

Internal distribution code:

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

**Datasheet for the decision
of 23 February 2012**

Case Number: T 2140/08 - 3.5.02

Application Number: 02792004.0

Publication Number: 1578020

IPC: H03M 7/42

Language of the proceedings: EN

Title of invention:

Data compressing method, program and apparatus

Applicant:

FUJITSU LIMITED

Headword:

-

Relevant legal provisions:

EPC Art. 84, 111(1)

Relevant legal provisions (EPC 1973):

-

Keyword:

"Clarity - no (main request and first auxiliary request);
- yes (amended second auxiliary request)"

Decisions cited:

-

Catchword:

-



Case Number: T 2140/08 - 3.5.02

D E C I S I O N
of the Technical Board of Appeal 3.5.02
of 23 February 2012

Appellant:
(Applicant)

FUJITSU LIMITED
1-1, Kamikodanaka 4-chome
Nakahara-ku
Kawasaki-shi
Kanagawa 211-8588 (JP)

Representative:

HOFFMANN EITLE
Patent- und Rechtsanwälte
Arabellastraße 4
D-81925 München (DE)

Decision under appeal:

Decision of the Examining Division of the
European Patent Office posted 16 June 2008
refusing European patent application
No. 02792004.0 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman: M. Rognoni
Members: M. Léouffre
P. Mühlens

Summary of Facts and Submissions

I. The appellant (applicant) appealed against the decision of the examining division refusing European Patent application No. 02 792 004.0.

II. In the contested decision, the examining division considered, *inter alia*, that claim 1 then on file lacked some essential features of the invention and that some of the features recited in claim 1 were not clear. The same applied, *mutatis mutandis*, to claim 10. Hence, the examining division found that the application did not meet the requirements of Article 84 EPC.

III. Oral proceedings before the Board were held on 23 February 2012.

IV. The appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of the main request filed with the grounds of appeal, or on the basis of the first auxiliary request filed with letter of 23 January 2012, or on the basis of the amended second auxiliary request filed at the oral proceeding of 23 February 2012.

V. Claim 1 of the main request reads as follows:

"A data compression method that generates compressed data from a data string to be compressed, comprising:

an input step of inputting and retaining, by an input unit, the data string to be compressed in the input buffer;

a list generating step of generating and retaining, by a recent - match - position - list generating unit, a recent match position list having stored therein a relative position where each character string having a predetermined length starting at each address in the input buffer has most recently appeared;

a candidate acquiring step of acquiring, by a repetition candidate acquiring unit, with the use of the recent match position list, a repetition candidate at a position where a character string at a coding position has previously appeared;

a match detecting step of comparing, by a match detecting unit, a character string starting at the position of the acquired repetition candidate and the character string at the coding position, and detecting a matching character string from the position of the repetition candidate; and

a code generating step of coding, by a code generating unit, the detected matching character string."

Claim 1 according to the first auxiliary request reads as follows:

"A data compression method that generates compressed data from a data string to be compressed, comprising:

an input step of inputting and retaining, by an input unit, the data string to be compressed in an input buffer;

a list generating step of generating and retaining, by a recent-match-position-list generating unit, a recent match position list having stored therein position differences at recent-match-position-list-addresses, corresponding to respective input-buffer-addresses, the position differences indicating the distances between the respective input-buffer-addresses where a character string having a predetermined length starts and the input-buffer-addresses where said character string having the predetermined length has most recently appeared;

a candidate acquiring step of acquiring, by a repetition candidate acquiring unit, with the use of the recent match position list, a first repetition candidate and at least one further repetition candidate from said input buffer, said first and at least one further repetition candidate representing the starting position of a string in said input buffer, wherein the first repetition candidate represents a position where a character string at a coding position has most recently appeared;

a match detecting step of comparing, by a match detecting unit, character strings starting at the positions of the first and the at least one further acquired repetition candidates and the character string at the coding position, and detecting a matching character string as the character string having the

longest match length with the character string starting at the coding position; and

a code generating step of coding, by a code generating unit, the detected matching character string with the relative position and match length of the matched character string."

