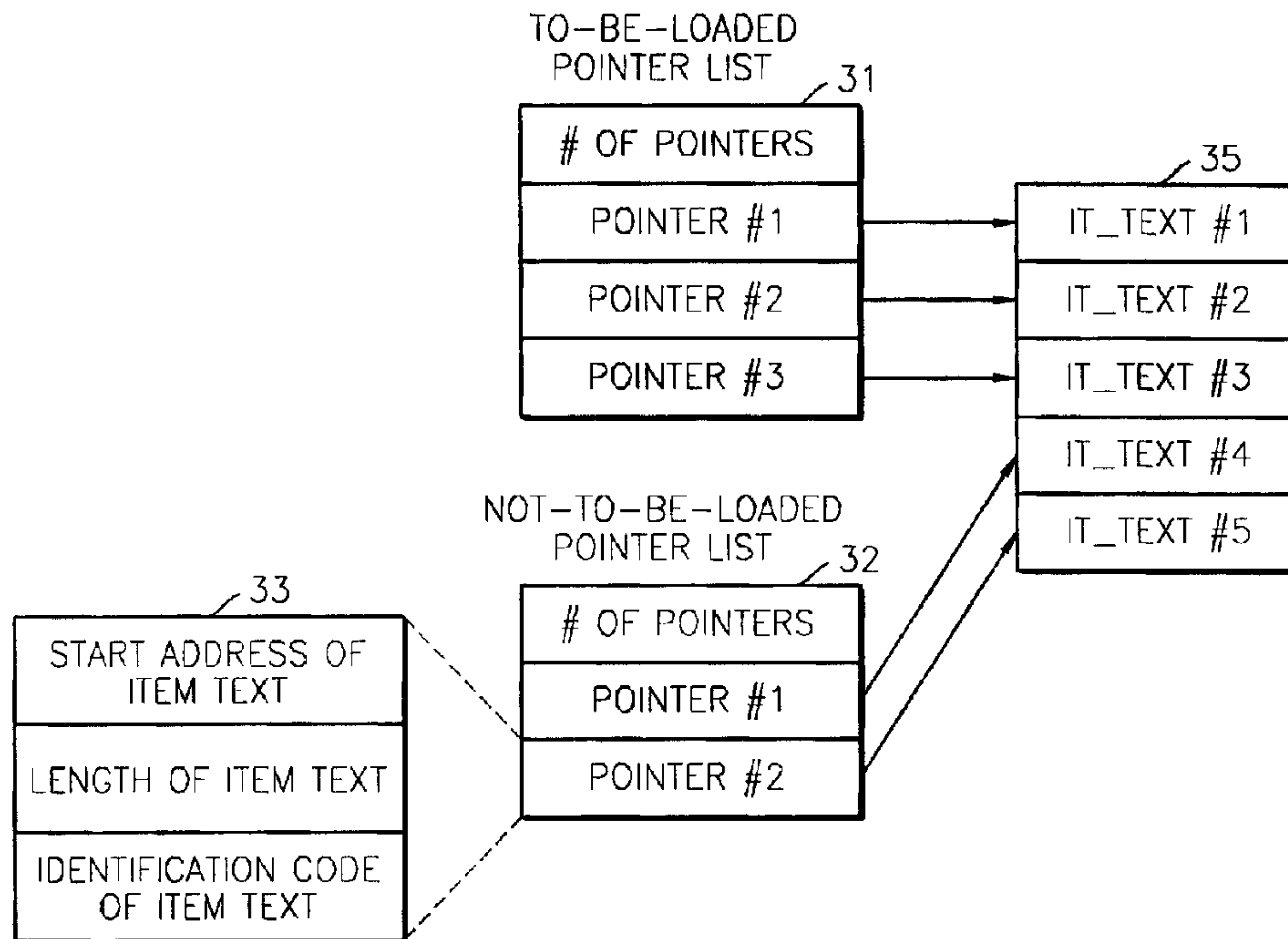




(22) Date de dépôt/Filing Date: 2002/10/22  
 (41) Mise à la disp. pub./Open to Public Insp.: 2003/04/23  
 (45) Date de délivrance/Issue Date: 2007/02/20  
 (30) Priorités/Priorities: 2001/10/23 (KR2001-65392);  
 2002/05/17 (KR2002-27350)

(51) Cl.Int./Int.Cl. *H04N 7/16* (2006.01),  
*G11B 20/12* (2006.01), *G11B 27/10* (2006.01),  
*G11B 27/32* (2006.01), *H04N 5/445* (2006.01),  
*H04N 5/765* (2006.01), *H04N 7/173* (2006.01)  
 (72) Inventeur/Inventor:  
 CHUNG, HYUN-KWON, KR  
 (73) Propriétaire/Owner:  
 SAMSUNG ELECTRONICS CO., LTD., KR  
 (74) Agent: RIDOUT & MAYBEE LLP

(54) Titre : SUPPORT DE STOCKAGE D'INFORMATIONS A STRUCTURE DE DONNEES FACILITANT LA RECHERCHE D'INFORMATIONS TEXTUELLES, ET APPAREIL ET METHODE DE LECTURE DE CE SUPPORT  
 (54) Title: INFORMATION STORAGE MEDIUM HAVING DATA STRUCTURE THAT ALLOWS EASY SEARCHING OF TEXT INFORMATION, AND APPARATUS AND METHOD FOR REPRODUCING THE INFORMATION STORAGE MEDIUM



(57) **Abrégé/Abstract:**

An information storage medium having a data structure that allows easy searching of text data, and an apparatus and a method for playing the information storage medium. The information storage medium includes: AV data that includes audio data and video data, a markup document displayed together with the AV data, and a reproduction control information file containing text data that has a data structure where the text information is classified into a text to be loaded to a memory and a text not to be loaded to the memory for easy search. Therefore, the text data can be searched easily and the memory of the reproducing apparatus can be utilized effectively.

