

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
28 August 2008 (28.08.2008)

PCT

(10) International Publication Number  
**WO 2008/103321 A2**

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

(21) International Application Number:  
PCT/US2008/002100

(22) International Filing Date:  
19 February 2008 (19.02.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
11/677,511 21 February 2007 (21.02.2007) US

(71) Applicant (for all designated States except US): **INTELLECTUAL VENTURES HOLDING 35 LLC** [US/US];  
7251 W. Lake Mead Boulevard, Suite 300, Las Vegas, NV  
89128 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **MONRO, Donald, Martin** [CA/GB]; 6, The Lays, Goose Street, Beckington  
Somerset BA116RS (GB).

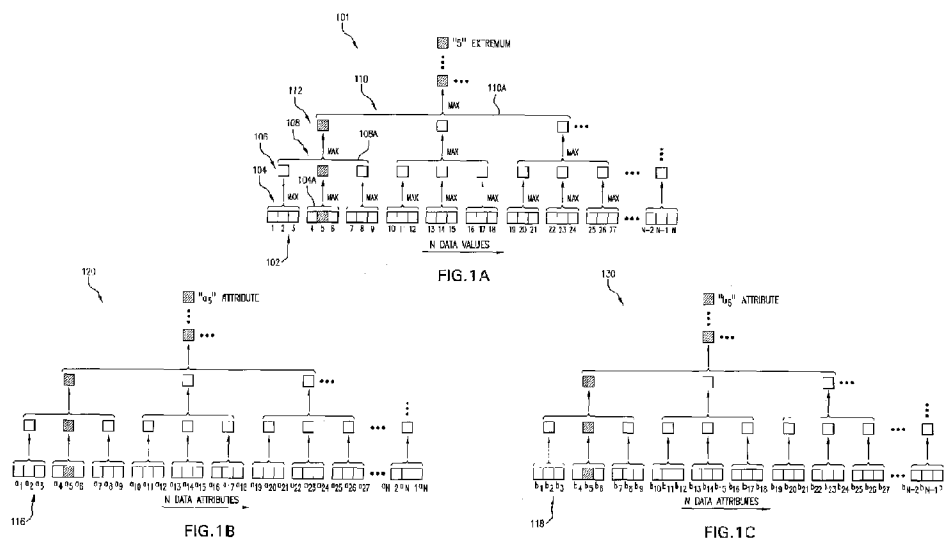
(74) Agents: **SPECHT, Michael D.** et al.; Sterne, Kessler,  
Goldstein & Fox P.L.L.C., 1100 New York Avenue, N.W.,  
Washington, DC 20005-3934 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

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

(54) Title: HIERARCHICAL UPDATE SCHEME FOR EXTREMUM LOCATION



(57) Abstract: Implementations related to hierarchical update schemes for extremum location are disclosed.

## HIERARCHICAL UPDATE SCHEME FOR EXTREMUM LOCATION

Inventors: Donald M. Munro

### BACKGROUND

[0001] The task of finding an extremum or extreme value (e.g., a maximum and/or minimum data value) of a data set is commonly undertaken using computing systems. For example, system performance may be evaluated by calculating a cost function over the ranges of P parameters yielding a P-dimensional data set of cost function values to be searched. It may be necessary to search the entire resulting data set at least once to find a desired number of extrema. If, however, one or more data values subsequently change, the entire data set may need to be searched again to find any new extrema.

[0002] Some applications, for example, some video encoding schemes, involve searching large data sets for extrema. Such data sets may be subject to repeated, sparse updating of the data values. Repeatedly searching the entirety of such updated data sets for new extrema wastes computing resources and may be too slow for many applications.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Subject matter is particularly pointed out and distinctly claimed in the concluding portion of the specification. Claimed subject matter, however, both as to organization and method of operation, together with objects and features thereof, may best be understood by reference of the following detailed description if read with the accompanying drawings in which:

[0004] Figures 1A — 3B are conceptualizations of a hierarchical data structure schemes;

[0005] Figure 4 is a flow diagram of a process for employing a hierarchical data structure;

[0006] Figure 5 illustrates an example encoding system;

[0007] Figure 6 is a flow diagram of a process for employing a hierarchical data structure for video encoding; and

[0008] Figures 7 — 8 illustrate example systems.

## DETAILED DESCRIPTION

[0009] In the following detailed description, numerous specific details are set forth to provide a thorough understanding of claimed subject matter. However, it will be understood by those skilled in the art that claimed subject matter may be practiced without these specific details. In other instances, well-known methods, procedures, components and/or circuits have not been described in detail.

[0010] Some portions of the following detailed description are presented in terms of algorithms and/or symbolic representations of operations on data bits and/or binary digital signals stored within a computing system, such as within a computer and/or computing system memory. These algorithmic descriptions and/or representations are the techniques used by those of ordinary skill in the data processing arts to convey the substance of their work to others skilled in the art. An algorithm is here, and generally, considered to be a self-consistent sequence of operations and/or similar processing leading to a desired result. The operations and/or processing may involve physical manipulations of physical quantities. Typically, although not necessarily, these quantities may take the form of electrical, magnetic and/or electromagnetic signals capable of being stored, transferred, combined, compared and/or otherwise manipulated. It has proven convenient, at times, principally for reasons of common usage, to refer to these signals as bits, data, values, elements, symbols, characters, terms, numbers, numerals and/or the like. It should be understood, however, that all of these and similar terms are to be associated with appropriate physical quantities and are merely convenient labels. Unless specifically stated otherwise, as apparent from the following discussion, it is appreciated that throughout this specification discussions utilizing terms such as “processing”, “computing”, “calculating”, “determining” and/or the like refer to the actions and/or processes of a computing platform, such as a computer or a similar electronic computing device, that manipulates and/or transforms data represented as physical electronic and/or magnetic quantities and/or other physical quantities within the computing platform’s processors, memories, registers, and/or other information storage, transmission, and/or display devices.

[0011] Figures 1A-1C depict an example scheme 100. Scheme 100 is presented for explanatory purposes and no arrangement, structure and/or illustration of any quantities

and/or elements in Figures 1A-1C should necessarily be construed to limit claimed subject matter in any way.

[0012] In scheme 100, a data value hierarchy and/or hierarchical data structure 101 includes, at its base, a one-dimensional (1 D) list and/or data set 102 of N data values (labeled 1, 2, 3... N) divided into partitions 104 having a dimension and/or size S of three data values. Claimed subject matter is not, however, limited in scope to any specific partition size. Data set 102 may also be referred to as a base level of structure 101, as base level data and/or a base level data set.

[0013] To begin populating structure 101, a total of  $(S - 1)$ , or two, comparisons may be made among data values in each partition 104, and subsequently identified extreme data values in partitions 104 may be carried into partitions 108 of a first level data set 106 in structure 101. Thus, for example, if set 102 includes fifty-four data values divided into eighteen partitions 104, then a total of thirty-six comparisons may suffice to determine all extrema of partitions 104. In various implementations, extrema in level 106 may comprise largest positive values, largest negative values, or largest absolute values of partitions 104. Claimed subject matter is not limited in this regard, however, and data values of level 106 may comprise, for example, lowest magnitude or smallest absolute values of partitions 104. Throughout this description and/or in the claims that follow, the phrases “carried into”, “abstracted to”, “be used to populate”, and/or “be placed in” may all be used to describe movement of extrema from partitions of a lower level in a hierarchical data structure to a next higher level of that structure. Also, throughout this description and/or in the claims that follow, the term “extremum” and its plural form “extrema” may be used interchangeably with the respective phrases “extreme data value” and “extreme data values.”

[0014] Like partitions 104, partitions 108 of level 106 may hold three data values each. Hence, in this example, each partition 108 holds three extrema taken from three corresponding partitions 104. To populate a second level data set 112 of structure 101, a total of  $(S-1)$ , or two, comparisons of data values of each partition 108 may be undertaken, and subsequently identified extreme data values of partitions 108 may be placed in partitions 110 of level 112. Thus, for example, if level 106 comprises eighteen values divided into six partitions 108, then a total of twelve comparisons may be used to determine all extrema of partitions 108.

[0015] First level 106 may be described as a coarse representation of extrema of the base level partitions 104, while second level 112 may be considered a coarse representation of extrema of the first level partitions 108. In this sense, upper levels of structure 101 may be described as successively coarser representations of extrema where each successive level 106, 112, etc., may be formed by finding and storing extrema of a previous level's partitions. Hence, a hierarchical data structure in accordance with claimed subject matter may be described as a hierarchy of successively coarser representations comprising a hierarchy of arrays and/or a tree.

[0016] In general, for a base level of  $N$  data values and for single values of  $S$ , a number of levels  $L$  in a data structure in accordance with some implementations of claimed subject matter may be provided by

$$L = \log_s (N) \quad (1)$$

[0017] Value  $L$  may comprise an integer if  $N$  comprises a power of  $S$ . For example, referring to structure 101 ( $S$  equal three), a number of data levels may be equal to  $\log_3(N)$ . Thus, for twenty-seven base level values, structure 101 may have three levels in a scheme with a single partition size of three. While for eighty-one base level values, structure 101 may have four levels in a scheme with a single partition size of three.

[0018] Claimed subject matter is not, however, limited to data structures having a specific number of levels and, thus, data structures in accordance with claimed subject matter may include as many levels as desired. For example, for a given number of data values  $N$ , there may be a certain number of data levels, not depicted in Fig. 1, between second level 112 and a top level or apex 114 of structure 101.

[0019] It may be recognized that, in accordance with equation (1), for some combinations of  $N$  and  $S$ , a number of levels  $L$  in a data structure in accordance with some implementations of claimed subject matter may have non-integer values. In particular, if  $N$  does not comprise a power of  $S$ ,  $L$  may not comprise an integer value and a next higher number of levels may be required to build a structure with a single apex. Thus, in some schemes, a penultimate level (e.g., the data level below the apex) may not include a total of  $S$  data values. However, claimed subject matter is not limited to full hierarchical data structures having a number of levels consistent with equation (1). Thus, for example, in some implementations of claimed subject matter, data structures may be employed having less levels than would be consistent with equation (1). In addition, while Fig. 1 may

illustrate a pyramidal data structure having same sized partitions in each level in accordance with some implementations of claimed subject matter, claimed subject matter is not limited in this regard and data structures with levels having different sized partitions SL may be employed in accordance with other implementations. Moreover, in accordance with some implementations of claimed subject matter, partitions within a level may be differently sized (e.g., may hold different numbers of data values).

[0020] Apex 114 of structure 101 may contain an extremum of base level 102. Overall, a total of (N-1) comparisons may be undertaken to initially populate a pyramidal data structure such as structure 101 with extrema. For example, for a set 102 of fifty-four data values, a total of fifty-three comparisons made among those values may be used to populate apex 114 with an extremum of set 102. Similarly fifty-two comparisons may be used to populate apex 114 for a base level set having fifty-three values, fifty-one comparisons may be used to populate apex 114 for a base level set having fifty-two values, and so forth.

[0021] In some cases, for example, with a sufficiently large data value set, an extremum of the data may not be unique (i.e., a set may contain multiple identical extreme data values or extrema). In other words, a base level of a data structure in accordance with claimed subject matter may contain multiple extremum data values. Thus, in various implementations, it may be valid to carry any one of a plurality of extreme data values up to an apex of a data structure, to carry a specific one of the extreme data values (e.g., first, last, or using some other criteria for selecting a value to carry) up to an apex, or to carry all such extreme data values up to an apex of a data structure.

[0022] In accordance with some implementations of claimed subject matter, each data value of a base level set may be associated with one or more attributes. For example, as shown in Figs. 1B and 1C, two attribute sets are shown: a first set 116 of N attributes. ( $a_1, a_2, \dots, a_N$ ) and a second set 118 of N attributes ( $b_1, b_2, \dots, b_N$ ). Any given  $k^{\text{th}}$  pair ( $a_k, b_k$ ) of attributes in attribute sets 116/118 being associated with a data value of set 102. In accordance with some implementations of claimed subject matter, parallel hierarchical attribute data structures may be employed to hold those attributes. For example, scheme 100 includes parallel attribute data structures 120 and 130, respectively, for sets 116 and 118, where, for a data value populating levels 102, 106, or 112 of structure 101, an

attribute of a corresponding pair of attributes has been carried into like levels of structure 120 and the other attribute into like levels of structure 130.

