

(51) International Patent Classification:
H04N 7/26 (2006.01)(21) International Application Number:
PCT/JP2011/059455(22) International Filing Date:
11 April 2011 (11.04.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/798,708 9 April 2010 (09.04.2010) US(71) Applicant (for all designated States except US): **SHARP KABUSHIKI KAISHA** [JP/JP]; 22-22, Nagaïke-cho, Abeno-ku, Osaka-shi, Osaka, 5458522 (JP).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **ZHAO, Jie. SEGALL, Christopher A.**(74) Agent: **HARAKENZO WORLD PATENT & TRADE-MARK**; Daiwa Minamimorimachi Building, 2-6, Tenjin-bashi 2-chome Kita, Kita-ku, Osaka-shi, Osaka, 5300041 (JP).

(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, CL, 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, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

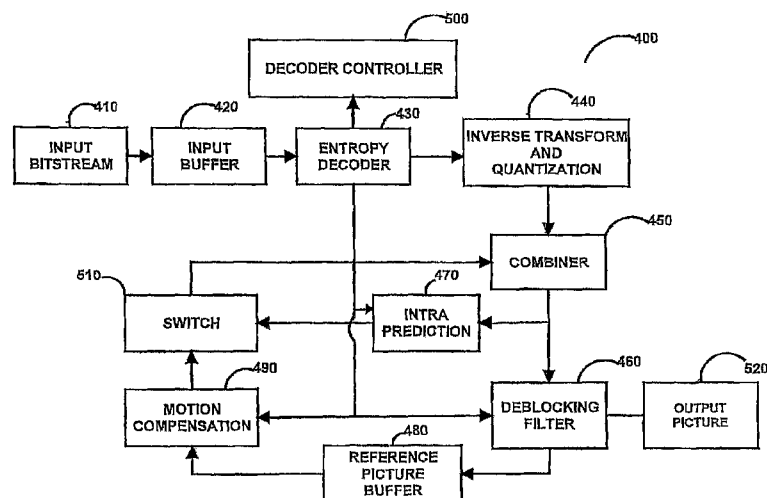
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: SUPER-BLOCK FOR HIGH PERFORMANCE VIDEO CODING

FIG. 2



(57) Abstract: A system for encoding and/or decoding video that includes the use of super blocks. The use of super blocks permits a reduction in the bit-rate of the video bit stream.

DESCRIPTION**TITLE OF INVENTION: SUPER-BLOCK FOR HIGH PERFORMANCE VIDEO CODING****TECHNICAL FIELD**

5 The present invention relates generally to a video encoder and/or a video decoder.

CROSS-REFERENCE TO RELATED APPLICATIONS

Not applicable.

BACKGROUND ART

10 The present invention relates generally to a video encoder and/or a video decoder.

 The transmission of video across a network typically includes a video encoder and a video decoder. The encoding of the video includes a lossy compression technique to achieve a lower bit rate for transmission while still
15 providing a perceptually good video quality. By way of example, digital video discs used a MPEG-2 video compression standard, hereby incorporated by reference in its entirety.

 Video compression typically operates based upon the grouping of neighboring pixels together, generally referred to as macroblocks. A macroblock,
20 or other group of pixels, are compared from one frame to another frame, where the differences between the frames are transmitted. In the presence of motion, the video compression transmits data indicative of the motion of the macroblock, or other group of pixels, from one frame to another frame together with the differences between the frames.

25 H.264/AVC (formally known as ISO/IEC 14496-10 – MPEG-4 Part 10, Advanced Video Coding) video compression standard, hereby incorporated by reference herein in its entirety, is used for many applications, such as Blu-ray discs. The H.264 standard is a block based compression standard that typically results in good video quality at substantially lower bit rates than MPEG-2.

While the H.264 standard provides a good result there is a desire for ever increasing reduction in the bit rate, especially for high definition content, while not significantly decreasing the perceived image quality.

SUMMARY OF THE INVENTION

5 A preferred embodiment is a method for decoding video comprising:
(a)receiving a super block flag indicating a super block consisting of a plurality of smaller blocks of pixels having shared decoding information with said super block; (b)receiving a coded block pattern flag indicating whether said super block has a residual; (c)decoding said super block based upon said super block flag
10 and said coded block pattern flag.

 Another preferred embodiment is a method of decoding video comprising:
(a)receiving a super block consisting of a plurality of smaller blocks of pixels having shared decoding information with said super block in a bit stream of encoded said video; (b)extracting said shared decoding information from one of
15 said smaller blocks of pixels; (c)applying said shared decoding information to another one of said smaller blocks of pixels of said super block; (d)decoding said one of said smaller blocks based upon said shared decoding information;
(e)decoding said another one of said smaller blocks based upon said shared decoding information.

20 The foregoing and other objectives, features, and advantages of the invention will be more readily understood upon consideration of the following detailed description of the invention, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

25 FIG. 1 illustrates a video encoder.
FIG. 2 illustrates a video decoder.
FIG. 3 illustrates block encoding.
FIG. 4 illustrates mapping of super blocks.
FIG. 5A illustrates syntax for slice data processing.

FIG. 5B illustrates syntax for slice data processing.

FIG. 6A illustrates syntax for macroblock processing.

FIG. 6B illustrates syntax for macroblock processing.

FIG. 7 illustrates extract, copy, and save for super-blocks.

- 5 FIG. 8 illustrates a data structure, (a) illustrates a data structure of a super-block and (b) illustrates a data structure of a macro-block.

FIG. 9 illustrates an example of syntax for a super-block header

DESCRIPTION OF EMBODIMENTS

Referring to FIG. 1, an exemplary H.264 encoder 200 is described for
10 purposes of illustration. It is to be understood that any video encoder may be used. The input video 210 is provided to a buffer suitable to reorder frames, or portions thereof, as necessary 220. A combiner 230 modifies a portion of the suitable reordered frame in a manner suitable for a transform and quantization process 240. The transform and quantization process 240 provides a signal to
15 an entropy coder 250. The entropy coder 250 provides a signal to an output buffer 260 for the output bit stream 270. An encoder controller 280 that receives the input video 210 provides control signals to all the modules of the encoder 200.