Abstract of the Disclosure

An information storage medium having a data structure that allows easy searching of text data, and an apparatus and a method for playing the information storage medium. The information storage medium includes: AV data that includes audio data and video data, a markup document displayed together with the AV data, and a reproduction control information file containing text data that has a data structure where the text information is classified into a text to be loaded to a memory and a text not to be loaded to the memory, for easy search. Therefore, the text data can be searched easily and the memory of the reproducing apparatus can be utilized effectively.

**INFORMATION STORAGE MEDIUM HAVING DATA STRUCTURE THAT  
ALLOWS EASY SEARCHING OF TEXT INFORMATION, AND APPARATUS  
AND METHOD FOR REPRODUCING THE INFORMATION STORAGE MEDIUM**

5

BACKGROUND OF THE INVENTION

This application claims the benefits of Korean Application No. 2002-27350, filed May 17, 2002, and Korean Application No. 2001-65392, filed October 23, 2001, in the Korean Intellectual Property Office.

10

1. Field of the Invention

The present invention relates to an information storage medium that supports an interactive mode, and a reproducing apparatus and a reproducing method, and more particularly, to text data in an information storage medium containing AV data that is reproduced and displayed on a display window defined by a markup document, and an apparatus and a method for playing the information storage medium.

15

2. Description of the Related Art

A personal computer (PC)-based interactive digital versatile disc (DVD) has been suggested, which can be reproduced in an interactive mode. The interactive DVD refers to the DVD where audio/video (AV) data and a markup document are recorded. The AV data stored in the interactive DVD can be reproduced in two ways, that is, in a video mode or an interactive mode. In the video mode, the AV data is displayed in the same way as video on a regular DVD is displayed, whereas in the interactive mode, the reproduced AV data is displayed in a display window defined by a markup document. If the interactive mode is selected by a user, a browser built in the PC interprets and displays the markup document recorded in the interactive DVD and the AV data selected by the user in the display window defined by the markup document. The representative markup document is an HTML document.

25

30

For example, in a movie whose content is AV data, video of the movie is played in the display window of the HTML document, and in the remaining part of the display screen, a variety of supplementary information including scripts, stories,

photos of actors and actresses, etc., can be displayed. The supplementary information is displayed in synchronization with a title (AV data). For example, when an actor is on stage, a markup document containing his brief history is invoked and displayed together with his video.

5           However, in the interactive mode, a data structure of a text data manager provided by the existing DVD-video standard is disadvantageous in that it is not easy to search text data on the markup document and utilize a memory efficiently.

### SUMMARY OF THE INVENTION

10           Accordingly, it is an object of the present invention to provide an information storage medium having a data structure that allows easy searching of text data in an interactive mode, and an apparatus and a method for playing the information storage medium.

15           It is another object of the present invention to provide an information storage medium having a data structure that allows efficient utilization of a memory, and an apparatus and a method for playing the information storage medium.

          Additional objects and advantages of the present invention will be set forth in part in the description that follows, and, in part, will be obvious from the description, or may be learned by practicing the present invention.

20           To achieve the above and other objects, an information storage medium according to the present invention includes AV data for including audio data and video data. The information storage medium also includes a markup document displayed together with the AV data. The information storage medium also includes a reproduction control information file containing text data which has a data structure  
25           where the text information is classified into a text to be loaded into a memory and a text not to be loaded into the memory for easy searching.

          The markup document contains a source code that requests a search of the reproduction control information file.

30           The source code has an application program interface (API) that has a text type, that identifies whether the text information to be searched is to be loaded into the memory, and a text information identification code, as parameters. The source code has the API that searches text data which is an  $n^{\text{th}}$  previously recorded one from previously searched text data. The source code has the API that searches text data which is the  $n^{\text{th}}$  next recorded one from previously searched text data.

In an embodiment, the API is a script API, for example, a JAVASCRIPT API.

The reproduction control information file includes multiple item texts containing text data which is classified into text to be loaded into the memory and text not to be loaded into the memory. The reproduction control information file includes multiple item texts, multiple pointers, and a pointer list. The multiple item texts contain text data. The multiple pointers indicate the item texts. The pointer list includes pointers indicating text to be loaded into the memory and pointers indicating text not to be loaded into the memory.

The pointer includes a start address, data length, and an identification code of the item text.

To achieve the above and other objects of the present invention, there is provided a reproducing apparatus for reproducing an information storage medium including AV data including audio data and video data, a markup document displayed together with the AV data, and a reproduction control information file containing text data that has a data structure where the text information is classified into a text to be loaded to a memory and a text not to be loaded to the memory for easy search. The information storage medium includes a reader, a cache memory, and a presentation engine. The reader reads data from the information storage medium. The presentation engine interprets the reproduction control information file and caches the text information to the cache memory if the text information is to be loaded into the memory.

In an aspect, the presentation engine identifies whether the text data requested by the API on the script source code is to be loaded or not to be loaded into the memory, reads the text data from the cache memory if the text data is to be loaded into the memory, and reads the text data from the information storage medium through the reader or receives the text data from the Internet if the text data is not to be loaded into the memory.