[0023] Claimed subject matter is not limited to any particular type and/or number of attributes and two attributes sets are shown in Figs. I B and IC simply to illustrate the principle. Attributes in accordance with some implementations of claimed subject matter may include a position or location of an extremum in a base level data set and/or other attributes of data such as, in the case of video data, a color or a dictionary entry for an associated basis function to name a few examples.

[0024] If one or more data values in a base level data set change, then, by employing a data structure in accordance with some implementations of claimed subject matter, effort required to identify any new extremum in a data set may be limited to making comparisons among data values in only those partitions affected by changed data value(s). For example, if a data value labeled "5" (shown hatched in Fig. IA) of set 102 changes, then comparisons among data values of a partition 104A holding value 5 may be undertaken to see if partition 104A contains a new extreme data value.

[0025] If no other data values change in set 102 then only comparisons within partition 104A of structure 101 may need to be undertaken at base level 102. Thus, in accordance with some implementations of claimed subject matter, a total of  $(S - 1)$ , or two, comparisons of data values of partition 104A would suffice to determine that partition's new extreme data value if any. Any new extreme data value may then be carried into a corresponding partition 108A of level 106. Likewise,  $(S - 1)$ , or two, comparisons of data values of newly updated partition 108A may then be undertaken to determine if a corresponding partition 110A of level 112 needs to be updated and so on. Thus, similar in manner to how structure 101 was initially populated as described above, and in accordance with some implementations of claimed subject matter, structure 101 may be subsequently updated with new extrema in any and/or all upper level partitions impacted by new extrema appearing in any of partitions 104. For example, if changing one or more data values results in a data value "5" becoming a new extremum of set 102, then that value may be propagated into apex 114 as shown in Fig. IA displacing an old extremum value.

[0026] If, on the other hand, carrying an updated extremum from partition 104A into partition 108A does not change that partition's extremum then, in accordance with some

implementations of claimed subject matter, updating of structure 101 may end and, hence, an extreme data value in apex 114 may remain unchanged and/or no additional comparisons may be needed. If, on the other hand, an extremum of partition 108A does change, then an extremum value of partition 110A may change as well and additional comparisons may be carried out.

[0027] In general, for N base level data values, a number of comparisons C undertaken to update and/or rebuild a pyramidal data structure when one base level data value changes may, in accordance with some implementations of claimed subject matter, be provided by

$$C = (S-1) \log_S (N) = (S-1)L \quad (2)$$

where S and L are, again, partition size and number of levels in a pyramidal data structure. In accordance with some implementations of claimed subject matter, if data values change in more than one base level partition, then a data structure may be updated or rebuilt in a similar manner above all partitions having a changed value.

[0028] While it may be convenient to choose values of N and S such that N comprises a power of S, claimed subject matter is not limited in this regard. Further, while a binary hierarchy and/or data structure having a partition size of two may minimize a number of comparisons required to populate or update a structure, claimed subject matter is not limited to specific partition sizes. Thus, for example, for a given set size N, larger values of S may be chosen in order to provide data structures having smaller values of L..

[0029] In accordance with some implementations of claimed subject matter, whenever a hierarchical data structure, such as structure 101, is updated, attributes associated with updated data values may be likewise updated in corresponding attribute data structure(s). Thus, for example, for any. new extreme data value propagated into partitions 108 and/or 110 of scheme 100, associated attributes of that data value may be propagated into corresponding partitions of attribute data structures 120 and 130 of scheme 100.

[0030] Although data structure 101 has been described as a full data structure having a numbers of levels L in accordance with equation (1), in other implementations of claimed subject matter, truncated hierarchical data structures having fewer levels may be employed. In such data structures, a top level or apex may contain multiple extreme data values comprising extrema of a penultimate layer's partitions. In yet other implementations, when, for example, equation (1) provides non-integer values of L, data structures may include a number of levels consistent with a next higher integer value of L.



- [0031] Figures 2A and 2B are diagrams depicting an example truncated hierarchical data scheme 200. Scheme 200 is presented for explanatory purposes and no arrangement, structure and/or illustration of any quantities and/or elements in Figures 2A/B should necessarily be construed to limit claimed subject matter in any way.
- [0032] As shown in Fig. 2A, a highest level 202 of a truncated data structure 201 contains extrema derived from a base level 204 of data values (labeled 1, 2, 3...N) and carried up through first and second levels numbered 206 and 208 respectively. In structure 201, each of levels 204 and 206 have partition sizes of three, while a penultimate level 208 has, in this example, a partition size of five. Because top level 202 of structure 201 contains more than one extreme data value, structure 201 may, in contrast to structure 101, provide a list or set of extrema rather than a single extremum. However, implementations in accordance with claimed subject matter may employ various combinations of hierarchical data structures having various partition sizes and . numbers of levels and data structures described herein represent only a small subset of possible hierarchical data structures in accordance with claimed subject matter.
- [0033] As with scheme 100 of Fig. 1, each of N data values in base level data set 204 of scheme 200 may be associated with one or more attributes. For example, in Fig. 2B, a single attribute set 216 of N attributes ( $a_1, a_2, \dots, a_N$ ) is shown where any one of attributes ( $a_1, a_2, \dots, a_N$ ) is associated with a single data value of set 204. Thus, in accordance with some implementations of claimed subject matter, a parallel attribute data structure 220 may be constructed for attribute set 216, where, for a data value populating one of levels 204 – 208 of structure 201, an associated attribute may be carried into like levels of structure 220. Again, claimed subject matter is not limited to any particular type and/or number of attributes and one set of attributes is shown in Fig. 2B simply to illustrate the principle.
- [0034] Figures 3A and 3B are diagrams depicting an example scheme 300. Scheme 300 is presented for explanatory purposes and no arrangement, structure and/or illustration of any quantities and/or elements in Figures 3A/B should necessarily be construed to limit claimed subject matter in any way.
- [0035] Scheme 300, includes a two-dimensional (2D) data, structure 301 having a 2D base level set 302 of sixty-four data values (labeled 1–64) divided into sixteen two-by-two partitions 304 of four data values each. Upper levels of data structure 301 comprise a

first level 306 holding a total of sixteen extrema in four two-by-two partitions 308, a penultimate or second level 310 comprising a single partition holding four extrema, and an apex 312 holding the extremum of set 302.

[0036] While set 302 as shown comprises a regular rectangular array, claimed subject matter is not limited in this regard and base level sets having any number of data values arranged or grouped in any manner may be employed. Further, while set 302 is shown as a 2D set, and sets 102/202 of schemes 100/200 as ID sets, claimed subject matter is not limited to base level sets of any particular dimensionality. Thus, for example, three-dimensional (3D) data structures in accordance with some implementations of claimed subject matter may be built above 3D base data sets.

[0037] Further, partitions may include any number of values and/or may have any shape. Thus, while partitions of structure 301 are shown having a two-by-two rectangular shape, partitions in accordance with claimed subject matter may be non-rectangular and/or may contain more or less than four values.

[0038] To build structure 301, extreme data values of each of partitions 304 may be carried into a separate one of four two-by-two partitions 308 of first level 306. To populate partitions 308 a total of (S-1), or three, comparisons may be made among data values of each partition 304. Thus, as shown in Fig. 3A, each partition 308 contains four extreme data values determined from among a corresponding four partitions 304 of set 302. Three comparisons may then be made among values in each partition 308 to establish the extrema of level 306 which may then be carried into second level 310.

[0039] Finally, three, comparisons among values of level 310 may establish an extremum of level 310, and hence of set 302, which may then be carried into apex 312 of structure 301. Thus, if, for example, a 27th data value comprises an extremum of set 302, then, in accordance with some implementations of claimed subject matter, this 27th data value may be propagated to apex 312 by undertaking a total of sixty-three comparisons: three comparisons for each of sixteen partitions 304 to populate level 306, three comparisons for each of four partitions 308 to populate level 310, and, finally, three comparisons of data values of levels 310 to provide the extremum value in apex 312. Hatched boxes in Fig. 3 illustrate propagation of this extremum through data structure 301 to apex 312.

[0040] As with schemes 100/200, a data value in structure 301 may be associated with one or more attributes. For example, in Fig. 3, one attribute set 316 of attributes (a,

$a_2, \dots, a_n$ ) is shown. Where a given  $k^{\text{th}}$  attribute in set 316 may be associated with a data value of set 302. Thus, as shown in Fig. 3B, and in accordance with some implementations of claimed subject matter, a parallel hierarchical attribute data structure 320 may be constructed, respectively, for or over set 316, where, for a data value carried into levels 302, 306 and/or 310 of structure 301, an associated attribute may be carried into like levels of structure 320. Again, claimed subject matter is not limited in scope to any particular type and/or number of attributes and one set of attributes is shown in Fig. 3B simply to illustrate the principle.

[0041] If one or more data values in base level 302 change, then updating of structure 301 may proceed in a manner similar to that described above with respect to schemes 100 and/or 200. Thus, if changing a data value results in a new extreme data value for set 302, then that extremum may be carried all the way to apex 312 of structure 301 using a total of nine comparisons consistent with equation (2) with a partition size of four. As described above with respect to schemes 100 and 200, and in accordance with some implementations of claimed subject matter, updating of structure 301 may be carried out in full, or updating may be terminated at a particular level if updating that level does not result in a new extremum or if a list of multiple extrema is sought.

[0042] While data structure 301 may allow for nine comparisons to achieve full updating for a single changed value, other schemes having ID data structures similar, for example, to scheme 100 of Fig. 1, may be employed in accordance with some implementations of claimed subject matter to reduce a number of comparisons needed to determine a new extremum. For example, if values of 2D set 302 were to be transformed into a 1 D set and a 1 D data structure having  $S = 2$  built over the ID set, then this ID data structure may be fully updated using only six comparisons after a single value changes.

[0043] If, however, two values change, for example, values labeled "26" and "27" in set 302, then, depending on how 2D set 302 was transformed, for example by scanning in some order, to generate a ID base level set, values 26 and 27 might be in different partitions so that updating a ID data structure of  $S = 2$  might require two partitions to be updated each requiring six comparisons, or twelve comparisons total. On the other hand, if the same two changes occur in scheme 300, where both changed values 26 and 27 occupy a same partition 304, then only one stage of updating may be required using, again, only nine comparisons. Clearly, there may be many ways, in accordance with

claimed subject matter, to optimize how data may be organized in a base level set and/or how a hierarchical data structure may be organized overall in order to reduce the number of comparisons and/or stages of updating required, claimed subject matter not being limited in scope to any particular ordering of a base level set and/or hierarchical data structure.

[0044] In accordance with some implementations of claimed subject matter, a base level data set may comprise data representative of image data, of video data, or of a signal such as an audio signal. Further, a base level data set in accordance with some implementations of claimed subject matter may be 1D, 2D, 3D or of higher dimensionality. In addition, a data structure in accordance with some implementations of claimed subject matter may comprise a hierarchy of arrays, or a tree. Moreover, implementations in accordance with claimed subject matter may employ various combinations of hierarchical data structures having various partition sizes and numbers of levels, and those data structures described herein with respect to Figs. 1-3 represent only a small subset of the possible hierarchical data structures or schemes in accordance with claimed subject matter.

[0045] Figure 4 is a flow diagram of a process 400 for employing a hierarchical data structure. In block 405, a base data value set may be provided. In blocks 410 and 415 a hierarchical data structure and one or more hierarchical attribute data structure(s) may, respectively, be created or built. For example, block 410 may involve creating a hierarchical data structure in a manner similar to that described with respect to Figs. 1-3. Block 410 may involve, for example, determining, given N data values in a base set of a particular dimensionality provided in block 405, a number of data levels L and partition size(s) SL for those levels. Data structure levels may then be populated with data values in a manner similar to that described above with respect to Figs. 1-3 to complete block 410.

