(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 June 2017 (01.06.2017)

(10) International Publication Number
WO 2017/088170 A1

(51) International Patent Classification:
H04N 19/96 (2014.01)

(21) International Application Number:
PCT/CN2015/095775

(22) International Filing Date:
27 November 2015 (27.11.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **MEDIATEK INC.** [CN/CN]; No.1, Dusing Road 1st, Science-Based Industrial Park, Hsin-Chu, Taiwan 300 (CN).

(72) Inventors: **HSIANG, Shih-Ta**; 6F., No.50, Ln. 85, Baoqiao Rd., Xindian Dist., New Taipei City, Taiwan 231 (CN). **AN, Jicheng**; No. 44, Zhanlanguan Road, Xicheng District, Beijing 100037 (CN).

(74) Agent: **BEIJING SANYOU INTELLECTUAL PROPERTY AGENCY LTD.**; 16th Fl., Block A, Corporate Square, No.35 Jinrong Street, Beijing 100033 (CN).

(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, BN, 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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, 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, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*

(54) Title: ENTROPY CODING THE BINARY TREE BLOCK PARTITIONING STRUCTURE

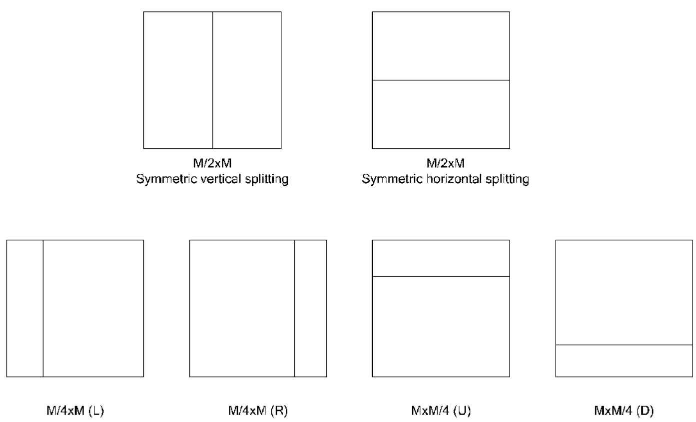

Fig. 3

(57) Abstract: A binary tree block partitioning structure is proposed. The proposed flexible block partitioning structure plays an important role in the substantial performance gains relative the HEVC.

WO 2017/088170 A1

ENTROPY CODING THE BINARY TREE BLOCK PARTITIONING STRUCTURE

TECHNICAL FIELD

[0001] The invention relates generally to video processing. In particular, the present invention relates to methods and apparatuses for block partitioning structure in video and image coding.

BACKGROUND

[0002] The High Efficiency Video Coding (HEVC) standard is the most recent joint video project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations, working together in a partnership known as the Joint Collaborative Team on Video Coding (JCT-VC).

[0003] In HEVC, one slice is partitioned into multiple coding tree units (CTU). In main profile, the minimum and the maximum sizes of CTU are specified by the syntax elements in the sequence parameter set (SPS) among the sizes of 8x8, 16x16, 32x32, and 64x64. Inside a slice, a raster scan method is used for processing the CTU.

[0004] The CTU is further partitioned into multiple coding units (CU) to adapt to various local characteristics. A quadtree denoted as the coding tree is used to partition the CTU into multiple CUs. Let CTU size be MxM where M is one of the values of 64, 32, or 16. The CTU can be a single CU or can be split into four smaller units of equal sizes of M/2xM/2, which are nodes of coding tree. If units are leaf nodes of coding tree, the units become CUs. Otherwise, the quadtree splitting process can be iterated until the size for a node reaches a minimum allowed CU size specified in the SPS. This representation results in a recursive structure specified by a coding tree as shown in Fig. 1. The solid lines indicate CU boundaries. The decision whether to code a picture area using interpicture (temporal) or intrapicture (spatial) prediction is made at the CU level. Since the minimum CU size can be 8x8, the minimum granularity for switching different basic prediction type is 8x8.