The presentation engine includes a reproduction control module, a markup document/script language interpretation module, and an object interface module. The reproduction control module searches the reproduction control information file. The markup document/script language interpretation module interprets the markup document, and interprets and executes the API recorded in the markup document. The object interface module interfaces with the reproduction control module and the markup document interpretation module through the API.

To achieve the above and other objects of the present invention, there is provided a method of reproducing an information storage medium including AV data for including audio data and video data, a markup document displayed together with the AV data, and a reproduction control information file containing text data which  
5 has a data structure where the text information is classified into a text to be loaded into a memory and a text not to be loaded into the memory for easy search. In the reproducing method, the reproduction control information file is searched for the text information.

In an aspect, in the reproducing method, the text information is searched from  
10 the reproduction control information file, and the searched text data is cached in the cache memory. In an aspect, an identification is made as to whether the text information requested by an API on a script source code is to be loaded into the memory, and the text data is read from the memory if the text information is to be loaded into the memory.

#### BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and advantages of the invention will be set forth in part in the description which follows and, in part, will be obvious from the description, or may be learned by practice of the invention.

20 FIG. 1 shows a directory structure of a disc, according to an embodiment of the present invention;

FIGS. 2A and 2B show a data structure of a reproduction control information file according to an embodiment of the present invention;

25 FIG. 3 shows a data structure of a text data manager, according to an embodiment of the present invention;

FIG. 4 shows an embodiment of the text data manager shown in FIG. 3;

FIG. 5 shows an embodiment of the text data information (TXTDTI) shown in FIG. 4;

FIG. 6 shows an embodiment of the IT\_TXT\_SRP shown in FIG. 4;

30 FIG. 7 is a block diagram of a reproducing apparatus, according to an embodiment of the present invention;

FIG. 8 shows an operation procedure of a presentation engine, according to an embodiment of the present invention in detail; and

FIG. 9 is a flowchart explaining a reproducing method, according to an

embodiment of the present invention.

### DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the present preferred embodiments  
5 of the present invention, examples of which are illustrated in the accompanying  
drawings, wherein like reference numerals refer to the like elements throughout.  
The embodiments are described below in order to explain the present invention by  
referring to the figures.

A markup document defined in this specification means not only a  
10 document prepared in a markup language such as hypertext markup language  
(HTML) and extensible markup language (XML), but also a document linked with or  
including a source code or execution code prepared in JAVASCRIPT and JAVA, and  
further means a file used being linked to or inserted into the markup document.

FIG. 1 shows a directory structure of a DVD, according to an embodiment of  
15 the present invention.

With reference to FIG. 1, a root directory 102 includes a video directory,  
VIDEO\_TS 104, which contains AV data. In VIDEO\_TS 104, there is a file  
VIDEO\_TS.IFO 106 that contains reproduction control information on all of the video  
titles. In the file VIDEO\_TS.IFO 106, language information is designated as a  
20 default value with regard to the video title. In addition, there is the file  
VTS\_01\_0.IFO 108 that contains reproduction control information on a first video title  
set. In addition, there are the files VTS\_01\_0.VOB 110 and VTS\_01\_1.VOB 112,  
..., which are video objects (VOBs) that make up video title sets. In each VOB,  
there is a video object unit (VOBU) that contains a navigation pack, and at least one  
25 of both a video pack and an audio pack. For more detailed configuration  
information, the DVD-Video standard 「DVD-Video for Read Only Memory disc 1.0」  
should be referred to.

In the root directory, there is also an interactive directory, DVD\_ENAV 114,  
which contains the information for supporting the interactive mode. The  
30 DVD\_ENAV 114 contains DVD\_ENAV.IFO 116, which is a reproduction control  
information file according to an embodiment. For example, the reproduction control  
information file includes definition and configuration of the interactive directory, the  
number of included titles, basic information on the titles, information on languages,  
captions and fonts used for the titles, markup document display information such as

resolution and color, and information on a copyright. Text data of the information is recorded in the data structure according to the present invention, and the detailed structure will be described below. Further, the DVD\_ENAV 114 contains the file INDEX.XML 118 for displaying an initial start markup document, and other markup documents. In addition, in the DVD\_ENAV 114, a file, which is to be linked to and displayed together with markup documents such as the INDEX.XML 118, can be stored. Further, other files 120 may be included in the root directory.

FIGS. 2A and 2B show a directory structure of a reproduction control information file according to an embodiment of the present invention.

With reference to FIG. 2A, the reproduction control information file DVD\_ENAV.IFO 116 contains a text data manager 202. With reference to FIG. 2B, the reproduction control information file DVD\_ENAV.IFO 116 can include an ENAV management information table 204, an ENAV HTML displayable language unit table 206, and a text data manager 208. In an embodiment, text data manager 202 and text data manager 208 refer to the same thing. The ENAV management information table 204 is a management table designed to manage the ENAV HTML displayable language unit table 206 and the text data manager 208. The ENAV management information table 204 can include an identifier of the reproduction control information file, the size of the reproduction control information file, a DVD identifier, a start address and an end address of the ENAV HTML displayable language unit table, and a start address and an end address of the text data manager. The ENAV HTML displayable language unit table 206 designates the types of displayable languages for the markup document. However, because the table that designates the types of displayable languages for a markup document is not within the scope of the present invention, a detailed explanation about the table will be omitted.

In the text data manager 208, there is a data structure that contains the text data which allows for more efficient searching according to the present invention.

FIG. 3 shows a data structure of the text data manager 208.