[0046] Block 415 may involve creating one or more hierarchical attribute data structures having a same total number of data levels L and partition size(s) SL as a data structure created in block 410. Any such hierarchical attribute data structures may then be populated with attributes of data values of a data structure created in block 410 in a manner similar to that described with respect to Figs. 1-3.

[0047] In respective blocks 420 and 425, one or more extreme data values and any associated attribute(s) may be determined. In some implementations of claimed subject

matter, blocks 420 and 425 may be performed upon completion of respective blocks 410 and 415 when top levels of a data structure and any associated attribute data structure(s) may be populated, respectively, with one or more extreme data values and associated attribute(s), if any.

[0048] At block 430, one or more data values may be changed in the base data set received in block 405. Block 430 may involve one or more data values getting larger or smaller. If multiple values change then some of values may increase in magnitude while others decrease in magnitude.

[0049] In block 440 the data structure may be updated to in response to value(s) changed in block 430. As described previously with respect to Figs. 1-3, updating a data structure in accordance with some implementations may involve making comparisons among base level partitions holding changed data values, providing associated first level partitions with any new extreme data values of those base level partitions, and then carrying that process forward in a similar manner for some if not all levels of a data structure. At block 445, associated attribute data structure(s) may be updated in a parallel manner as was described previously with respect to Figs. 1-3.

[0050] Claimed subject matter is not, however, limited to updating an entire data structure in block 440 and, thus; in some implementations, block 440 may involve updating a data structure only for partitions of those levels above partitions having new extrema. For example, in some implementations, changing one or more data values in block 430 may not result in a new extreme data value in a base data set and, thus, updating of a data structure in block 440 may not be carried through to a highest level and/or apex of a data structure. In other implementations, extrema or multiple extreme data values of a base data set, rather than a single extremum, may be provided. In such implementations, updating in block 440 may be terminated before a highest level and/or apex of a data structure. In accordance with some implementations of claimed subject matter, updating of any associated attribute data structures in block 445 may be carried out to a same extent as that of a data structure in block 440.

#### Application to Video Encoding

[0051] Encoding video data may comprise an application suitable for employing hierarchical data structures in accordance with claimed subject matter. In some video encoding schemes, algorithms, such as matching pursuits (MP) algorithms, may be

employed to transform 2D image data into coded information describing the data in terms of various known signals or basis functions having discrete amplitudes. Claimed subject matter is not, however, limited to video encoding, or to video encoding schemes employing MP processes.

[0052] An MP method was first described with respect to coding of raw 1D audio signals. See, for example, S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries", IEEE Trans. Signal Processing, vol. 41, pp. 3397-3415, Dec. 1993. MP methods have also been applied in 2D to video coding. See, for example, R. Neff and A. Zakhor, "Very low bit rate video coding based on matching pursuits", IEEE Trans. Circuits and Systems for Video Tech., vol. 7, pp. 158-171, Feb. 1997; and A. Zakhor and R. Neff, Method and apparatus for compression of very low bit rate video signals, US Patent 5,699,121, 16 December 1997.

[0053] An MP algorithm may include repeatedly determining, for different locations or positions in a data set, full inner products between data to be coded and members of a dictionary of basis functions, and then identifying basis functions yielding largest inner products at different positions. At any particular position, a dictionary entry of an identified basis function may describe the data locally and may be termed an "Atom." To find a particular Atom, a maximum of absolute values of inner products may need to be identified. Amplitudes of Atoms thus identified may be quantized using one of any number of well-known quantization techniques, claimed subject matter not being limited in scope in this regard. For example, Atom amplitudes may be quantized using a Precision Limited Quantization (PLQ) method (see, for example, D. M. Monroe, J-L Aufranc, M. A. Bowers and W Poh, "Visual embedding of wavelet transform coefficients", IEEE Int. Conf. Image Process. (ICIP 2000), Sept. 2000), or some other method.

[0054] When initially undertaking an MP process in accordance with some implementations of claimed subject matter, a hierarchical data structure may be created or built over a base level data set comprising absolute values of inner products in a manner similar to that described above with respect to Figs. 1-3. In an MP process a base level may comprise extrema determined, at each position, over all basis functions in a dictionary, and it may be necessary to record which dictionary entry is associated with each value in the base level data set. Hence, attributes associated with absolute values of

inner products, such as dictionary entries, signs, positions or locations in image data, etc, may be used to initially populate hierarchical attribute data structure(s) in a manner similar to that described above with respect to Figs. 1-3.

[0055] Once identified using, at least in part, a data structure in accordance with some implementations of claimed subject matter and quantized, an Atom may be removed or subtracted from associated image data. Removing an Atom from a location in an image data set may change image data in a local region. Inner products may then be recomputed and another position and dictionary entry yielding a maximum absolute value of an inner product may be identified by, at least in part, updating a base level of a data structure and then applying a hierarchical update in accordance with some implementations of claimed subject matter in a subsequent iteration of an MP process. Thus, data may be altered hundreds or thousands of times when coding using MP methods as successive Atoms are identified and removed, and a new search for a maximum absolute inner product may be carried out when identifying each Atom. It may be recognized that results of MP processing may be improved if a maximum of all absolute inner products are determined with each iteration or step of an MP process.

[0056] An iteration of an MP process may carry out new inner product calculations only in a locality where the image data has been changed as a result of a previous iteration. This may be termed 'repairing' inner products to those familiar with the field. Having repaired inner products in a locality, a next iteration of an MP method may include examining repaired inner products to determine, at each repaired position, a dictionary entry that provides a maximum absolute value. Doing so may result in a new quantized amplitude and dictionary entry for each repaired position. Newly determined quantized amplitudes may then, in accordance with implementations of claimed subject matter, be used to update a base level of an associated hierarchical data structure. Dictionary entries associated with the newly determined quantized amplitudes may, likewise, be used to update a base level of an associated attribute hierarchical data structure holding dictionary entries. The hierarchical data structure and associated attribute hierarchical data structure(s) may then be updated to locate a next Atom and its attribute(s).

[0057] At any particular stage or iteration of an MP process, data being processed may be described by codes of Atoms found up to that stage, and a remaining data residual. However, when identifying and removing an Atom, a subset of inner products values may

change in regions that overlap an area where a previous Atom has been subtracted from the image data. This region may be termed a “footprint” of a previous Atom and may span tens of image data pixels.

[0058] An MP process may be described in pseudocode as:

Initialize compute full set of inner products

Repeat

Find Atom. Full search or reduced complexity strategy

Atom Update. Subtract quantized Atom from image

Repair. Recompute required inner products only in Atom footprint.

Until distortion or bit rate criterion met

[0059] Claimed subject matter is not, however, limited to a particular MP process such as described by the above pseudocode.

[0060] An MP process may be terminated at some stage and codes of a certain number of Atoms stored or transmitted by a further lossless coding process. Atoms used may describe a signal with some loss of information, while any unused Atoms plus a residual complete a signal's description. A lossless coding process employed may, for example, be a MERGE code employing PLQ quantization or some other method. See, for example, Yuan Yuan and Monro, D. M., “Improved Matching Pursuits Image Coding”, IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2005, Philadelphia, March 2005. Claimed subject matter is not, however, limited in scope to any particular lossless coding process and/or quantization process. A decoder may reconstruct transmitted coded Atoms to form a lossy signal description.

[0061] For some implementations, a dictionary of basis functions may comprise 2D bases. Other implementations may employ dictionaries comprising 1D bases that can be combined separably to form 2D bases. A dictionary of  $n$  basis functions in one dimension may provide a dictionary of  $n^2$  basis functions in two dimensions. In some implementations, 2D data, such as a portion of a frame of video data, may be transformed, for example by scanning in some suitable order, to yield a 1D signal and a 1D dictionary may be applied. In some implementations, a dictionary may comprise a set, group and/or collection of Gabor functions although claimed subject matter is not limited in scope in this regard.



- [0062] To better understand application of some implementations of claimed subject matter to video encoding, an example MP implementation may be described with reference to scheme 300 although those skilled in the art may recognize that MP encoding may be performed using much larger data sets than set 302.
- [0063] In some implementations, video data subjected to a MP process may comprise a portion or region of a video frame. In some implementations, video data may comprise a Displaced Frame Difference (DFD) image generated during motion compensation processing of a video frame. As described above, an MP process may include searching a data set comprising absolute values of inner products for new extrema. Thus, for example, set 302 may comprise a set of absolute of inner product values determined for a region of image data. Claimed subject matter is not, however, limited in scope to inner products data or, for that matter, to any particular type of data, MP related video data or otherwise.
- [0064] Thus, at a position of a maximum inner product value in set 302 identified by an extreme data value of a hierarchical data structure in accordance with some implementations of claimed subject matter, a dictionary entry associated with that maximum inner product may describe video data locally. In this sense, a particular basis function (i.e., a dictionary entry) may be described as an attribute associated with, and/or representing video data associated with a location in set 302. In accordance with some implementations of claimed subject matter, other attributes of a data value may include its position or location in a base level data set. For example, a position of a data value in set 302 may be indicated by a position index, a row and column index, etc. Moreover, in addition to a position or location of a value in the base level data set, an absolute inner product value of base level set, such as set 302 may be associated with attributes such as a dictionary entry of an associated Atom and/or quantized amplitude of an associated Atom. In implementations where data values comprise absolute inner product values, attributes may also comprise a value's sign (e.g., positive or negative valued):
- [0065] Figure 5 is a block diagram of an example video encoder and/or encoding system 500. Encoding system 500 may be included in any of a wide range of electronic devices, including digital cameras or other image forming devices, although claimed subject matter is not limited in this respect. System 500 may receive data 501 for a current original image. For this example implementation, current original image 501 may comprise a frame from a digital video stream or sequence of image frames. A motion

estimation block 510 may receive current original image 501 and a reference or previous reconstruction frame 513. Motion estimation block 510 may perform motion compensation on image 501 to produce motion data 515 and prediction data 503. Motion data 515, which may include motion vectors and/or motion vector corrections, may be encoded by a code motion block 522 to produce coded motion data. Claimed subject matter is not limited in scope to any particular motion compensation method and/or any particular method used to encode motion data. Prediction data 503 may be subtracted from current original image data 501 to form an error or DFD image 505.

[0066] DFD image 505 may be received at an MP block 514. In some cases, DFD image 505 may be transformed before being provided to MP block 514. For example, DFD image 505 may be wavelet transformed before being provided to MP block 514. Claimed subject matter is not, however, to a particular type and/or format of data in general or in particular as provided to MP block 514.

[0067] MP block 514 may perform an MP process on DFD image 505 in a manner similar to that described above. In accordance with some implementations of claimed subject matter, MP block 514 may, in the process of MP encoding DFD image 505, use hierarchical data structures (e.g., pyramidal data structures similar to structure 301 of Fig. 3) to, for example, determine extreme absolute inner, product values as successive Atoms are identified and removed from DFD image 505. In doing so, MP block 514 may store data values of the one or more hierarchical data structures in memory 513 coupled to MP block 514 and/or may access memory 513 to receive extrema and/or to update data values. Memory 513 may comprise any type of memory such as, but not limited to, Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), or the like. Further, in some implementations of claimed subject matter, MP block 514 may employ logic including comparator logic while using a hierarchical data structure to determine extreme inner product values. Those skilled in the art will recognize that comparator logic comprising one or pyramidal arrays of comparators may be used to undertake comparisons of data values.

[0068] Those skilled in the art may recognize that inner product data sets (e.g., inner product data sets derived from DFD image 505) may comprise much larger base data sets than are shown in the example of Fig. 3. Moreover, when identifying extreme inner product values, MP block 514 may also employ one or more separate attribute data

structures, similar to structures 320 and 330, holding attributes associated with the new extrema. Attributes populating those data 'structures may also be held in memory 513.

