

(19)



(11)

EP 2 661 081 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
04.03.2015 Bulletin 2015/10

(51) Int Cl.:
H04N 19/85 (2014.01)

(21) Application number: **12173534.4**

(22) Date of filing: **26.06.2012**

(54) **Processing of image**

Bildverarbeitung

Traitement d'image

(84) Designated Contracting States:
**AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR**

(30) Priority: **01.05.2012 US 201213461029**

(43) Date of publication of application:
06.11.2013 Bulletin 2013/45

(73) Proprietor: **Gurulogic Microsystems OY
20100 Turku (FI)**

(72) Inventors:
• **Kärkkäinen, Tuomas
20100 Turku (FI)**
• **Kalevo, Ossi
20100 Turku (FI)**

(74) Representative: **Kolster Oy Ab
Iso Roobertinkatu 23
PO Box 148
00121 Helsinki (FI)**

(56) References cited:
WO-A1-2006/016007 US-A- 6 160 848

EP 2 661 081 B1

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description

FIELD OF THE INVENTION

[0001] The present invention relates to processing of an image such that after compressing, the amount of image data to be stored on a storage medium, or alternatively the amount of data to be further transmitted over a communications network, can be minimized.

DESCRIPTION OF THE PRIOR ART

[0002] Previously there are known image/video coders, such as DivX, MPEG4, JPEG, JPEG2000, etc., which are able to compress an image such that the file size is considerably reduced as compared with what the file size would be if the image had been stored directly in the format received from a camera, video camera, scanner, screen, file or any other image source. For instance, the MPEG compression utilizes a solution in which all image frames are not directly forwarded from the coder as image frames, but only in given situations a single image frame is forwarded from the coder. Frames subsequent to the image frame are not forwarded, but only the changed frame parts are retrieved and they are forwarded from the coder. In this way there will be no need to continuously send information regarding a portion of the image which remains unchanged. In connection with reproduction, a single frame is displayed, after which change information collected from subsequent frames is utilized to change the single image frame at the same rate as the original images. Thus, the viewer will see an almost identical image as the one the camera originally shot.

[0003] The coders utilizing the above-described known compression algorithms have a drawback. That is, more often than necessary they interpret the image as having changed to the extent that it necessitates transmission of a whole new image frame. Frequent forwarding of a whole frame from the coder leads to a large file size. For instance, WO 86/03922 teaches that there is also previously known a solution in which memories of an image transmitter and receiver maintain the same earlier-shot reference image and in which, as the image changes, it is first checked to determine whether the changed areas have changed to correspond to the respective blocks in the stored reference image. If they have changed, a code word is sent from the receiver, on the basis of which code word the receiver can retrieve from the reference image in its memory the changed movement blocks. Thereafter, if the changed movement blocks do not correspond to the respective blocks in the reference image, the changed movement blocks are transmitted from the transmitter to the receiver. The solution known from WO 86/03922 has a drawback, however, in that it works effectively only when the background of the frame remains unchanged. In other words, in practice the camera should stay still. In addition, a solution based on code word trans-

mission is not compatible with known coders, and consequently compression algorithms of the known coders that have been found effective cannot be utilized.

[0004] Previously there is also known from WO 2006/016007 A1 a solution for processing a video image. In this solution movement blocks detected as unchanged as compared to a previous processed frame are indicated by setting the colour value of their pixels to a predetermined value. However, standard video codecs are unable to correctly handle unchanged frames indicated in such a way.

[0005] Previously there is also known from US - A - 6 160 848 a conditional replenishment device implemented as a separate module which is connected to a H.263 encoder. In this solution the option of declaring blocks as "not-coded" is utilized as defined in the H.263 standard by the conditional replenishment device for indicating blocks where no significant and meaningful changes have occurred since the last corresponding block. A problem with such a solution is that it can be implemented only by encoders supporting the H.263 standard.

SUMMARY OF THE INVENTION

[0006] It is an object of the present invention to solve the above problems associated with the known solutions by providing a solution for processing an image in a novel manner which enables fast and efficient video encoding and in connection with which it is possible to use coders utilizing prior art compression algorithms for image compression.

[0007] In order for the solution to be effectively utilized in connection with the known coder compression algorithms, the invention always forwards or stores all the movement blocks of the whole frame. In a solution according to independent claims 1, 9 and 13 standard and most probably also future video codecs can efficiently detect and encode zero motion vectors when the block in the current and in the reference frame is same or very similar. For that reason the pixel values in the block that is detected to be unchanged is copied from the reference frame.

[0008] Preferred embodiments of the processor and the method of the invention are disclosed in the dependent claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] In the following the invention will be described in greater detail, by way of example, with reference to the accompanying drawings, in which:

- Figure 1 is a block diagram of a processor in accordance with the invention;
- Figure 2 is a flow chart of a processing method in accordance with the invention;
- Figure 3 is a block diagram of a reproduction device in accordance with the invention;

Figure 4 illustrates production of an image with the reproduction device, and

Figure 5 is a flow chart of a reproduction method in accordance with the invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0010] Operations set forth in the blocks of a processor 1 of the block diagram in Figure 1 can be implemented by software, circuit solutions or combinations thereof. Operationally the blocks can be implemented in the presented manner or alternatively the operations of one or more blocks can be integrated.

[0011] In practice the processor 1 can be integrated with the camera with which the image is obtained. This camera may be, for instance, a conventional camera, video camera, a control camera or a mobile station (for example, a mobile phone, computer tablet, etc.) with an integrated camera. It is advantageous that the processor processes the image directly from the camera, i.e., prior to storing it in a file or transmitting it further over a communications network, because in that case it will be possible to minimize the noise appearing in the image at the time of processing. This kind of noise increases the size of the file to be stored or forwarded.

[0012] In the exemplary case of Figure 1 the processor 1 receives a signal 3 produced by a camera. On the basis of the processed signal the processor 1 generates and further supplies a signal 4. The latest frame received from the camera 2, i.e., the frame under process, is stored in a memory 5. A second memory 6 stores the previous processed frame. A comparison block 7 compares the contents of the memories 5 and 6 one movement block at a time and detects the changed movement blocks on the basis of colour changes in the pixels of the movement blocks.

[0013] A modification block 8 modifies the frame under process on the basis of the comparison performed by the comparison block 7. Modification is performed on the movement blocks stored in the memory 5 which are unchanged as compared with the frame stored in the memory 6. In the modification the characteristic values, such as colour values of the unchanged movement block pixels are set to a characteristic value that is not present in the changed movement blocks. In this example it is by way of example assumed that the characteristic value is a colour value, and it is set to correspond to the value of a specific colour. The colour values of the pixels may therefore be set to define the pixel colour as black.

[0014] The modification block 8 also modifies the frame stored in the memory 6. This is done by copying the pixels of all changed movement blocks from the memory 5 and said pixels are stored on the corresponding movement blocks of the frame in the memory 6. Said changed frame in the memory 6 is then used as a reference frame for a subsequent frame to be received from the camera 2 so as to detect changed movement blocks.

[0015] The processor 1 utilizes the changed and un-

changed movement blocks of the frame stored in the memory 5 for generating a signal 4 to be further transmitted. Said signal is supplied to a coder 9, which may be a prior art coder that compresses the received image into a compact format (for instance MPEG4, Div X, JPEG, or JPEG2000). After the compression by the coder 9 the obtained file is stored, for instance, on a computer hard disc 10 or another storage medium readable with an image reproduction device. Alternatively, instead of being stored, the obtained file is forwarded from the coder 9 over a communications network to an image reproduction device.

[0016] In Figure 1 the coder 9 is depicted by way of example as a separate part from the processor. Alternatively, another option is that the coder 9 is integrated with the processor, and consequently the processor and the coder can be implemented by one microcircuit, for instance.

[0017] Figure 2 is a flow chart revealing in greater detail the operation of the processor in Figure 1.

