



- (51) International Patent Classification:
H03M 7/32 (2006.01)
- (21) International Application Number:
PCT/US2012/021882
- (22) International Filing Date:
19 January 2012 (19.01.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
13/009,175 19 January 2011 (19.01.2011) US
- (71) Applicant (for all designated States except US): **EXAR CORPORATION** [US/US]; 48720 Kato Road, Fremont, California 94538 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **ZHANG, Yuhong** [CN/US]; 1785 Castlegate Drive, San Jose, California 95132 (US). **WANG, Jiebing** [US/US]; 728 Mairwood Court, San Jose, California 95120 (US).
- (74) Agents: **FLIESLER, Martin C.** et al.; 650 California Street, 14th Floor, San Francisco, California 94108 (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, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: APPARATUS AND METHOD OF DELTA COMPRESSION

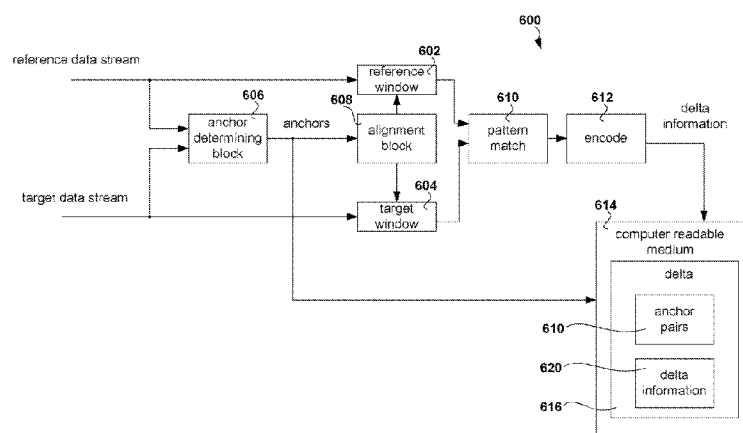


FIGURE 6

(57) Abstract: A method includes aligning a reference window and target window for compression of a target data stream in terms of a reference data stream. The anchors are determined by examining the target data stream and reference data streams. The target data stream is aligned with respect to the reference data streams using the anchors. Pattern matching between the aligned target data stream and reference data stream is done to delta compress the target data stream.

APPARATUS AND METHOD OF DELTA COMPRESSION**CLAIM OF PRIORITY**

[0001] This application claims the benefit of priority to U.S. Patent Application No. 13/009,175, filed January 19, 2011, entitled "APPARATUS AND METHOD OF DELTA COMPRESSION", by Zhang et al., the specifications and drawings of which are incorporated herein by reference.

BACKGROUND

[0002] Most compression techniques are concerned with processing a single data stream. Delta compression on the other hand is the compression of one data stream, referred to as the target data stream, in terms of another data stream, called the reference data stream, by computing a delta. The delta can be viewed as an encoding of the difference between the target and the reference data stream. The target data stream can be later recovered from the delta and the reference data stream. Delta compression can be based on byte-to-byte comparisons. Delta compression is different from hash-based deduplication methods. Delta compression can provide for a finer comparison result than hash-based deduplication methods.

[0003] Delta compression is used in revision control systems. By storing deltas of different versions instead of the actual data, these systems are able to reduce storage requirements significantly. For example, Xdelta File System (XDfs) developed by Joshua MacDonald is a file system implemented with delta compression. Another application of delta compression is software distribution; especially the software that is distributed over the Internet. By distributing deltas, or essentially patches, one can significantly reduce network traffic. Delta compression can also be used to improve HTTP performance. By exploiting the similarity between different pages on a given website or between the different versions of a given web page, one can reduce the latency for web access. VCDIFF is defined in RFC 3284 to support this kind of usage.

[0004] But in many cases, due to deleting or inserting operations, the reference data is no longer aligned with target data. If reference data and target data are misaligned too much, the incoming target data can't find matched pattern in reference data window. The compression ratio will then be dramatically degraded. There are already several delta compressors available including xdelta, vdelta (and its newer variant VCDIFF) and zdelta. None of them avoids the problem.

SUMMARY OF THE INVENTION

[0005] Delta compression logic pattern matches using a reference window and a target window. The reference and target data is aligned during delta compression so that the reference and target windows contain similar data. In this way, a better compression ratio can be achieved.

5 [0006] An intelligent alignment can be implemented by indentifying one or more anchor pairs by examining the target and reference data streams. The anchor pairs can be determined by using Rabin-Karp or a similar rolling hash algorithm. Each byte in the target or reference data stream has a rolling hash result that corresponds to a hash of a multiple byte window.

10 [0007] A reference anchor candidate is located when a feature pattern is found in the rolling hash results of reference data stream. The rolling hash result is also referred to as fingerprint value of the reference anchor candidate. If an anchor candidate from the target data has the same fingerprint value as a counterpart from the reference data, an anchor pair is identified.

15 [0008] With such anchor pair, the invention can use much smaller reference window than other tools. This can simplify the computation complexity and improve performance. The use of a smaller reference window also makes hardware implementation feasible by saving memory resources on a chip.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Figure 1 shows an example of inserting some text into target data.

