BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS	**BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE**	**CHAMBRES DE RECOURS DE L'OFFICE EUROPÉEN DES BREVETS**

Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 2 October 2018

Case Number:	T 0650/13 - 3.5.07
Application Number:	02700074.4
Publication Number:	1397866
IPC:	H03M7/30
Language of the proceedings:	EN

Title of invention:
Method and apparatus for adaptive data compression

Applicant:
Google LLC

Headword:
Adaptive data compression/GOOGLE

Relevant legal provisions:
EPC Art. 54(1), 54(2), 56, 84

Keyword:
Novelty - claims 1 and 8 of the main request (yes)
Inventive step - claims 1 and 8 of the main request (yes)
Remittal to the department of first instance - (yes)

Decisions cited:
T 0107/87

This datasheet is not part of the Decision.
It can be changed at any time and without notice.

Europäisches Patentamt
European Patent Office
Office européen des brevets

Beschwerdekammern

Boards of Appeal

Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 0650/13 - 3.5.07

DECISION
of Technical Board of Appeal 3.5.07
of 2 October 2018

Appellant:
(Applicant)

Google LLC
1600 Amphitheatre Parkway
Mountain View, CA 94043 (US)

Representative:

Betten & Resch
Patent- und Rechtsanwälte PartGmbB
Postfach 10 02 51
80076 München (DE)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted on 15 October 2012 refusing European patent application No. 02700074.4 pursuant to Article 97(2) EPC

Composition of the Board:

Chairman R. Moufang
Members: R. de Man
P. San-Bento Furtado

Summary of Facts and Submissions

I. The applicant (appellant) appealed against the decision of the Examining Division refusing European patent application No. 02700074.4, filed as international application PCT/CA02/00143 and published as WO 02/065646.

II. The decision was issued as a so-called decision on the state of the file, referring to a communication dated 9 May 2012 for the reasons for the refusal. That communication cited the following documents:

D2: D. Salomon: "Data compression: the complete reference", 1998, ISBN: 0-387-98280-9, pp. 101-162 and 357-360;

D4: US 5 455 576 A, published on 3 October 1995;

D5: JP 6 161705 A, published on 10 June 1994; and

D6: O. Pentakalos and Y. Yesha: "Online Data Compression in a Mass Storage File System", 1995, retrieved from the Internet at http://wwwinfo.cern.ch/asd/cernlib/storage/papers/14ieee/abstracts/28.

The Examining Division decided that the subject-matter of claims 1, 2 and 8 of the then sole substantive request infringed Article 123(2) EPC and that the subject-matter of claims 1 to 24 was not new in view of document D2.

III. In the statement of grounds of appeal, the appellant replaced the sole substantive request with a main request and an auxiliary request. The main request was the same as the sole substantive request considered in the decision under appeal, with the exception of minor amendments. The auxiliary request was to be considered by the Board only if the main request was found to violate Article 123(2) EPC. The appellant requested oral proceedings "in the event that the Board of Appeal will otherwise refuse either the Main Request or the Auxiliary Request".

IV. In a communication accompanying a summons to oral proceedings, the Board expressed the preliminary view that the main request complied with Article 123(2) EPC but that its independent claims were not clear and lacked support in the description. It appeared to be the case that the subject-matter of an appropriately clarified independent claim 1 would be new over document D2. The Board also noted that the following documents had been cited as "X" documents in the international search report:

D1: EP 0 788 239 A2, published on 6 August 1997; and
D3: US 5 389 922, published on 14 February 1995.

V. In its written submissions, the appellant replaced its requests with an amended main request and first and second auxiliary requests, and it gave reasons why the subject-matter of claim 1 of each request was new and inventive over the cited prior art. The Board then cancelled the oral proceedings.

VI. The appellant requests that the decision under appeal be set aside and that a patent be granted on the basis of the claims of the main request or, in the alternative, on the basis of the claims of either the first or the second auxiliary request.