With reference to FIG. 3, text data manager 208 contains multiple item texts IT\_TEXT 35 where text data is recorded. In addition, there are pointer lists 31 and 32 including multiple pointers that indicate the time text IT\_TEXT 35. The pointer lists, according to an embodiment of the present invention, are classified into memory-loadable text data and memory-unloadable text data. The reference number 31 indicates the list of the pointers indicating the item text IT\_TEXT 35

where the text data to be loaded to the memory is recorded. The reference number 32 indicates the list of the pointers indicating the item text IT\_TEXT 35 where the text data not to be loaded to the memory is recorded. Each pointer includes a start address of an item text, a data length, and an identification code. That is, a  
 5 separate item text search table is prepared, which helps to identify whether the text data is to be loaded to the memory or not.

FIG. 4 shows an embodiment of the text data manager 208 shown in FIG. 3.

With reference to FIG. 4, in the text data manager, text data information (TXTDTI) 402, which contains header information about pointers, is first stored.

10 The text data information is followed by LD\_IT\_TXT\_SRP #n 404, pointers which indicate an item text loadable to a memory, and UL\_IT\_TXT\_SRP #m 406, pointers which indicate an item text unloadable to the memory. Also included are IT\_TXT 408 that are item texts.

FIG. 5 shows an embodiment of TXTDTI 402 shown in FIG. 4.

15 With reference to FIG. 5, the TXTDTI field 402 includes CHRS 502, a reserved field 504, LD\_IT\_TXT\_SRP\_Ns 506, UL\_IT\_TXT\_SRP\_Ns 508, and TXTDT\_MG\_EA 510. CHRS 502 contains a code for designating a character set of a character used in the item text. LD\_IT\_TXT\_SRP\_Ns 506 denotes the number of pointers that indicate the item text that is to be loaded to the memory.  
 20 UL\_IT\_TXT\_SRP\_Ns 508 denotes the number of pointers that indicate the item text that is not to be loaded to the memory. TXTDT\_MG\_EA 510 indicates the end address of the text data manager.

FIG. 6 shows an embodiment of the IT\_TXT\_SRP fields 404 and 406 shown in FIG. 4.

25 With reference to FIG. 6, the IT\_TXT\_SRP fields 404 and 406 include IT\_TXT\_SA 602, IT\_TXT\_SZ 604, and IT\_TXT\_IDCD 606. IT\_TXT\_SA 602 indicates the start address of the item text. IT\_TXT\_SZ 604 indicates the size of the item text. IT\_TXT\_IDCD 606 indicates the identification code of the item text.

30 Here, the size of the to-be-loaded text data of the entire memory should be determined. If IT\_TXT\_SA 602 of a first to-be-loaded item text is searched through the pointer indicating a first to-be-loaded item text, IT\_TXT\_SA 602 of a first not-to-be-loaded item text is searched through the pointer indicating a first not-to-be-loaded item text and, the size of the to-be-loaded text data can be calculated using the difference between the IT\_TXT\_Sas 602. Further, the length of

the to-be-loaded/not-to-be-loaded text data is limited to several characters. The item text is described and generated in the designated character set.

FIG. 7 is a block diagram of a reproducing apparatus 200 according to an embodiment of the present invention.

5 With reference to FIG. 7, the reproducing apparatus 200 decodes AV data recorded in the disc 100, reproduces it as an AV data stream and displays the AV data stream on the display window defined by the markup document in the interactive mode. The reproducing apparatus 200 includes a reader 1, a buffer memory 2, a cache memory 3, an AV decoder 4, a presentation engine 5, and a  
10 blender 7.

The reader 1 reads a markup document or AV data from the disc 100. The buffer memory 2 buffers the AV data read by the reader 1. The cache memory 3 caches the markup document read by the reader 1. Especially, the cache memory 3 caches memory-loadable text data among the text data stored in the reproduction control information file. The presentation engine 5 interprets the read markup  
15 document and identifies the location of the display window, then displays the AV data stream output by the AV decoder 4 on the display window. Further, the presentation engine 5 searches the reproduction control information file for memory-loadable text data and caches the memory-loadable text data on the cache memory 3. If an  
20 application program interface (API) on a script source code requests text data, the presentation engine 5 identifies whether the text data is loadable or not loadable to the memory. If the text data is loadable to the memory, the presentation engine 5 reads the text data from the cache memory 3. If the text data is not to be loaded to the memory, the presentation engine 5 reads the text data from the disc 100 through  
25 the reader 1 or from the outside. The blender 6 blends the decoded AV data and the markup document, and outputs them to the display device 300.

FIG. 8 shows an operation procedure of the presentation engine.

30 With reference to FIG. 8, from the perspective of software, the presentation engine 5 includes a reproduction control module 802, an object interface module 804, and a markup document/script language interpretation module 806, which connects an API and the object interface module. API is a special method pre-determined on a script language in which a requested control operation is written when a source code written in a script language is interpreted by the script language interpretation module and then executed.

The reproduction control module includes text data 812 made up of item text, and a text search engine 814 designed to search the text data 812. The text search engine 814 searches text data, DVD\_ENAV.IFO 116, which has the data structure according to the embodiment.

5 The object interface module 804 performs interfacing in which the script language interpretation module 806 transfers parameters and methods defined on an object interface on the basis of an API to the reproduction control module. To do so, the object interface module 804 has a text data search interface 808 and a text data extraction interface 810. The text data search interface 808 sends a parameter  
10 provided through the API by the markup document/script language interpretation module 806, that is, the information about the text data 812 to be searched, to the text search engine 814. The text data extraction interface 810 transmits the text data 812 searched by the text search engine 814 to the markup document/script language interpretation module 806.