[0005] One or more prediction units (PU) are specified for each CU. Coupled with the CU, the PU works as a basic representative block for sharing the prediction information. Inside one PU, the same prediction process is applied and the relevant information is transmitted to the decoder on a PU basis. A CU can be split into one, two, or four PUs according to the PU splitting type. HEVC defines eight shapes for splitting a CU into PU as shown in Fig. 2. Unlike the CU, the PU may only be split once.

[0006] After obtaining the residual block by prediction process based on PU splitting type, a CU can be partitioned into transform units (TU) according to another quadtree structure which is analogous to the coding tree for the CU as shown in Fig. 1. The solid lines indicate CU boundaries and dotted lines indicate TU boundaries. The TU is a basic representative block having residual or transform coefficients for applying the integer transform and quantization. For each TU, one integer transform having the same size to the TU is applied to obtain residual coefficients. These coefficients are transmitted to the decoder after quantization on a TU basis.

[0007] The terms coding tree block (CTB), coding block (CB), prediction block (PB), and transform block (TB) are defined to specify the 2-D sample array of one color component associated with CTU, CU, PU, and TU, respectively. Thus, a CTU consists of one luma CTB, two chroma CTBs, and associated syntax elements. A similar relationship is valide for CU, PU, and TU.

[0008] The tree partitioning is generally applied simultaneously to both luma and chroma, although exceptions apply when certain minimum sizes are reached for chroma.

[0009] However, due to some restrictions in the current HEVC block partitioning structure, e.g., the quadtree based partitioning from CTU to CU and from CU to TU, the limited depth of PU partitioning, the coding efficiency is still not sufficient to cope with the ever increasing demands for storage and transmission of video content.

SUMMARY

[0010] In light of the previously described problems, there exists a need for an apparatus and method, in which a more flexible block partitioning structure is used.

[0011] A binary tree structure is proposed for partitioning a unit into multiple smaller units such as partitioning a slice into CTUs, a CTU into CUs, a CU into PUs, or a CU into TUs, and so on.

[0012] The binary tree structure can be combined with quadtree structure, which is a quadtree plus binary tree (QTBT) structure. In the QTBT structure, the root unit is firstly partitioned by a quadtree structure, then the leaf nodes of the quadtree is further partitioned by a binary tree structure. Some parameters for restriction of the quadtree and binary tree can be defined in high level such as in SPS.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention can be more fully understood by reading the subsequent detailed description and examples with references made to the accompanying drawings, wherein:

[0001] Fig. 1 illustrates the subdivision of a CTU into CUs and TUs (left) and its corresponding quadtree (right) in HEVC.

[0002] Fig. 2 illustrates the types for splitting a CU into PUs in HEVC.

[0003] Fig. 3 illustrates the possible splitting types in the proposed binary tree.

[0004] Fig. 4 illustrates an example of block partitioning (left) and its corresponding binary tree (right).

[0005] Fig. 5 illustrates an example of block partitioning (left) and its corresponding quadtree plus binary tree (right).

DETAILED DESCRIPTION

[0006] The following description is of the best-contemplated mode of carrying out the invention. This description is made for the purpose of illustrating the general principles of the invention and should not be taken in a limiting sense. The scope of the invention is best determined by reference to the appended claims.

[0007] A binary tree block partitioning structure is proposed. In the proposed binary tree partitioning structure, a block can be recursively split into two smaller blocks. There are kinds of splitting types as shown in Fig. 3. The most efficient and simplest ones are the symmetric horizontal and vertical split as shown in the top two splitting types in Fig. 3. Therefore, as one embodiment, we only use these two splitting types. For a given block of size MxN, a flag signals whether it is split into two smaller blocks. If yes, another syntax element is signaled to indicate which splitting type is used. If the horizontal splitting is used then it is split into two blocks of size MxN/2, otherwise if the vertical splitting is used then it is split into two blocks of size M/2xN. The binary tree splitting process can be iterated until the size (width or height) for a splitting block reaches a minimum allowed block size (width or height) that can be defined in high level syntax such as SPS. Since the binary tree has two splitting types horizontal and vertical, so the minimum allowed block width and height should be both indicated. Not horizontal splitting is implicit when splitting would result in a block height smaller than the indicated minimum. Not vertical splitting is implicit when splitting would result in a block width smaller than the indicated minimum. Fig. 4 illustrates an example of block partitioning (left) and its corresponding binary tree (right). In each splitting (i.e., non-leaf) node of the binary tree, one flag indicates which splitting type (horizontal or vertical) is used, 0 indicates horizontal splitting and 1 indicates vertical splitting.