[0069] MP block 514 may use a dictionary 516 to construct a series of Atom parameters 517 which may be delivered to a code Atoms block 520. Atom parameters 517 may, for example, comprise one or more of the data attributes held in hierarchical attribute data structures. Code Atoms block 520 may encode the Atom parameters using any of a wide range of encoding techniques, claimed subject matter not being limited in scope in this regard. MP block 514 may also produce a coded residual 509 that that may be added to motion prediction information 503 to form a current reconstruction image 511 corresponding to current image data. Image 511 may be delayed by a delay block 518 before being provided to motion estimation block 510 as a previous reconstruction image 513 to be used in connection with motion estimation operations for a next original image.

[0070] Coded Atoms from block 520 and coded motion data from block 522 may be formed into a bitstream 526 that, in turn, may be transmitted to any of a wide range of devices, such as devices incorporating video decoders, using any of a wide range of interconnect technologies, including wireless interconnect technologies, the Internet, local area networks, etc., although claimed subject matter is not limited in this respect.

[0071] The various blocks and units of encoding system 500 may be implemented using software, firmware, and/or hardware, or any combination of software, firmware, and hardware. Further, although Figure 5 depicts, an example system having a particular configuration of components, other implementations are possible using other configurations. In addition, while Fig. 5 is directed to a video encoding system, claimed subject matter is not limited to video encoding applications, and, thus, other systems adapted for the encoding of still images or for the encoding of audio signals, to name two examples, may, in accordance with other implementations of claimed subject matter, employ hierarchical data structures holding data values to determine extrema.

[0072] Figure 6 is a flow diagram of a video coding process 600. In block 605, a base set of video data values may be provided. A data set provided in block 605 may comprise a set similar to set 302 of Fig. 3 and may comprise absolute inner product values determined in an MP process, although claimed subject matter is not limited in this regard. Thus, referring to Figs. 3 and 5, Block 610 may, for example, involve having MP block 514 employ logic to, in part; determine inner product values over a dictionary of

basis functions for a set of image data such as DFD image 505 and then provide absolute values of those inner products as set 302.

[0073] At block 610, a hierarchical data structure may be built. For example, referring to Figs. 3 and 5, block 610 may involve having MP block 514 divide set 302 in partitions 304, determine a number of levels  $L$  and associated partition sizes  $SL$ , and employ logic to, in part, compare the inner product values within partitions to each other to initially populate a hierarchical data structure with extreme inner product values.

[0074] While scheme 300 may provide a useful example, those skilled in the art may recognize that common video data applications may involve much larger data sets than set 302. For example, in television broadcasting, a set of image data may comprise 720 horizontal by 560 vertical rows of pixel data yielding a data set of 414,720 pixel values. MP encoding of such video data may entail searching for extreme data values in a number of inner product data sets each having 414,720 inner product data values derived from those pixel values. Thus, in some implementations, schemes in accordance with claimed subject matter may employ a plurality of data structures each having a base level data set of 414,720 absolute inner product values and each associated with one or, more corresponding attribute structures of the same size. However, this is only one example image data set size, there being many possible image data set sizes, and claimed subject matter is not limited to any particular base level data set size whether comprising video data or any other data.

[0075] For example, referring also to Fig. 5, in block 610, MP block 514 may build a data structure over a base level of 414,720 absolute inner product values where higher levels of the data structure may be populated with extreme data values by defining partitions within the base level, comparing data values within partitions, and so on as described previously above with respect to Figs. 1-3. In undertaking block 610, MP block 514 may store data values that populate the data structure(s) in memory 513 and/or may use comparator logic to undertake comparisons.

[0076] In block 620 one or more hierarchical attribute data structures may be built. In some implementations, block 620 may involve: creating one or more hierarchical attribute data structures having the same total number of data levels  $L$  and partition size(s)  $SL$  as a data structure created in block 610, and then populating those attribute data structures with attributes of data values that populate a data structure created in block 610.

- [0077] In accordance with some implementations of claimed subject matter, at least two attribute data structures may be created in block 620 by MP block 514. One of those attribute data structures may be built on a base level comprising the signs of a corresponding 414,720 absolute inner product values of a data structure created in block 610, while another attribute structure may be built on a base level comprising dictionary entries of basis functions associated with those inner product absolute values. In undertaking block 620, MP block 514 may store and/or access values populating the attribute data structure(s) in memory 513.
- [0078] At block 630, one or more data values may be changed in a base data set of absolute inner product values. For example, block 630 may occur when MP block 514, performing an MP process, subtracts an Atom from image data and repairs affected inner products thereby altering one or more values in the base data set
- [0079] Those skilled in the art may recognize that a size of a footprint generated by subtracting an Atom may depend on a size or extent of basis functions employed in an MP process. For example, if a maximum basis size of an MP basis function comprises nine pixel units, then, when an Atom is subtracted from an image, a total of eighty-one pixel values may be altered. However, because of overlap between inner product determinations, inner product values over a larger 17x17 window, or a 289-pixel region, this example, may change in a base data set when an Atom is removed from image data. Those skilled in the art may further recognize that repairing an affected region may involve calculating new inner product values in that affected region and best matching basis functions may then be determined for each location in a repaired region. Thus, in this example, block 630 may involve changing amplitudes of 289 data values out of 414,720 in a base data set when an Atom is subtracted from image data.
- [0080] In block 640, the data structure may be updated. As described previously with respect to Figs. 1-3, updating a data structure in accordance with some implementations of claimed subject matter may involve making comparisons among those base level partitions holding changed data values, providing associated partitions of a first level with any new extreme data values of base level partitions, and then carrying that process forward in a similar manner for some if not all levels of a data structure.

For a pyramidal two-dimensional hierarchical data structure having a 2x2 partition size similar to structure 301 but a base set of 414,720 values, a total of ten levels above a

base level may need to be updated in block 640 if one or more values change in a base level in block 630. A base level data set may represent an image comprising 720 horizontal by 576 vertical pixels. At each level above the base, a number of partitions may be halved in each direction. Therefore, no more than 10 levels may be required for such an image, since starting from an apex and doubling ten times would produce 1024 by 1024 partitions, which is larger than the base level data set in this example.

[0081] Continuing this example, if 289 base level values change in block 630, MP block 514 may need to search eighty-one partitions, out of a total of 103,680 base level partitions, for new extrema. In other words, in block 640, MP block 514 may continue performing a MP process by comparing four data values, which may require three comparison operations, in each of eighty-one base level partitions to determine if any partitions have new extreme data values. Hence, updating eighty-one partitions may require a total of 243 comparisons. If some base level partitions have new extreme data values, then, given a particular data structure used in this example, MP block 514 may need to examine a total of twenty-five of 25,920 first level partitions by using seventy-five comparisons to search for any new extrema on that level. As long as updating a level results in one or more new extreme data values on that level, then updating of a data structure may continue in block 640 with updating of corresponding partitions of a next higher level, etc. If updating a specific level does not result in any new extrema in that level's partitions then updating of a data structure, in block 640 may end with that level.

[0082] Overall, using this example, if changing 289 base values in block 630 result in a new extreme value in a base level set then, after eighty-one partitions are examined in a base level, twenty-five partitions may be searched at a first level, nine at a second level, four at a third level, and one at each level thereafter yielding a total of 375 comparisons in block 640 to identify a new extreme value in 414,720 total base level values. By comparison, without benefit of claimed subject matter, establishing a new base set extreme value may require a total of  $(N - 1)$ , or 414,719, comparisons.

[0083] Again, claimed subject matter is not limited in block 640 to updating an entire data structure and, thus, in some implementations block 640 may involve updating a data structure only for those levels having new extrema. For example, some implementations may occur wherein changing one or more data values in block 630 does not result in a new overall extremum in a base data set and, thus, updating of a data structure in block

640 may not be carried through to a highest level and/or apex of a data structure. In other implementations, a set of extrema or multiple extreme data values may be sought rather than a single extremum. In these implementations, updating in block 640 may also be terminated before a highest level and/or apex of a data structure.

[0084] In block 650, an attribute data structure(s) may be updated. As described previously with respect to Figs. 1-3, updating an attribute data structure in accordance with some implementations may involve updating attribute data structure(s) to reflect any changes in an associated data structure. In other words, if, for example, any data values change in partitions of a base level in block 630, and if any of those new values are carried into partitions of a first level in block 640, then attributes of those data values may likewise be carried into like first level partitions of associated attribute data structure(s) by MP block 514 in block 650. Similarly, MP block 514 may continue block 650 by updating upper levels of associated attribute data structure(s) with attributes of any new extreme data values propagated into like levels of a data structure in block 640.

[0085] Thus, in accordance with some implementations of claimed subject matter, either initially building data structures in blocks 610 and 620 or updating such structures in block 640 and 650 may result in identification of an extremum inner product value and associated attributes such as a dictionary entry and/or a quantized amplitude. of an associated Atom. Attributes may also include a position or location of an inner product value in the base level data set where that location may correspond to a position of an associated Atom in the image data. Attributes may also include a sign value .(e.g., positive or negative valued) for base level sets comprising absolute inner product values.

[0086] Figure 7 is a block diagram of an example computer system 700. System 700 may be used to perform some or all of the various functions discussed above in connection with Figures 1-6. System 700 includes a central processing unit (CPU) 710 and a memory controller hub 720 coupled to CPU 710. Memory controller hub 720 may further coupled to a system memory 730, to a graphics processing unit (GPU) 750, and to an input/output hub 740. GPU 750 may be further coupled to a display device 760, which may comprise a Cathode Ray Tube (CRT) display, a Liquid Crystal Display (LCD) flat panel display, or other type of display device. Although example system 700 is shown with a particular configuration of components, other implementations are possible using any of a wide range of configurations.

[0087] Figure 8 is a block diagram of an example video transmission system 800. System 800 includes a video encoder 802 (e.g., system 700) that may transmit or convey information 804 (e.g., in a bitstream) to a video decoder 806 (e.g., system 700) where that information includes video data that has been compressed using an encoding process that employs hierarchical data structures in accordance with some implementations of claimed subject matter. For example, encoder 802 may, in accordance with some implementations of claimed subject matter and while employing a MP encoding scheme to generate information 804, use data structures having base data sets of inner product absolute values to determine inner product extrema. Although example system 700 is shown with a particular configuration of components, other implementations are possible using any of a wide range of configurations.

[0088] It will, of course, be understood that, although particular implementations have just been described, claimed subject matter is not limited in scope to a particular embodiment or implementation. For example, one embodiment may be in hardware, such as implemented to operate on a device or combination of devices, for example, whereas another embodiment may be in software. Likewise, an embodiment may be implemented in firmware, or as any combination of hardware, software, and/or firmware, for example. Likewise, although claimed subject matter is not limited in scope in this respect, one embodiment may comprise one or more articles, such as a storage medium or storage media. This storage media, such as, one or more CD-ROMs and/or disks, for example, may have stored thereon. instructions, that when executed by ‘a system, such as a computer system, computing platform, or other system, for example, may result in an embodiment of a method in accordance with claimed subject matter being executed, such as one of the implementations previously described, for example. As one potential example, a computing platform may include one or more processing units or processors, one or more input/output devices, such as a display, a keyboard and/or a mouse, and/or one or more memories, such as static random access memory, dynamic random access memory, flash memory, and/or a hard drive.

[0089] Reference in the specification to “an implementation,” “one implementation,” “some implementations,” or “other implementations” may mean that a particular feature, structure, or characteristic described in connection with one or more implementations may be included in at least some implementations, but not necessarily in all . .



implementations. The various appearances of “an implementation,” “one implementation,” or “some implementations” in the preceding description are not necessarily all referring to the same implementations. Also, as used herein, the article “a” includes one or more items. Moreover, when terms or phrases such as “coupled” or “responsive” or “in response to” or “in communication with” are used herein or in the claims that follow, these terms should be interpreted broadly. For example, the phrase “coupled to” may refer to being communicatively, electrically and/or operatively coupled as appropriate for the context in which the phrase is used.