20 [0010] Figure 2 is an example of prior art delta compression solution.

[0011] Figure 3 is a diagram of an embodiment of the present invention which uses anchors to align the delta compression.

[0012] Figure 4 illustrates a procedure to find an anchor candidate.

[0013] Figure 5 shows a flow chart of one embodiment of the invention.

25 [0014] Figure 6 shows an exemplary delta compressor.

[0015] Figure 7 shows an exemplary delta decompressor.

[0016] Figure 8 shows an exemplary anchor pair determination.

DESCRIPTION

30 [0017] Adding or deleting data from an old file version to a new file version can happen daily. The advantage of delta compression is that only the difference need be stored. Figure 1 shows an example. A data segment y is inserted in the target data stream.

[0018] The segment y can be identified and encoded in delta compression. Delta compression saves storage space by referring to data in the reference stream.

[0019] All delta compressors compare the incoming target data stream with a reference data stream. Some compressors also compare the incoming target data with the previous target data (target history).

[0020] Figure 2 is an example of a prior art system. Source data (also called as target data) 201 is compared to reference data using a reference window 203 and target window 204 to find pattern matching. The best compression ratio is achieved when reference window is big enough to hold the whole reference data stream. But for cost effectiveness reasons, typically not the all of the reference data is compared with the incoming target data. Instead, as in most compression systems, only a part of reference data is stored in reference window and participates in comparison. So if the data pattern in reference data stream happens to be not in reference window, no pattern will be found. The compression ratio will degrade dramatically if target data can't find matched pattern in reference window. For example, if the data segment y in figure 1 is bigger than the size of reference window, the target data segment after data y will not find matched pattern in reference window.

[0021] A delta compression method can comprise determining anchors to align a reference window and target window for compression of a target data stream in terms of a reference data stream. The anchors can be determined by examining the target data stream and reference data stream. The target data stream can then be aligned with respect to the reference data stream. Pattern matching between the aligned target data stream and reference data stream can be used to delta compress the target data stream.

[0022] A delta decompression method can comprise using anchors to align a reference window and target window for decompressing of a target data stream in terms of a reference data stream using a delta. The anchors can be previously determined by examining the target data stream and reference data stream during compression of the target data stream. The target data stream can be decompressed using the aligned reference and target windows.

[0023] A delta compressor 600 can include a reference window 602, a target window 604, and an anchor determining block 606 to determine anchors by examining the target data stream and reference data stream. As discussed below, the anchor determining block 606 can use a rolling hash algorithm. An aligning block 608 can align the target data stream with respect to the reference data stream in the reference and target windows using the anchors. A pattern matching block 610 can pattern match between the aligned target data stream and reference data stream to delta compress the target data stream using encoder 612.

[0024] The compressed delta 616 can be stored on a computer readable medium 614. The delta 616 can be later used to decompress the target data stream along with the reference data

stream. The delta 616 can include anchor pairs 618 from the anchor determining block 606 indicating where to align the reference and target data streams. Delta information 620 from the encoder 612 can indicate how to decompress the aligned target data stream with respect to the reference data stream. Thus, the delta 616 can be decompressed to produce the target data stream using the computer readable medium.

[0025] A delta decompressor 700 can use a computer readable medium 614 to provide the delta information 620 and anchor pairs 610. The anchor pairs 610 are provided to an alignment block 702 that aligns the reference window 704 and target window 706. The decompressor block 708 receives the delta information 620 and the aligned reference data and produces the target data stream that is sent to the target window 706.

[0026] The anchors can be selected by using a hash method, such as a rolling hash method. The hash method can be implemented in hardware.

[0027] Figure 8 shows an exemplary anchor pair determination. The reference data stream and target data stream can be streamed into a reference hash window 802 and target hash window 804 respectively. The size of the reference and target hash windows can be different from the size of the reference and target windows. The reference hash window 802 and target hash window 804 can correspond to different offset positions in the reference and target data stream respectively. The hash outputs of the reference and target hash windows will typically often match since the target and reference streams will typically be similar when aligned. In the example of figure 8, both reference and target hash values are "00001101 00000000". An anchor pair can be determined when at least a portion of the reference and target hash values matches a predetermined pattern, such as when the last portion of the hash values match the value "00000000" as in figure 8. The anchor pair can correspond to the offsets when there is a match with the predetermined pattern.

[0028] The reference data stream and target data stream can be streamed into the reference window and target window respectively until an anchor value is reached. At that time, one of the reference or target data streams is stalled until the reference and target data streams are aligned.

[0029] Anchors label the same parts of content between reference and target data stream. Anchors can be represented as the pair (offset of reference anchor, offset of target anchor).

[0030] Figure 3 shows an embodiment of the present invention. Anchors can be read in by the compressor before pattern match. During pattern match, the compressor can adjust reference window pointers according to the anchors 302. The compressor pulls in reference data at a faster pace if the reference offset is bigger than target offset, which would be the result of text being deleted in the target data stream with respect to the reference data. The compressor stalls the

reference window if reference offset is smaller than target offset, which would be the result of text being inserted in the target data stream with respect to the reference data stream.