Claim 1 according to the amended second auxiliary request reads as follows:

"A data compression method that generates compressed data from a data string to be compressed, comprising:

an input step of inputting and retaining, by an input unit, the data string to be compressed in an input buffer;

a list generating step of generating and retaining, by a recent-match-position-list generating unit, a recent match position list having stored therein position differences at recent-match-position-list-addresses, corresponding to respective input-buffer-addresses, the position differences indicating the distances between the respective input-buffer-addresses where a character string having a predetermined length starts and the input-buffer-addresses where said character string having the predetermined length has most recently appeared;

a candidate acquiring step of acquiring, by the repetition candidate acquiring unit, with the use of the recent match position list, a first candidate for a repetition position of a character string and at least

one further candidate from said input buffer, said first and at least one further candidate representing the starting position of a string in said input buffer, wherein the first candidate represents a position where a character string at a coding position has most recently appeared, and wherein said at least one further candidate is selected

either by subtracting from a coding position the position difference stored in one of the respective recent-match-position-list-addresses following the coding position, when said position difference exceeds an evaluation value, the evaluation value representing a position difference either between the coding position and the first candidate or between the coding position and a previously acquired candidate,

or by subtracting from the position of the first candidate or of a further candidate the position difference stored in the recent-match-position-list-address at the address corresponding to said first candidate or to said further candidate;

a match detecting step of comparing, by a match detecting unit, character strings starting at the positions corresponding to the first and the at least one further acquired candidates and the character string at the coding position, and detecting a matching character string as the character string having the longest match length with the character string starting at the coding position; and

a code generating step of coding, by a code generating unit, the detected matching character string with the relative position and match length of the matched character string."

Claims 2 to 9 are directly or indirectly dependent on claim 1.

Claim 10 relates to a "computer-readable storage medium which stores a program for compressing data [sic] allowing a computer to execute" steps as recited in claim 1.

Claims 11 to 18 are directly or indirectly dependent on claim 10.

VI. The appellant's arguments relevant to the decision can be summarized as follows:

The present application was concerned with the generation of compressed data from a data stream to be compressed and related to a data compression method which achieved a higher performance than known techniques.

According to a key aspect of the present invention, the search for a string of characters in an input buffer which matched a character string at a coding position did not end as soon as a matching character string was identified. On the contrary, it was extended to areas in the buffer further removed from the coding position. Once the search for matching character strings was completed, the identified character strings were compared with the character string beginning at the

coding position in order to find the string with the longest match. In so doing, better data compression could be achieved.

Claim 1 of the main request comprised a list generating step that generated a recent match position list. Such list, which was based on character strings read into the input buffer, stored, at an address corresponding to an input buffer address, the relative position where a character string identical to the character string of a predetermined length starting at said input buffer address had most recently appeared. The relative position could be the distance counted in address-numbers from the address position where the search started to the address of the input buffer where said character string had recently appeared. However, the skilled person would recognise that the relative position could be determined also in other ways, for instance through the use of pointers when the memory was handled dynamically. As specified in the description (application as filed, page 20, line 14 to page 23 line 12 in conjunction with Figures 7 and 8), what was essential to the invention was generating a recent match position list. Different embodiments, as depicted in the figures, defined, in conjunction with corresponding passages of the description, examples of how to implement such step.

The candidate acquiring step determined repetition candidates which were defined as those elements where a character string found at a coding position had previously appeared. This was in line with the general concept of the invention set out, for instance, on page 21, lines 9 to 13. The embodiments described in

the remaining part of the description and figures represented examples of possible ways of implementing the acquiring step.

The match detecting step compared the character string starting at the candidate address of the repetition candidate with the character string at the coding position and detected a matching character strings starting at the address of the repetition candidate. The skilled person understood from the wording of the claim that the coding position was a position where the method was currently evaluating matching character strings for coding purposes. Furthermore, the skilled person understood from the definition of the detecting step that said step sought to narrow down the number of eligible candidates for a matching character string. Finally, the code generating step performed a coding of the matched character wherein the matched character was determined in the previous step.