VII. Independent claim 1 of the main request reads as follows:

"A method for compressing an input string (102) of symbols comprising the steps of:

searching an encoder dictionary (108) for each symbol received in the input string (102), the symbol width being selected dependent on a type of data in the input string; and

upon detecting the symbol is not stored in the encoder dictionary (108), learning the symbol by storing the symbol at a next sequential index in the encoder dictionary (108) and transmitting the symbol in a code word to a decoder (112);

upon detecting that the symbol is stored in the dictionary (108):

> concatenating the symbol with a next symbol in the input string until the concatenated symbols are not found in the encoder dictionary,
>
> transmitting in the code word an index at which a longest prefix match, identified by the concatenating, is stored in the encoder dictionary (108), and
>
> learning the concatenated symbols, not found in the encoder dictionary, by storing the concatenated symbols in the encoder dictionary;

wherein the code word includes a prefix field (404) identifying the contents of the code word, a state of the prefix field (404) indicating whether the code word is a plain symbol (406) to be learned or is an index (408),

the method further comprising:

upon detecting a compression ratio based on the current symbol width is low relative to a threshold which determines the compression ratio at which to change the width, modifying the symbol width and communicating a new symbol width to the decoder in order to increase the compression ratio."

Claims 2 to 7 are directly or indirectly dependent on claim 1.

Independent claim 8 reads as follows:

"An apparatus (110) for compressing an input string (102) of symbols comprising:

an encoder dictionary (108) for storing each symbol received in the string of symbols, a symbol width selected dependent on a type of data in the input string; and

control logic adapted to:

search the encoder dictionary (108) for each symbol;

transmit each symbol in a code word upon detecting the symbol is not stored in the encoder dictionary (108) and learning the symbol by storing the symbol in the encoder dictionary (108); and

upon detecting the symbol is stored in the encoder dictionary:

concatenating the symbol with a next symbol in the input string until the concatenated symbols are not found in the encoder dictionary,

transmitting in the code word an index at which a longest prefix match, identified by the concatenating, is stored in the encoder dictionary (108), and

learning the concatenated symbols, not found in the encoder dictionary, by storing the concatenated symbols in the encoder dictionary;

wherein the code word includes a prefix field (404) identifying the contents of the code word, a state of the prefix field (404) indicating whether the code word is a plain symbol (406) to be learned or is an index (408), the control logic being further adapted to modify the symbol width and communicate a new symbol width to a decoder (112) upon detecting a compression ratio based on the current symbol width is low relative to a threshold which determines the compression ratio at which to change the width in order to increase the compression ratio."

Claims 9 to 12 are directly or indirectly dependent on claim 8.

Independent claim 13 reads as follows:

"A method for decompressing a sequence of code words comprising the steps of:

receiving a code word including a prefix field (404) identifying the contents of the code word, the state of the prefix field indicating whether the code word is a plain symbol (406) to be learned or is an index (408);

upon detecting from the state of the prefix field (404) that a symbol is stored in the code word, learning the symbol by storing the symbol in a next sequential index in a decoder dictionary (108) and providing the symbol as decoded data; and

upon detecting from the state of the prefix field (404) that an index is stored in the code word, accessing the symbol at the index position in the decoder dictionary (108) and providing the symbol as decoded data,

wherein the width of the symbol is variable, the method further comprising:

modifying the symbol width in response to a notification from an encoder (110)."

Claims 14 to 18 are directly or indirectly dependent on claim 13.

Independent claim 19 reads as follows:

"A decoder apparatus (112) for decompressing a sequence of code words comprising:

a decoder dictionary (108) for storing at a next sequential index a plain symbol received in a code word, the code word including a prefix field (404) identifying the contents of the code word, the state of the prefix field (404) indicating whether the code word is a plain symbol (406) to be learned or is an index (408), and

logic adapted to:

receive a code word; detect a plain symbol (406) stored in the code word; store the symbol (406) in the decoder dictionary (108); and provide the symbol as decoded data,

receive a code word; detect an index (408) stored in the code word; access the symbol at the index location in the decoder dictionary (108); and provide the symbol as decoded data; and

modify the symbol width in the decoder apparatus in response to a notification from an encoder (110)."