The transform and quantization process 240 also provides its output to an inverse transform and quantization 300 so that the corresponding decoder can
20 be simulated. A picture-type decision process 310 is interconnected with the frame ordering buffer 220. The picture-type decision process 310 is also interconnected to a macro-block-type decision 320. In this manner, control over the frame ordering buffer 220 may be achieved. In addition, control over the type of macro-block may be achieved.

25 The inverse transform and quantization 300 provides a signal to a combiner 330, which in combination with the macro-block type decision 320, provides a signal to an intra coding prediction module 340 and a deblocking filter 350. The deblocking filter 350 is interconnected to a reference picture buffer 360. The reference picture buffer 360 provides a signal to a motion estimation process

370 and a motion compensation process 380. The motion estimation 370 provides a signal to the motion compensation 380 and to the entropy coder 250. A selector 390 selects between the output of the motion compensation 380 and the output of the intra-coded prediction 340 for the combiner 230. In this manner,
5 the combiner 230 receives information related to whether the macro-block is intra coded 340 or motion-compensation coded 380.

The decision made by the selector 390 relates to the macro-block type decision 320. For example, if the macro-block type decision 320 decides that the macro-block should be intra-coded, then the selector should select a form of
10 intra-prediction. For example, if the macro-block type decision 320 decides that the macro-block should be motion compensated, then the selector should select a form of motion compensation. The decisions made by the macro-block type decision 320, the picture-type decision 310, the selector 390, and the selection among one or more intra-prediction techniques 340, are all included within the
15 bit-stream by the entropy coding 250. In addition, the combiner 330 may receive an input from the selector 390 to provide information about the selection made.

Any suitable decoder may be used. An exemplary video decoder 400 for an input bit stream 410 includes an input buffer 420. The input buffer 420 provides a signal to an entropy decoder 430. The entropy decoder 430 provides
20 a signal to an inverse transform and quantization process 440. The inverse transform and quantization process 440 provides a signal to a combiner 450. The combiner 450 provides a signal to a deblocking filter 460 and an intra-prediction module 470. The deblocking filter 460 provides a signal to a reference picture buffer 480. The reference picture buffer 480 provides a signal to a motion
25 compensator 490.

The entropy decoder 430 provides a signal to the motion compensation 490 and the deblocking filter 460. The entropy decoder 430 also provides a signal to a decoder controller 500. The decoder controller is interconnected with the other modules of the decoder 400. The motion compensator 490 provides a

signal to a switch 510. The intra-prediction module 470 provides a signal to the switch 510. The switch 510 selectively provides a signal to the combiner 450. The deblocking filter 460 provides an output picture 520.

Referring to FIG. 3, different frames, or portions thereof, of video are typically encoded using different techniques. One such technique includes the use of picture types generally referred to as I-frames, P-frames, and B-frames. I-frames do not require other video frames to decode. P-frames may use data from a previously transmitted frame to decode. B-frames may use two or more previously transmitted frames to decode. The encoding of the video may likewise be based upon one or more different sized blocks of pixels from within the frame. Also, the encoding of the video may likewise be based upon motion estimation, slices, spatial prediction of blocks, or otherwise between one or more frames. Therefore, in general there is decoder prediction information transmitted with the video bitstream which indicates the type of encoding of the frames, the type of prediction of the frames, the direction(s) of the predictions, which frames are used, motion estimation information between the frames, frame size information, block sizing information within the frame, spatial prediction information, and/or other suitable parameters. Accordingly, the decoder 400 decodes the frames of the video based upon the prediction information provided with the bit-stream by the encoder 200.

Referring to FIG. 4, in existing video coding systems, such as ITU-T H.264 or MPEG-4 AVC, a macro-block (MB) refers specifically to a 16x16 block of pixels. In different video coding systems it is desirable to support the 16x16 macro-block structure of such video coding systems while simultaneously supporting a "super-block" that refers to a group of $N \times N$ such 16x16 macroblocks, where $N \geq 2$. For the case that $N=2$, the super-block would define a 32x32 block of pixels. For example, in the case that $N=4$, the super-block defines a 64x64 block of pixels. The use of common information structure permits effective coding of macro-blocks and super-blocks.

In any exemplary implementation macro-blocks are generally considered to be partitions of super-blocks, just as blocks of 4x4 pixels are generally considered to be partitions of macro-blocks. As such, the four macro-blocks within a super-block may have common characteristics (super-block type) and super-block type includes parameters, such as a macro-block type (prediction mode), a transform type, merge flag, weight parameter, reference indices, mode decision, quantization parameter (QP) and motion vectors. The video encoder encodes the common characteristics that are contained within a super-block. The video decoder decodes the common characteristics that are contained within a super-block. Since the video encoder encodes the super-block type for the super-block instead of encoding it for each of the four macro-blocks within the super-block, it is possible to cut a processes to encode same information for each of the four macro-blocks within the single super-block. This can reduce an amount of encoded data.

To support the super-block at both the encoder and the decoder, images may be divided into super-blocks and processed in a $N \times N$ macro-block group order. The following description discusses a case in which $N = 2$. The intra prediction mode, the motion vectors, the reference indices, and/or the mode decision consistent with macro-blocks may be included with the super-block type. For intra super-block encoding, the macro-block within a super-block type may be restricted to have the same macro-block type and/or the same prediction modes as the super-block. By way of example, macro-block types may include intra-coded 4x4, intra-coded 8x8, intra-coded 16x8, intra-coded 8x16, and/or intra-coded 16x16. That is, prediction modes of the respective macro blocks within the single super block are identical with a prediction mode (macro-block type) encoded for that super block. For an inter-coded super-block, a partition of 32x32 may be used, two partitions of 32x16 may be used, and/or two partitions of 16x32 may be used. In addition, a super-block based skip mode may be used and a super-block based direct mode may be used. A super-block based intra mode may be used and a super-block based merge mode may be used. In the super-block based skip mode, only a prediction index in a motion vector is