Thus, the defined method could be summarised as performing the steps of determining candidate characters that matched characters which had previously appeared, narrowing down the list of candidate strings and detecting among them matching characters that matched the characters at the coding position, *i.e.* at the current position where coding was about to be performed. The matched character string was then coded in order to obtain data compression.

In view of the above considerations, claim 1 according to the main request defined all the technical features necessary for achieving the technical effect of finding

the best match among a list of candidate character strings located in an input buffer.

Claim 1 according to the first auxiliary request recited, *inter alia*, that the candidate acquiring step acquired a first repetition candidate and at least one further repetition candidate. The first and further repetition candidates represented starting positions of the string in the buffer. Furthermore, claim 1 recited that the first repetition candidate represented a position where a character string at a coding position had most recently appeared, as evident from Figures 9A, 12A, 15A and 18 A in combination with corresponding passages of the description. The expression "*most recently*" identified the repetition of the string closest to the coding position among all repetitions of the same string in the input buffer. Since coding was a process progressing over time from a coding position to following coding positions, the expression *most recently* was clear without requiring further amendments.

The feature of the match detecting step had been then amended to recite that a comparison was performed between character strings starting at the positions indicated by the candidates and the character string at the coding position. Moreover, this feature had been amended to clarify that it was for detecting a matching character string as the character string having the longest match length with the character strings starting at the coding position.

In the amended claims, the term "*relative position*" had been replaced by "*position differences*" accompanied by

an extensive definition. Moreover, the whole feature clarified the recent match position list and its addresses. The term "*most recently appeared*" was clearly understood by the skilled person in the context of the claim, since coding (and its related steps of reading in buffers, writing in buffers, coding positions in buffers etc.) was a process which progressed from a certain position to subsequent positions. Thus, the term "*most recently appeared*" referred to the position where the character most recently appeared relative to the position where the process was currently working at.

As claim 1 according to the first auxiliary request defined in a clear manner all the essential features of the invention as disclosed in the application as originally filed, it complied with the requirements of Article 84 EPC.

The claims according to the amended second auxiliary request addressed all the outstanding clarity objections and thus complied with Article 84 EPC.

Reasons for the Decision

1. The appeal is admissible.
- 2.1 The application relates, *inter alia*, to a data compression method for generating compressed data from a data string to be compressed (paragraph [0001] - all citations in the decision refer to the published application).

As explained in paragraph [0013] and illustrated in Figures 5A to 5D, data compression according to the present invention starts by creating a "rank list" 214 and "recent match position list" 216. The "rank list" is obtained "by sorting three-character strings starting at each address in the input buffer 212 in the order of a numerical value". The "recent match position list" 216 shown in Figure 5C relates to the character strings identified in the "rank list" 214 and contains a "relative position" of the "most-recently appearing address", which is the address of a matching character string immediately preceding a given character string located in the input buffer. In the example of Figure 5C, the "relative position" represents the distance, expressed as the difference between the input buffer addresses of the first characters of two consecutive matching strings. Thus, the term "most-recently" implies a sequence of characters in the input buffer, as expressed, for instance, by their respective buffer addresses (see Figure 5A "1, 2, 3 34").

- 2.2 According to the first embodiment of the invention illustrated in Figures 9A and 9B, a "first candidate for a repetition position of a character string" starting at a particular coding position (address 19 of the input buffer of Figures 9A and 9B) is the (relative) position at which such character string (*i.e.* *abc*) appears. It precedes the coding position in the input buffer and is defined as the distance (6) to the coding position (address 19). In this example (see paragraph [0033]), the distance to the coding position of the first character string found to the left of the coding position which matches the character string identified

by the coding position is taken as an "evaluation value".

The "subsequent candidates" are determined as follows:

- starting from the coding position (address 19) and moving to positions with higher addresses, the value stored in the "recent match position list" 24 (Figure 9B), which represents the relative distance between a three-character string starting at that position and the same "recently appeared" character string, is compared with the evaluation value (*i.e.* 6 in the cited embodiment);
- the first stored value greater than the evaluation value (6) defines the "second candidate for a repetition position";
- the next stored value greater than the evaluation value (6) defines the "third candidate for a repetition position" and so on.