Claims 20 to 24 are directly or indirectly dependent on claim 19.

VIII. The text of the auxiliary requests is not relevant to the outcome of the appeal.

Reasons for the Decision

1. The appeal complies with the provisions referred to in Rule 101 EPC and is therefore admissible.

2. *The application*

2.1 The application relates to lossless data compression. Its background section describes the well-known Lempel-Ziv-Welch (LZW) compression algorithm. This algorithm employs a dictionary table which is initialised with the symbols of the source alphabet. The number of symbols of the source alphabet is dependent on the symbol width used: a symbol width of 1 byte/8 bits corresponds to 256 (=2^8) table entries, and a symbol width of 2 bytes/16 bits corresponds to 65,536 (=2^{16}) table entries. Increasing the symbol width can improve the compression ratio but quickly leads to an impractical table size.

2.2 The application proposes a compression method and a corresponding decompression method that do not require initialisation of the dictionary table with all the source symbols. Instead, symbols are added to the dictionary as they are encountered in the input data. The symbol width can be selected based on the type of data being compressed and can be modified during the encoding process if the compression ratio drops below a threshold value.

Main request

3. *Added subject-matter - independent claims 1 and 8*

3.1 The objections under Article 123(2) EPC raised in the decision under appeal no longer apply to the claims of the main request. In particular, the feature of independent claims 1 and 8 specifying that the symbol width is modified "upon detecting [that] a compression ratio based on the current symbol width is low relative to a threshold which determines the compression ratio at which to change the width" now adheres closely to the text on page 11, lines 22 to 28, of the published application, and a typographical mistake in claim 2 has been corrected.

3.2 Independent method claim 1 is based on a combination of original claims 1, 3 and 5 to 7 and on original page 9, lines 2 to 7, page 10, lines 2 and 3, page 11, lines 15 to 27, and page 13, lines 9 to 24, of the published application. The subject-matter of claim 1 therefore does not extend beyond the content of the application as filed, in accordance with Article 123(2) EPC.

3.3 Corresponding apparatus claim 8, based on original independent apparatus claims 11, 13 and 15 to 17 and the description passages cited for claim 1, likewise complies with Article 123(2) EPC.

4. *Clarity and support - claims 1 and 8*

The objections of lack of clarity and lack of support raised in the Board's communication no longer apply to independent claims 1 and 8 of the amended main request.

5. *The invention as defined by claim 1*

5.1 Claim 1 is directed to a method for compressing an input string. The input string is divided into input symbols having a particular width, the symbol width being selected based on the type of data in the input string.

5.2 Each input symbol is searched for in an encoder dictionary. If the input symbol is not found in the dictionary, it is added to the dictionary "at a next sequential index", and a code word containing the symbol is transmitted to the decoder.

5.3 If the input symbol is found in the dictionary, the input symbol is concatenated with further symbols from the input string until the string of concatenated symbols is no longer found in the dictionary. When the "longest prefix match" has been determined, a code word containing the index of the longest prefix match in the dictionary is transmitted. The string formed by concatenating the longest prefix match and the next input symbol is then added to the dictionary.

5.4 Each transmitted code word includes a "prefix field" indicating whether (the remaining part of) the code word is a plain symbol or the index of a longest prefix match.

5.5 If the compression ratio based on the current symbol width falls below a threshold, the symbol width is modified in order to increase the compression ratio. The new symbol width is then communicated to the decoder.

The Board notes that changing the symbol width does not necessarily lead to an improved compression ratio, but it accepts that the skilled person attempting to carry out the claimed invention is able to determine circumstances in which changing the symbol width may be expected to be beneficial, for example by detecting the type of the input data currently being processed.