[0018] To initiate processing there are first performed preparatory operations. Raw image material or part of the raw image material to be captured from the camera is set unchanged in format, e.g., RGB, YUV, CMYK, B/W, Y. Movement blocks (MB) are defined to be rectangles having a width (MBW) and a height (MBH), i.e., both are preferably four pixels. MBW and/or MBH can also be 1 pixel. The size of the movement block (MBSIZE) is calculated by multiplying the width of the movement block (MBW) by its height (MBH), $(MBW * MBH) = MBSIZE$.

[0019] The width (FW) and the height (FH) of a frame (FRAME) are set to be divisible by the width (MBW) and the height (MBH) of the movement block (MB). The number of fixed movement blocks (MB) is calculated by multiplying the frame width (FW) by the frame height (FH) and the product is divided by the size of the movement block (MBSIZE), i.e. $((FH * FW) = FRAMESIZE / MBSIZE) = MBCOUNT$.

[0020] A matrix (MBMATRIX) is created according to the number (MBCOUNT) of the movement blocks (MB) for instance such that a frame is divided into sections (4x4) starting from the upper left corner and ending at the lower right corner. For each (MBCOUNT) movement block (MB) in the frame there are calculated corner coordinates X and Y of the rectangle, P1, P2, P3, P4, which are placed in the matrix in accordance with the order number (MBINDEX) of the movement block. The corner coordinates of the movement block are disposed as follows: P1 is the upper left corner, P2 is the upper right corner, P3 is the lower right corner and P4 is the lower left corner.

[0021] There are created a frame buffer (FRAMEBUFFER) and a last frame buffer (LASTFRAMEBUFFER) that are equal in size. The size to be reserved for the buffers (FRAMEBUFFERSIZE, LASTFRAMEBUFFERSIZE) is calculated by multiplying the frame width (FW), the frame height (FH) and the colour depth (COLORDEPTH) in bytes or bits, i.e. $(FH * FW * COLOR-$

DEPTH) = FRAMEBUFFERSIZE = LASTFRAMEBUFFERSIZE.

[0022] For pixel changes there is created a common colour sensitivity value (COLORSENSITIVITY), which defines the pixel change. The higher the value, the greater the colour change in the compared pixels must be in order for the current pixel to be found changed.

[0023] A common trigger value (MBTRIGGERVALUE) is created for the changes in the movement blocks, which value defines a change in the movement block. The higher the value, the larger the number of pixels which must change, in order for the current movement block to have changed. The common trigger value may be set to one, in which case a given movement block is detected as changed if in said movement block there is at least one pixel whose characteristic value, in this example colour value has changed.

[0024] A counter value of changed pixels (CHANGECOUNTER) is created, which value indicates the number of changed pixels in the current movement block.

[0025] A list (CHANGEDLIST) of changed movement blocks is created, which list indicates an index of the changed movement blocks.

[0026] After these preparatory operations, reception of a signal from a camera starts in block A. The image arriving from the camera or camera module is copied in the FRAMEBUFFER memory. The last frame buffer (LASTFRAMEBUFFER) memory is zeroed, if it has not been used before. To zero the memory denotes that the value of each byte is set to zero. The list of the changed movement blocks (CHANGEDLIST) is reset. The value of the colour sensitivity of the movement block (COLORSENSITIVITY) is set, for instance, to have a value within the range of 4096 to 8192. The value can be set to be more sensitive (e.g., 2048 to 4096) if the target of the image is in the dark, or correspondingly, higher (e.g. 8192 to 16384) if the target is better lit than normal. The value varies also based on the bit depth of the pixel values. The smaller the bit depth (COLORDEPTH) of the pixel value is, the smaller the COLORSENSITIVITY value is, and vice versa. For one colour component the COLORSENSITIVITY value can vary from 1 to 255, if 8 bits (= 1 byte) are used for colour component value. A zero value means that there is no change in pixel value. The trigger value of a change in the movement blocks (MBTRIGGERVALUE) is set to be 4 to 8, for instance. The value can also be set lower (e.g. 1 to 4) if the target is more remote, or higher (e.g. 8 to 16) if there is a lot of noise or interference in the target. The value varies also based on the size of the block. The bigger the block is, the bigger the MBTRIGGERVALUE value is, and vice versa.

[0027] Block C starts movement-block-specific comparison with the corresponding movement blocks in the preceding frame. Operations of blocks D to G are performed on each movement block.

[0028] Block D finds out whether the current movement block includes at least a given number of pixels whose colour value change exceeds a predetermined threshold.

This can be carried out in the following manner:

[0029] First the counter of changes in the movement block pixels (CHANGECOUNTER) is set to zero. A sub-loop is created where each (MBSIZE) pixel of the movement block (MB) to be processed (MBINDEX) is examined in accordance with the matrix (MBMATRIX) starting from the coordinates P1 through to the coordinates P4, however in rectangle-shape form not to go over the coordinates P2, nor to go under the coordinates P3. The following operations are performed on each pixel:

[0030] RGB values of pixels indicated by the coordinates are picked up from the frame under process (FRAMEBUFFER) and the preceding frame (LASTFRAMEBUFFER).

[0031] RG values (PIXELRG) of a pixel in the frame under process (FRAMEBUFFER) are compared with the RG values (LASTPIXELRG) of the pixel in the preceding frame (LASTFRAMEBUFFER) so as to obtain a change in the absolute value of colour components red (RED) and green (GREEN), i.e., an absolute (ABS) change (ABSRGDIFFERENCE). Counting is performed such that the difference between the RG values of pixels in the frame under process (FRAMEBUFFER) and the preceding frame (LASTFRAMEBUFFER) is placed in an ABS function, i.e. $ABS(PIXELRG - LASTPIXELRG) = ABSRGDIFFERENCE$. If the absolute change (ABSRGDIFFERENCE) in the colour components red (RED) and green (GREEN) is greater than the colour sensitivity (COLORSENSITIVITY) of the movement blocks, the value of the pixel change counter (CHANGECOUNTER) is raised by one.

[0032] If the change in pixels was not detected in the colour components red (RED) and green (GREEN), then the colour components of green (GREEN) and blue (BLUE) are compared in the same manner as described above for the RG values. The GB values (PIXELGB) of the pixel in the frame under process (FRAMEBUFFER) are in that case compared with the GB values (LASTPIXELGB) of the pixel in the preceding frame (LASTFRAMEBUFFER) so as to obtain an absolute (ABS) change (ABSGBDIFFERENCE) in the colour components green (GREEN) and blue (BLUE). Counting is performed such that the difference between the GB values of pixels in the frame under process (FRAMEBUFFER) and the preceding frame (LASTFRAMEBUFFER) are placed in an ABS function, i.e. $ABS(PIXELGB - LASTPIXELGB) = ABSGBDIFFERENCE$. If the absolute change (ABSGBDIFFERENCE) in the colour components green (GREEN) and blue (BLUE) is greater than the colour sensitivity (COLORSENSITIVITY) of the movement blocks, the value of the pixel change counter (CHANGECOUNTER) is raised by one. The previous RG and BG values are just examples and other colour values can be used when the change is detected and the selection of the used value also affects the COLORSENSITIVITY value.

[0033] When all pixels in the movement block have been examined in the above manner, the pixel change counter is checked. If the value of the pixel change coun-

ter (CHANGECOUNTER) is higher than the trigger value of a movement block change (MBTRIGGERVALUE), the movement block is detected as changed, whereby the process proceeds to block F, where a movement block index (MBINDEX) is set on a list of changed items (CHANGEDLIST). Whereas, if the value of the pixel change counter is lower than the trigger of the movement block change, the predetermined threshold was not exceeded, whereby the movement block is detected as and the process proceeds to block E.

[0034] In block G it is checked whether all the movement blocks of the frame to be processed have been examined. If not, the process returns to block C, from which the processing of the following movement block starts.

[0035] In block H the colour values of pixels in the movement blocks detected as unchanged are set to a predetermined value. One alternative is to set the colour values to indicate the movement block pixels as being black, because experiments have shown that prior art compression algorithms of an image are then capable of processing these movement blocks quickly, and additionally, the file size of the produced compressed image is small. However, it is possible to set the colour values to correspond to another colour that enables achievement of corresponding advantages with the compression algorithm used. The colour values can also be selected based on the colours that are not available or common in the image. Information of the selected colour can be pre-defined for the reproduction device or the reproduction device can get the information e.g. from image processor.