[0090] In the preceding description, various aspects of claimed subject matter have been described. For purposes of explanation, specific numbers, systems and/or configurations were set forth to provide a thorough understanding of claimed subject matter. However, it should be apparent to one skilled in the art having the benefit of this disclosure that claimed subject matter may be practiced without the specific details. In other instances, well-known features were omitted and/or simplified so as not to obscure claimed subject matter. While certain features have been illustrated and/or described herein, many modifications, substitutions, changes and/or equivalents will now, or in the future, occur to those skilled in the art. It is, therefore, to be understood that the appended claims are intended to cover all such modifications and/or changes as fall within the true spirit of claimed subject matter.

## WHAT IS CLAIMED:

## 1. A method comprising:

forming a data structure comprising: a base level including data values partitioned into first partitions; a first level including extreme data values of the first partitions; and an apex including at least one extreme data value of the first level corresponding to an extremum of the base level; and

in response to at least one first partition having a new extreme data value, updating the data structure, at least in part, by placing the new extreme data value in the first level;

wherein updating the data structure includes placing the new extreme data value in the apex if the new extreme data value comprises an extremum of the base level.

## 2. The method of claim 1, further comprising:

partitioning the first level into second partitions; and

forming a second level of the data structure, the second level including extreme data values of the second partitions.

## 3. The method of claim 2, wherein the first partitions and the second partitions include different numbers of extreme data values.

## 4. The method of claim 1, further comprising: forming an attribute data structure comprising:

a base level including attributes of data values of the base level of the data structure; a first level including attributes of data values of the first level of the data structure; and an apex including at least an attribute of an extremum of the base level of the data structure.

## 5. The method of claim 4, wherein attributes of data values comprise positions of data values in the base level of the data structure.

## 6. The method of claim 4, wherein attributes of data values comprise one of dictionary entries of Atoms, quantized amplitudes of Atoms, or sign values.

7. The method of claim 1, wherein the base level comprises two-dimensional data.
8. The method of claim 7, wherein the base level comprises data values representative of image data or video data.
9. The method of claim 1, wherein the base level comprises one-dimensional data.
10. The method of claim 9, wherein the base level comprises data values representative of audio data.
11. The method of claim 1, wherein the data structure comprises a hierarchy of arrays or a tree.
12. The method of claim 1, wherein the first partitions include different numbers of extreme data values.
13. The method of claim 1, further comprising:
  - storing extreme data values of the first partitions in memory.
14. The method of claim 1, wherein the base level comprises absolute values of inner products determined in a matching pursuits process.
15. A method, comprising:
  - partitioning a base level data set;
  - forming a coarse representation of extrema of partitions of the base level data by finding and storing the extrema of the partitions;
  - updating the coarse representation in response to altering one or more data values of the base level data, wherein updating the coarse representation comprises finding and storing new extrema only for corresponding partitions of the base level data including altered data values.
16. The method of claim 15, further comprising:
  - forming a hierarchy of successively coarser representations of extrema, each successive level in the hierarchy formed by finding and storing extrema of a previous level's partitions;

selectively updating the hierarchy from the base level to a highest level for partitions of the base level data containing altered data values; and

identifying extrema of the base level data by processing data held in the highest level of the hierarchy.

17. The method of claim 16, wherein the hierarchy of successively coarser representations comprises a hierarchy of arrays or a tree.

18. The method of claim 16, wherein the hierarchy of successively coarser representations comprises a hierarchy of levels, wherein each level above the base level includes extrema of a next lower level, the method further comprising:

partitioning data values of each level, wherein partitions comprise different numbers of values within levels of the hierarchy and between different levels of hierarchy.

19. The method of claim 16, wherein the partitions include two or more data values.

20. The method of claim 15, further comprising:

forming a coarse representation of attributes of extrema of partitions of the base level data by finding and storing the attributes of the extrema of the partitions;

updating the coarse representation of attributes in response to altering one or more data values of the base level data, wherein updating the coarse representation of attributes comprises finding and storing attributes of new extrema only for corresponding partitions of the base level data including altered data values.

21. The method of claim 15, wherein the base level comprises data values representative of at least one of an image, of video data, or of a signal.

22. The method of claim 21, wherein the signal comprises an audio signal.

23. The method of claim 15, wherein the base level comprises one-dimensional data.

24. The method of claim 15, wherein the base level comprises two-dimensional or more than two-dimensional data.

25. The method of claim 15, wherein the base level comprises absolute values of inner products determined by a matching pursuits process.
26. A method of video encoding, comprising:
  - partitioning a set of data values into a plurality of first partitions;
  - storing a set of first extrema corresponding to extreme data values of the first partitions;
  - altering one or more data values to produce one or more altered first partitions;
  - updating extreme data values of the set of first extrema corresponding to the one or more altered first partitions.
27. The method of claim 26, wherein storing the set of first extrema comprises:
  - placing the extreme data values of the first partitions in memory.
28. The method of claim 27, wherein updating extreme data values comprises:
  - replacing one or more of the extreme data values of the first partitions in memory.
29. The method of claim 26, further comprising:
  - partitioning the set of first extrema into a plurality of second partitions; and
  - storing a set of second extrema corresponding to extreme data values of the ' second partitions.
30. The method of claim 29, wherein altering one or more data values produces one or more altered second partitions, the method further comprising:
  - updating extrema values of the set of second extrema corresponding to the one or more altered second partitions.
31. The method of claim 26, wherein updating extreme data values of the set of first extrema comprises comparing data values of the one or more altered first partitions.
32. The method of claim 26, wherein updating extreme data values of the set of first extrema comprises replacing one or more extreme data values of the set of first extrema with one or more new extrema of the one or more altered first partitions.

33. The method of claim 26, further comprising:
- storing a set of attributes corresponding to the set of first extrema;
  - altering one or more data values to produce one or more altered first partitions;
  - and
  - updating the set of attributes in response to new extrema associated with the one or more altered first partitions.
34. An apparatus, comprising: a video encoder adapted to use a hierarchical data structure to identify an extremum of a data set.
35. The apparatus of claim 34, wherein the encoder is further adapted to update the data structure in response to a new extremum of the data set.
36. The apparatus of claim 34, wherein the data structure comprises:
- a base level including, data values of the data set partitioned into first partitions; a first level including extreme data values of the first partitions; and
  - an apex including at least one extreme data value of the first level corresponding to an extremum of the base level.
37. The apparatus of claim 36, wherein the encoder is further adapted to update the data structure in response to a new extremum of the data set by, at least in part, placing the new extreme data value in the first level.
38. The apparatus of claim 36, wherein the encoder is further adapted to update the data structure in response to a new extremum of the data set by placing the new extreme data value in the apex if the new extreme data value comprises an extremum of the base level.
39. The apparatus of claim 36, wherein the first partitions and the second partitions include different numbers of extreme data values.
40. The apparatus of claim 36, wherein the encoder is further adapted to:
- form an attribute data structure comprising: a base level including attributes of data values of the base level of the data structure; a first level including attributes of data

values of the first level of the data structure; and an apex including at least an attribute of an extremum of the base level of the data structure.

41. The apparatus of claim 40, wherein attributes of data values comprise positions of data values in the base level of the data structure.
42. The apparatus of claim 40, wherein attributes of data values comprise one of dictionary entries of Atoms, quantized amplitudes of Atoms, or sign values.
43. The apparatus of claim 36, wherein the base level comprises data values representative of one of image data, video data, or audio data.
44. The apparatus of claim 36, wherein the base level comprises one of one-dimensional data, two-dimensional data, or three-dimensional data.
45. The apparatus of claim 36, wherein the first partitions include different numbers of extreme data values.
46. The apparatus of claim 36, further comprising:
  - memory;
  - wherein the encoder is further adapted to store extreme data values of the first partitions in the memory.
47. The apparatus of claim 36, wherein the base level comprises absolute values of inner products determined in a matching pursuits process.
48. The apparatus of claim 34, wherein the data structure comprises a hierarchy of arrays or a tree.
49. An article comprising: a storage medium having stored therein instructions that, if executed, result in:

forming a data structure comprising: a base level including data values partitioned into first partitions; a first level including extreme data values of the first partitions; and an apex including at least one extreme data value of the first level corresponding to an extremum of the base level; and

in response to at least one first partition having a new extreme data value, updating the data structure, at least in part, by placing the new extreme data value in the first level;

wherein updating the data structure includes placing the new extreme data value in the apex if the new extreme data value comprises an extremum of the base level..

50. The article of claim 49, wherein said instructions, if executed, further result in:

partitioning the first level into second partitions; and

forming a second level of the data structure, the second level including extreme data values of the second partitions.

51. The article of claim 49, wherein said instructions, if executed, further result in:

forming an attribute data structure comprising: a base level including attributes of data values of the base level of the data structure; a first level including attributes of data values of the first level of the data structure; and an apex including at least an attribute of an extremum of the base level of the data structure.

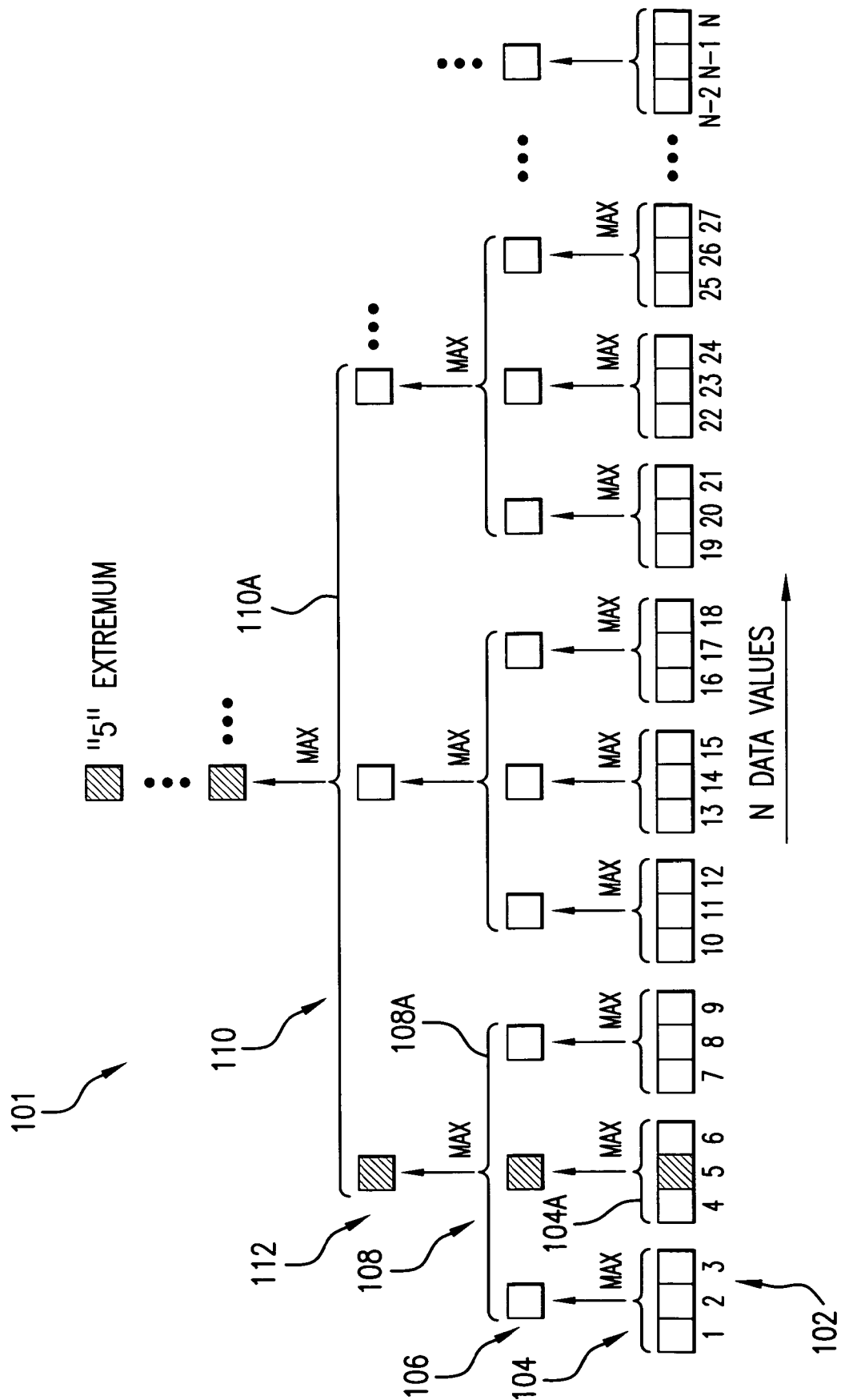
52. The article of claim 51, wherein attributes of data values comprise positions of data values in the base level of the data structure.

