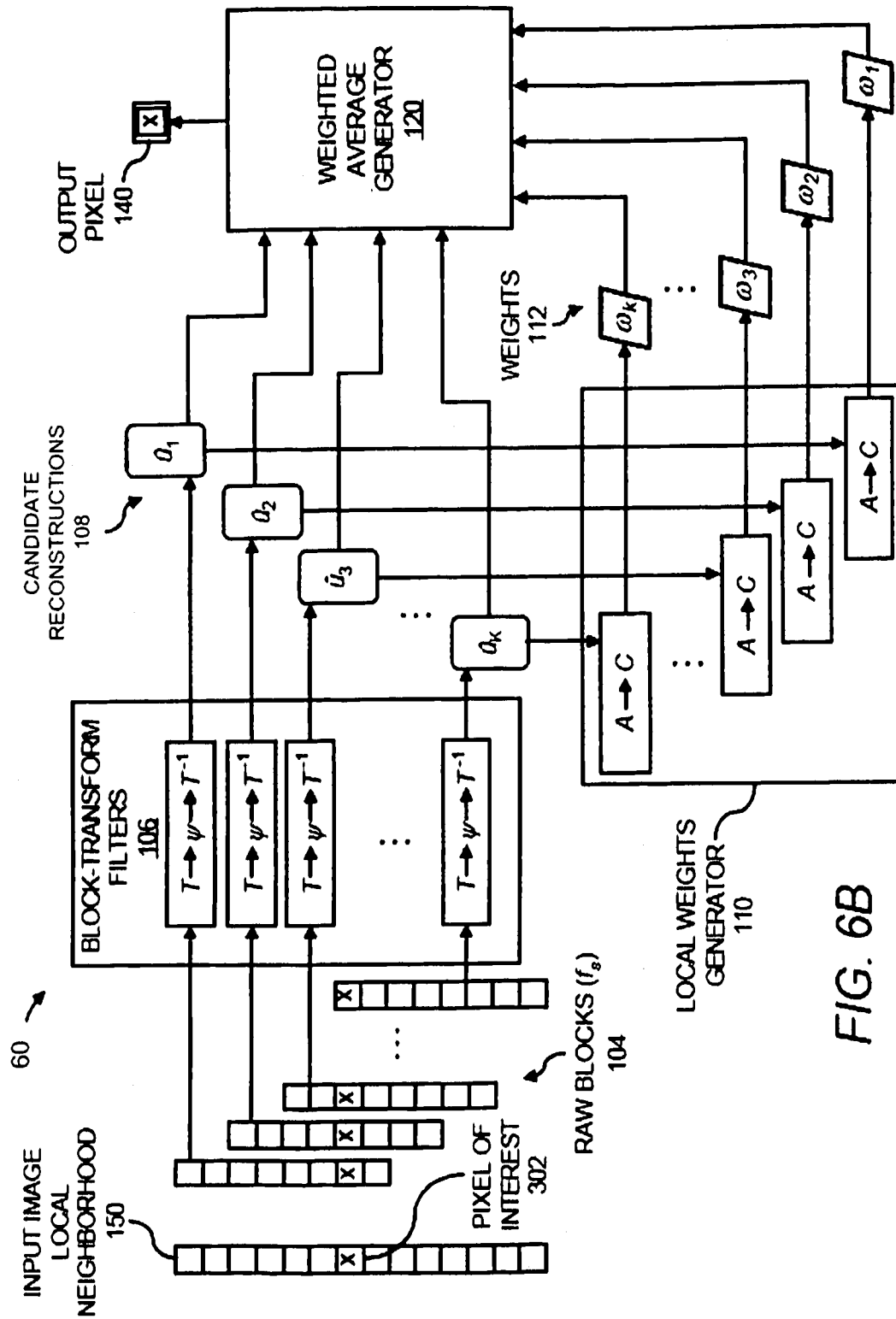


FIG. 6B

300

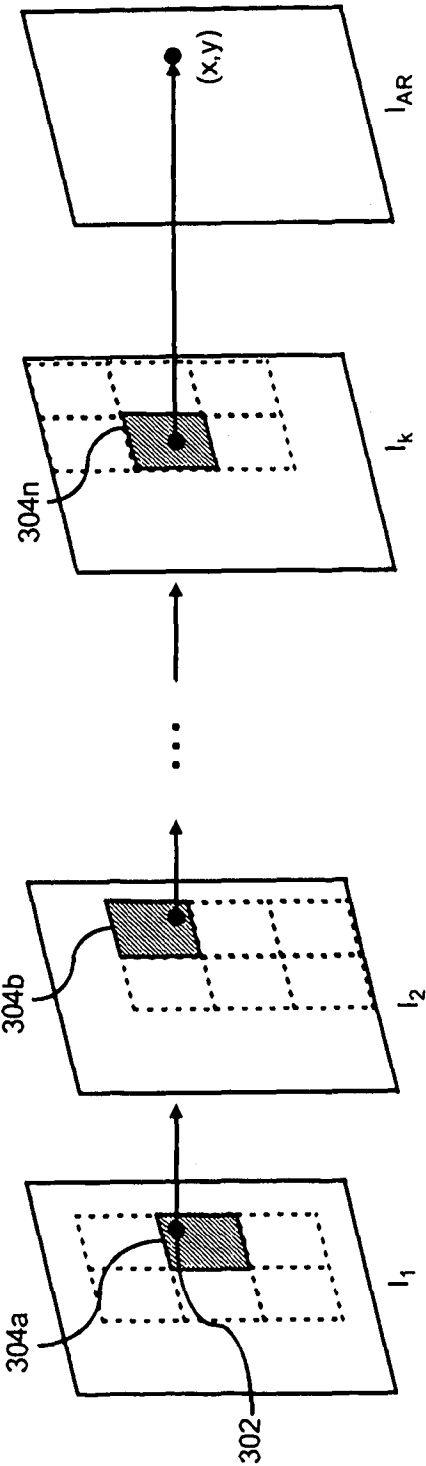