encoded and no other information is encoded. In the super-block based direct mode, only a transform coefficient is encoded. In the super-block based merge mode, information of a target super-block is determined among information of neighboring super-blocks. This super-block description is for $N=2$, while other values of N will have additional and larger partitions. Macro-blocks within the same partition preferably have same motion vectors, prediction modes (macro-block type) and references indices as those of super-block type. A "super-block flag" may be included within the bit-stream indicating whether a particular group of macro-blocks is a super-block or not. That is, information indicating whether or not the super-block is divided into macro-blocks is given at a head of data stream for the super-block or given in data stream for a first one of a group of macro-blocks within the super-block. If so, an alternative syntax decoding process may be employed as described below. The super-block flag may also be used to control the transform size and which transform (or transforms) should be used. For example, different transform sizes are used based on whether one super-block is further partitioned into macro-blocks or not.

Improved coding efficiency using a system that includes super-blocks primarily results from two aspects. The first aspect is that the system is capable of providing improved prediction. The second aspect is that the system has a reduction in the syntax necessary to describe a bit-stream. In particular, a significant portion of the super-block system based coding efficiency is the result of a reduction in the syntax signaling.

There are two primary functions that enable the efficient signaling of the super-block. The first function is a flag indicating a super-block and a flag indicating a particular super-block coded block pattern (hereinafter CBP). For each group of macro-blocks, a super-block flag is sent to indicate if the group should be decoded with the alternative, super-block process. If the flag is equal to 1 (or other value), then a super-block CBP flag is additionally sent to indicate whether the super-block has a residual.

The second function is the embedding of super-block information into a first macro-block (or a selected macro-block) of a group of corresponding macro-blocks of the super-block. In ITU-T H.264 and MPEG-4 AVC, macro-block type and other high level information is sent for each macro-block. For the super-block, the system reduces this signaling overhead by mapping super-block information into a macro-block and only transmitting the macro-block header for the first (or selected) macro-block or transmitting the super block header. The macro-block type, motion vector difference (hereinafter referred to as MVD), and reference indices, merge flag, weight parameter, prediction mode, transform size, quantization parameter (QP) of a super-block are compacted and mapped to a 16x16 macro-block, and transmitted at the start of the first macro-block or header of the super-block. An exemplary mapping is illustrated in FIG. 4, where the arrows represent MVD for that partition. For example, a 32x16 super-block is mapped to a 16x8 macro-block, and the super-block information is sent as a 16x8 macro block. At the decoder, the mapping is reversed and the 16x8 macro-block is converted to a 32x16 super-block for reconstruction. Reference indices and reconstructed motion vectors are filled to corresponding macro-blocks within the super-block.

Macro-blocks within a super-block may share additional common characteristics, including macro-block skip, transform size, merge flag, prediction modes, and delta quantization. This common information is also only sent with the first (or selected) macro-block within a super-block (Fig.8 (b)) or header of a super-block (Fig.8 (a)). For the non-first macro-blocks within a super-block, macro-block skip, transform size, delta-quantization, etc., are copied from the other macro-block.

The suitable use of super-blocks result in a bit rate savings for signaling macro-block information. Detailed exemplary syntaxes are illustrated in FIGS. 5A and 5B. The syntaxes are based upon the syntaxes of slice_data in ITU-T H.264 and MPEG-4 AVC but are modified to process macro-block in a group of macro-blocks order. In FIG. 5A and 5B, some common syntaxes such as

slice_data in H.264/AVC are omitted for purposes of clarity. The additional syntax includes slice data semantics.

A superblock_flag specifies whether this group of macro-blocks is a super-block or not.

- 5 The superblock_cbp_1bit specifies whether this super-block has any coefficients. If superblock_cbp_1bit equals 1 means at least 1 macro-block within the superblock has coefficients. In this case, information indicative of cbp is also given in data stream for the at least 1 macro-block within the super-block. If superblock_cbp_1 bit equals 0 means that none of the macro-blocks within the
- 10 superblock has coefficients.

- The superblock_skip_run specifies the number of consecutive skipped super-blocks for which, when decoding a P or SP slice, mb_type of macro-blocks within the super-block may be inferred to be P_Skip and the macro-block type is collectively referred to as P macro-block type, or for which, when decoding a B
- 15 slice, mb_type may be inferred to be B_Skip and the macro-block type is collectively referred to as B macro-block type. The value of superblock_skip_run may be in the range of 0 to PicSizeInSuperblocks – Curr MbAddr, inclusive.

- If superblock_skip_flag is equal to 1 specifies that for the current super-block, when decoding a P or SP slice, mb_type of macro-blocks within the super-
- 20 block may be inferred to be P_Skip and the macro-block type is collectively referred to as P macro-block type, or for which, when decoding a B slice, mb_type may be inferred to be B_Skip and the macro-block type is collectively referred to as B macro-block type. If superblock_skip_flag equal to 0 specifies that the current super-block is not skipped. FIG.9 illustrates an example of
- 25 another syntax.

 The variable superblock_size denotes the number of macro-blocks in the super-block. For example, for a 32x32 super-block, the superblock_size is 4 except at the picture boundary where it may not be a multiple of 32.

The `extract_and_save_superblock_info()` and `copy_macroblock_info_from_superblock()` refer to functions to get the super-block syntaxes, save, and fill them into the macro-blocks.

5 The `nextSuperblockAddress()` returns the start macro-block address of the next super-block.

Yet additional syntaxes refer to macro-block layer semantics. For the macro-block layer, its syntax may be similar to the `macroblock_layer` in H.264/AVC standard with a modification to reading `coded_block_pattern` as illustrated in FIGS. 6A and 6B.