[0008] The proposed binary tree structure can be used for partitioning a block into multiple smaller blocks such as partitioning a slice into CTUs, a CTU into CUs, a CU into PUs, or a CU into TUs, and so on. As one embodiment, we use the binary tree for partitioning a CTU into CUs, i.e., the root node of the binary tree is a CTU and the leaf node of the binary tree is CU, the leaf nodes are further processed by prediction and transform coding. And for simplification as another embodiment there is no further partitioning from CU to PU or from CU to TU. That means CU equal to PU equal to TU, therefore, it is also to say that the leaf node of the binary tree is the basic unit for prediction and transform coding.

[0009] Binary tree structure is more flexible than quadtree structure, since much more partition shapes can be supported which is also the source of coding efficiency improvement. However, the encoding complexity will also increase in order to select the best partition shape. In order to balance the complexity and coding efficiency, it is proposed to combine the quadtree and binary tree structure, which is called as quadtree plus binary tree (QTBT) structure. In the proposed QTBT structure, a block is firstly partitioned by a quadtree structure, the quadtree splitting can be iterated until the size for a splitting block reaches the minimum allowed quadtree leaf node size. If the leaf quadtree block is not larger than the maximum allowed binary tree root node size, it can be further partitioned by a binary tree structure, the binary tree splitting can be iterated until the size (width or height) for a splitting block reaches the minimum allowed binary tree leaf node size (width or height) or the binary tree depth reaches the maximum allowed binary tree depth. In the QTBT structure, the minimum allowed quadtree leaf node size, the maximum allowed binary tree root node size, the minimum allowed binary tree leaf node width and height, and the maximum allowed binary tree depth can be indicated in the high level syntax such as in SPS. Fig. 5 illustrates an example of block partitioning (left) and its corresponding QTBT (right). The solid lines indicate quadtree splitting and dotted lines indicate binary tree splitting. In each splitting (i.e., non-leaf) node of the binary tree, one flag indicates which splitting type (horizontal or vertical) is used, 0 indicates horizontal splitting and 1 indicates vertical splitting.

[0010] The proposed QTBT structure can be used for partitioning a block into multiple smaller blocks such as partitioning a slice into CTUs, a CTU into CUs, a CU into PUs, or a CU into TUs, and so on. As one embodiment, we use the QTBT for partitioning a CTU into CUs, i.e., the root node of the QTBT is a CTU which is partitioned into multiple CUs by a QTBT structure, and the CUs are further processed by prediction and transform coding. And for simplification as another embodiment there is no further partitioning from CU to PU or from CU to TU. That means CU equal to PU equal to TU, therefore, it is also to say that the leaf node of the QTBT structure is the basic unit for prediction and transform.

[0011] For the proposed QTBT structure, we can take a specific example here. We have a CTU with size 128x128, and the minimum allowed quadtree leaf node size equal to 16x16, the maximum allowed binary tree root node size equal to 64x64, the minimum allowed binary tree leaf node width and height both equal to 4, and the maximum allowed binary tree depth equal to 4. Firstly, the CTU is partitioned by a quadtree structure and the leaf quadtree unit may have size from 16x16 (minimum allowed quadtree leaf node size) to 128x128 (equal to CTU size, i.e., no split), if the leaf quadtree unit is 128x128 it cannot be further split by binary tree since the size exceeds the maximum allowed binary tree root node size 64x64, otherwise the leaf quadtree unit can be further split by binary tree, the leaf quadtree unit which is also the root binary tree unit has binary tree depth as 0. When the binary tree depth reaches 4 (the indicated maximum), not splitting is implicit, when the binary tree node has width equal to 4, not horizontal splitting is implicit, and when the binary tree node has height equal to 4, not vertical splitting is implicit. The leaf nodes of the QTBT are further processed by prediction (intrapicture or interpicture) and transform coding.