These "candidates" are used to determine the starting positions of the character strings to be compared with the character string beginning at the coding position. Each starting position is given by the difference between the coding position (19) and the various "candidates" (10, 18). In the example of Figure 9B, the character string (cbcd...) identified by the "second candidate" starts at input buffer address 9 (*i.e.* 19 - 10), the character string (abcde...) identified by the "third candidate" starts at address 1 (*i.e.* 19 - 18).

- 2.4 The second embodiment illustrated in Figures 12A and 12B differs from the first embodiment only in the way the "evaluation value" is determined.

The value 6 at the coding position (address 19) identifies the starting position (address 13) of the character string (*abc*) matching the three-character string beginning at the coding position. The value 6 at the address 13 identifies a previous position (address 7) where the same character string (*abc*) has appeared. The distance between address 7 and the coding position (address 19) is taken as "evaluation value" (12) for determining the "candidates".

- 2.5 The third embodiment shown in Figures 15A and 15B differs from the previous embodiments in that the "first candidate" is not taken as an "evaluation value" for determining the other candidates, but the "second candidate" is identified by adding the value found in the "recent match position list" at the position defined by the "first candidate" and the "third candidate" by adding the value found at the position defined by "second candidate". No evaluation value, which identifies a position within the buffer prior to which candidates are to be selected, is used in this embodiment.

- 2.6 In the fourth embodiment described in paragraph [0045] and illustrated in Figures 18A and 18B, a character string having the longest match is acquired according to the first embodiment of Figures 9A and 9B. A candidate for that acquired string is newly taken as a first candidate ("revised first candidate"), and then

the process in the third embodiment of Figures 15A and 15B is applied.

Main request

- 3.1 The "*candidate acquiring step*" recited in claim 1 of the main request specifies that "*a repetition candidate*" is acquired "*at a position where a character string at a coding position has previously appeared*". As a "*candidate*" is indicative of the position of a character string relative to the input address of a matching character string (see paragraph [0033]), it is not clear what meaning should be attributed in claim 1 to the step of acquiring a "*repetition candidate*" at a certain position. Furthermore, this definition does not appear to apply to the first embodiment of the invention illustrated in Figure 9A and 9B. In fact, the character string (*abc*) at the coding position 26 is not repeated at the position 30 identified by the "*second candidate*" 30 where the character string (*cbc*) starts (see published application, column 13, lines 2 to 11).
- 3.2 As to the "*match detecting step*" recited in claim 1, it is specified in paragraph [0034] that the "*match detecting unit 18 of Fig. 7 compares a character string starting at each of the addresses of the first candidate and its subsequent candidates and the character string starting at the coding position, and acquires a character string having the longest match length to cause the code generating unit 20 to perform coding*" (underlining added).

Claim 1, however, does not indicate that the match detecting step involves all the acquired repetition

candidates and that the actual purpose of this match detecting step is to identify the character with the longest match length.

3.3 As to the "*code generating step*" recited in claim 1 of the main request, it is specified in the description (column 13, lines 11 to 17) that it is the character string having the longest match length which is acquired for coding, and that coding is performed on the basis of the relative position and match length. In other words, coding according to the present application consists essentially in replacing a character string at a coding position with two parameters (*i.e.* distance to the first string character and match length) which identify the same character string at a previous location within the input buffer. From the wording of claim 1, however, it is not clear according to which criteria a character string is coded and which kind of coding scheme is utilized.

3.4 In summary, claim 1 does not comprise some essential features of the present invention as illustrated in the embodiments of Figures 9A and 9B, 12A and 12B, 15A and 15B, 18A and 18B. Moreover, it contains unclear expressions which do not appear to find full support in the description of the original application.

For these reasons, claim 1 according to the main request does not comply with Article 84 EPC.

First auxiliary request

4.1 According to the "*candidate acquiring step*" recited in claim 1 of the first auxiliary request, a "*first*

repetition candidate" and at least "one further repetition candidate from said input buffer" are acquired "by a repetition candidate acquiring unit", with the use of the recent match position list".