10 The semantics of a `coded_block_pattern` may be defined as follows. `Coded_block_pattern` may specify which of the six 8x8 blocks - luma and chroma - may contain non-zero transform coefficient levels. For macroblocks with prediction mode not equal to `Intra_16x16`, the `coded_block_pattern` is included in the bitstream and the variables `CodedBlockPatternLuma` and
 15 `CodedBlockPatternChroma` may be derived as follows.

$$\text{CodedBlockPatternLuma} = \text{coded\_block\_pattern} \% 16$$

$$\text{CodedBlockPatternChroma} = \text{coded\_block\_pattern} / 16$$

When the `coded_block_pattern` is present, the `CodedBlockPatternLuma` may specify, for each of the four 8x8 luma blocks of the macroblock, one of the
 20 following cases. First, that all transform coefficient levels of the four 4x4 luma blocks in the 8x8 luma block are equal to zero. Second, that one or more transform coefficient levels of one or more of the 4x4 luma blocks in the 8x8 luma block are non-zero valued.

In the case of superblock, when `superblock_cbp_1bit == 0`, the
 25 `coded_block_pattern` of Macroblock may be set to 0.

Any suitable mapping process from the super-block to the macro-block may be used. Referring to FIG. 7, a pseudo code to

extract_and_save_superblock_info and copy_macroblock_info_from_superblock referred in the syntax is illustrated.

```

    extract_and_save_superblock_info ( )
{
5    if (superblock_flag)
    {
        Save a copy of current Macroblock, let's denote it as SMb
        Get superblock MV predictor from neighbor MBs.
        Reconstruct superblock MV by adding superblock MVD and superblock
10    MV predictor
        copy_macroblock_info_from_superblock(0);
    }
}

    copy_macroblock_info_from_superblock(N)
15 {
    if (superblock_flag)
    {
        Copy mb_type, Qp, luma_transform_size_8x8_flag, skip_flag from SMb
        if (mb_type == 16x8 || mb_type == 8x16)
20    Set current Macroblock's mb_type to 16x16
        Otherwise

```

Keep the mb_type same as SMb

Get the MV, reference index at the Nth 8x8 block of SMB, copy them and fill to the current 16x6 macroblock.

}

5 }

The terms and expressions which have been employed in the foregoing specification are used therein as terms of description and not of limitation, and there is no intention, in the use of such terms and expressions, of excluding equivalents of the features shown and described or portions thereof, it being
10 recognized that the scope of the invention is defined and limited only by the claims which follow.

CLAIMS

1. A method for decoding video comprising:
- 5 (a) receiving a super block flag indicating a super block consisting of a plurality of smaller blocks of pixels having shared decoding information with said super block;
- (b) receiving a coded block pattern flag indicating whether said super block has a residual;
- (c) decoding said super block based upon said super block flag and said coded block pattern flag.
- 10 2. The method of claim 1 wherein said super block is 32x32 pixels.
3. The method of claim 2 wherein said of said smaller blocks are 16x16 pixels.
- 15 4. The method of claim 1 wherein said super block is 64x64 pixels.
5. The method of claim 1 wherein said shared decoding information includes at least one of (a) block type; (b) transform type; and (c) motion vector.
- 20 6. The method of claim 5 wherein said decoding information includes at least two of (a), (b), and (c).
- 25 7. The method of claim 6 wherein said decoding information includes at least three of (a), (b), and (c).
8. The method of claim 1 wherein said super block has a block order and said shared decoding information is included together with the first of said smaller blocks of said block order.
- 30

9. The method of claim 1 wherein said shared decoding information is not included in other ones of said smaller blocks of said super block.

10. The method of claim 1 wherein said shared decoding information includes at least one of (a) macro block skip; (b) transform size; and (c) delta quantization.

11. The method of claim 10 wherein said decoding information includes at least two of (a), (b), and (c).

12. The method of claim 11 wherein said decoding information includes at least three of (a), (b), and (c).

13. The method of claim 1 wherein said shared decoding information includes (a) block type; (b) transform type; (c) motion vector; (d) macro block skip; (e) transform size; and (f) delta quantization.

14. A method of decoding video comprising:

(a) receiving a super block consisting of a plurality of smaller blocks of pixels having shared decoding information with said super block in a bit stream of encoded said video;

(b) extracting said shared decoding information from one of said smaller blocks of pixels;

(c) applying said shared decoding information to another one of said smaller blocks of pixels of said super block;

(d) decoding said one of said smaller blocks based upon said shared decoding information;

(e) decoding said another one of said smaller blocks based upon said shared decoding information.

15. The method of claim 14 further receiving a super block flag indicating said super block.

16. The method of claim 16 further comprising receiving a coded block pattern flag indicating whether said super block has a residual.
- 5 17. The method of claim 16 further comprising decoding said super block based upon said super block flag and said coded block pattern flag.
18. The method of claim 14 wherein said super block is 32x32 pixels.
- 10 19. The method of claim 18 wherein said of said smaller blocks are 16x16 pixels.
20. The method of claim 14 wherein said super block is 64x64 pixels.
- 15 21. The method of claim 14 wherein said shared decoding information includes at least one of (a) block type; (b) transform type; and (c) motion vector.
- 20 22. The method of claim 21 wherein said decoding information includes at least two of (a), (b), and (c).
23. The method of claim 22 wherein said decoding information includes at least three of (a), (b), and (c).
- 25 24. The method of claim 14 wherein said super block has a block order and said shared decoding information is included together with the first of said smaller blocks of said block order.
- 30 25. The method of claim 14 wherein said shared decoding information is not included in other ones of said smaller blocks of said super block.

26. The method of claim 14 wherein said shared decoding information includes at least one of (a) macro block skip; (b) transform size; and (c) delta quantization.

5 27. The method of claim 26 wherein said decoding information includes at least two of (a), (b), and (c).

28. The method of claim 27 wherein said decoding information includes at least three of (a), (b), and (c).

10

29. The method of claim 14 wherein said shared decoding information includes (a) block type; (b) transform type; (c) motion vector; (d) macro block skip; (e) transform size; and (f) delta quantization.