[0036] The setting of colour values can be implemented, for instance, by creating a loop, in which all changed movement blocks (CHANGEDLIST) are examined. In the loop there are examined the movement blocks that do not appear on the list of changed items (CHANGEDLIST). Each pixel of these unchanged movement blocks is set as black in the frame buffer (FRAMEBUFFER).

[0037] In block I the pixels of the changed movement blocks are copied to the preceding frame in the corresponding movement blocks. This can be implemented by examining in the loop the movement blocks that appear on the list of changed items (CHANGEDLIST). Each pixel of these movement blocks is copied unchanged to the same location in the preceding frame buffer (LASTFRAMEBUFFER).

[0038] In block J a signal is generated on the basis of the information (FRAMEBUFFER) on the changed and unchanged movement blocks. The signal thus conveys a pre-processed image that can subsequently be compressed by means of a compression algorithm.

[0039] In block K it is monitored whether more data is arriving from the camera or another image source. In the affirmative, the process returns to block A, from which the processing of a new received frame starts.

[0040] Unlike in the flow chart of Figure 2 which is not

an embodiment of the invention, after detecting a specific movement block changed, it is possible in connection with block F, to record the movement blocks (8 blocks) surrounding said changed movement block as changed as well, for instance, by adding them to the list of changed items (CHANGEDLIST). Experiments have shown that in some cases there may occur interference in a reproduced image in these surrounding movement blocks.

[0041] Figure 3 is a block diagram of a first preferred embodiment of a reproduction device in accordance with the invention. Operations set forth in the blocks of the reproduction device 11 presented in the block diagram of Figure 3 can be implemented by computer software, circuit solutions or combinations thereof. Operationally the blocks can be implemented in the presented manner, or alternatively, the operations of one or more blocks can be integrated. In practice, the reproduction device 11 may consist of a computer, a television peripheral, a mobile station or any other reproduction device which is capable of reproducing an image.

[0042] In the case of Figure 3, it is assumed, by way of example, that compressed data is retrieved from a hard disk 10, after which a decoder 12 decodes compression (e.g. MPEG 4 or DivX) and supplies to the reproduction device 11 a signal 13 containing image information. The images contained in the signal 13 are stored in a memory 14, from which the reproduction device 11 retrieves and processes them one by one.

[0043] Modification block 15 identifies from the frame stored in the memory 14 the movement blocks comprising at least a given number of pixels whose characteristic values correspond with a given accuracy to values of a given characteristic. In this example it is assumed that the characteristic value is a colour value, and consequently, movement blocks with at least a given number of pixels whose colour values correspond with a given accuracy to those of a given colour are identified. The modification block 15 identifies these movement blocks as unchanged movement blocks. The modification block 15 identifies all other blocks as changed movement blocks. The modification block 15 copies the pixels of the changed movement blocks from the frame in the memory 14 to the corresponding movement blocks of the preceding frame to be maintained in the memory 16. The reproduction device 11 utilizes the image produced in the memory 16 by including it in a signal 17 to be applied to a display 18.

[0044] Figure 3 shows, by way of example, that the hard disk 10 (or another storage medium of a file), the decoder 12 and the display 18 are separate from the reproduction device 11. In practice, however, one or more of these parts can be integrated with the reproduction device. Moreover, it is an option that the data to be reproduced is not stored on a storage medium at all, but is received from a communications network from which it is supplied to the decoder 12.

[0045] Figure 4 illustrates production of an image with the reproduction device. On the left in Figure 4 there is

a previous, processed and reproduced frame. In the middle there is a frame under process, wherein unchanged movement blocks are black, and whereby only the changed movement blocks are in the original form. When the changed movement blocks of the middle frame are copied to the corresponding locations in the previous frame on the left, the result will be a new, changed frame shown on the right.

[0046] Figure 5 is a flow chart of a first preferred embodiment of a reproduction method in accordance with the invention. At first, some preliminary operations are performed to reproduce an image.

[0047] A decoded image is received in, e.g., RGB format. The movement blocks are rectangles whose width (MBW) and height (MBH) are, i.e., both four pixels. MBW and/or MBH can also be one pixel. The size of a movement block (MBSIZE) is calculated by multiplying the width (MBW) of the movement block by its height (MBH), $(MBW * MBH) = MBSIZE$. The width (FW) and the height (FH) of a frame (FRAME) are set to be divisible by the width (MBW) and the height (MBH) of the movement block. The number of fixed movement blocks (MB) is calculated by multiplying the frame width (FW) by the frame height (FH) and the product is divided by the size of the movement block (MBSIZE), i.e. $((FH * FW) = FRAME-SIZE / MBSIZE) = MBCOUNT$.

[0048] A matrix (MBMATRIX) is created in accordance with the number (MBCOUNT) of the movement blocks (MB), for instance, such that a frame is divided in sections (4x4) starting from the upper left corner and ending at the lower right corner. For each (MBCOUNT) movement block (MB) in the frame there are calculated corner coordinates X and Y of the rectangle, P1, P2, P3, P4, which are placed in the matrix in accordance with the order number (MBINDEX) of the movement block. The corner coordinates of the movement block are disposed as follows: P1 is the upper left corner, P2 is the upper right corner, P3 is the lower right corner and P4 is the lower left corner.

[0049] There are created a frame buffer (FRAMEBUFFER) and a buffer of a frame to be interpreted (PREFRAMEBUFFER) that are equal in size. The size to be reserved for the buffers (FRAMEBUFFERSIZE, PREFRAMEBUFFERSIZE) is calculated by multiplying the frame width (FW), the frame height (FH) and the colour depth (COLORDEPTH) in bytes or bits, i.e. $(FH * FW * COLORDEPTH) = FRAMEBUFFERSIZE = PREFRAMEBUFFERSIZE$.

[0050] There is created a common trigger value (MBTRIGGERVALUE) (for instance 4), which determines a change in the movement block. The higher the value, the more pixels need to change in order for the movement block to be considered changed. There is created a counter value of changed pixels (CHANGECOUNTER), which indicates the number of changed pixels in the movement block to be processed.

[0051] Reception of the signal containing image information starts in block L. The image information contained

in the received signal is stored in a memory (PREFRAMEBUFFER). The memory (FRAMEBUFFER) of a completed frame to be reproduced is zeroed, if it has not yet been used. To zero the memory denotes that the value of each byte is set to zero.

[0052] In block M the pixels of the frame to be examined are divided into movement blocks. Thus, the pixels that belong to a subsequent movement block to be examined are searched one movement block at a time.

[0053] In blocks N to O all the movement blocks of the frame are examined one by one so as to find out whether in the movement block there are at least a given number of pixels whose characteristic values correspond with a given accuracy to values of a given characteristic. In this example it is examined if the colour values correspond with a given accuracy to those of a given colour. In the affirmative, an unchanged movement block is in question. Otherwise a changed movement block is in question. This can be implemented such, for instance, that the following operations are performed on each movement block:

[0054] The pixel change counter (CHANGECOUNTER) of a movement block is set to zero;

[0055] A sub-loop is created in which each (MBSIZE) pixel of the movement block (MB) to be processed (MBINDEX) is examined in accordance with the matrix (MBMATRIX), starting from the coordinates P1 through to the coordinates P4, however, in rectangle-shape form not to go over the coordinates P2, nor to go under the coordinates P3;

[0056] The pixel colour indicated by the coordinates is established from the interpreted frame (PREFRAMEBUFFER);

[0057] If the unchanged movement blocks are indicated by setting their colour values to correspond to those of black colour, then, if the value of the pixel colour (PIXELCOLOR) of the frame to be interpreted (PREFRAMEBUFFER) exceeds the black sensitivity (BLACKSENSITIVITY) value of the changing pixels (for instance 4096), the value of the pixel change counter (CHANGECOUNTER) is raised by one. (In other words, if the pixel colour is not black, the value of the pixel change counter is raised by one.);

[0058] Whereas, if the unchanged movement blocks have been indicated by setting their colour values to correspond to those of another colour, then:

[0059] If the sum of pixel colour components red (RED) and green (GREEN) differs at most for a predetermined error margin from the sum of the colour components red (RED) and green (GREEN) of a given colour, the value of the pixel change counter (CHANGECOUNTER) will be raised by one.