[0012] The proposed tree structure is applied separately to luma and chroma for I slice, and applied simultaneously to both luma and chroma (except when certain minimum sizes are reached for chroma) for P and B slice. That is to say that, in I slice, the luma CTB has its QTBT-structured block partitioning, and the two chroma CTBs has another QTBT-structured block partitioning, wherein as another embodiment, the two chroma CTBs can also have their own QTBT-structured block partitionings.

[0013] The representation of the proposed binary tree structure can be entropy coded by the context-based adaptive binary arithmetic coding (CABAC) mode. The context modeling can consider information on the binary tree depths of the neighboring blocks and the current block subject to signaling the split decision. The contextual information can further consider the size and shape of the current block. When combined with the quadtree structure, the context modeling can further consider information on the quadtree depths of the neighboring blocks and the current block. A binary syntax element bt_split_flag is coded to signal if the current block is to be further split or not. If the syntax element bt_split_flag is true, a syntax element bt_split_mode is next coded to indicate the split/partitioning mode to be adopted by the current block. In one embodiment for the proposed QTBT structure with the two binary split modes, context modeling for entropy coding the syntax element bt_split_flag is conditioned on the number of the neighboring blocks, consisting of the top block and left block, with the higher combined QTBT depth than the current block. The combined QTBT depth of a block is defined to be equal to the sum of its quadtree depth scaled by 2 and its binary tree depth. Context modeling for entropy coding the syntax element bt_split_mode is conditioned on the shape of the current block. **Tables 1** and **2** provide the example mapping tables for context selection for such an embodiment for coding syntax elements bt_split_flag and bt_split_mode, respectively. In another embodiment, the context modeling is further conditioned on the binary tree depth of the current block. **Tables 3** and **4** provide the example mapping tables for context selection for such an embodiment for coding syntax elements bt_split_flag and bt_split_mode, respectively, where “W” and “H” denotes the width and height of the current block. In another embodiment, context modeling is further conditioned on the quadtree depth of the current block. In another embodiment, context modeling for different quadtree depths or binary tree depths can share the same set of modeling contexts.

Table 1

ctxInc	0	1	2
# of neighboring CUs with higher combined QTBT depth	0	1	2

Table2

ctxInc	0	1	2
block shape	square	width > height	width < height

Table 3

binary tree depth	0			1		
ctxInc	0	1	2	3	4	5
# of neighboring CUs with higher combined QTBT depth	0	1	2	0	1	2
binary tree depth	2			>=3		
ctxInc	6	7	8	9	10	11
# of neighboring CUs with higher combined QTBT depth	0	1	2	0	1	2

Table 4

binary tree depth	0			1		
ctxInc	0				1	2
block shape	square				W > H	W < H
binary tree depth	2			>=3		
ctxInc	3	4	5	6	7	8
block shape	square	W > H	W < H	square	W > H	W < H

[0014] The methods described above can be used in a video encoder as well as in a video decoder. Embodiments of the methods according to the present invention as described above may be implemented in various hardware, software codes, or a combination of both. For example, an embodiment of the present invention can be a circuit integrated into a video compression chip or program codes integrated into video compression software to perform the processing described herein. An embodiment of the present invention may also be program codes to be executed on a Digital Signal Processor (DSP) to perform the processing described herein. The invention may also involve a number of functions to be performed by a computer processor, a digital signal processor, a microprocessor, or field programmable gate array (FPGA). These processors can be configured to perform particular tasks according to the invention, by executing machine-readable software code or firmware code that defines the particular methods embodied by the invention. The software code or firmware codes may be developed in different programming languages and different format or style. The software code may also be compiled for different target platform. However, different code formats, styles and languages of software codes and other means of configuring code to perform the tasks in accordance with the invention will not depart from the spirit and scope of the invention.

[0015] The invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described examples are to be considered in all respects only as illustrative and not restrictive. To the contrary, it is intended to cover various modifications and similar arrangements (as would be apparent to those skilled in the art). Therefore, the scope of the appended claims should be accorded the broadest interpretation so as to encompass all such modifications and similar arrangements.