6. *Technicality of claims 1 and 8*

6.1 In particular in view of the feature of claim 1 "transmitting the symbol in a code word to a decoder", the Board is satisfied that the method of claim 1 involves the use of technical means. The subject-matter of independent claim 1 therefore has technical character and is thus not excluded from patentability under Article 52(2) and (3) EPC. The same holds true for the subject-matter of independent apparatus claim 8.

6.2 However, the method of claim 1 and the apparatus of claim 8 can be implemented on a notorious general-purpose computer by means of a computer program. It is important to establish whether the algorithmic features of claims 1 and 8 contribute to the solution of a technical problem. If they do not, they relate to a computer program "as such" and have to be ignored in the assessment of inventive step.

6.3 In decision T 107/87 of 26 April 1991, the board there took the view that a "redundancy-reducing coding rule", enabling a redundant data sequence to be converted into a more compact form, did not have technical character, since it required no technical means for its implementation and achieved no technical effect. A technical result was achieved only if the rule was applied in a technical process such as the storage or transmission of data (see points 2 and 3 of the decision's reasons in combination with section VI of the facts and submissions).

6.4 In the present case, the compression algorithm underlying independent claims 1 and 8 can likewise be regarded as a redundancy-reducing coding rule, more particularly a rule for identifying and eliminating statistical redundancy. This Board agrees with the finding in decision T 107/87 that such a rule contributes to the solution of a technical problem where it is used for the purpose of reducing the amount of data to be stored or transmitted. That condition is complied with, since claims 1 and 8 are in fact explicitly limited to a process for "transmitting" the compressed data. And the Board considers that, where used in an appropriate technical context such as here, the term "compression" itself implies the storage or transmission of the encoded data.

7. *The cited prior art*

7.1 *Document D2*

7.1.1 The Examining Division decided that the subject-matter of claim 1 of the then pending request was not new in view of document D2. It relied, in particular, on section 3.5, which describes the well-known LZ78 compression algorithm.

7.1.2 The LZ78 algorithm is a dictionary-based lossless compression algorithm. As explained in document D2 on page 112, lines 11 to 33, the algorithm starts with an (almost) empty dictionary, to which strings are added as input symbols are processed. When an input symbol, e.g. "x", is read, the dictionary is searched for a corresponding one-symbol string. If it is not found, the symbol is added to the dictionary as a one-symbol string, and a token such as (0, "x") is output, indicating to the decoder a concatenation of the null string and "x". If the one-symbol string is found, the next input symbol, e.g. "y", is read, and the dictionary is searched for the two-symbol string "xy". If that string is not found, it is added to the dictionary, and a token such as (37, "y") is output, consisting of the position in the dictionary of the one-symbol string "x" (here 37) and the symbol "y". This process continues until the end of the input stream is reached.

7.1.3 For a possible way of dealing with the problem that the memory area reserved for the LZ78 dictionary will eventually fill up, document D2 refers on page 115, line 34, to the "UNIX compress utility" described in section 3.15. This section, on page 147, lines 9 to 13, includes the following passage:

"When the largest allowed dictionary fills up, the program continues without changing the dictionary (which then becomes static), but with monitoring the compression ratio. If the ratio falls below a predefined threshold, the dictionary is deleted, and a new, 512-entry dictionary is started. This way, the dictionary never gets 'too out of date'".

7.1.4 The compression method of claim 1 differs from the LZ78 compression algorithm in that each transmitted code word includes a prefix field indicating whether the remainder of the code word is a plain symbol to be learned or an index into the dictionary. In the LZ78 compression algorithm disclosed in document D2, each transmitted code word consists of a position (or index) in the dictionary and a plain symbol, the position being set to zero if no match is found.