FIG. 1

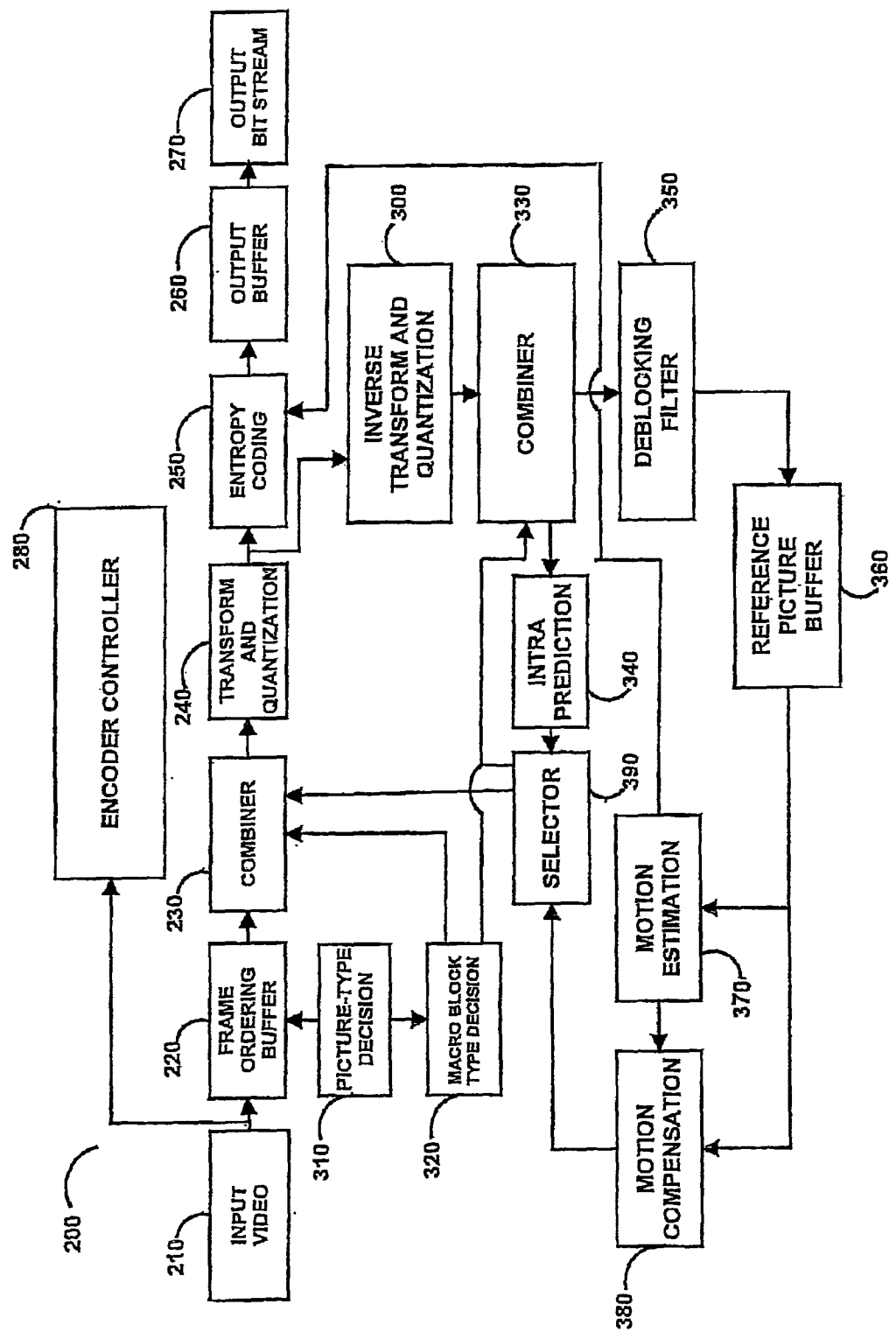


FIG. 2

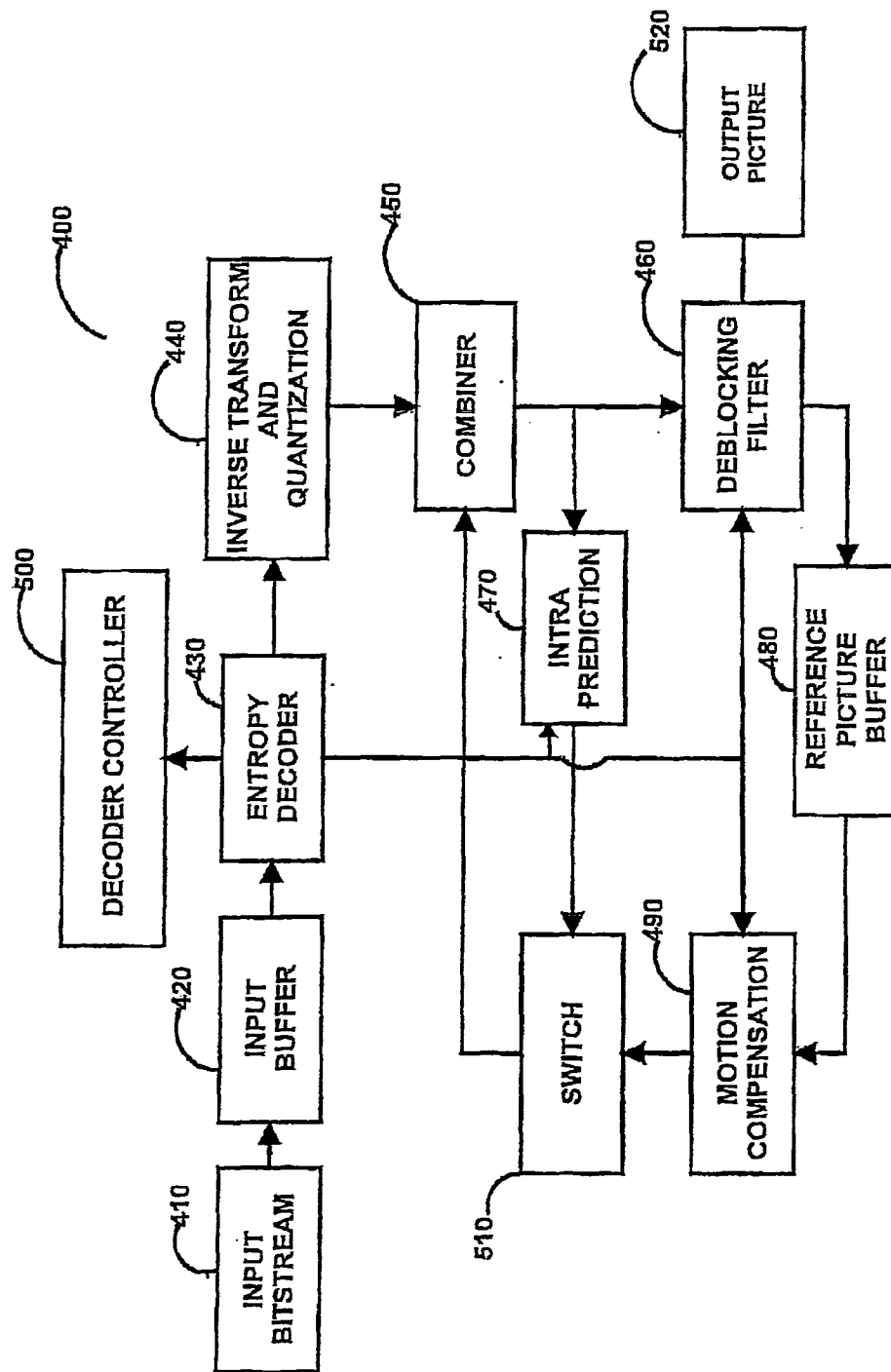


FIG. 3

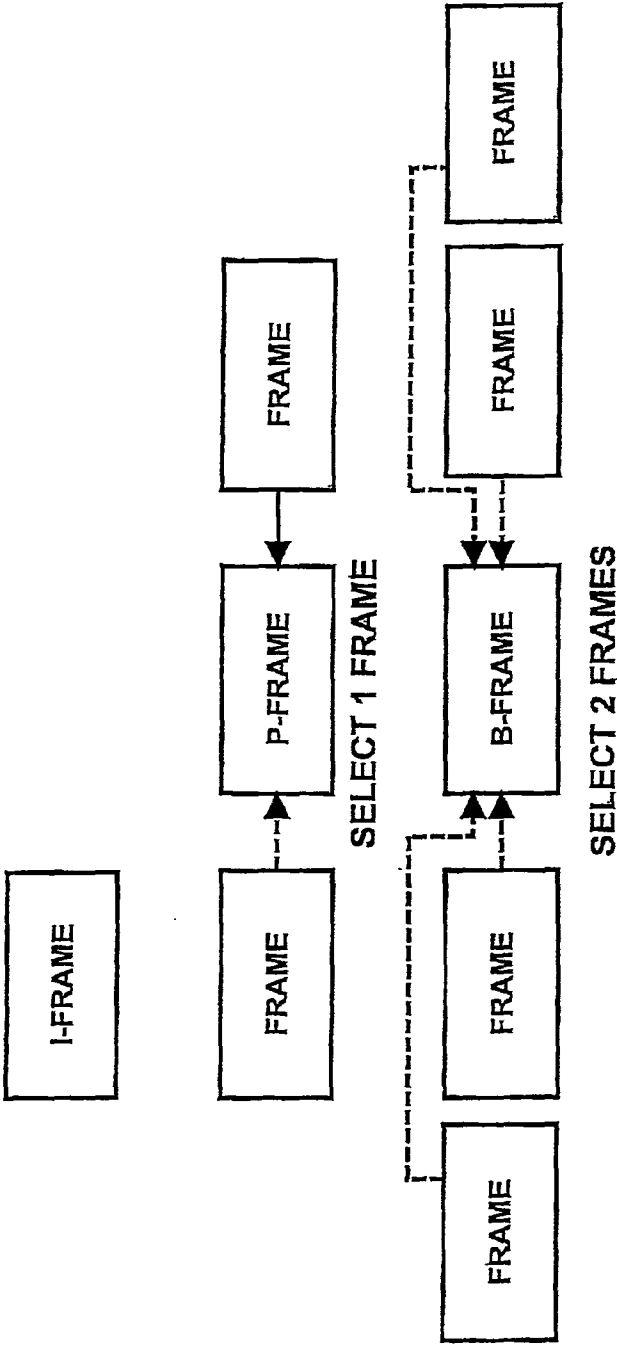


FIG. 4

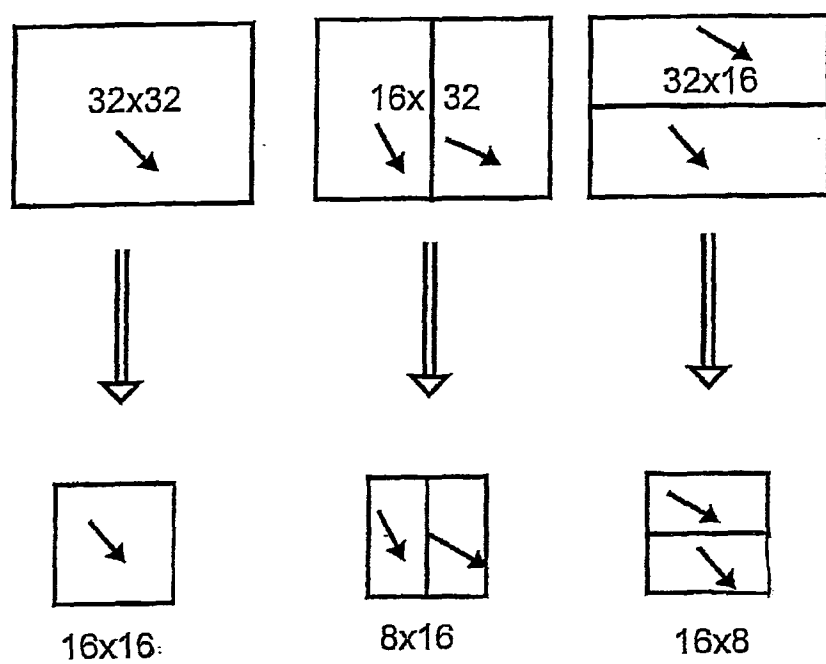


FIG. 5A

slice_data() {	C	Descriptor
.....		
moreDataFlag = 1		
prevMbSkipped = 0		
superblock_flag = 0		
do { // process a group of macro-blocks		
if(slice_type != I && slice_type != SI) {		
superblock_flag	2	u(1) ae(v)
if (superblock_flag) {		
superblock_cbp_1bit	2	u(1) ae(v)
if(lentropy_coding_mode_flag) {		
superblock_skip_run	2	ue(v)
prevMbSkipped = (superblock_skip_run > 0)		
for(i=0; i<superblock_skip_run; i++)		
CurrMbAddr = NextSuperblockAddress(		
CurrMbAddr)		
moreDataFlag = more_rbsp_data()		
} else {		

FIG. 5B

superblock_skip_flag	2	u(1) ae(v)
moreDataFlag = !superblock_skip_flag		
}		
for (mb=0; mb < superblock_size && moreDataFlag; mb++) {		