CLAIMS

1. A video or image coding method, the method comprising a binary tree block partitioning structure possibly combined with a quadtree block partitioning structure, an arithmetic coding engine, a plurality of probability models built up from the different contextual conditions, and a entropy coder for coding the said block partitioning structure using context-based adaptive binary arithmetic coding conditioned on the contextual information related to the binary tree depths of the neighboring blocks and the current block and the size and shape of the current block.

2. The method according to Claim 1, wherein the entropy coder comprising a binary syntax element, indicated by bt_split_flag, to signal if the current block is to be further split or not and entropy coded conditioned on the number of the neighboring blocks with the higher combined QTBT depth than the current block.

3. The method according to Claim 1, wherein the entropy coder comprising a syntax element, indicated by bt_split_mode, to signal the split/partitioning mode to be adopted by the current block and entropy coded conditioned on the shape of the current block.

4. The method as claimed in claim 3, wherein the shape of the current block is classified by the relationship between width and height of the block.

5. The method according to Claim 2, wherein the syntax element bt_split_flag is entropy coded further conditioned on the binary tree depth of the current block.

6. The method according to Claim 3, wherein the syntax element bt_split_mode is entropy coded further conditioned on the binary tree depth of the current block.

7. The method as claimed in Claim 2, wherein the combined QTBT depth is replaced by the binary tree depth.

8. The method as claimed in Claim 5, wherein the binary tree depth is replaced by the combined QTBT depth.

9. The method as claimed in Claim 6, wherein the binary tree depth is replaced by the combined QTBT depth.

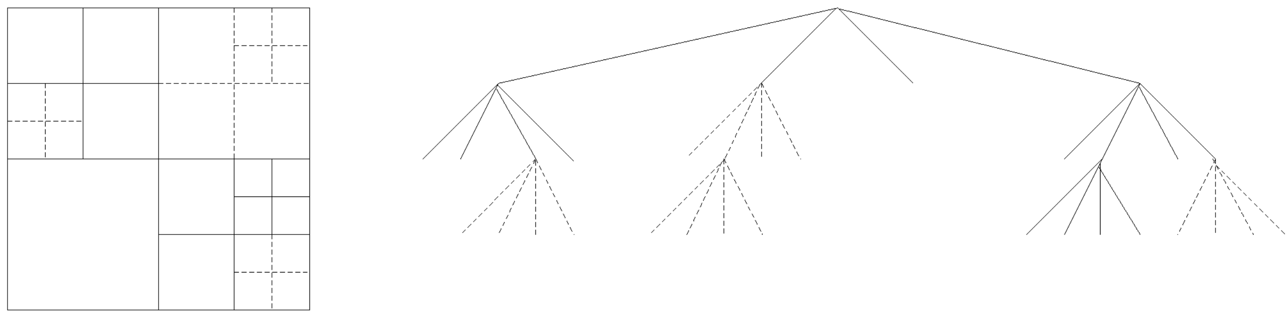

Fig. 1

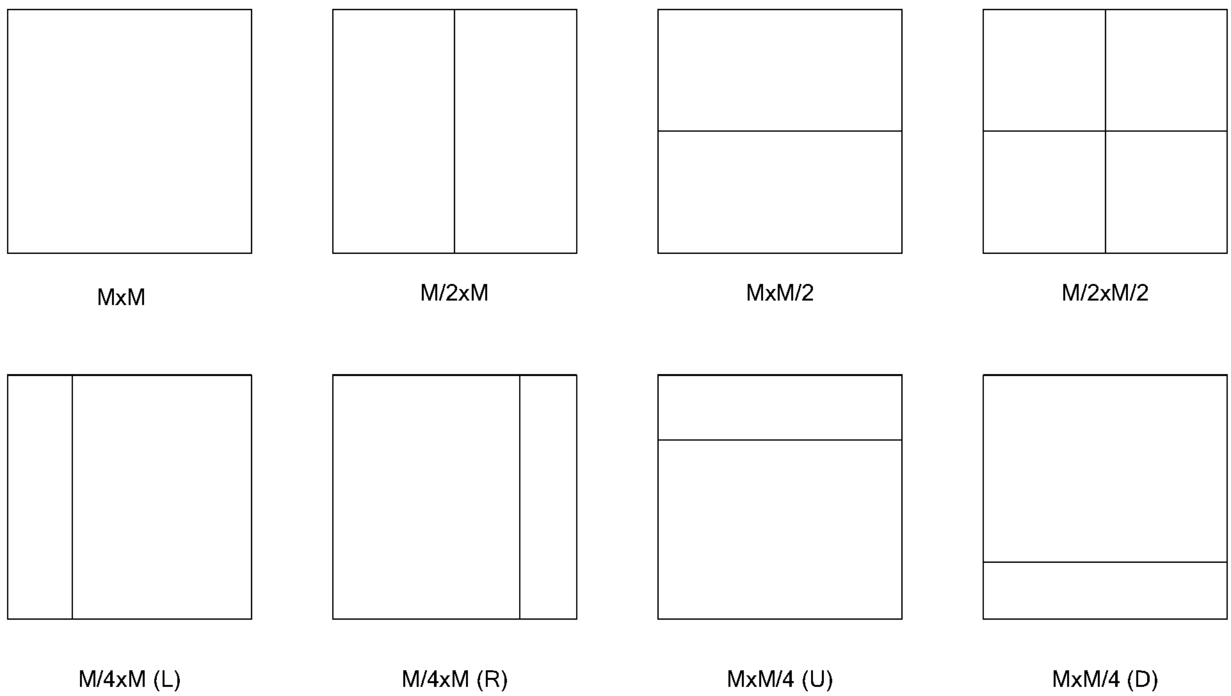

Fig. 2

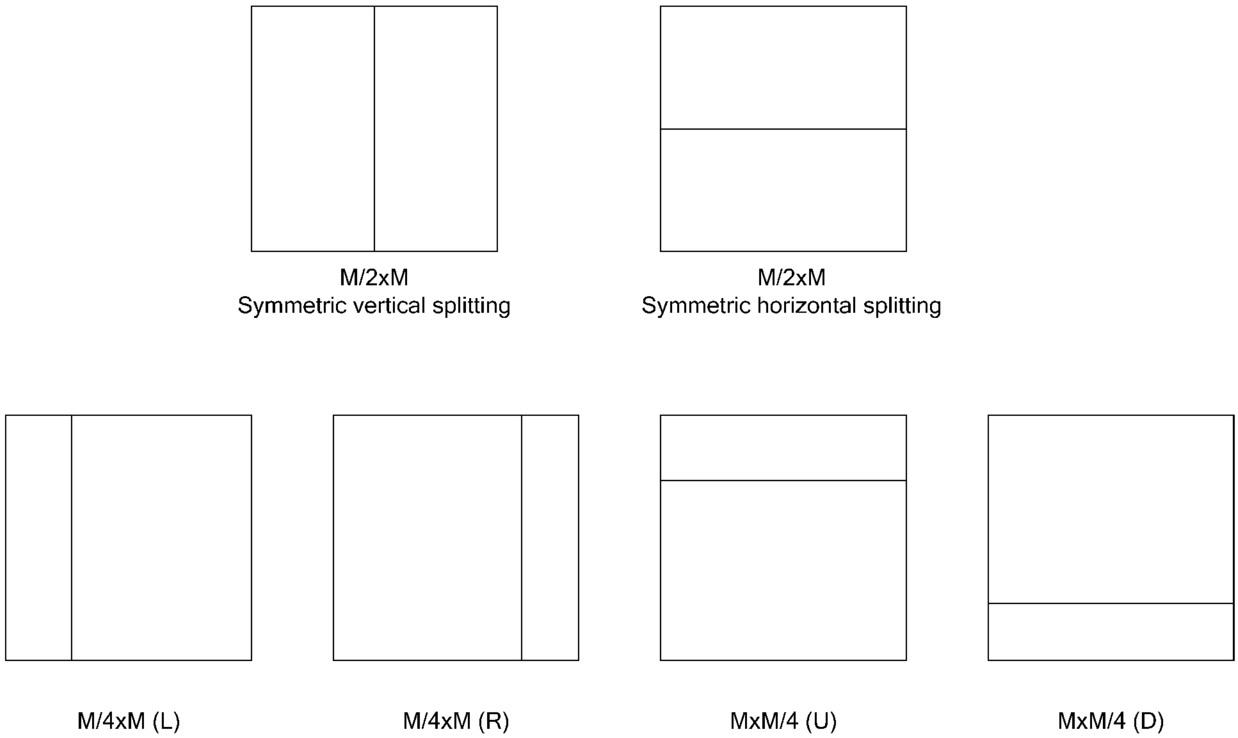
M/2xM
Symmetric vertical splitting
M/2xM
Symmetric horizontal splitting
M/4xM (L)
M/4xM (R)
MxM/4 (U)
MxM/4 (D)