53. The article of claim 51, wherein attributes of data values comprise one of 47 dictionary entries of Atoms, quantized amplitudes of Atoms, or sign values.

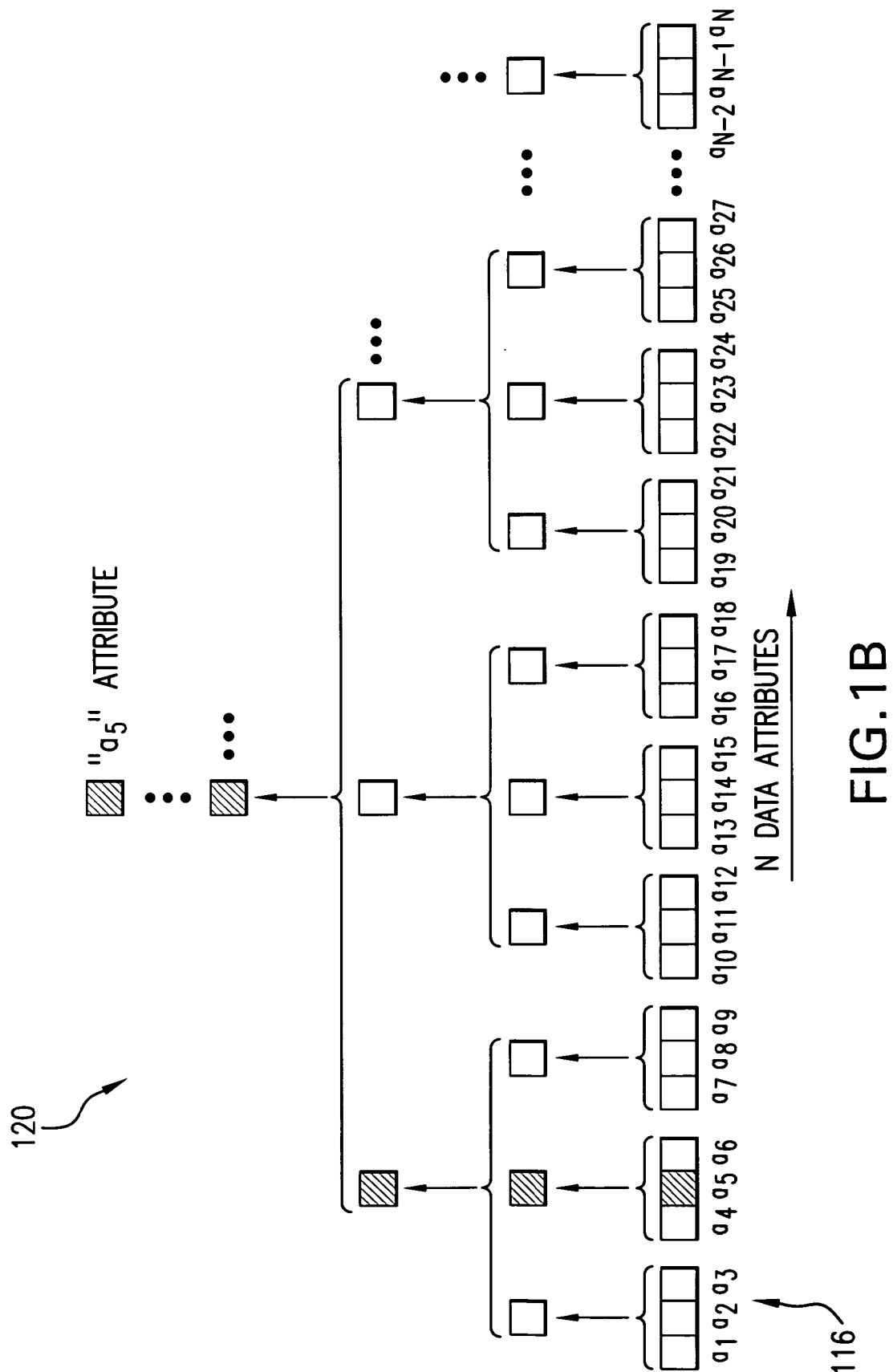
54. The article of claim 49, wherein said instructions, if executed, further result in: storing extreme data values of the first partitions in memory.

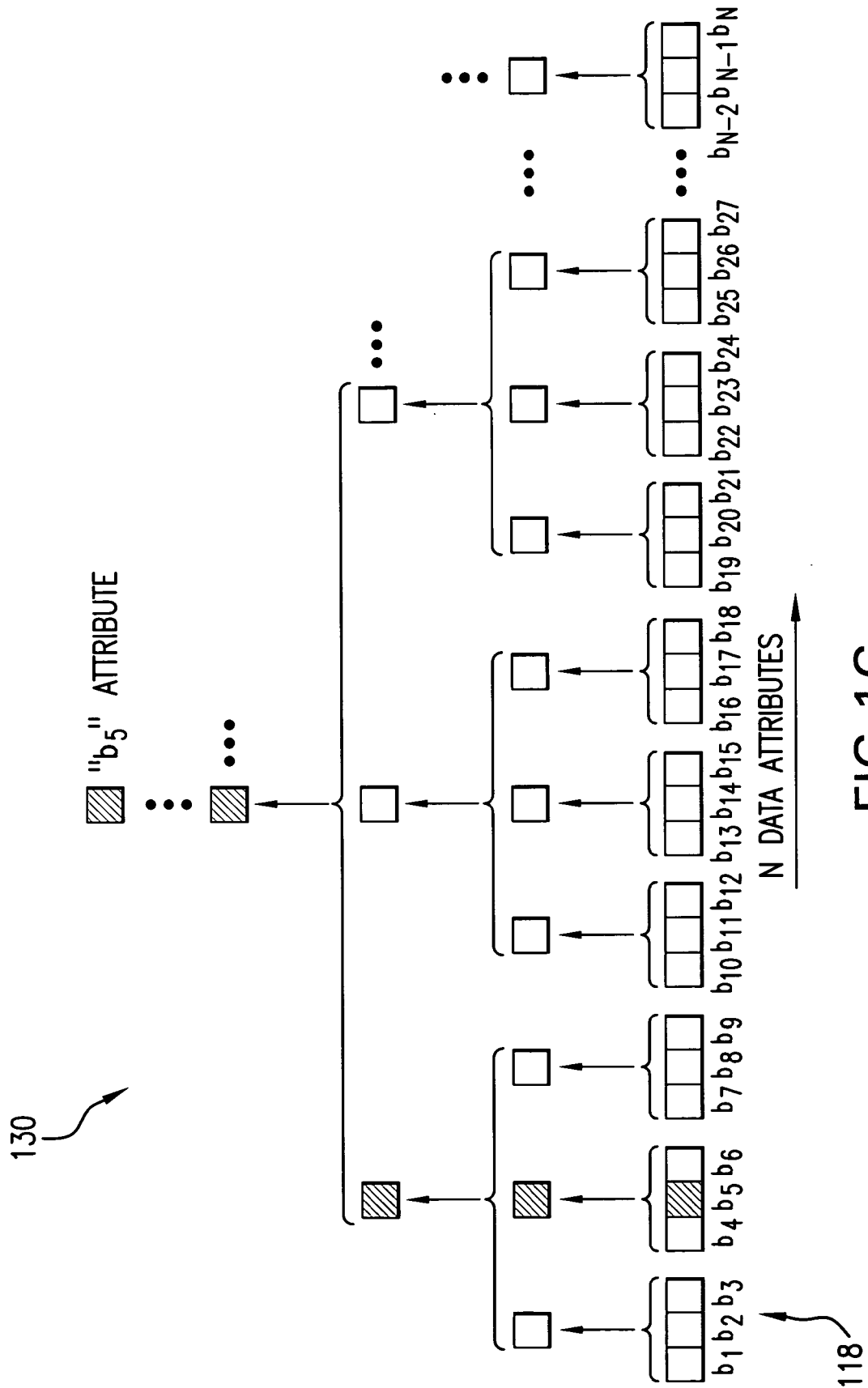
55. The article of claim 49, wherein the base level comprises absolute values of inner products determined in a matching pursuits process.