The reason why the LZ78 compression algorithm always transmits a plain symbol is that this allows the decoder that decompresses the received compressed sequence of code words to reconstruct the encoder's dictionary by storing in its dictionary the string formed by concatenating the string found at the received dictionary position and the received symbol. In the application, the decoder reconstructs the encoder's dictionary in a different manner: it adds to its dictionary either the received input symbol or the string formed by concatenating the string found at the received dictionary position and the first symbol of the string encoded by the code word received next (see Figure 8B and page 20, lines 20 to 22, and page 21, lines 28 to 30, of the description).

7.1.5 The method of claim 1 further differs from the disclosure of the LZ78 compression algorithm in document D2 in that it includes a step of changing the symbol width when the compression ratio falls below a threshold.

7.2 *Document D3*

7.2.1 Document D3 was cited in the international search report but was not referred to by the Examining Division in the first-instance proceedings. The Board nevertheless considers it to be a relevant document.

7.2.2 In column 6, lines 32 to 64, document D3 explains the LZW compression algorithm. This algorithm first initialises a dictionary with all the possible singlesymbol strings and assigns a distinct index code word to each string. It then reads input symbols and concatenates them until it arrives at a phrase Wa, where W is an initial string of input symbols forming a phrase that is in the dictionary, a is the next input symbol, and the string Wa is not found in the dictionary. Now a code word containing the index in the dictionary of the "longest prefix match" W is transmitted to the decoder, and the phrase Wa is added to the dictionary.

7.2.3 Document D3, in column 7, lines 30 to 41, explains that the initialisation step is inefficient in applications where the alphabet size is large. For example, in an application where the "natural" alphabet for the input data consists of 16-bit symbols, the initial dictionary has 65,536 entries. As explained in column 10, lines 29 to 32, 16 bits is the natural symbol width in certain medical imaging applications, and the best compression is obtained when the natural symbol width is used.

7.2.4 To overcome the problem of the large initial dictionary, a known variation on the algorithm starts with an empty dictionary and uses a special code word to indicate that the next portion of the compressed data represents a new symbol to be learned (column 7, lines 42 to 48).

7.2.5 Document D3 proposes a generalisation of this variation, in which 2^k special code words are reserved for encoding k bits of the new input symbol (column 7, line 60, to column 8, line 3). For k=0, this generalisation corresponds to the known variation (column 8, lines 4 to 7).

7.2.6 In these variations, phrases are added to the dictionary "at a next sequential index" (Figure 5 and column 8, lines 8 to 28; "next available code (NAC)").

7.2.7 Hence, in the variations on the LZW compression algorithm described in document D3, a string of input symbols is compressed by searching a dictionary for each input symbol. If the input symbol is not found, it is added to the dictionary and a code word containing it is transmitted to a decoder in the form of a special code followed by a copy of the new symbol. If the input symbol is found in the dictionary, further input symbols are read until a "longest prefix match" is found. The index of the longest prefix match in the dictionary is transmitted to the decoder, and a string formed by concatenating the longest prefix match and the next input symbol is added to the dictionary.

7.2.8 Document D3 also discloses that the symbol width is "selected dependent on a type of data in the input string" in that it explains that the "natural" symbol width giving the best compression depends on the type of application that produces the data (see point 7.2.3 above) and in that it presents solutions to the problem of the large initial dictionary size resulting from the use of larger symbol widths (see point 7.2.4 and D3, column 4, lines 8 to 11).

7.2.9 When the dictionary fills up, it is either frozen or re-initialised (column 7, lines 2 to 5). As a third approach, the application suggests freezing the dictionary initially and resetting it when the compression ratio deteriorates (column 2, lines 22 to 24).

7.2.10 The subject-matter of claim 1 therefore differs from what is disclosed in document D3 in that:

- a code word includes a "prefix field" indicating whether the remainder of the code word is a plain symbol or an index; and

- when the compression ratio falls below a threshold, the symbol width is changed and communicated to the decoder to increase the compression ratio.

7.3 None of the remaining prior-art documents on file is a more relevant starting point for assessing whether the subject-matter of claim 1 involves an inventive step. In particular, none of them discloses a step of changing the width of input symbols during the encoding process.