Fig. 3

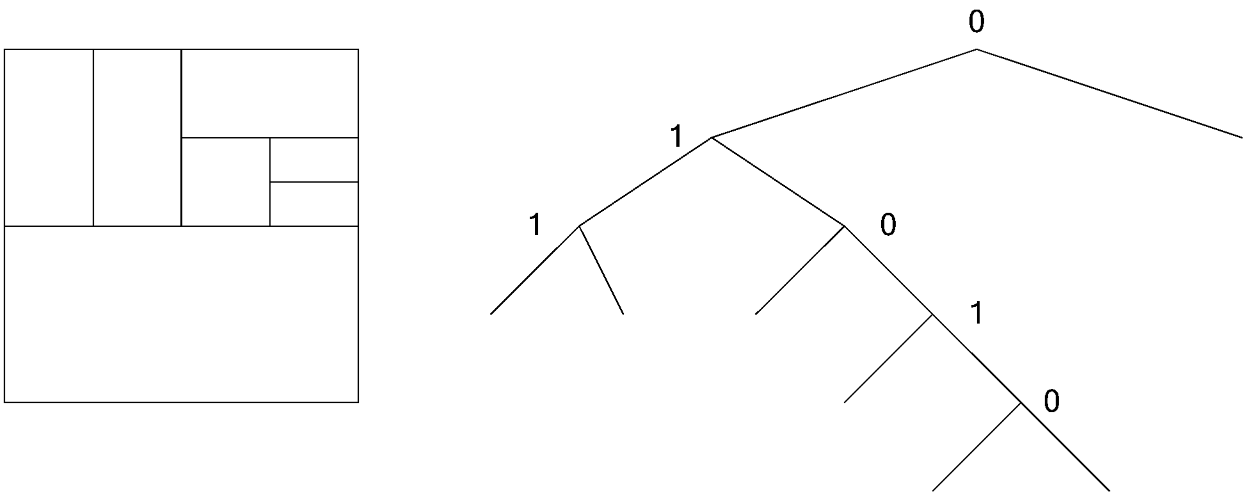
0
1
1
0
1
0

Fig. 4

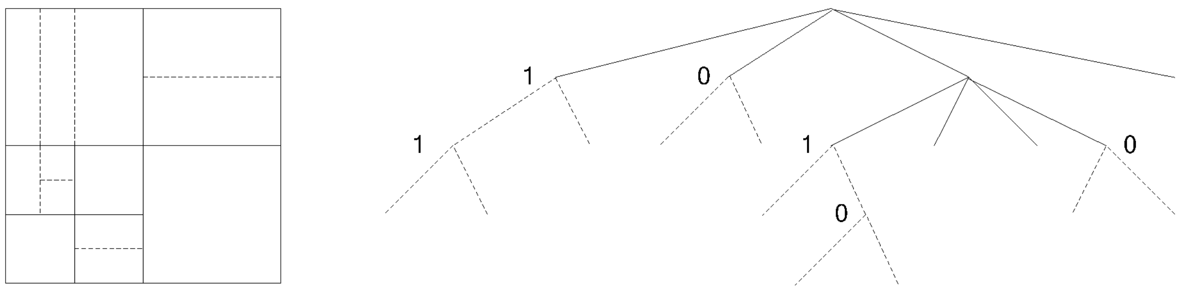

Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2015/095775

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/96(2014.01)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)