- 4.2 As pointed out above, for a certain coding position, the *"first candidate"* represents the value stored in the recent match position list at the address corresponding to the *"coding position"*. As to the *"further candidates"*, the description of the present application relates to an embodiment (see Figures 15A and 15B) where the *"at least one further candidate"* corresponds to the value stored in the recent match position list at the address corresponding to the *"first candidate"*. According to the other embodiments (see Figures 9A, 9B and 12A,12B) however, the acquisition of at least *"one further candidate"* involves the determination of an *"evaluation value"* and of a position in the input buffer, following the coding position, on the basis of the evaluation value and of the *"relative positions"* stored in the recent match position list at addresses following the address corresponding to the coding position. As these essential features of the invention are not specified in claim 1 according to the first auxiliary request, the requirements of Article 84 EPC are not fulfilled.

Amended second auxiliary request

- 5.1 The method according to claim 1 of the second auxiliary request comprises the following features:

- (a) an input step of inputting and retaining, by an input unit, the data string to be compressed in an input buffer;

- (b) a list generating step of generating and retaining, by a recent-match-position-list generating unit, a recent match position list having stored therein position differences at recent-match-position-list-addresses, corresponding to respective input-buffer-addresses,
 - (b') the position differences indicating the distances between the respective input-buffer-addresses where a character string having a predetermined length starts and the input-buffer-addresses where said character string having the predetermined length has most recently appeared;

- (c) a candidate acquiring step of acquiring by a repetition candidate acquiring unit, with the use of the recent match position list, a first candidate for a repetition position of a character string and at least one further candidate from said input buffer, said first and at least one further candidate representing the starting position of a string in said input buffer

- (d) wherein the first candidate represents a position where a character string at a coding position has most recently appeared, and

- (e) wherein said at least one further candidate is selected
 - (e') either by subtracting from a coding position the position difference stored in one of the respective recent match-position-list-addresses following the coding position, when said position difference exceeds an evaluation value,
 - (i) the evaluation value representing a position difference either between the coding position and the first candidate
 - (ii) or between the coding position and a previously acquired candidate,
 - (e'') or by subtracting from the position of the first candidate or a further candidate the position difference stored in the recent-match-position-list-address at the address corresponding to said first candidate or to said further candidate;
- (f) a match detecting step of comparing, by a match detecting unit, character strings starting at the positions corresponding to the first and the at least one further acquired candidates and the character string at the coding position, and detecting a matching character string as the character string having the longest match length with the character string starting at the coding position; and

(g) a code generating step of coding, by a code generating units, the detected matching character string with the relative position and match length of the matched character string.

5.2 Step (a), which corresponds to the first step recited in claim 1 as originally filed, is self-explanatory.

5.3 Step (b) corresponds essentially to one of the steps also recited in claim 1 as originally filed and relates to the fact that a *"recent-match-position-list generating unit"* generates and retains a *"recent match position list"* of *"position differences"* (defined in (b')) which are stored at addresses corresponding to respective input buffer addresses.

5.4 Feature (b'), which departs from the wording of claim 1 of the filed application, defines the position difference essentially as the distance between the input buffer address of a character string of a predetermined length and the input buffer address of the first identical character string which precedes said character string in the input buffer or, in the wording of the claim, the input buffer address where the same character string *"has most recently appeared"*.

In the letter dated 23 January 2012, the appellant pointed out that, since coding was a process progressing over time from a coding position to the following coding position, the expression *"most recently"* was clear and identified, in fact, the closest repetition of a certain character string at a given input buffer address among all the repetitions of

such character string, which were located before that input buffer address.

An illustrative example of an input buffer and a recent match position list is given in Figures 5A and 5C. As explained in paragraph [0013], a "rank list" shown in Figure 5B is first created by sorting three-character strings starting at each address in the input buffer *"in the order of a numerical value"*. Hence, *"the recent match position list 216 has stored therein a relative position of the most-recently appearing address. For example, a character string "com" from an address 15 has most recently appeared at an address 1 and a relative position 14. Therefore, the relative position 14 is stored in the address 15 in the recent match position list 216"*. In other words, there is a one-to-one correspondence between the addresses of the input buffer and the addresses of the recent match position list.