15 The markup document/script language interpretation module 806 interprets and displays the markup document. That is, the markup document/script language module 806 analyzes the markup document and performs text data searching if an API for searching for the text data 812 is called. The text data 812 is for searching the AV data, and displaying the text information (for example, names of actors and  
20 scripts, and etc.) related to the AV screen. The parameters of the API includes the information that identifies whether the item text is to be loaded or not to be loaded to the memory, that is, text type (textType) and the identification code (IDCD) of the item text which contains text information to be searched. To be more specific, the markup document/script language interpretation module 806 identifies the  
25 identification code (IDCD) of the text data upon the request of an API on a script source code, and sends the textType and the IDCD to the reproduction control module 802 via the object interface module 804 through the FindTextData, API. The navigation module searches the relevant text search table depending on the received text type. If the navigation module finds the identification code identical to the  
30 identification code to be searched, it searches the start address of the relevant item text, fetches the requested text data and sends the text data back to the markup document/script language interpretation module 806 through the object interface module 804. Likewise, if the navigation module uses FindPreviousData(n) and FindNextData(n), it can find the text data which is an n<sup>th</sup> previously recorded one or

which is an  $n^{\text{th}}$  next recorded one at the position where the previously found text data is recorded.

The API used in the embodiment will be described in detail.

### <Example of API>

5

#### 1. Method

##### 1.1. .FindTextData(textType,IDCD)

###### 1) Summary

A text data (character string) is searched according to the parameter provided by the text data manager.

10

###### 2) Parameter

textType = 0 : item text loadable to the memory

1 : item text not loadable to the memory

IDCD : identification code

###### 3) Example

15

obj.FindTextData(0,30) // Find the item text whose IDCD is 30 out of the item texts loadable to the memory.

##### 1.2. .FindPreviousData(n)

###### 1) Summary

$N^{\text{th}}$  previously recorded item text is searched at the location where the previously found item text is recorded.

20

###### 2) Parameter

n :  $N^{\text{th}}$  previous text from the previously found item text

###### 3) Example

25

obj.FindPreviousData(10) // Find the item text which is  $10^{\text{th}}$  previously recorded one from the previously found item text.

##### 1.3. .FindNextData (n)

###### 1) Summary

$N^{\text{th}}$  next recorded item text is searched at the location where the previously found item text is recorded.

30

###### 2) Parameter

n :  $N^{\text{th}}$  next text from the previously found item text

###### 3) Example

obj.FindNextData (10) // Find the item text which is  $10^{\text{th}}$  next recorded one from the previously found item text.

## 2. Property

### 2.1. .TextData

#### 1) Summary

Indicates the character string of the searched item text.

#### 2) Return value

Character String.

#### 3) Example

`x = obj.TextData // Copy the searched text data on x.`

### 2.2. .TextDataIDCD

#### 1) Summary

Indicates the identification code value of the searched item text.

#### 2) Return value

Number

#### 3) Example

`x = DvdVideo.TextDataIDCD // Copy the IDCD of searched text on x.`

Based on the above configuration, the reproducing method according to an embodiment of the present invention will be described below.

FIG. 9 is a flowchart explaining a reproducing method.

With reference to FIG. 9, a reproducing apparatus searches and reads the reproduction control information file recorded on the disc 700, at 901. At 902, the reproducing apparatus caches only memory-loadable text data in the cache memory 3 by interpreting the read reproduction control information file. If the API on a script source code within the displayed markup document requests text information at 903, the reproducing apparatus identifies whether the requested text information is to be loaded to the memory at 904. If the text information is to be loaded to the memory, the reproducing apparatus 200 reads the relevant text data from the cache memory 3, at 905. If the text information is not to be loaded to the memory, the reproducing apparatus reads the relevant text data from the disc 100 or from the outside, at 906.

The above reproducing method can be implemented as a computer program. The codes and code segments making up the program can be inferred easily from the computer program in a relevant field. In addition, the program is saved in a computer readable medium, and read and executed by the computer to be implemented as a method of recording text information to and reproducing it from

an information storage medium. The computer readable medium includes magnetic recording media, optical recording media, and carrier wave media.

As described above, the present invention provides the information storage medium having a data structure that allows easy searching of text data, and an apparatus and a method for playing the information storage medium. Therefore, the memory of the reproducing apparatus can be utilized efficiently. In addition, the user can search easily the huge amounts of the text information as if he is browsing the web. As a result, if the markup document of the same content is provided in multiple languages, or the text information such as the script of the video title is provided in multiple languages, the data structure according to the present invention helps to record the text data.

Especially, according to the present invention, a hash table structure, a linked list structure, and a binary tree structure can be applied for easy data searching.

Although a few embodiments of the present invention have been shown and described, it would be appreciated by those skilled in the art that changes may be made in these embodiments without departing from the principles and spirit of the invention, the scope of which is defined in the claims and their equivalents.