if (mb==0 is_intra) {		
macroblock_layer();		
extract_and_save_superblock_info();		
}		
else {		
copy_macroblock_info_from_superblock(mb);		
if (superblock_cbp_1bit != 0) {		
coded_block_pattern	2	me(v) ae(v)
residual()		
}		
}		
} // end for loop		
}		
else { // not super-block		
for (mb=0; mb < num_mb_in_superblock; mb++) {		
// process as a regular macro-block as usual		
...		
macroblock_layer();		
....		
}		
CurrMbAddr = NextSuperblockAddress(CurrMbAddr)		
} while(moreDataFlag)		
}		

FIG. 6A

macroblock_layer() {	C	Descriptor
mb_type	2	ue(v) ae(v)
if(mb_type == I_PCM) {		
while(lbyte_aligned())		
pcm_alignment_zero_bit	2	f(1)
for(i = 0; i < 256; i++)		
pcm_sample_luma[i]	2	u(v)
for(i = 0; i < 2 * MbWidthC * MbHeightC; i++)		
pcm_sample_chroma[i]	2	u(v)
} else {		
noSubMbPartSizeLessThan8x8Flag = 1		
if(mb_type != I_NxN && MbPartPredMode(mb_type, 0) != Intra_16x16 && NumMbPart(mb_type) == 4) {		
sub_mb_pred(mb_type)	2	
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8) {		
if(NumSubMbPart(sub_mb_type[mbPartIdx]) > 1)		
noSubMbPartSizeLessThan8x8Flag = 0		
} else if(ldirect_8x8_inference_flag)		
noSubMbPartSizeLessThan8x8Flag = 0		
} else {		
if(transform_8x8_mode_flag && mb_type == I_NxN)		
transform_size_8x8_flag	2	u(1) ae(v)

FIG. 6B

mb_pred(mb_type)	2	
}		
if(MbPartPredMode(mb_type, 0) != Intra_16x16 && superblock_cbp_1bit != 0) {		
coded_block_pattern	2	me(v) ae(v)
if(CodedBlockPatternLuma > 0 && transform_8x8_mode_flag && mb_type != I_NxN && noSubMbPartSizeLessThan8x8Flag && (mb_type != B_Direct_16x16 !irect_8x8_inference_flag))		
transform_size_8x8_flag	2	u(1) ae(v)
}		
if(CodedBlockPatternLuma > 0 CodedBlockPatternChroma > 0 MbPartPredMode(mb_type, 0) == Intra_16x16) {		
mb_qp_delta	2	se(v) ae(v)
residual()	3 4	
}		
}		
}		