- 5.5 Step (c) relates to the operation of a *"repetition candidate acquiring unit"* which is responsible for identifying, in a certain area of the input buffer, the positions (*"first candidate"* and *"at least one further candidate"*) of character strings to be compared with the character string starting at the coding position.

With respect to claim 1 of the original application, the claim now distinguishes between *"first candidate"* and *"at least one further candidate"* and clarifies that *"candidates"* identify starting positions of character strings in the input buffer. This is in conformity with the description of the various embodiments, where the position of a character string is identified either in

terms of a position in the input buffer relative to a coding position or in terms of a corresponding input buffer address.

- 5.6 Step (d) specifies that the first candidate represents the position of a character string matching a character string which starts at a coding position.

According to paragraph [0033] (column 11, lines 13 to 15) "a position acquired from the recent match position list 24 is taken as a first candidate for a repetition position of a character string" (underlining added). Furthermore, the "first candidate is taken as an evaluation value" (column 11, lines 16 and 17). It appears, however, that in other parts of the description the term "candidate" is used to designate either the position of a character string in terms of input address or the character string itself. For example, in paragraph [0035] (column 12, lines 21 to 24) it is specified that, "by referring to the recent match position list 24 with the address 19 of the input buffer 12, a stored value at the address 19 is taken as a first candidate for a character-string repetition position" (underlining added). In the following sentence, however, the term "first candidate" appears to relate to the character string: "The position of this first candidate is obtained by $19-6=13$ from the address 19 and its stored value 6, and is therefore a position of an address 13 as indicated by an arrow 36. This means that a repetition character string from the address in the input buffer 12 is taken as a first candidate 28" (underlining added).

However, in view of the overall disclosure in the application and, in particular of the figures illustrating the various embodiments, the meaning attributed to the term "*first candidate*" in the claims is clear and supported by the description.

- 5.7 Feature (e) together with features (e') and (e'') relate to the selection of a "*further candidate*". It distinguishes between a first case, which uses an "*evaluation value*" and "*position differences*" stored in the recent match-position-list at addresses following the coding position in the input buffer, and a second case where no evaluation value is involved.

Feature (e'') intends to cover the possibility that a further character is obtained by increasing the distance from the coding position by:

- (j) moving away from the position corresponding to the first candidate by the position difference stored in correspondence with the first candidate, or
- (jj) moving away from the position corresponding to a further candidate by the position difference stored in correspondence with the further candidate.

Case (j) relates to the embodiment of Figure 15A and Figure 15B. Case (jj) corresponds to Figures 18A and 18B where the further candidate 58 identifying the address 1 in the buffer is obtained by subtracting from the address "4" (a "*further candidate*") the position difference "3" stored at the address "4" in the recent match position list.

5.8 According to (e') the "*further candidate*" is selected by subtracting from the coding position (*i.e.* from its input buffer address) the position difference stored in one of the addresses of the recent match position list following the coding position, when such position difference is greater than an "*evaluation value*". Thus, the evaluation value defines the boundary in the input buffer address where the search for further candidates begins. This boundary can coincide with the first candidate, *i.e.* with the position of the first character string identical to the character string of predetermined length which begins at the coding position and is located in the buffer prior to the coding position in the direction of decreasing addresses.

According to (e''), however, the boundary defined by the evaluation value may coincide with a "*previously acquired candidate*".

The exact wording of feature (e'') is not in the application as originally filed. However, it is supported by the embodiment of Figure 12A and 12B where the evaluation value coincides with a further candidate which would be acquired according of feature (e''). Furthermore, as convincingly argued by the appellant, the teaching of the present application is not limited to setting this boundary at the input buffer address identified by the first candidate or by the second candidate, as shown in the examples according to Figures 9A, 9B and 12A, 12B. Any candidate other than the first or the second candidate could be taken as a boundary for limiting the search of matching character

strings to regions of the input buffer which were farther removed from the coding position.

In fact, it is evident for the skilled reader of the application that the search for further candidates could also be carried out in two separate stages, for instance, by selecting some candidates as specified in step (e'') or (e') together with step (i) and then by taking one of the acquired candidates as an evaluation value to limit the search for further candidates to an area of the buffer which is farther removed from the coding position. As this possibility is directly and unambiguously derivable from the disclosure in the application as originally filed, feature (e'') does introduce subject-matter extending beyond the content of the application as filed (Article 123 (2) EPC).