[0060] If the error margin was not exceeded in the comparison of the red (RED) and the green (GREEN) colour components, the green (GREEN) and the blue (BLUE) colour components are also compared. If the sum of the pixel colour components green (GREEN) and blue (BLUE) of the interpreted frame differs at most for a given

error margin from the sum of the green (GREEN) and the blue (BLUE) colour components of a given colour, the value of the pixel change counter (CHANGE-COUNTER) will be raised by one.

[0061] When all the pixels have been examined, the value of the pixel change counter (CHANGE-COUNTER) is checked. If the value exceeds the trigger value of the change in the movement block (MBTRIGGERVALUE), a changed movement block is in question, whereby the process proceeds via block P to block Q. Whereas, if the value is not higher than the change trigger value, the process proceeds via block O to block R.

[0062] In block Q the pixels of the changed movement blocks are copied to the preceding frame. This can be done by copying each pixel of the movement block under process unchanged to the same location in the frame buffer (FRAMEBUFFER).

[0063] From block R the process returns to block M to process a subsequent movement block if there are still unexamined movement blocks in the frame under process.

[0064] In block S a signal is generated on the basis of the modified frame (FRAMEBUFFER).

[0065] If the signal reception goes on, the signal containing more frames to be interpreted, the process returns from block T to block L so as to process the image information of a subsequent frame received.

[0066] When the above-described method is used, the standard or currently implemented image or video compression methods (codecs) can be used without modification. New methods can be used to improve image compression efficiency or image quality when the used codec solution can also be modified.

[0067] With non-modified codecs it is sometimes also difficult to indicate unchanged movement blocks by colour values when the used colour is also in the image in the changed area or the changed image area contains almost all colours so that it is difficult to change to a colour value suitable for the unchanged detection.

[0068] When the codec being used is modified, the information of unchanged movement blocks can be delivered differently rather than changing the pixel colour values of the movement block to the predefined colour value. For example, information can be separately delivered from the image processor to the video encoder and decoder as well as to the reproduction device. When the encoder or decoder knows that some block is unchanged, it does not need to compress or decompress that block at all and so the compression efficiency can be improved. In addition, the processing requirements of the encoder and decoder are reduced. As a result, the reproduction device then modifies only the decoded pixels and does not touch the unchanged blocks at all, but keeps the information the same as it was in the previous image frame.

[0069] When a situation arises wherein the same colour is available in the changed movement block as that used as an indicator colour for the unchanged movement

block, then there are multiple solutions available to solve the conflict.

[0070] A first solution changes the indicator colour to the colour that is not available in the changed areas and delivers this changed colour to the reproduction device. If the delivery is not possible or if there is no suitable colour available, then other solutions are needed.

[0071] A second solution detects if there is the indicator colour used in the changed blocks and slightly modifies those pixel values in the changed blocks so that the image quality is not damaged. However, the values used in the changed movement block are not detected as the unchanged block in the reproduction device. This means that all of the colour values are modified or at least some of the colour values are modified inside the movement block. It is also possible that other pixel and colour values, other than those originally used as the colour indicator, inside the movement block are modified, if that makes detection more robust or it improves the image quality.

[0072] It is also possible to modify unchanged movement block colour pixels and send some other information within the unchanged movement block, as long as the detection of movement block is still unchanged and it does not require too much additional bandwidth in the codec. In that way, some additional information improving coding efficiency or image quality can be delivered without any changes to the normal operation of the image processor and the image reproduction device.

[0073] A third solution uses a modified image/video codec that enables the indication of an unchanged movement block by signals other than the indication colour in the image. This information can be delivered by using one bit for every movement block to indicate whether the movement block is changed or remains unchanged. This information can also be compressed by using different coding methods, for example, run length coding. This information has to be used in the encoder and in the decoder. This information also needs to be delivered to the decoder before the image information of the movement block tries to decode. The method of the present invention is able to encode and decode this one bit per movement block information separately from the other information. This information does not contain any address changes or motion vector information, but only the information that the movement block is changed or unchanged.

[0074] In the present method the previous image or one selected image (Note: there can be multiple frames in the memory and one of them is selected to be as LAST-FRAMEBUFFER) is always used and the present method is therefore suitable for all image areas (not only for background areas). Note: the movement block can be unchanged inside the moving object or it can be a different object in different images, whereas the image information inside the movement block is similar enough in the current and reference image.

[0075] It is also possible that the detection of the movement block being changed or unchanged is made based

on some of the colour components and some of the colour components are still coded differently. For example, if Y and U colour channels are detected as unchanged and their values are set as an indication colour, but B channel is still coded as a changed movement block, this issue has to be taken into account in both the image processor and the reproduction device.

[0076] It is also possible that the indication of the movement block being unchanged is made based on some colour components in the reproduction device, but still all colour components or some other colour components are processed as unchanged movement areas. This means that the image information is copied to unchanged colour components from the reference memory and the new image information of that colour might be used for other colour components or for some other purposes. This issue has to be taken into account in both the image processor and the reproduction device. It is also possible that the indication of the movement block being changed is made based on some colour components in the reproduction device, but still all colour components or some other colour components are processed as changed movement areas. This enables e.g. delivery and reception of the indication colour as changed movement block.

[0077] It is also possible that the detection of changed/unchanged is made differently in the image processor than using CHANGECOUNTER and MB-TRIGGERVALUE. It can be based on, for example, sum/mean of differences, sum/mean of absolute differences, variance/standard deviation of differences, variance/standard deviation of absolute differences, or any other detection method. The image processor still indicates the movement block to be unchanged by using an indication colour in the image or the image processor delivers the changed/unchanged status by any other means that can be used in the image/video reproduction device (and also in encoder and in decoder).

[0078] It is also possible that the threshold of changed/unchanged block detection in the image processor varies block by block. The reproduction device (and e.g. encoder and decoder) will get the information based on the indication colour or other mean, but the image processor can arrive at a decision based on the current and previous frames (e.g., if the mean of the absolute differences is used as a changed/unchanged detection method, then the variance of the pixels in the reference frame and/or in the current frame can change the threshold). The bigger the variance of the movement block pixels, the bigger the threshold for changed/unchanged detection. Also noise level in the image can change the threshold. The bigger the noise level, the bigger the threshold. The noise level can be calculated also from the previous frame as well as from the current frame. It is also possible that the mean of the block variances can be used as a noise level detector. It is also possible that only some blocks (e.g., 10 %) are used for noise detection; that is, the blocks that have the smallest variances. Also, many other noise detection methods can be used,

and the mean of the pixel values can change the threshold. There might not always be any linear change of the threshold and the other parameters available, but the change can be also nonlinear.

[0079] It is possible that the detection of the changed/unchanged movement block is done differently in the reproduction device. The detection can be based on the number of pixels that have the indication colour value. The detection can also be, for example, the mean of the pixel values in the movement block. It can also be the mean and variance of the pixel values in movement block. As mentioned earlier, it is also possible that the pixel values are not used at all (pixels might not be even delivered/received), but the information is delivered/received by another mean.

[0080] It is also possible to use movement blocks that are different in size in different parts of the image and then MBCOUNT is also calculated differently. In this case, the image processor and the image reproduction device have to know what sizes the movement blocks are in the different parts of the image. This information can be pre-defined or sent by the image processor and received by the reproduction device using many different methods. In practice this kind of method requires a modified image encoder and decoder that can understand the used block sizes and optimize functionalities based on them.