FIG. 7

8 / 8

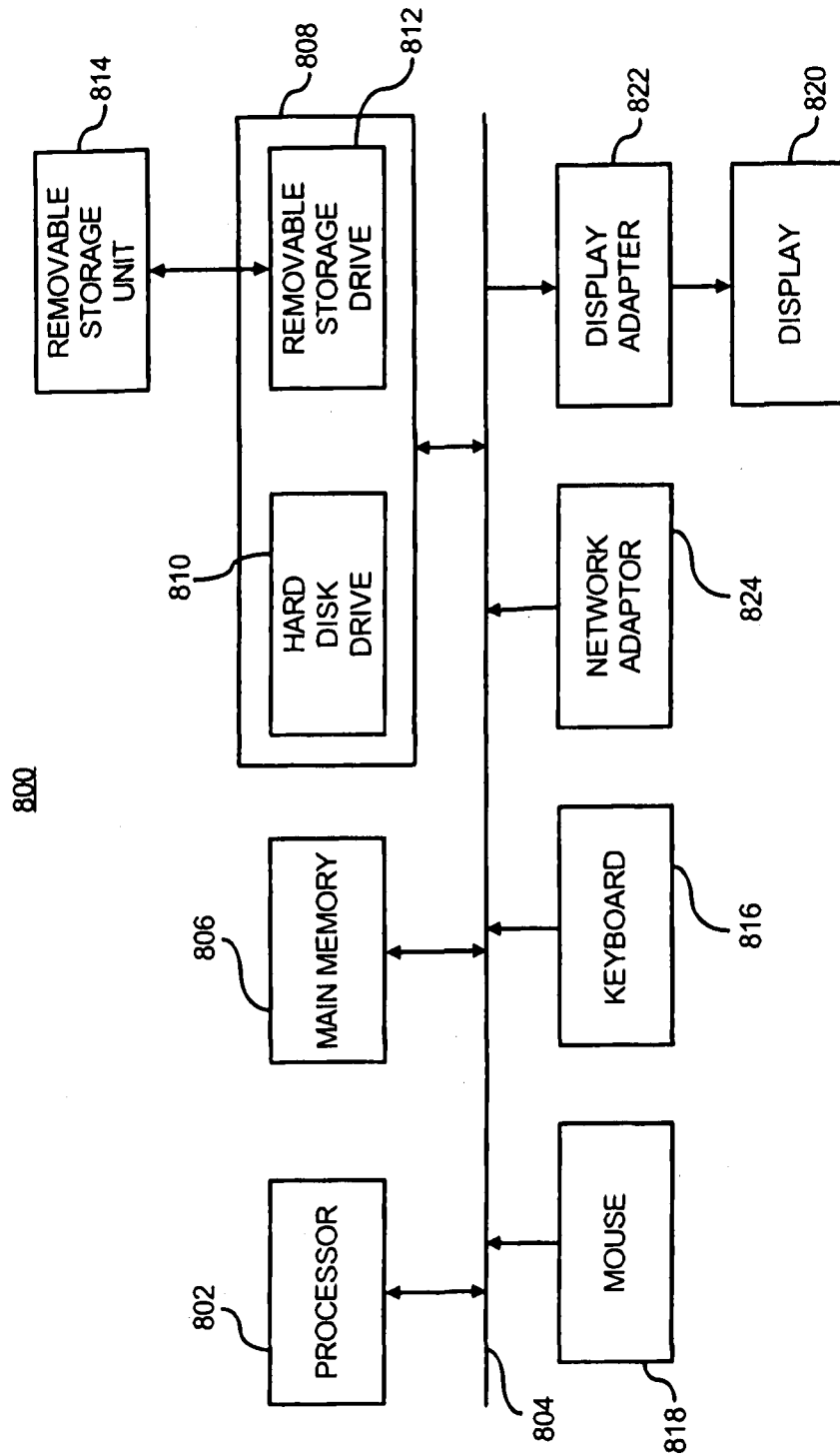


FIG. 8