5.9 Feature (f) differs from the corresponding feature of claim 1 as originally filed in that

The comparison involves character strings starting at positions corresponding to the first and the at least one further acquired candidate

The purpose is to detect the character string having the longest match length with the character string starting at the coding position. This finds for instance support in paragraph [0048]. It is also evident that any number of candidates may be involved in this comparison in accordance with the length of the buffer, in particular with the length of the buffer region to be searched, and also in accordance with the available search time.

- 5.10 Feature g) finds support, for instance in paragraph [0043].
- 5.11 The Board is satisfied that the subject-matter of claim 1 reflects in a manner sufficiently clear and complete the teaching disclosed in the application as originally filed. It thus satisfies the requirements Article 84 EPC and does not offend against Article 123 (2) EPC.
- 6.1 According to the contested decision (see item 1.3), the invention was based on working through the input buffer and thus iterating through coding positions as disclosed in Figure 10. Looping as disclosed in Figure 10 was then an essential feature of the invention.
- 6.2 The Board agrees with the examining division that, as described in the present application, iteration is an integral part of a method for data compression. In fact, the application seeks to provide a method for finding character strings of a predetermined length, obviously greater than 1, which are located in a certain region of the input buffer and match a character string of equal length which starts at a position in the input buffer. The essential purpose of the method recited in this claim is to replace a character string starting at a certain address in the input buffer with the address of the same character string, identified in a certain region of the buffer, and its length. As stressed by the appellant, the gist of the invention consists essentially in extending the search for a matching character string after a first candidate string is found into a region of the buffer that is further

removed from the coding position, in order to increase the chance of finding a character string with a longer match. The Board is satisfied that claim 1 of the amended second auxiliary request comprises all the steps required for defining this essential aspect of the invention, although, in practical terms, it would certainly not make much sense to look for a matching character string only once for all the data stored in the buffer.

Summarizing, the Board agrees that iteration may indeed be considered as an essential feature of the invention, as far as its practical implementation is concerned. However, it is not a feature essential for defining in a clear and concise manner the subject-matter for which protection may be legitimately sought in accordance with Article 84 EPC.

- 7.1 Claim 10 is directed to a "*computer-readable storage medium which stores a program for compressing data [sic] allowing a computer to execute*" the steps recited in claim 1.
- 7.2 In a communication dated 2 January 2006, the examining division essentially objected that the scope of a claim, such as claim 10 then on file, directed to a computer-readable storage medium which stored a program for compressing data allowing a computer to execute the procedural steps recited in this claim was essentially not clear. In fact, such claim defined its subject-matter with reference to a second entity, namely "a computer". According to the examining division, this case was treated in the Guidelines, CIII, 4.8a where it was stated that a lack of clarity was particularly the

case when the claim not only defined the entity itself (the storage medium with a program), but also specified its relationship to a second identity (a computer) which was not part of the claimed entity but was needed for determining the technical features of the first entity. In the present case, the skilled person had no means to determine the resultant restriction of the scope of protection on the first entity (the storage medium with the program) inferred by the second entity (computer), because "a computer" was neither a standardised entity nor was it clear which technical features were present in "a computer". Therefore, no clear restrictions on the use of a computer on the scope of the first entity (the storage media with the program) were present, rendering the scope of the computer readable storage medium which stored a program for compressing data unclear, contrary to Article 84. When considering that a computer could come in many different variations, it became clear that from the term "a computer" it was not possible to infer any restrictions on the claimed entity. In fact, a computer could comprise only an arbitrary processor, a computer system with unknown hardware or a computer system including software but an undefined software collection.

- 7.3 In the statement of ground of appeal (page 20), the appellant essentially argued that a person construing the claim with a mind willing to understand would clearly understand what kind of standard computer was referred to by the claim and which standard technical features of the computer were implied in order to execute the steps defined in such claim. Furthermore the appellant observed that the formulation of claim 10 was indeed a standard formulation for a claim directed

to "a computer-readable storage medium" widely used and largely recognised as clear and patentable under EPO practice.