[0081] Standard and most probably also future video codecs can efficiently detect and encode zero motion vectors when the block in the current and in the reference frame is same or very similar. For that reason, in the claimed invention, the pixel values in the block that is detected to be unchanged can be copied from the reference frame and that can be used as "indication colour" to the encoder. This method enables faster and more efficient video encoding, but still improves the coding result e.g. in noisy imaging environment and in case when the block is not wanted to be modified. When this kind of indication method is used then the reproduction device just uses decoder output and modifies reference image based on those pixel values as all the blocks would be changed movement blocks. Reproduction device can also detect all the zero motion blocks without any prediction error coding to be as unchanged movement blocks and then it can simplify its operations and does not need to do any unnecessary block copies to the reference frame when the block is not actually changed.

[0082] The input image can also be transformed into different dimension (without increasing or decreasing the quality of image information). The blocks of original image are moved so that image height is set to the height of motion block and image width is set using formula by multiplying total amount of changed blocks with the width of motion block. With this method only the changed blocks by motion remains in transformed image and all unchanged blocks are ignored. For this method, the information of changed and unchanged blocks must be within. This information can be delivered by using one

bit for every movement block to indicate whether the movement block is changed or remains unchanged. This information can also be compressed by using different coding methods, for example, run length coding, bzip2-specific RLE. This information has to be used in the encoder and in the decoder. This information also needs to be delivered to the decoder before the image information of the movement block are tried to be decoded. There are several ways to process this kind of transformed image. The transformed image can be encoded using any lossless or lossy image codec like jpg, png, etc.

[0083] The specified lossy or lossless image codec can be used to decode this kind of image as any image. After the decoding the transformed image is needed to be restored back to the full size image within dimensions of original image width and height. The restored image must be build using changed blocks in decoded transformed image and unchanged blocks that are already known. The bit header information of changed and unchanged blocks are decoded and used for this. For lossy or lossless image decoder all the changed blocks are copied into their positions by using bit header information and all unchanged blocks are copied from the reference image. For some cases the output image uses same memory as reference image and then only all changed blocks are copied to it and all unchanged blocks are ignored.

[0084] It is also possible to perform extremely fast and almost perfect motion detection for blocks by subsampling the current image. The subsampling can be calculated using mean or weighted mean. Implementation can use different masks. The size of mask is equal for the motion block size. This method is efficiently fast because the subsampled output image is smaller than the input image. For an example if 8x8 motion blocks are used for 128x128 input image then 16x16 output image values are used for changed or unchanged blocks detection.

[0085] It is also possible that the image or block has processed with different colour or pixel formats (e.g., Y, YUV420, YUV422, YUV444, RGB444, G). The format can also be planar, interleaved, line planar etc. It is also possible that during the processing the format is changed; that is, the original image is in interleaved RGB format and when it comes out from the image processor the format is planar YUV420. The size of the movement block for different colour channels can be the same or different, and also the amount of movement blocks in the different colour channels can be different. Colour and pixel format changes can be done in the image processor as well as in the reproduction device.

[0086] It is also possible to process only part of the image by using any of the methods described above. For example the image may contain some interesting area that can be called as region of interest (ROI). The blocks outside of the ROI are set as unchanged and so they do not need any special processing. This enables to speed-up the total processing and decreases the amount of used bits. This method uses bits mainly for ROI area and all other areas are coded within minimal number of bits.

Claims

1. An image processor (1) which is configured:
 - 5 to receive a signal (3),
 - to compare (7) an individual frame under process, included in the received signal (3), with a preceding processed frame,
 - to detect changes (7) by dividing pixels of the individual frame into movement blocks of a predetermined size and to detect a given movement block as changed if in said movement block there are at least one pixel whose characteristic value has changed as compared with a corresponding pixel of the preceding processed frame by a predetermined threshold, where the characteristic value is a colour value,
 - 10 to provide an indication of the movement blocks that are unchanged as compared with the preceding processed frame,
 - to generate a signal (4) which indicates the changes by including in said signal the information on movement blocks detected as changed and information on movement blocks detected as unchanged such that said signal includes all the movement blocks of the frame under process, and
 - 15 to supply said signal (4) to an encoder (9) that compresses the received signal, **characterized in that** the image processor is configured to provide said indication to said encoder (9) of the movement blocks that are unchanged by copying the pixel values in the movement blocks that are detected to be unchanged from the preceding processed frame.
2. The image processor (1) of claim 1, **characterized in that** the image processor is configured to divide only a portion of the pixels of an individual frame into movement blocks.
3. The image processor (1) of claim 1, **characterized in that** a given movement block is detected as changed if there is at least a given number of pixels whose characteristic value change exceeds a predetermined threshold.
4. The image processor (1) of claim 1 or 2, **characterized in that** the characteristic value is a colour value corresponding to one of RGB, YUV, CMYK, B/W and Y.
5. The image processor (1) according to one of claims 1 to 4, **characterized in that** the detection of whether the movement block is changed or unchanged is based upon at least one of sum/mean differences, sum/mean absolute differences, variance/standard deviation of differences, and variance/standard de-

viation of absolute differences.

6. The image processor according to one of claims 1 to 5, **characterized in that** if the processor (1) detects a given movement block as changed on the basis of a characteristic value check of pixels in the movement block, the processor (1) is configured to record also all the movement blocks surrounding the given movement block as having changed.
7. The image processor according to one of claims 1 to 6, **characterized in that** the processor (1) is configured to analyze a change in characteristic values such that if the absolute value of a change in a first characteristic value of a single pixel exceeds said predetermined threshold, the processor (1) finds the characteristic values of said pixel changed.
8. The image processor (1) according to one of claims 1 to 7, **characterized in that** the received signal (3) is a video signal generated by at least one of a camera, video camera, scanner, screen, file or other image source (2).
9. A method for processing a signal, the method comprising:
- receiving (A) a signal,
 - comparing a single frame included in the signal and being processed with a preceding processed frame
 - detecting (D) changes by dividing pixels in the single frame into movement blocks of a predetermined size and by detecting a given movement block as changed, if the number of pixels whose characteristic value changes as compared with the corresponding pixels in the preceding processed frame exceeds a predetermined threshold, where the characteristic value is a colour value,
 - providing an indication of the movement blocks that are unchanged, as compared with the preceding processed frame,
 - generating (J) a signal that indicates the changes by including in the signal the information on the movement blocks detected as changed as well as the information regarding the unchanged movement blocks, such that said signal includes all the movement blocks of the frame under process, and
 - supplying said signal to an encoder that compresses the received signal, **characterized in that** in said method
 - said indication (H) of the movement blocks that are unchanged is provided to said encoder by copying the pixel values in the movement blocks that are detected to be unchanged from the preceding processed frame.

10. The method of 9, **characterized in that** the movement block is a rectangle.
11. The method according to one of claims 9 to 10, **characterized in that** the method analyzes changes in characteristic values by analyzing changes in colour values and the colour values of a single pixel are detected as changed if the absolute value of a first colour value of the pixel exceeds said predetermined threshold.
12. The method according to one of claims 9 to 11, **characterized in that** the colour values correspond to at least one of RGB, YUV, CMYK, B/W and Y values.
13. Software for controlling a programmable device for implementing the method according to one of claims 9 to 12.

Patentansprüche

1. Bildprozessor (1), der konfiguriert ist zum:
- Empfangen eines Signals (3),
 - Vergleichen (7) eines in Verarbeitung befindlichen individuellen Einzelbilds, das in dem Empfangssignal (3) enthalten ist, mit einem vorhergehenden verarbeiteten Einzelbild,
 - Detektieren von Veränderungen (7) durch Unterteilen von Pixeln des individuellen Einzelbilds in Bewegungsblöcke vorbestimmter Größe und Detektieren eines gegebenen Bewegungsblocks als verändert, falls sich in dem Bewegungsblock mindestens ein Pixel befindet, dessen Kennwert sich im Vergleich mit einem entsprechenden Pixel des vorhergehenden verarbeiteten Einzelbilds um einen vorbestimmten Schwellenwert verändert hat, wobei der Kennwert ein Farbwert ist,
 - Ausgeben einer Angabe der Bewegungsblöcke, die im Vergleich mit dem vorhergehenden verarbeiteten Einzelbild unverändert sind,
 - Erzeugen eines Signals (4), das die Veränderungen angibt, indem in dem Signal die Information über als verändert detektierte Bewegungsblöcke und Information über als unverändert detektierte Bewegungsblöcke derart einbezogen ist, dass das Signal sämtliche Bewegungsblöcke des in Verarbeitung befindlichen Einzelbilds enthält, und
 - Zuführen des Signals (4) zu einem Kodierer (9), der das Empfangssignal komprimiert, **dadurch gekennzeichnet, dass** der Bildprozessor konfiguriert ist zum:
- Ausgeben des Hinweises auf die Bewegungsblöcke, die unverändert sind, an den

Kodierer (9), indem die Pixelwerte der als unverändert detektierten Bewegungsblöcken aus dem vorhergehenden verarbeiteten Einzelbild kopiert werden.