[0031] Anchors can be determined by rolling hash algorithms. A rolling hash is a hash function where the input is hashed in a sliding window that moves through the input. A few hash functions allow a rolling hash to be computed very quickly -- the new hash value is rapidly calculated given only the old hash value, the old value removed from the hash window, and the new value added to the hash window -- similar to the way a moving average function can be computed much more quickly than other low-pass filters. Hash functions can also be efficiently implemented in hardware. Figure 4 shows an exemplary rolling hash.

[0032] Let us take Rabin-Karp algorithm as example. The Rabin-Karp algorithm is normally used with a very simple rolling hash function that only uses multiplications and additions:

$$H_k = (c_1 a^{k-1} + c_2 a^{k-2} + c_3 a^{k-3} + \dots + c_k a^0) \bmod M, \text{ where } a \text{ is a constant and } c_1, \dots, c_k \text{ are the input characters.}$$

[0033] In order to avoid manipulating huge H values, all math is done modulo M.

[0034] Removing and adding characters simply involves adding or subtracting the first or last term. Shifting all characters by one position to the left requires multiplying the entire sum H_k by a . The calculation of H_{k+1} can be simplified as:

$$H_{k+1} = ((H_k - c_1 a^{k-1}) * a + c_{k+1}) \bmod M$$

[0035] So sweeping through the whole reference data stream, each rolling hash sliding window can generate a hash result. If the hash result is matched with the predefined feature pattern (e.g. a selected number of least significant bit "0"s), the hash result and reference offset are recorded as reference anchor candidate. The hash result is also referred as the fingerprint of the anchor candidate. An anchor candidate can be represented as the pair (anchor offset, anchor fingerprint).

[0036] The target anchor candidates can be determined in the same way. If the fingerprint of the target anchor candidate is same as a reference anchor candidate, an anchor pair is identified.

[0037] The hash result can be updated at the byte level such that a hash value is determined for each byte of the target and reference data stream. For example, for the following data stream: Byte0, Byte1, ..., ByteN-1, ByteN, ByteN+1 ..., if we define the window size to N, the first rolling hash result can be calculated on [Byte0, Byte1, ..., ByteN-1], the second rolling hash result can be calculated on [Byte1, Byte2, ..., ByteN] and the third result can be calculated on [Byte2, Byte3, ..., ByteN+1]. In this way, each byte can correspond to a rolling hash result.

Since the rolling hash drops the oldest byte each time, the complexity of the computation is linear.

[0038] Anchor density can be adjusted. For example, we can configure to identify an anchor pair every 2 KB in average by configuring the feature pattern with the least significant 11 bit “0”s. For density of 1 KB, by configuring the feature pattern with the 10 least significant “0”s. Higher density will result in better delta compression ratio, but more processing in the anchor determination.

[0039] The workflow of one embodiment is shown in figure 5. A rolling hash algorithm is calculated on target and reference data stream in step 501. Anchor candidates are recorded if they match predefined feature pattern so that anchor pairs can be identified later on in step 502.

[0040] Target data and reference data are streamed in for pattern match in step 503. During the pattern match processing in step 504, if an anchor pair is detected, the compressor has to align the reference window and target window. The anchor pair can be represented as the pair (offset of reference anchor, offset of target anchor). The compressor can maintain a reference offset counter and a target offset counter. The reference offset counter can be incremented when a new character is moved into the reference window. The target offset counter can be incremented when a new character is moved into the target window. An anchor is detected when either offset counter hits an anchor in step 506.

[0041] In the alignment process, if reference data stream is ahead of the target data stream, i.e., the compressor meets reference anchor before the corresponding target anchor 507, the compressor can stall the reference window, while target data is streamed in and do pattern match in step 508, until the target offset of the same anchor is met in step 509.

[0042] If target data stream is ahead of reference data stream, i.e., the compressor meets target offset of an anchor first in step 510, the compressor can stall the target window and stream reference data in the reference window in step 511 until the reference offset of the same anchor is met in step 512. No pattern match is performed.

[0043] The post pattern match result is encoded and output.

[0044] During decompression, the same anchor pairs are input to decompressor before decompression. When anchors are detected by decompressor during the processing, decompressor is able to align the reference window and target window to recover data back.

[0045] Experiments show that the invention can use much smaller reference window than other tools. This could simplify the computation complexity and improve performance. Smaller reference window also makes hardware implementation feasible by saving a lot of memory resources on chip.

[0046] The foregoing description of preferred embodiments of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the claims and their equivalents.

5

CLAIMS

1. A delta compression method comprising:
determining anchors to align a reference window and target window for compression of a
5 target data stream in terms of a reference data stream, the anchors being determined by
examining the target data stream and reference data stream;
aligning the target data stream with respect to the reference data stream; and
pattern matching between the aligned target data stream and reference data stream to
delta compress the target data stream.
- 10 2. The delta compression method of claim 1, wherein the anchors are selected by using a
hash method.
3. The delta compression method of claim 2, wherein the hash method is implemented in
15 hardware.
4. The delta compression method of claim 2, wherein the anchors are selected in a rolling
hash method.
- 20 5. The delta compression method of claim 4, wherein the anchors are selected when at least
a portion of the rolling hash values in hash windows of both the reference and target delta stream
match a predetermined value.
- 25 6. The delta compression method of claim 5, wherein the reference data stream and target
data stream are streamed into the reference window and target window respectively until an
anchor value is reached, at that time one of the reference or target data streams are stalled until
the reference and target data streams are aligned.
7. The delta compression method of claim 1, wherein the anchors are stored as the pair
30 (offset of reference anchor, offset of target anchor).
8. A delta decompression method comprising:
using anchors to align a reference window and target window for decompression of a
target data stream in terms of a reference data stream using a delta, the anchors being previously

determined by examining the target data stream and the reference data stream during compression of the target data stream;

decompressing the target data stream using the aligned reference and target windows.