What is claimed is:

1. A reproducing apparatus for reproducing an information storage medium including AV data including audio data and video data, a markup document displayed together with the AV data, and a reproduction control information file containing text data that has a data structure where text information is classified into a text to be loaded to a memory and a text not to be loaded to the memory for easy searching, the reproducing apparatus comprises:
- 5
- a reader that reads data from the information storage medium;
  - 10 a cache memory; and
  - a presentation engine that searches the reproduction control information file for the text information which is classified into a text to be loaded to the memory and caches the text information to the cache memory if the text information is to be loaded to the memory.
- 15
2. The reproducing apparatus of claim 1, wherein the presentation engine identifies whether the text data requested by an API on a script source code is to be loaded or not to be loaded to the memory, and reads the text data from the cache memory if the text data is to be loaded to the memory or reads the text data from the information storage medium through the reader if the text data is not to be loaded to the memory.
- 20
3. The reproducing apparatus of claim 2, further comprising:
- a network data transceiver for receiving a markup document from the Internet,
  - 25 wherein if the text data is not to be loaded to the memory, the presentation engine receives text data from the Internet via the network data transceiver.
- 30
4. The reproducing apparatus of claim 1, wherein the presentation engine comprises:

a reproduction control module that searches the reproduction control information file;

a markup document/script language interpretation module that interprets the markup document and also interprets and executes the API in a script language recorded in the markup document; and

an object interface module that interfaces the reproduction control module with the markup document/script language interpretation module through the API.

10 5. The reproducing apparatus of claim 4, wherein the reproduction control module includes a text search engine that searches the text data included in the reproduction control information file, wherein the markup document/script language interpretation module interprets and executes the markup document and the script source code, and wherein the object interface module sends a

15 text type and an identification code of the text information to be searched, which is provided by the markup document/script language interpretation module through the API, to the text search engine, and transmits the searched text data from the text search engine back to the markup document/script language interpretation module.

20

6. The reproducing apparatus of claim 5, wherein, if the text search engine searches a relevant text search table according to the text type sent by the object interface module, and finds the identification code identical to the identification code to be found, the text search engine searches the start

25 address of the relevant item text and reads the text data.

7. The reproducing apparatus of claim 5, wherein the text search engine reads  $n^{\text{th}}$  previously or next recorded text data from the location where the previously found text data is recorded.

30

8. The reproducing apparatus of claim 1 further comprising:  
a buffer memory buffering the read AV data;

an AV decoder decoding the read AV data; and  
a blender unit blending the decoded AV data so that the AV data is  
displayable on the display window defined by the markup document.

- 5 9. A method of reproducing from an information storage medium including  
AV data for including audio data and video data, a markup document  
displayed together with the AV data, and a reproduction control information  
file containing text data that has a data structure where text information is  
classified into a text to be loaded to a memory and a text not to be loaded to  
10 the memory for easy search, the reproducing method comprising:  
searching the reproduction control information file for the text  
information which is classified into a text to be loaded to the memory and  
caching the text data relating to the searched text information in a cache  
memory; and  
15 reading the text data from the cache memory or the information storage  
medium.
10. The method of claim 9, wherein the searching the reproduction control  
information file further comprises:  
20 identifying whether the text information requested by an API on a script  
source code is to be loaded to the memory; and  
reading the text data from the memory if the text information is to be  
loaded to the memory.
- 25 11. The method of claim 10, wherein the searching the reproduction control  
information file further comprises reading the text data from the information  
storage medium if the text data is not to be loaded to the memory.
12. The method of claim 10, wherein the searching the reproduction  
30 control information file further comprises receiving the text data from the  
Internet if the text data is not to be loaded to the memory.

13. An apparatus, comprising:  
a reader unit to read audio/video (AV) data, a markup document, and a reproduction control information file from an interactive digital versatile disc (DVD),

5 wherein the markup document comprises text information and a source code to request a search for the text information, and

wherein the reproduction control information file classifies the text information as to be loaded into a memory or not to be loaded into the memory;

10 a cache memory to store the text information that is to be loaded into the memory; and

a presentation engine to search the cache memory for the text information that is stored in the cache memory and retrieve the text information searched for in the cache.

15

14. The apparatus of claim 13, wherein the presentation engine is implemented in software.

15. The apparatus of claim 14, wherein the presentation engine comprises:

20 a reproduction control module to search the markup document;

a markup document/script language interpretation module to interpret the markup document for display; and

25 an object interface module to provide an interface between the reproduction control module and the markup document/script language interpretation module.

16. The apparatus of claim 13, wherein the reproduction control information file includes a text data manager, which includes a data structure for easy searching.

30

17. The apparatus of claim 16, wherein the data structure is a hash table, a linked list, or a binary tree.

18. The apparatus of claim 16, wherein the data structure comprises:  
a first plurality of pointers to a first portion of the text information that is  
classified as to be loaded into the memory; and  
a second plurality of pointers to a second portion of the text information  
5 that is classified as not to be loaded into the memory.