2. Bildprozessor (1) nach Anspruch 1, **dadurch gekennzeichnet, dass** der Bildprozessor derart konfiguriert ist, dass er nur einen Teil der Pixel eines Einzelbilds in Bewegungsblöcke unterteilt.
3. Bildprozessor (1) nach Anspruch 1, **dadurch gekennzeichnet, dass** ein gegebener Bewegungsblock als verändert detektiert wird, falls mindestens eine gegebene Anzahl von Pixeln existiert, deren Kennwert einen vorbestimmten Schwellenwert überschreitet.
4. Bildprozessor (1) nach Anspruch 1 oder 2, **dadurch gekennzeichnet, dass** der Kennwert ein Farbwert ist, der einem der Farbwerte RGB, YUV, CMYK, B/W und Y entspricht.
5. Bildprozessor (1) nach einem der Ansprüche 1 bis 4, **dadurch gekennzeichnet, dass** die Detektion, ob der Bewegungsblock verändert oder unverändert ist, auf mindestens einer aus Summen-/Mittelwert-Differenzen, Summen-/Mittelwert-Absolutdifferenzen, Varianz/Standardabweichung von Differenzen und Varianz/Standardabweichung von Absolutdifferenzen basiert.
6. Bildprozessor (1) nach einem der Ansprüche 1 bis 5, **dadurch gekennzeichnet, dass**, falls der Prozessor (1) einen gegebenen Bewegungsblock auf der Basis einer Kennwertprüfung von Pixeln in dem Bewegungsblock als verändert detektiert, der Prozessor (1) derart konfiguriert ist, dass er auch sämtliche Bewegungsblöcke, die den gegebenen Bewegungsblock umgeben, als verändert aufzeichnet.
7. Bildprozessor (1) nach einem der Ansprüche 1 bis 6, **dadurch gekennzeichnet, dass** der Prozessor (1) derart konfiguriert ist, dass er eine Veränderung in den Kennwerten dahingehend analysiert, dass, falls der Absolutwert einer Veränderung in einem ersten Kennwert eines einzelnen Pixels den vorbestimmten Schwellenwert überschreitet, der Prozessor (1) den Kennwert des Pixels als verändert erachtet.
8. Bildprozessor (1) nach einem der Ansprüche 1 bis 7, **dadurch gekennzeichnet, dass** das Empfangssignal (3) ein Videosignal ist, das mittels mindestens eines bzw. einer aus einer Kamera, einer Videokamera, einem Scanner, einem Bildschirm, einer Datei oder einer anderen Bildquelle (2) erzeugt wird.
9. Verfahren zum Verarbeiten eines Signals, wobei das

Verfahren umfasst:

Empfangen (A) eines Signals,
 Vergleichen eines in dem Signal enthaltenen und in Verarbeitung befindlichen individuellen Einzelbilds, mit einem vorhergehenden verarbeiteten Einzelbild,
 Detektieren (D) von Veränderungen durch Unterteilen von Pixeln in dem individuellen Einzelbild in Bewegungsblöcke vorbestimmter Größe und durch Detektieren eines gegebenen Bewegungsblocks als verändert, falls die Anzahl von Pixeln, deren Kennwert sich im Vergleich mit den entsprechenden Pixeln in dem vorhergehenden verarbeiteten Einzelbild verändert hat, einen vorbestimmten Schwellenwert überschreitet, wobei der Kennwert ein Farbwert ist,
 Ausgeben eines Hinweises auf die Bewegungsblöcke, die im Vergleich mit dem vorhergehenden verarbeiteten Einzelbild unverändert sind,
 Erzeugen (J) eines Signals, das die Veränderungen angibt, indem in dem Signal die Information über als verändert detektierte Bewegungsblöcke und die Information über die unveränderten Bewegungsblöcke derart einbezogen wird, dass das Signal sämtliche Bewegungsblöcke des in Verarbeitung befindlichen Einzelbilds enthält, und
 Zuführen des Signals zu einem Kodierer, der das Empfangssignal komprimiert, **dadurch gekennzeichnet, dass** bei dem Verfahren die Angabe (H) der Bewegungsblöcke, die unverändert sind, an den Kodierer übermittelt wird, indem die Pixelwerte in den als unverändert detektierten Bewegungsblöcken aus dem vorhergehenden verarbeiteten Einzelbild kopiert werden.

10. Verfahren nach Anspruch 9, **dadurch gekennzeichnet, dass** der Bewegungsblock ein Rechteck ist.
11. Verfahren nach einem der Ansprüche 9 bis 10, **dadurch gekennzeichnet, dass** bei dem Verfahren Veränderungen in den Kennwerten analysiert werden, indem Veränderungen in Farbwerten analysiert werden und die Farbwerte eines einzelnen Pixels als verändert detektiert werden, falls der Absolutwert eines ersten Farbwerts des Pixels den vorbestimmten Schwellenwert überschreitet.
12. Verfahren nach einem der Ansprüche 9 bis 11, **dadurch gekennzeichnet, dass** die Farbwerte mindestens einem der Werte RGB, YUV, CMYK, B/W und Y entsprechen.
13. Software für das Steuern einer programmierbaren Vorrichtung zum Implementieren des Verfahren

nach einem der Ansprüche 9 bis 12.

Revendications

1. Processeur d'image (1) qui est configuré :

pour recevoir un signal (3),
 pour comparer (7) une trame individuelle en
 cours de traitement, incluse dans le signal reçu
 (3), avec une trame traitée précédente,
 pour détecter des changements (7) en divisant
 des pixels de la trame individuelle en blocs de
 mouvement d'une taille prédéterminée et pour
 détecter un bloc de mouvement donné comme
 changé si dans ledit bloc de mouvement, il y a
 au moins un pixel dont la valeur caractéristique
 a changé comparé à un pixel correspondant de
 la trame traitée précédente d'un seuil prédéter-
 miné, où la valeur caractéristique est une valeur
 de couleur,
 pour fournir une indication des blocs de mouve-
 ment qui sont inchangés comparé à la trame
 traitée précédente,
 pour générer un signal (4) qui indique les chan-
 gements en incluant dans ledit signal les infor-
 mations sur des blocs de mouvement détectés
 comme changés et des informations sur des
 blocs de mouvement détectés comme inchan-
 gés de sorte que ledit signal inclue tous les blocs
 de mouvement de la trame en cours de traite-
 ment, et
 pour fournir ledit signal (4) à un encodeur (9) qui
 compresse le signal reçu, **caractérisé en ce
 que** le processeur d'image est configuré
 pour fournir ladite indication audit encodeur (9)
 des blocs de mouvement qui sont inchangés en
 copiant les valeurs de pixel dans les blocs de
 mouvement qui sont détectés comme étant in-
 changés à partir de la trame traitée précédente.

2. Processeur d'image (1) selon la revendication 1, **ca- ractérisé en ce que** le processeur d'image est con- figuré pour diviser uniquement une partie des pixels d'une trame individuelle en blocs de mouvement.

3. Processeur d'image (1) selon la revendication 1, **ca- ractérisé en ce qu'un** bloc de mouvement donné est détecté comme changé s'il y a au moins un nom- bre de pixels donné dont le changement de valeur caractéristique dépasse un seuil prédéterminé.