- 5 9. The delta decompression method of claim 8, wherein the anchors are selected in a rolling hash method.
10. The delta decompression method of claim 9, wherein the anchors are selected when at least a portion of the rolling hash values in hash windows of both the reference and target delta
- 10 streams match a predetermined value.
11. The delta decompression method of claim 8, wherein the anchors are stored as the pair (offset of reference anchor, offset of target anchor).
- 15 12. A delta compressor comprising:
- a reference window;
- a target window;
- an anchor determining block to determine anchors by examining the target data stream and reference data stream;
- 20 an aligning block to align the target data stream with respect to the reference data stream in the target and reference windows; and
- a pattern matching block to pattern matching between the aligned target data stream and reference data stream to delta compress the target data stream.
- 25 13. The delta compressor of claim 12, wherein the anchors are selected by using a hash method.
14. The delta compressor of claim 13, wherein the hash method is implemented in hardware.
- 30 15. The delta compressor of claim 13, wherein the anchors are selected in a rolling hash method.

16. The delta compressor of claim 14, wherein the anchors are selected when at least a portion of the rolling hash values in hash windows of both the reference and target delta streams match a predetermined value.
- 5 17. The delta compressor of claim 12, wherein the reference data stream and target data stream are streamed into the reference window and target window respectively until an anchor value is reached, at that time one of the reference or target data streams are stalled until the reference and target data streams are aligned.
- 10 18. The delta compressor of claim 12, wherein the anchors are stored as the pair (offset of reference anchor, offset of target anchor).
19. A computer readable medium containing a delta for decompressing a target data stream; the delta including:
- anchor pairs indicating where to align a reference and target data stream; and
- 15 delta information indicating how to decompress the aligned target data stream with respect to the reference data stream.
20. The computer readable medium of claim 19, wherein the anchors are stored as the pair (offset of reference anchor, offset of target anchor).