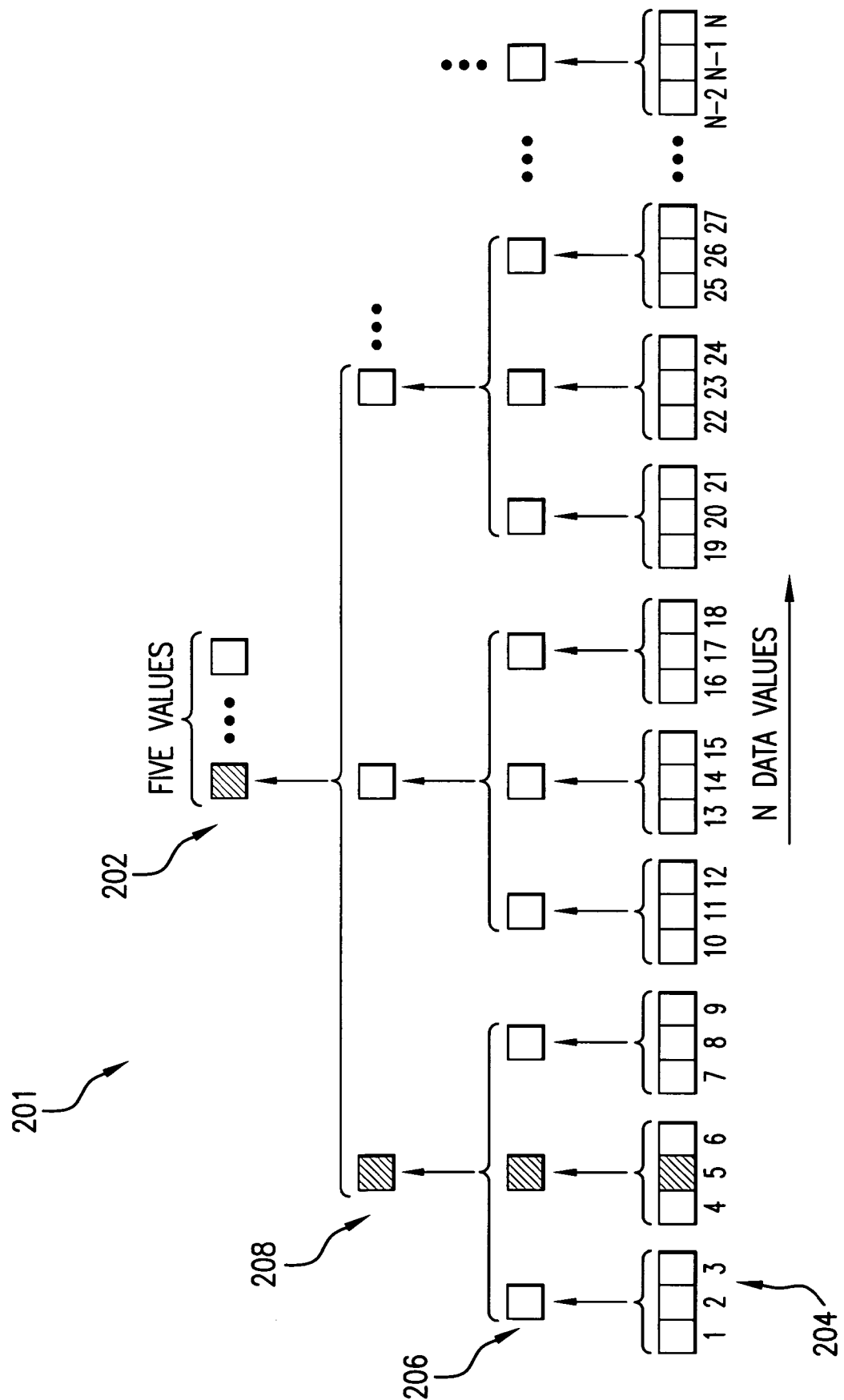




**FIG. 1A**







**FIG. 2A**

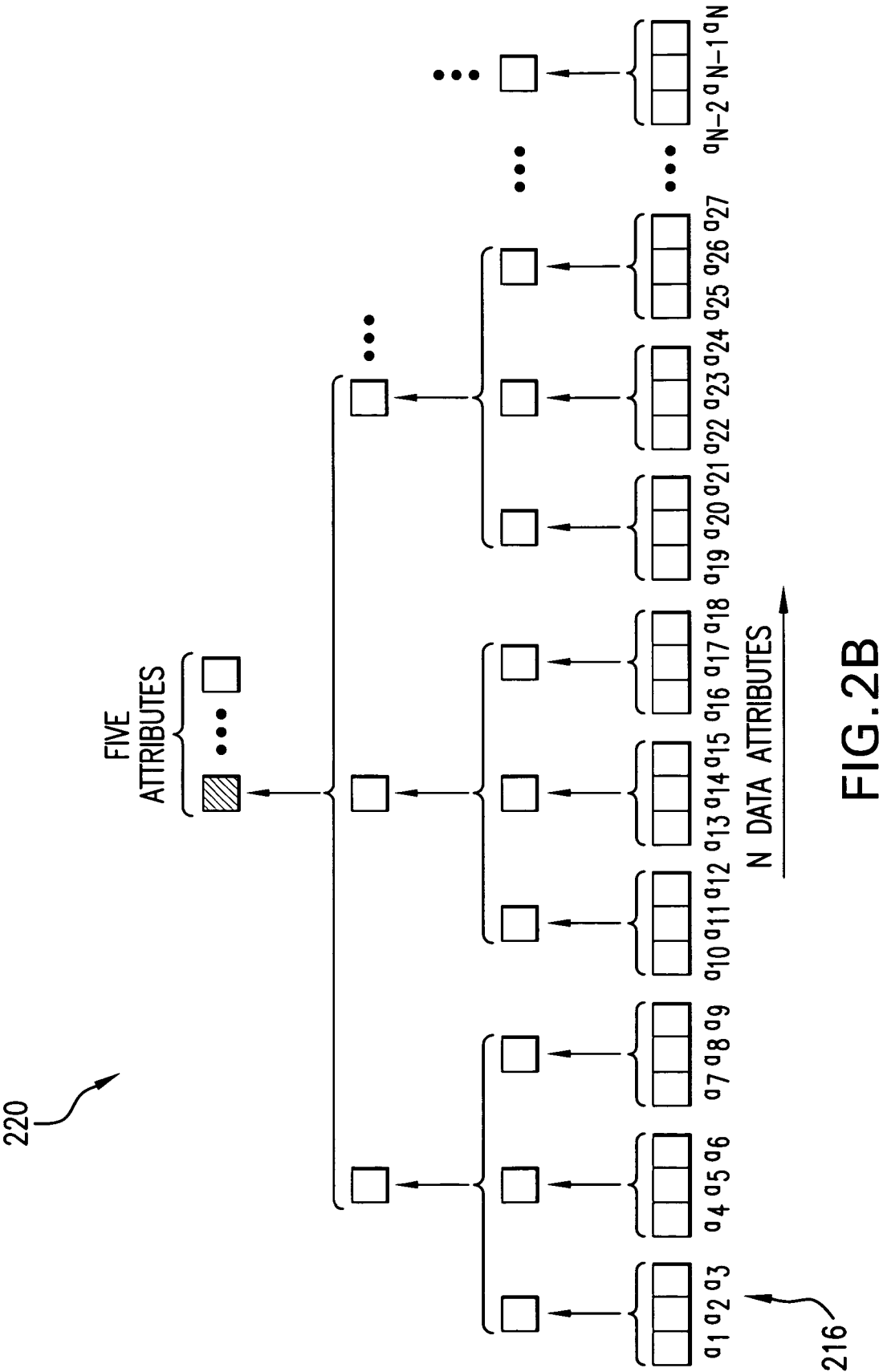


FIG. 2B

6/10

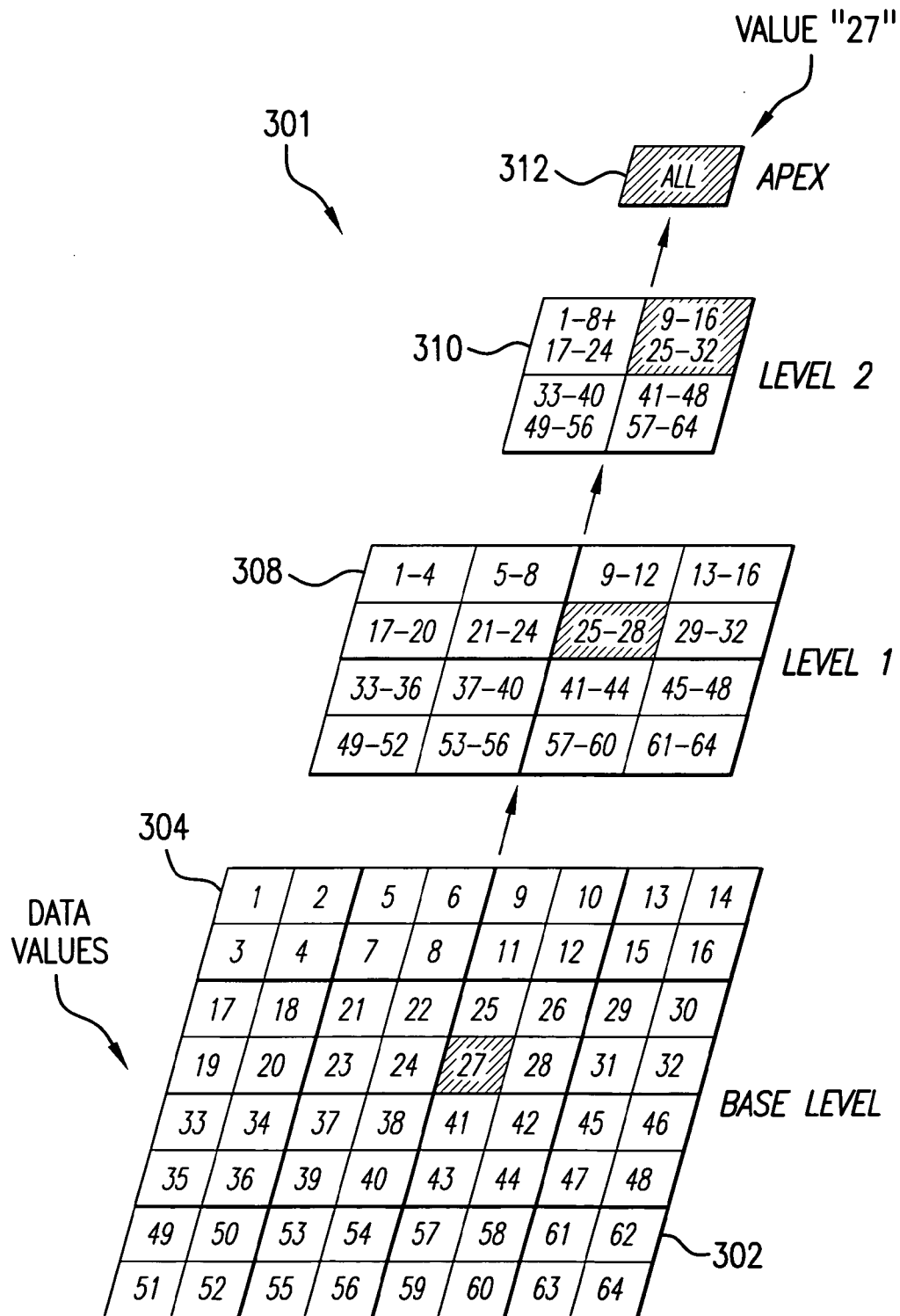


FIG.3A

7/10

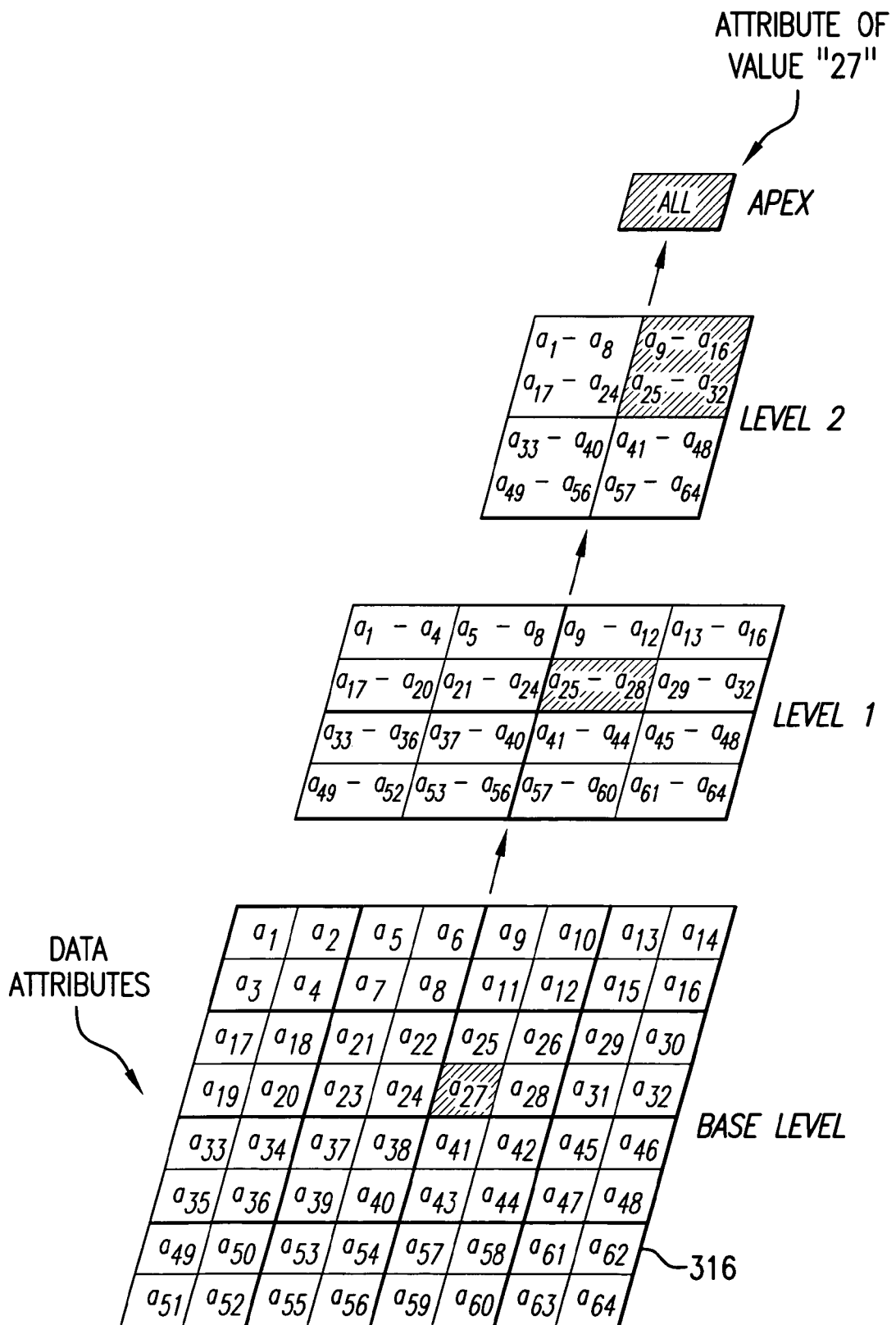


FIG. 3B

SUBSTITUTE SHEET (RULE 26)