4. Processeur d'image (1) selon la revendication 1 ou 2, **caractérisé en ce que** la valeur caractéristique est une valeur de couleur correspondant à une parmi RGB, YUV, CMYK, B/W et Y.

5. Processeur d'image (1) selon l'une des revendica-

tions 1 à 4, **caractérisé en ce que** la détection est
 si le bloc de mouvement est changé ou inchangé est
 basée sur au moins une parmi des différences som-
 me/moyenne, des différences absolues som-
 me/moyenne, une variance/écart-type de différen-
 ces, et une variance/écart-type de différences abso-
 lues.

6. Processeur d'image selon l'une des revendications 1 à 5, **caractérisé en ce que** si le processeur (1) détecte un bloc de mouvement donné comme chan- gé sur la base d'une vérification de valeur caracté- ristique de pixels dans le bloc de mouvement, le pro- cesseur (1) est configuré pour enregistrer également tous les blocs de mouvement entourant le bloc de mouvement donné comme ayant changé.

7. Processeur d'image selon l'une des revendications 1 à 6, **caractérisé en ce que** le processeur (1) est configuré pour analyser un changement de valeurs caractéristiques de sorte que si la valeur absolue d'un changement d'une première valeur caractéri- stique d'un seul pixel dépasse ledit seuil prédéter- miné, le processeur (1) trouve les valeurs caractéri- stiques dudit pixel changé.

8. Processeur d'image (1) selon l'une des revendica- tions 1 à 7, **caractérisé en ce que** le signal reçu (3) est un signal vidéo généré par au moins un parmi un appareil photo, une caméra vidéo, un scanner, un écran, un fichier ou une autre source d'image (2).

9. Procédé pour traiter un signal, le procédé comprenant :

la réception (A) d'un signal,
 la comparaison d'une seule trame incluse dans
 le signal et qui est traitée avec une trame traitée
 précédente,
 la détection (D) de changements en divisant des
 pixels dans la seule trame en blocs de mouve-
 ment d'une taille prédéterminée et en détectant
 un bloc de mouvement donné comme changé,
 si le nombre de pixels dont la valeur caractéri-
 stique change comparé aux pixels correspon-
 dants dans la trame traitée précédente dépasse
 un seuil prédéterminé, où la valeur caractéri-
 stique est une valeur de couleur,
 la fourniture d'une indication des blocs de mou-
 vement qui sont inchangés, comparé à la trame
 traitée précédente,
 la génération (J) d'un signal qui indique les chan-
 gements en incluant dans le signal les infor-
 mations sur les blocs de mouvement détectés com-
 me changés ainsi que les informations concer-
 nant les blocs de mouvement inchangés, de sor-
 te que ledit signal inclue tous les blocs de mou-
 vement de la trame en cours de traitement, et

la fourniture dudit signal à un encodeur qui compresse le signal reçu, **caractérisé en ce que** dans ledit procédé

ladite indication (H) des blocs de mouvement qui sont inchangés est fournie audit encodeur en copiant les valeurs de pixel dans les blocs de mouvement qui sont détectés comme étant inchangés à partir de la trame traitée précédente.

5

10

10. Procédé selon la revendication 9, **caractérisé en ce que** le bloc de mouvement est un rectangle.

11. Procédé selon l'une des revendications 9 à 10, **caractérisé en ce que** le procédé analyse des changements de valeurs caractéristiques en analysant des changements de valeurs de couleur et les valeurs de couleur d'un seul pixel sont détectées comme changées si la valeur absolue d'une première valeur de couleur du pixel dépasse ledit seuil prédéterminé.

15

20

12. Procédé selon l'une des revendications 9 à 11, **caractérisé en ce que** les valeurs de couleur correspondent à au moins une parmi des valeurs RGB, YUV, CMYK, B/W et Y.

25

13. Logiciel pour commander un dispositif programmable pour mettre en oeuvre le procédé selon l'une des revendications 9 à 12.