H04N

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

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

CNABS,CNTXT,CNKI,VEN,USTXT: binary tree, quadtree, quad tree, tree, combine, entropy, encode, code, context, syntax element, depth, size, shape

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 104883569 A (SAMSUNG ELECTRONICS CO., LTD.) 02 September 2015 (2015-09-02) the description paragraphs [0042]-[0074], figures 10-12	1-9
A	CN 103503461 A (MITSUBISHI ELECTRIC CORP.) 08 January 2014 (2014-01-08) the whole document	1-9
A	CN 102801976 A (SHANDONG ACAD SCI INFORMATION INST.) 28 November 2012 (2012-11-28) the whole document	1-9
A	US 2003202602 A1 (APOSTOLOPOULOS J. G. ET AL.) 30 October 2003 (2003-10-30) the whole document	1-9

☐ 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	Date of mailing of the international search report
11 August 2016	**18 August 2016**

Name and mailing address of the ISA/CN	Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China**	**LUO,Xinyao**
Facsimile No. **(86-10)62019451**	Telephone No. **(86-10)62089576**

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 104883569 A	02 September 2015	JP 2015167380 A	24 September 2015
		KR 20150052818 A	14 May 2015
		AU 2012278484 B2	05 May 2016
		KR 20140085395 A	07 July 2014
		US 2015195539 A1	09 July 2015
		CN 104994384 A	21 October 2015
		JP 2014521244 A	25 August 2014
		JP 2015159600 A	03 September 2015
		KR 20150000853 A	05 January 2015
		KR 20150052817 A	14 May 2015
		KR 20130004182 A	09 January 2013
		US 2014219335 A1	07 August 2014
		CN 104980746 A	14 October 2015
		JP 5917746 B2	18 May 2016
		WO 2013005968 A3	14 March 2013
		US 2015139333 A1	21 May 2015
		AU 2012278484 A1	06 February 2014
		JP 5917745 B2	18 May 2016
		JP 2015159601 A	03 September 2015
		RU 2014103482 A	10 August 2015
		WO 2013005968 A2	10 January 2013
		CA 2840887 A1	10 January 2013
		US 2015195585 A1	09 July 2015
		JP 5917744 B2	18 May 2016
		CN 104853200 A	19 August 2015
		KR 20150052816 A	14 May 2015
		JP 2015167379 A	24 September 2015
		US 2015195586 A1	09 July 2015
		CN 103765887 A	30 April 2014
		EP 2728873 A2	07 May 2014
		CN 104994385 A	21 October 2015
		MX 2014000162 A	28 February 2014
		MX 332580 B	20 August 2015
		MX 332245 B	10 August 2015
		MX 327686 B	09 February 2015
		MX 332579 B	20 August 2015
		ZA 201502922 A	29 June 2016
		ZA 201400767 A	24 February 2016
		VN 37885 A	26 May 2014
		SG 196419 A1	13 February 2014
		MX 334250 B	22 October 2015
		KR 1613978 B1	20 April 2016
		KR 1552909 B1	15 September 2015
		KR 1464978 B1	26 November 2014
		KR 1579114 B1	24 December 2015
		KR 1613977 B1	20 April 2016
CN 103503461 A	08 January 2014	US 8494290 B2	23 July 2013
		EP 2705665 A1	12 March 2014
		TW I504209 B	11 October 2015
		US 2012281928 A1	08 November 2012

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2015/095775

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
		WO 2012150693 A1	08 November 2012
		TW 201309025 A	16 February 2013
		JP 2014511628 A	15 May 2014
		US 2013279820 A1	24 October 2013
		US 8750634 B2	10 June 2014
CN 102801976 A	28 November 2012	CN 102801976 B	22 October 2014
US 2003202602 A1	30 October 2003	EP 1500050 A1	26 January 2005
		AU 2003224939 A1	17 November 2003
		JP 2005524319 A	11 August 2005
		WO 03094113 A1	13 November 2003
		US 7302006 B2	27 November 2007