FIG. 7

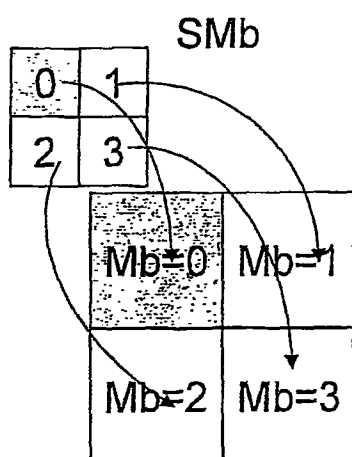


FIG. 8

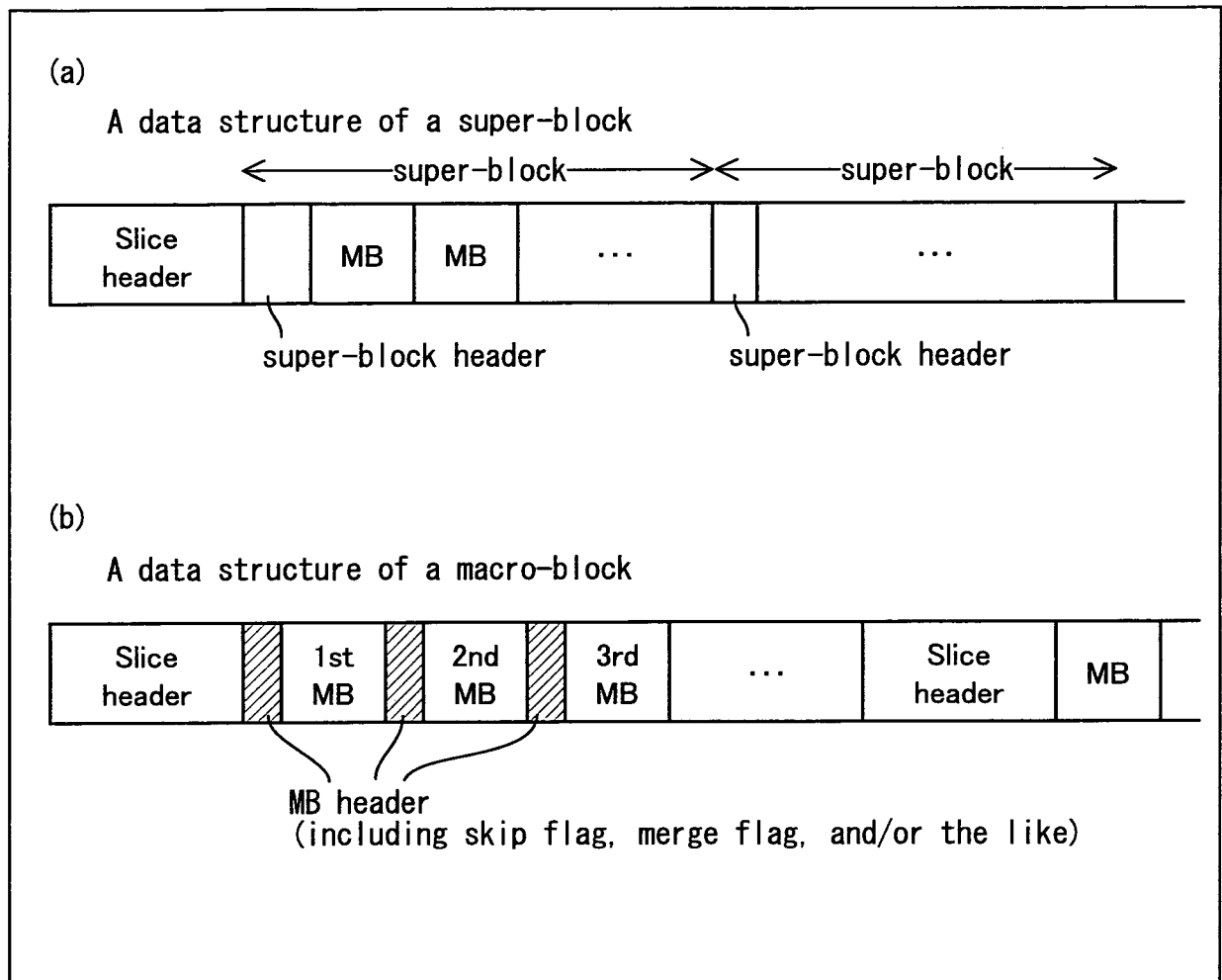


FIG. 9

An example of syntax for a super-block header

```

slice_header() {
    .....
}
slice_data() {
    .....
    while (! end of slice) {
        super_block()
    }
}

super_block() {
    superblock_flag
    if (superblock_flag) {
        if (slice_type != Intra) {
            skip_flag
        }
        if (skip_flag) {
            macroblock(skip_flag, superblock_flag)
        }
        else {
            super_block_cbp_1bit
            if (slice_type != Intra) {
                merge_flag
            }
            if (merge_flag) {
                macroblock(merge_flag, super_block_cbp_1bit)
            }
            else {
                pred_mode
                if (pred_mode == Intra) {
                    intra_split_flag
                    if (! intra_split_flag) {
                        macroblock(intra, super_block_cbp_1bit)
                    }
                    else {
                        macroblock (intra 1,super_block_cbp_1bit)
                        macroblock (intra 2,super_block_cbp_1bit)
                        .
                        .
                        macroblock (intra X,super_block_cbp_1bit)
                    }
                }
                else if (pred_mode == Direct) {
                    macroblock (direct,super_block_cbp_1bit)
                }
                else { /* pred_mode == inter */
                    motion_vector
                    reference_indices
                    weight_parameter
                    while (partition number) {
                        macroblock (Inter,super_block_cbp_1bit)
                    }
                }
            }
        }
    }
    else { /* ! superblock */
        while macroblock()
    }
}

```

pieces of information indicative of skip, merge, super_block_cbp_1bit, intra, and/or the like are inherited in data stream for macroblock()

divide

pieces of information indicative of skip, merge, intra, inter and/or the like are decoded for macroblock() like as H.264

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/059455

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. H04N7/26 (2006.01) i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. H04N7/26

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

Published examined utility model applications of Japan 1922-1996
 Published unexamined utility model applications of Japan 1971-2011
 Registered utility model specifications of Japan 1996-2011
 Published registered utility model applications of Japan 1994-2011

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 2010-502102 A (THOMSON LISING), 2010.01.21, page 18, line 31 to page 27, line 44, page 28, line 17 to 45, page 29, line 18 to 44, FIG.4 to 6, 8, 10 & US 2009/0196517 A1 & EP 2055108 A2 & WO 2008/027192 A2 & KR 10-2009-0057225 A & CN 101507280 A	1-29
A	Marta Karczewicz et al., Video coding using motion compensation with polynomial motion vector fields, Signal Processing: Image Communication, 1997.07, Vol.10, No.1-3, p.63-91	1-29



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

12.07.2011

Date of mailing of the international search report

19.07.2011

Name and mailing address of the ISA/JP

Japan Patent Office

3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan

Authorized officer

Takayuki Hatanaka

Telephone No. +81-3-3581-1101 Ext. 3541

5C

9468

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/059455

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Huipin Zhang and Frank Bossen, Region-Based Coding of Motion Fields for Low-Bitrate Video Compression, 2004 International Conference on Image Processing(ICIP '04), 2004.10.24, Vol.2, p.1117-1120	1-29
A	Cixun Zhang et al., Video Coding Using Pruned Transforms and Interleaving of Multiple Blocks, IEEE International Conference on Acoustics, Speech and Signal Processing, 2008.(ICASSP 2008), IEEE, 2008.03.31, p.1049-1052	1-29

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/059455

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

The "special technical features" of claim 1-13 relates to [decoding a super block based upon a super block flag and a coded block pattern flag] while the "special technical features" of claim 14-29 relates to [extracting shared decoding information from one of a plurality of smaller blocks of pixels; applying said shared decoding information to another one of said smaller blocks of pixels of a super block; decoding said one of said smaller blocks based upon said shared decoding information; decoding said another one of said smaller blocks based upon said shared decoding information]. There is no technical relationship among those inventions involving one or more of the same or corresponding technical features. Therefore, these groups of inventions are not so linked as to form a single general inventive concept.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☒ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.