30

35

40

45

50

55

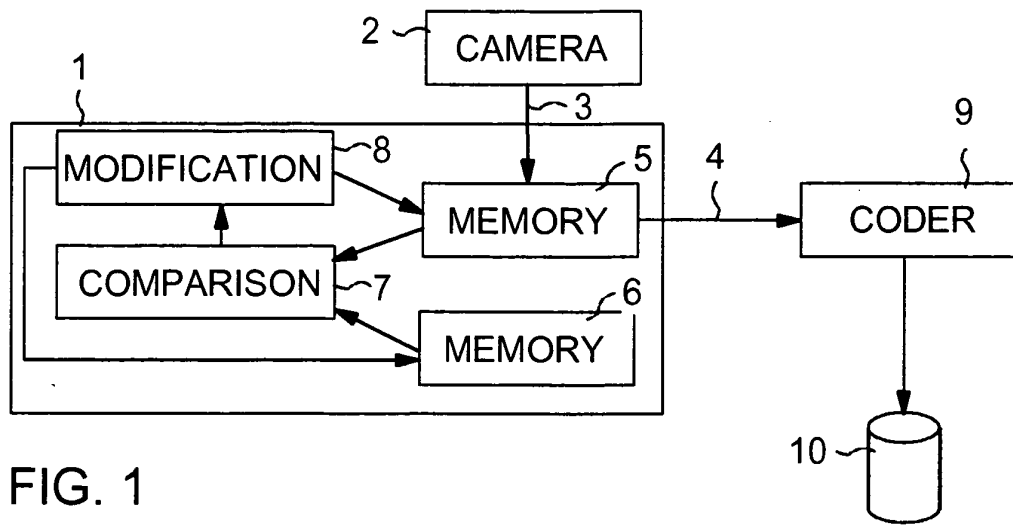


FIG. 1

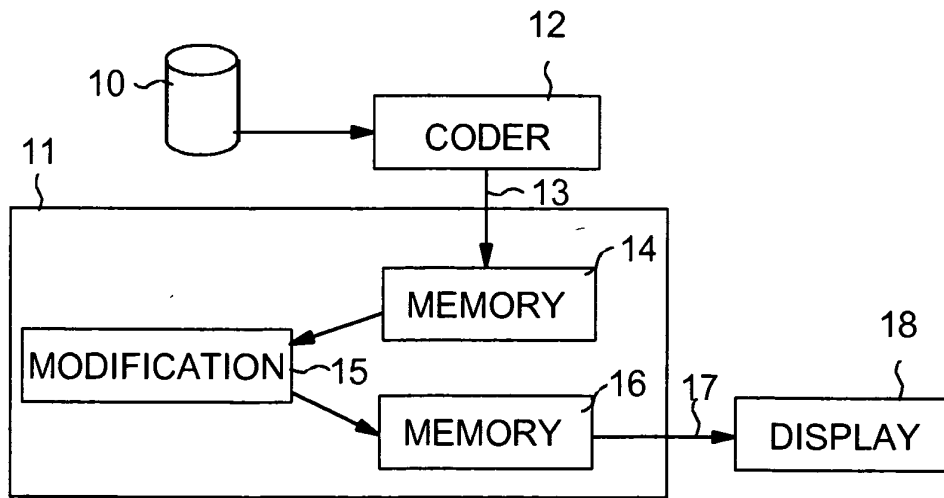


FIG. 3

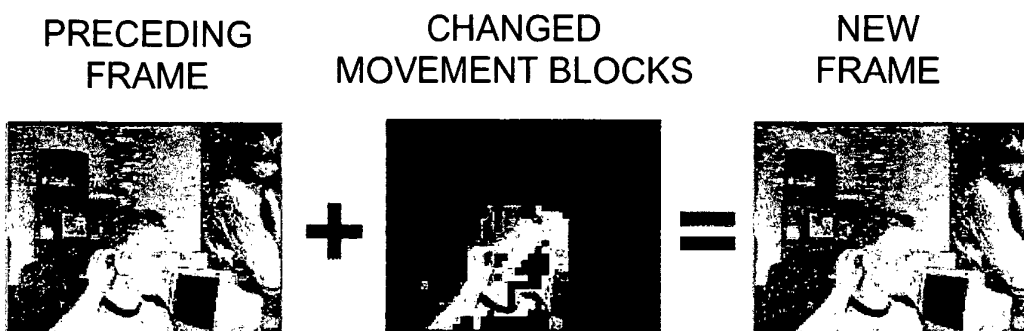


FIG. 4

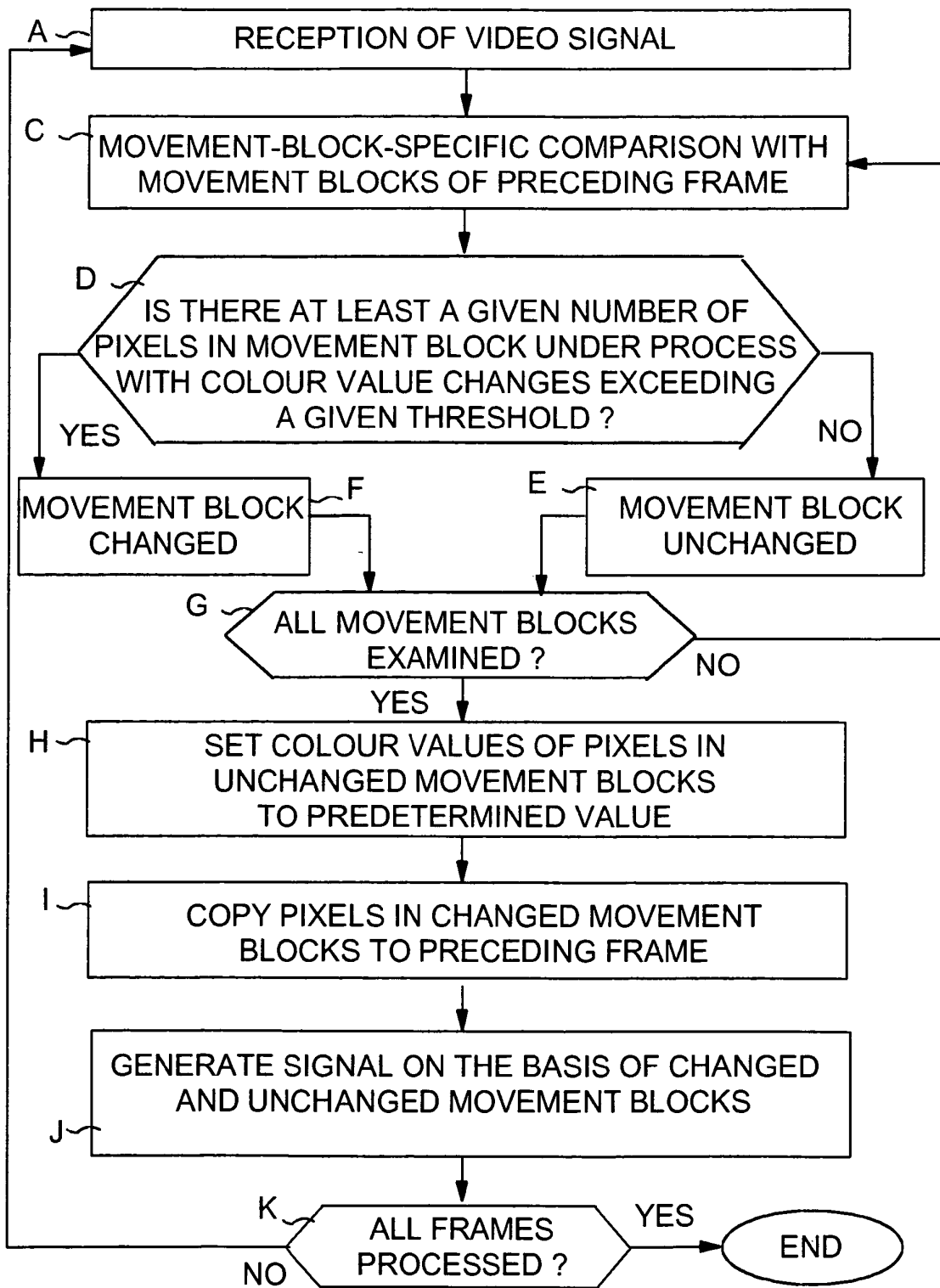


FIG. 2

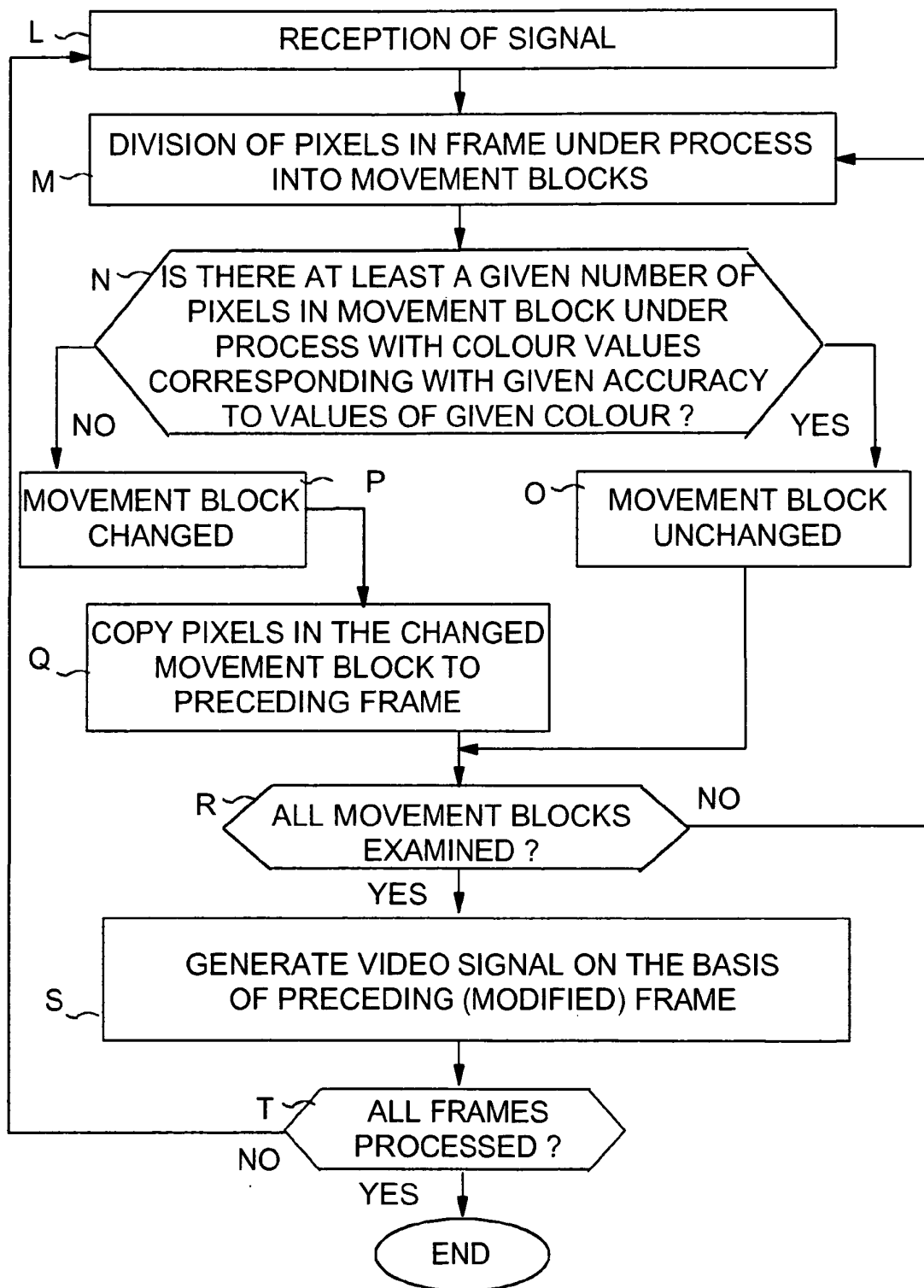


FIG. 5

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO 8603922 A [0003]
- WO 2006016007 A1 [0004]
- US 6160848 A [0005]