FIG. 1

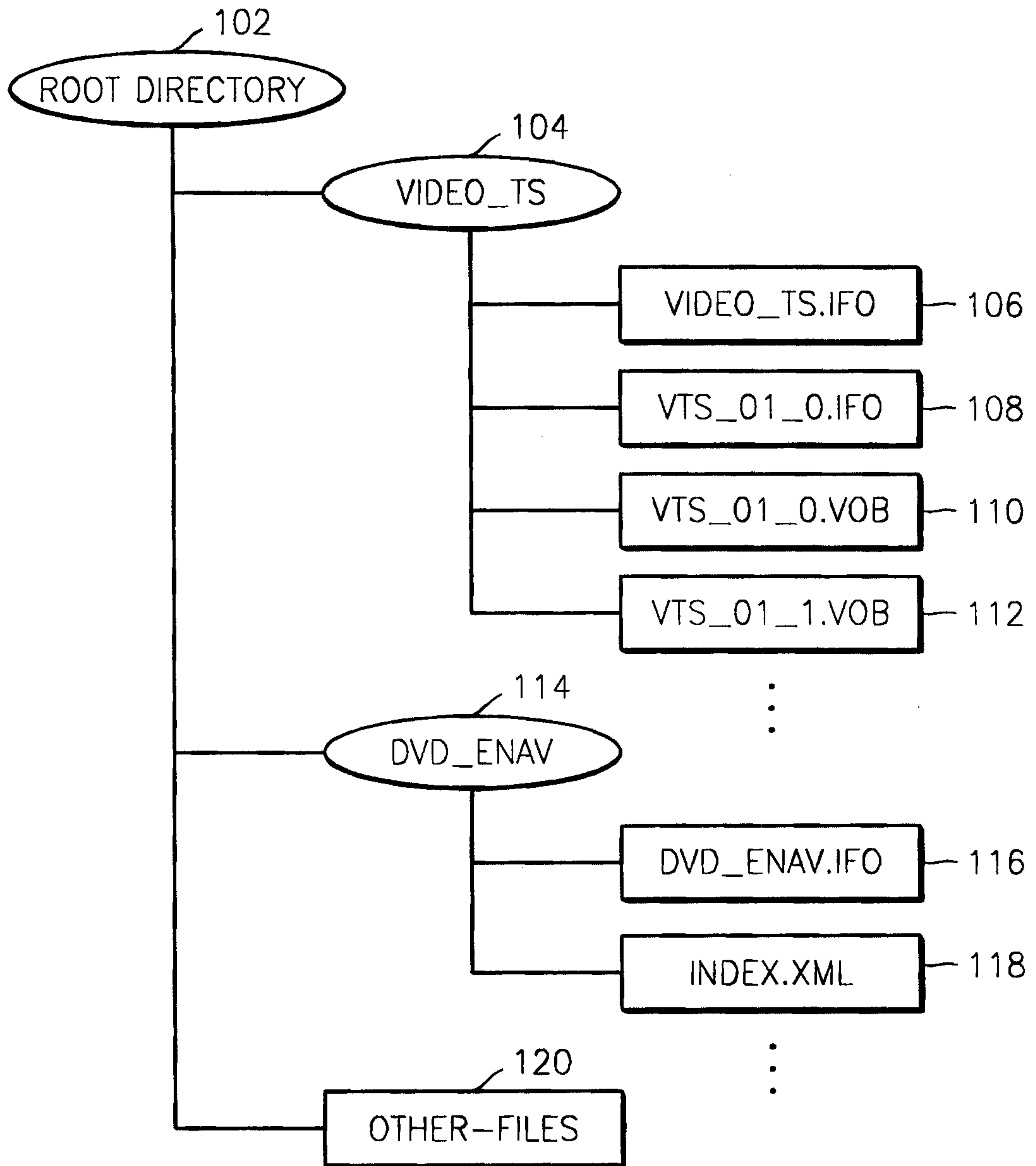


FIG. 2A

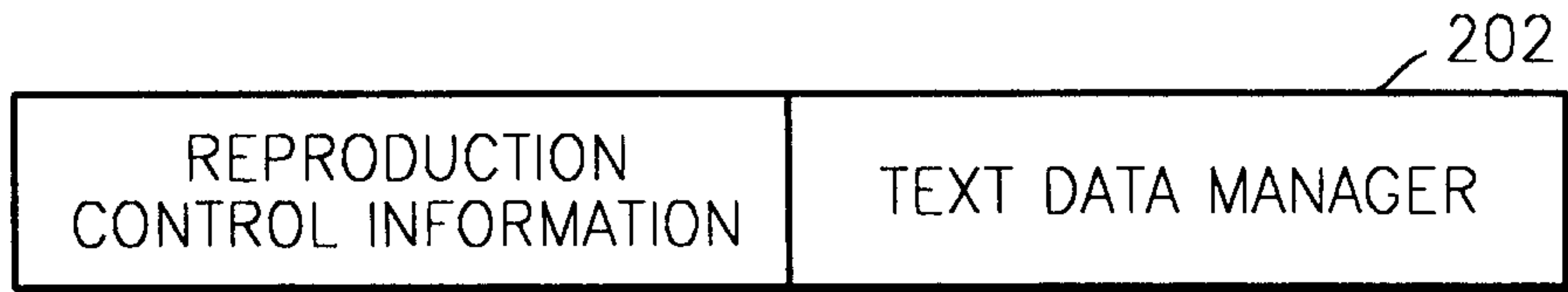