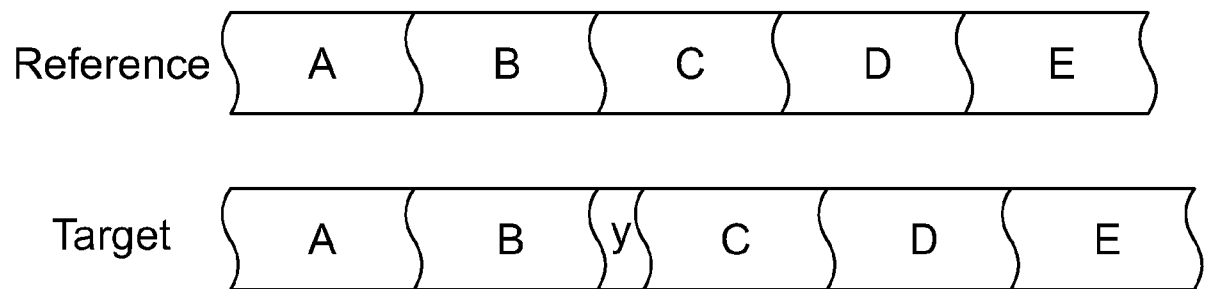


FIGURE 1

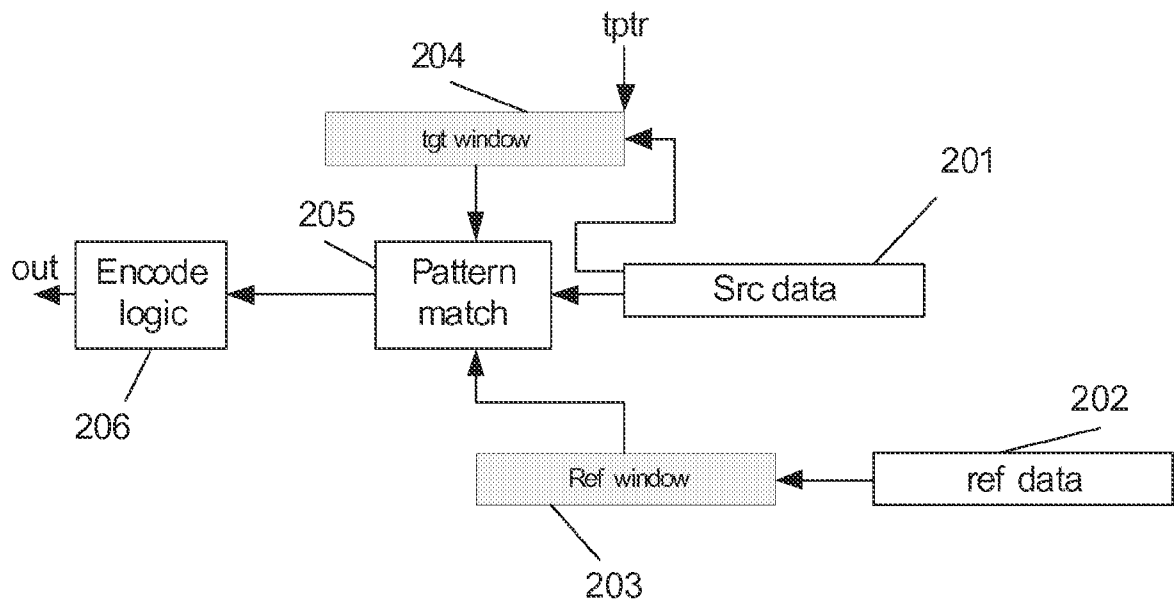


FIGURE 2

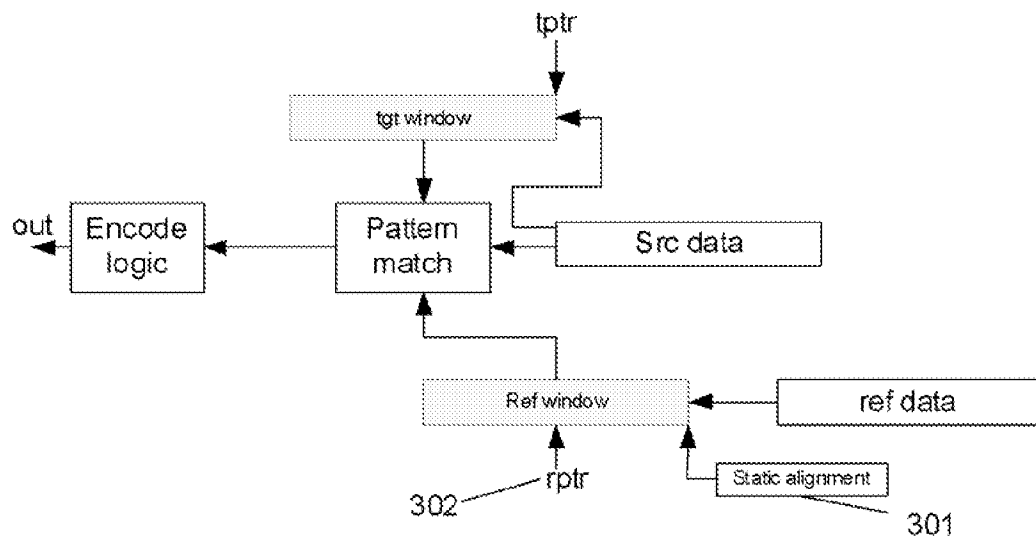


FIGURE 3

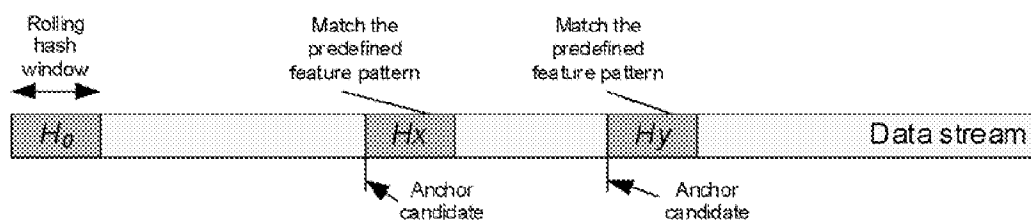


FIGURE 4

3/6

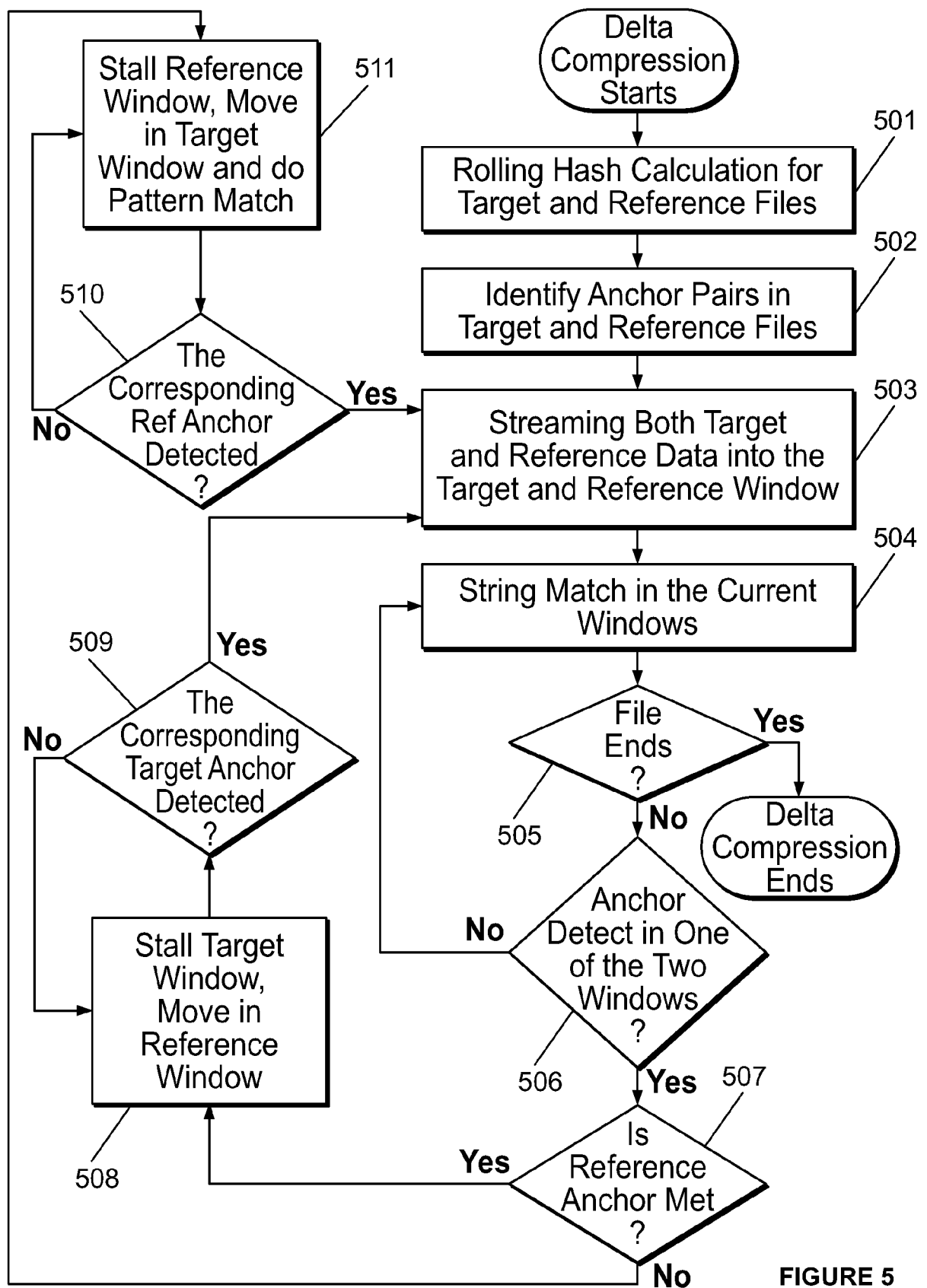


FIGURE 5

4/6

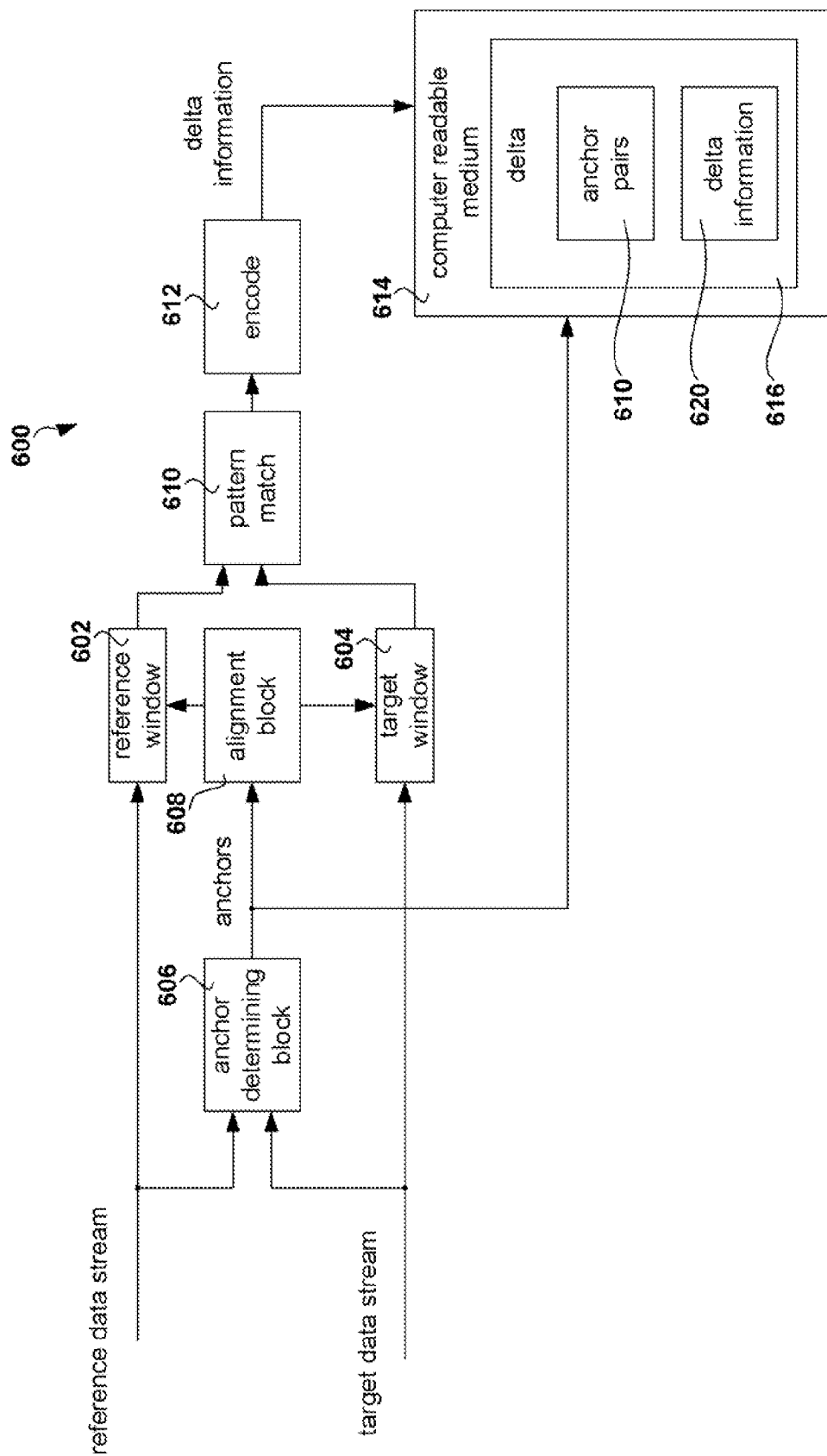


FIGURE 6

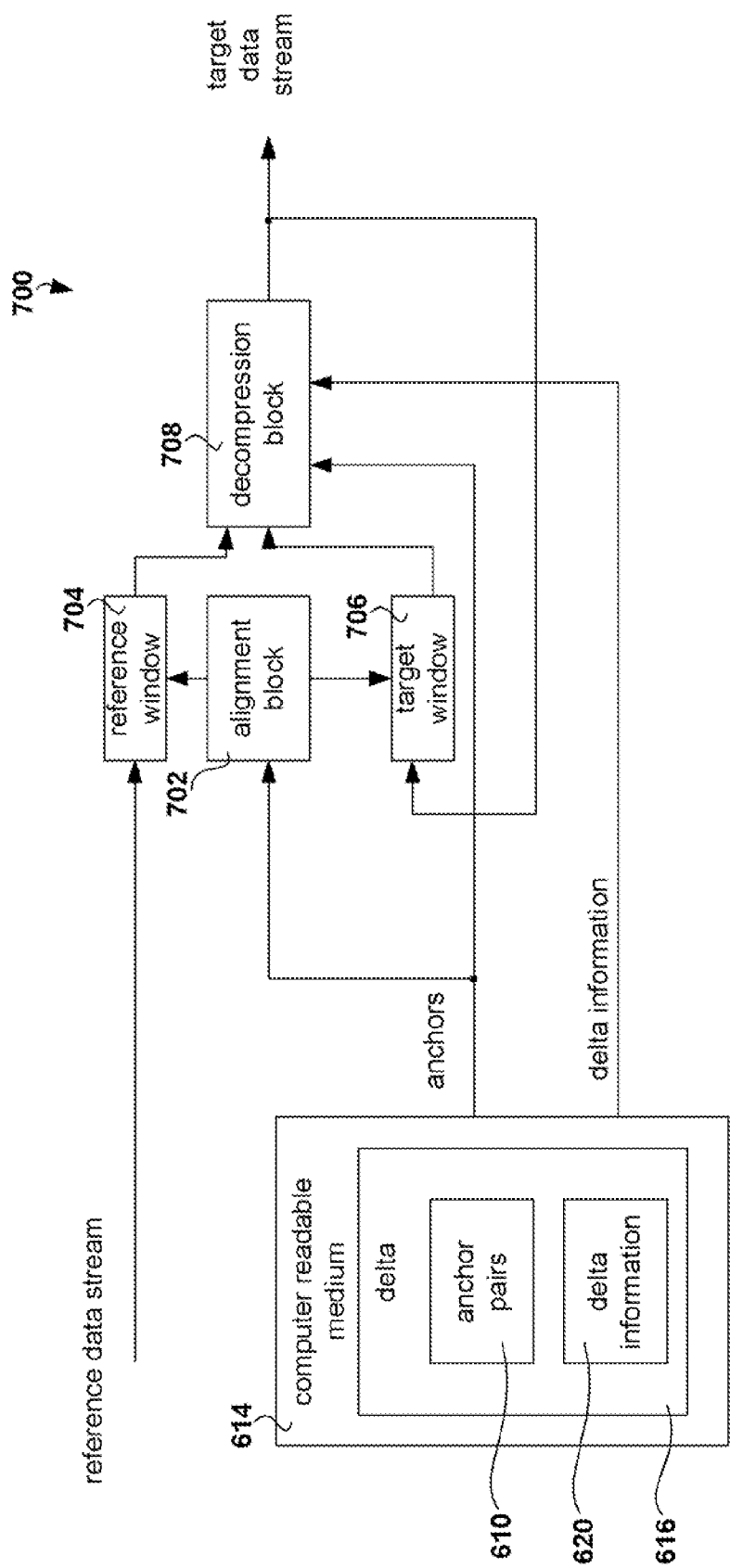


FIGURE 7

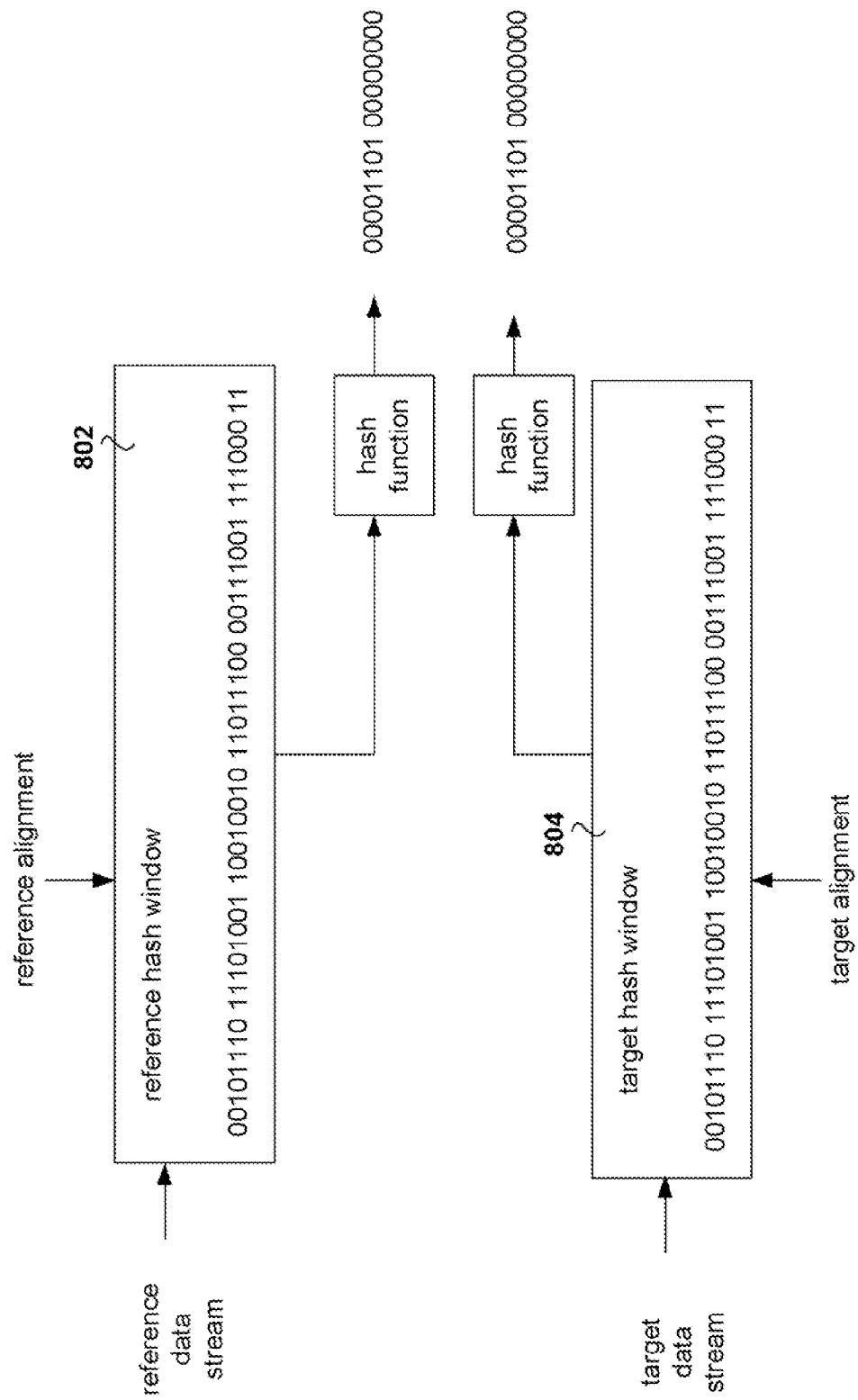