- 7.4 The "Guidelines for Examination in the European Patent Office" (published in April 2010) point out in C-III.4.14 (corresponding to C-III.4.8a of the previous edition referred to by the examining division) that a lack of clarity can indeed result where a claim in respect of a physical entity (product, apparatus) seeks to define the invention by reference to features relating to the entity's use. *"This is particularly the case where the claim not only defines the entity itself but also specifies its relationship to a second entity which is not part of the claimed entity (for example, a cylinder head for an engine, where the former is defined by features of its location in the latter). Before considering a restriction to the combination of the two entities, it should always be remembered that the applicant is normally entitled to independent protection of the first entity per se, even if it was initially defined by its relationship to the second entity. Since the first entity can often be produced and marketed independently of the second entity, it will usually be possible to obtain independent protection by wording the claims appropriately (for example, by substituting "connectable" for "connected"). If it is not possible to give a clear definition of the first entity per se, then the claim should be directed to a combination of the first and second entities (for example, "engine with a cylinder head" or "engine comprising a cylinder head")."*

It may also be allowable to define the dimensions and/or shape of a first entity in an independent claim by general reference to the dimensions and/or corresponding shape of a second entity which is not part of the claimed first entity but is related to it through use. This particularly applies where the size of the second entity is in some way standardised (for example, in the case of a mounting bracket for a vehicle number-plate, where the bracket frame and fixing elements are defined in relation to the outer shape of the number-plate). However, references to second entities which cannot be seen as subject to standardisation may also be sufficiently clear in cases where the skilled person would have little difficulty in inferring the resultant restriction of the scope of protection for the first entity (for example, in the case of a covering sheet for an agricultural round bale, where the length and breadth of the covering sheet and how it is folded are defined by reference to the bale's circumference, width and diameter, see T 455/92, not published in OJ). It is neither necessary for such claims to contain the exact dimensions of the second entity, nor do they have to refer to a combination of the first and second entities. Specifying the length, width and/or height of the first entity without reference to the second would lead to an unwarranted restriction of the scope of protection" (emphasis added).

In summary, the cited passages of the Guidelines point out that a lack of clarity may arise when some essential parameters of the entity for which protection is sought cannot be defined without specifying a second entity which is not part of the claimed entity.

7.5 In the Board's view, however, the lack of clarity objection identified in the cited item of the Guidelines does not apply to a claim directed to "a computer-readable storage medium" which stores a program allowing a computer to execute certain well-defined steps. Apart from the fact that this claim formulation is in principle accepted at the EPO, as correctly observed by the appellant (cf. Case Law of the Boards of Appeal, I.A.2.4.3 "Claims for a computer program product"), it should be interpreted as covering a computer-readable storage medium which stores a set of instructions readable by a general purpose computer, whereby the stored instructions allow the computer to perform the functions recited in the claim. In other words, it is immaterial for construing this kind of claim to consider which program code and/or which computer architecture may be actually used. Furthermore, as long as no special features are attributed to the computer, the skilled person would have no difficulty in inferring which effect, if any, the reference to a general purpose computer might have on the scope of protection for the claimed computer-readable storage medium.

7.6 In summary, the Board is of the opinion that, as long as the method recited in claim 1 complies with Article 84 EPC, there is no reason to question the clarity of claim 10 directed to a medium which stores a program for allowing a general purpose computer to execute the steps of the method recited in claim 1.

8. Some dependent claims have been amended to bring them into conformity with the independent claims 1 or 10. In

the opinion of the Board, all dependent claims now fulfil the requirements of Article 84 EPC.

9.1 In the examination proceedings and in particular in the contested decision, the examining division considered only the requirements of Article 84 EPC and in particular did not examine the claimed invention as to novelty and inventive step.

9.2 Under these circumstances, the Board considers it appropriate to make use of its power under Article 111(1) EPC and to remit the case to the department of first instance for further prosecution on the basis of the appellant's amended second auxiliary request.

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:

C. Moser

M. Rognoni