8. *Inventive step*

8.1 The compression method of claim 1 differs from the dictionary-based compression methods disclosed in documents D2 and D3 *inter alia* in that it includes a step of changing the symbol width during compression when the compression ratio falls below a threshold, in order to increase the compression ratio.

8.2 Neither document D2 nor D3, nor any of the other cited documents, discloses such a step. Document D3 comes closest, because it mentions that different types of data may have different "natural" symbol widths giving the best compression and addresses the problem of the large initial dictionary size resulting from the use of larger symbol widths. But document D3 contains no suggestion to change the symbol width dynamically during the encoding process.

8.3 In point 4 of the communication referred to in the appealed decision, the Examining Division argued that, if claim 1 were modified or interpreted to be new over document D2, its subject-matter would not be inventive because it was well known in the field of data compression to switch dictionaries and symbol width when different file/content types are to be compressed. It cited passages from document D4 (column 1, lines 33 to 40; column 4, lines 43 to 50; column 6, lines 35 to 50) and from document D5 (paragraphs [0045] and [0046]) as examples of such common general knowledge.

However, the cited passages do not support the Examining Division's contention. The passages in document D4 merely disclose that, in a dictionary-based compression scheme, a dictionary optimised for one type of data is unlikely to lead to efficient compression of another type of data (column 1, lines 33 to 40); that a dictionary may be reset (or switched to a "standby dictionary") when the dictionary is full or when the compression ratio falls below a threshold (column 4, lines 43 to 50; column 6, lines 42 to 50); and that resetting the dictionary frees up storage locations, making them available for storing new character strings (column 6, lines 35 to 42). Nor do paragraphs [0045] and [0046] of document D5, which relate to LZW compression, disclose a step of changing the symbol width.

In fact, although, in the context of dictionary-based compression methods, it was clearly conventional to reset the dictionary under certain conditions, none of the cited documents hints at changing any kind of compression parameter, let alone the symbol width, at the same time as resetting the dictionary.

8.4 Hence, there is no evidence on file that the skilled person, starting from either of documents D2 and D3, would have considered dynamically changing the symbol width when looking for a measure to be taken in response to a deterioration of the compression ratio.

8.5 In sum, the subject-matter of independent method claim 1 and corresponding apparatus claim 8 is new (Article 54(1) and (2) EPC) and involves an inventive step (Articles 52(1) and 56 EPC) over the cited prior art.

9. *Remittal to the Examining Division*

9.1 Although the Board has come to positive conclusions with respect to independent claims 1 and 8 of the main request, the application is not yet ready for grant.

9.2 In particular, independent method claim 13 and independent apparatus claim 19, both directed to the decompression process, have not been amended in view of the lack-of-clarity and lack-of-support objections raised to the then claim 1 in the Board's communication. It therefore still has to be examined whether claims 13 and 19 are clear and supported, in particular whether they need to include features expressing that strings consisting of multiple symbols are stored in the dictionary (see in this respect Figure 8B, steps 815, 842 and 818 and the corresponding passages on pages 20 and 21 of the description; see also point 6.2 of the Board's communication).

9.3 It also still has to be examined whether, in view of the amendments made to the independent claims, the dependent claims are still clear, supported and concise (such that, for example, they do not merely repeat features now present in the independent claims).

9.4 The description may need adapting to the amended claims and may need to acknowledge document D3.

9.5 Hence, it is appropriate to remit the case to the Examining Division for further prosecution on the basis of the main request.

9.6 Since the Board is not refusing any of the appellant's substantive requests, the conditional request for oral proceedings does not apply. This decision can therefore be taken in the written proceedings.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.

2. The case is remitted to the department of first instance for further prosecution.

The Registrar: The Chairman:

Beschwerdekammern des Europäischen Patentamts
Boards of Appeal of the European Patent Office
Chambres de recours de l'Office européen des brevets

I. Aperribay R. Moufang

Decision electronically authenticated