FIGURE 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2012/021882

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - H03M 7/32 (2012.01)

USPC - 341/77

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)

IPC(8) - H03M 3/02, 7/32, 7/36 (2012.01)

USPC - 341/77, 143; 707/693

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)

PatBase, Google Patents, Google Scholar

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7,454,431 B2 (VO et al) 18 November 2008 (18.11.2008) entire document	1-7
---		-----
Y		8-20
Y	Applicant's Admitted Prior Art, para 0002, 0003, 0020	8-11, 19, 20
Y	US 2004/0126021 A1 (SULL et al) 01 July 2004 (01.07.2004) entire document	12-18
A	US 2008/0133561 A1 (DUBNICKI et al) 05 June 2008 (05.06.2008) para 0020, 0038-0044, 0047-0056, fig 2	1-20
A	US 7,079,051 B2 (STORER et al) 18 July 2006 (18.07.2006) col. 4, lines 43-67, fig 2	1-20
A	US 2007/0217649 A1 (LOWE et al) 20 September 2007 (20.09.2007) para 0039-0055, fig 1-4	1-20

☐ Further documents are listed in the continuation of Box C.


* 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

13 March 2012

Date of mailing of the international search report

26 MAR 2012

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents

P.O. Box 1450, Alexandria, Virginia 22313-1450

Facsimile No. 571-273-3201

Authorized officer:

Blaine R. Copenheaver

PCT Helpdesk: 571-272-4300

PCT OSP: 571-272-7774