8/10

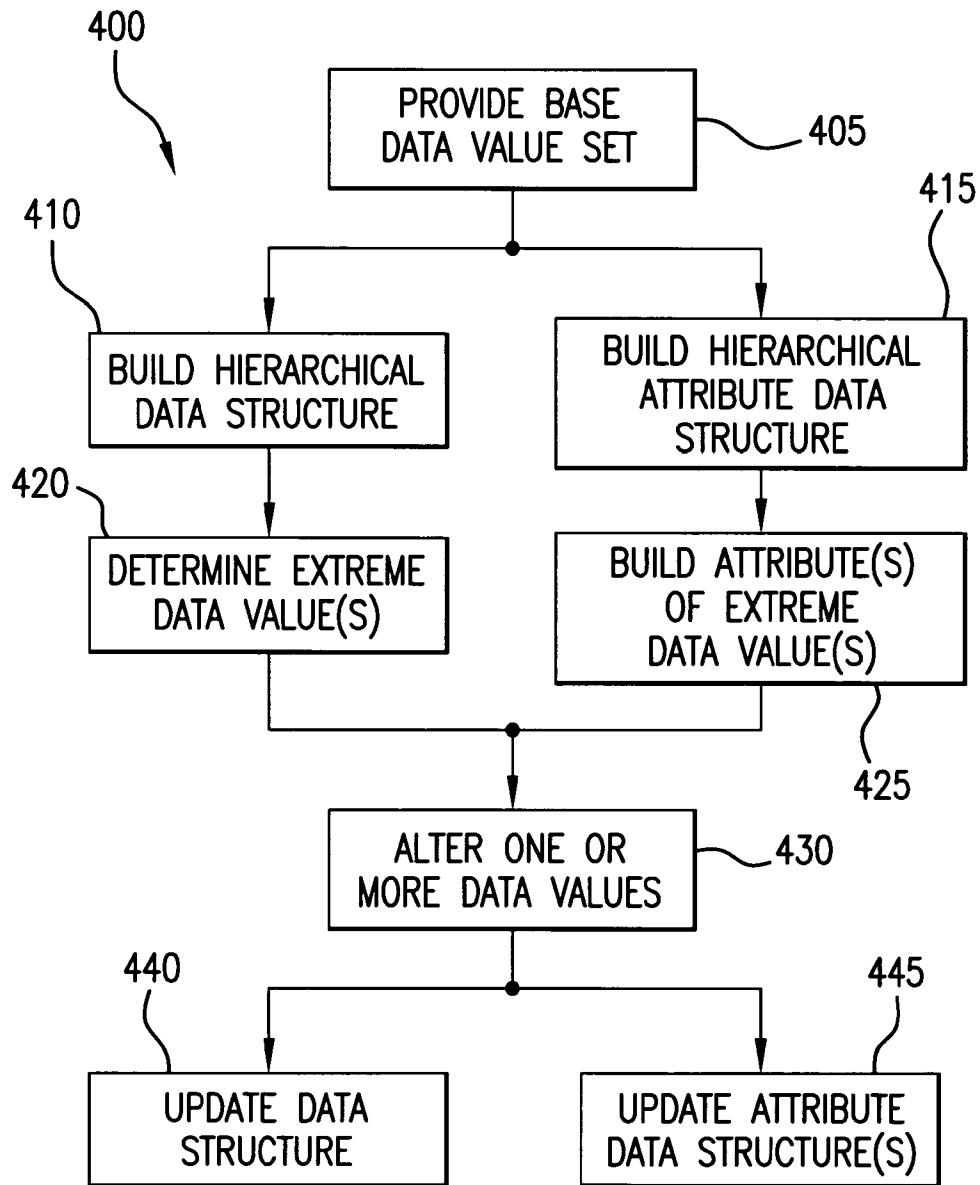


FIG.4



9/10

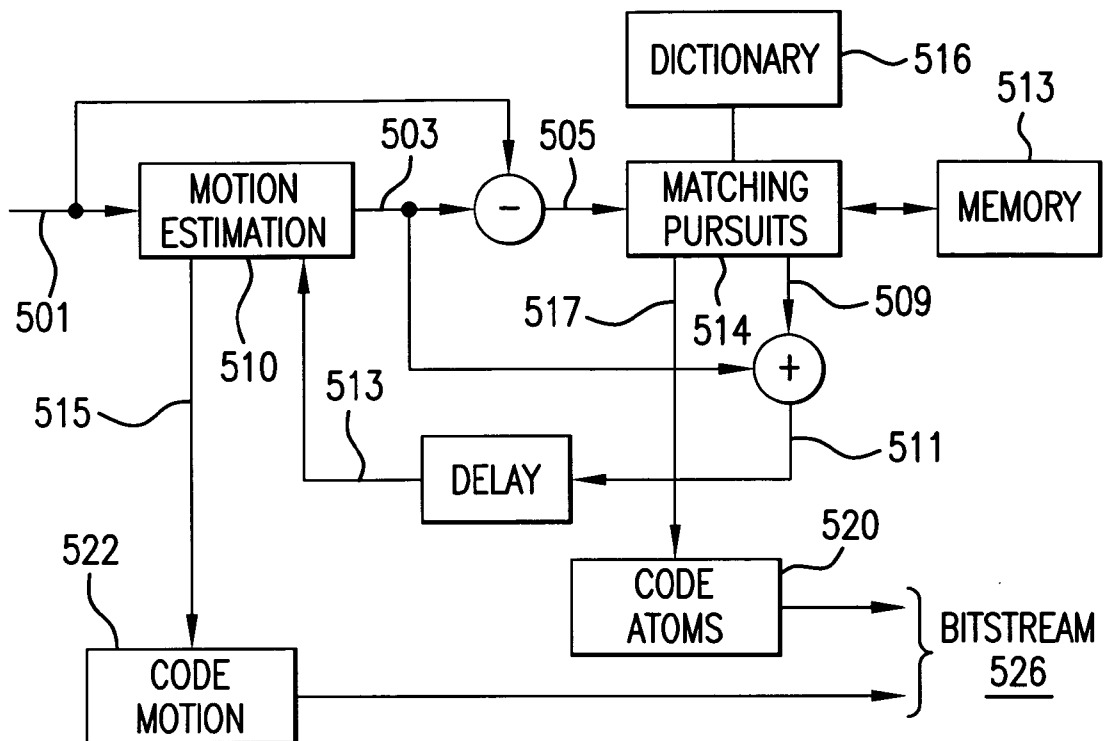


FIG. 5

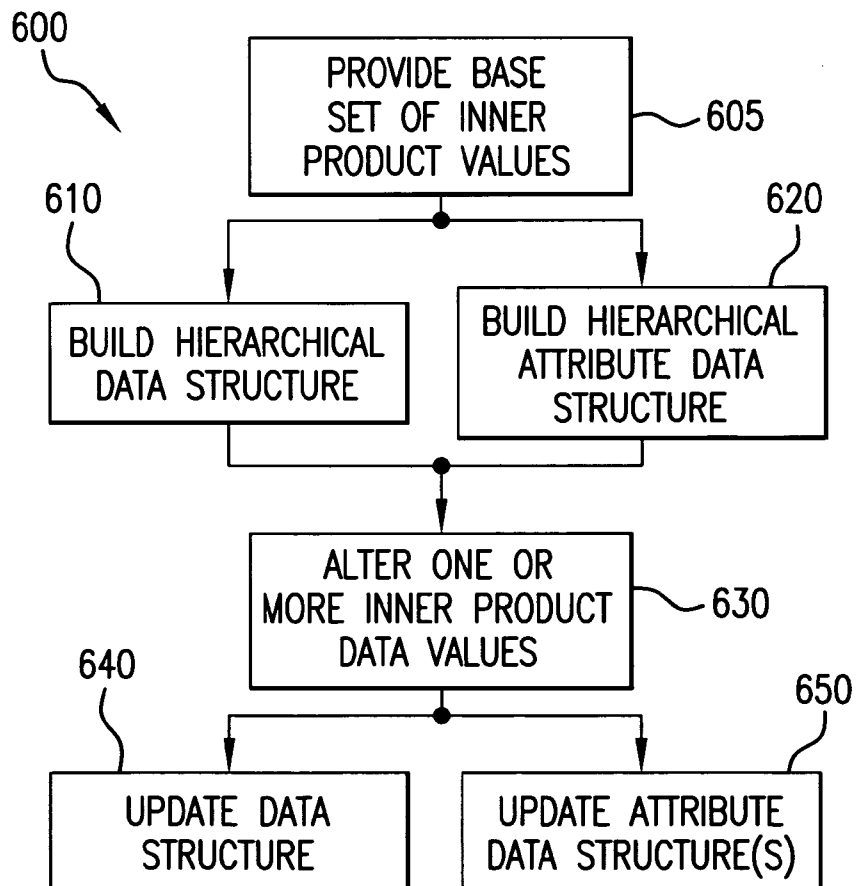


FIG. 6

10/10

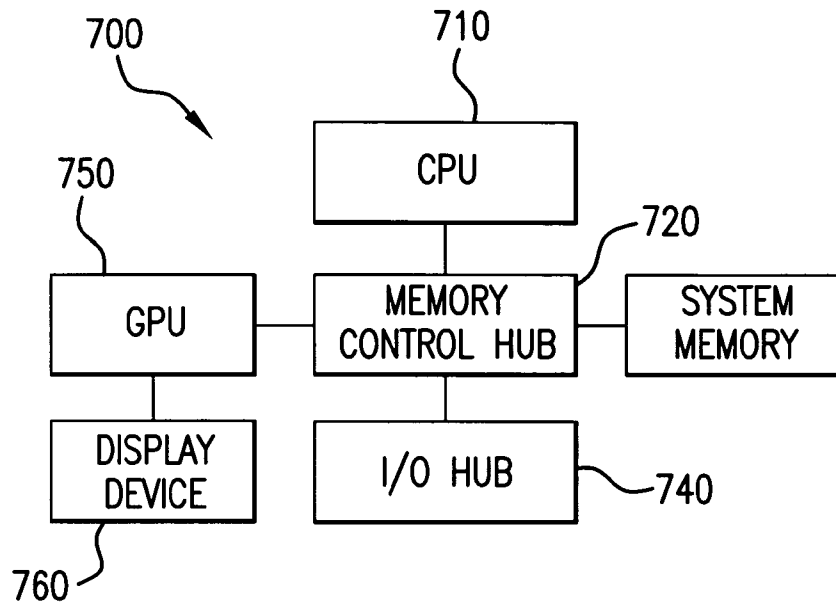


FIG. 7

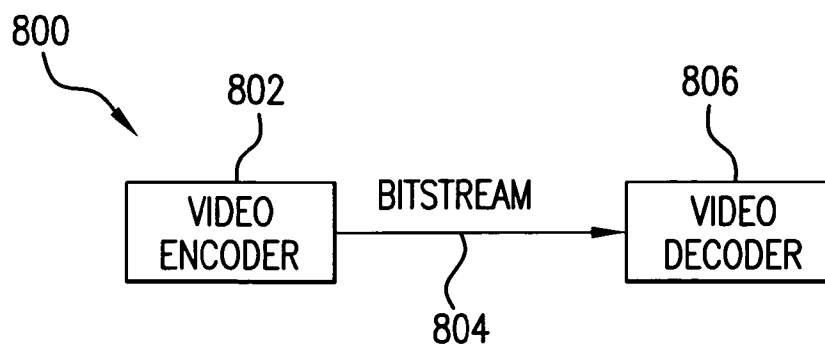


FIG. 8