FIG. 2B

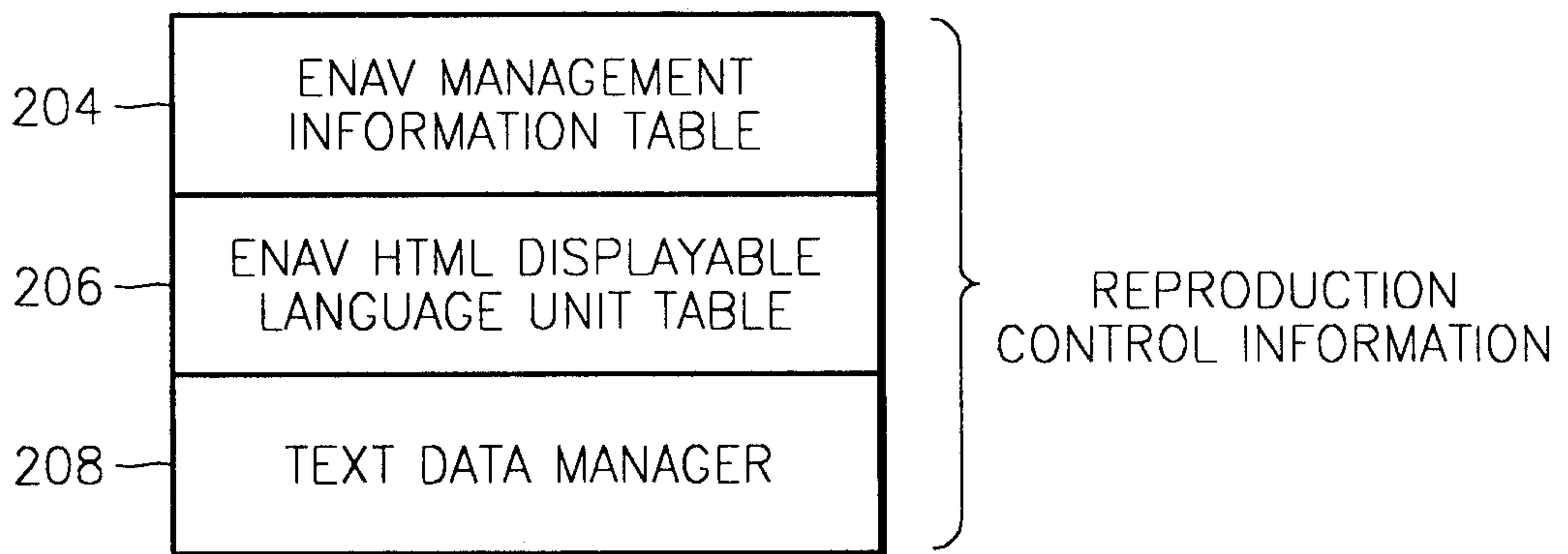


FIG. 3

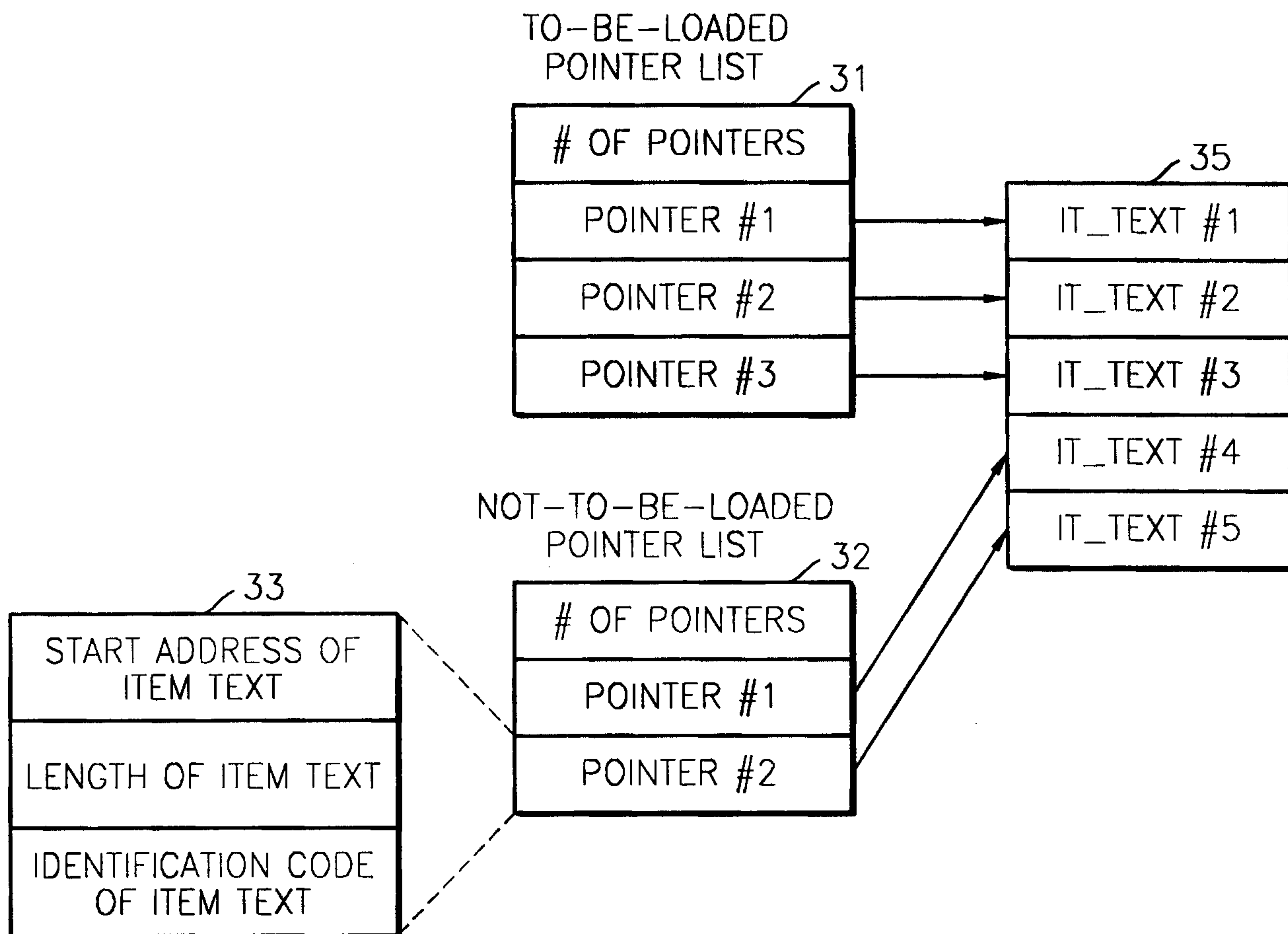


FIG. 4

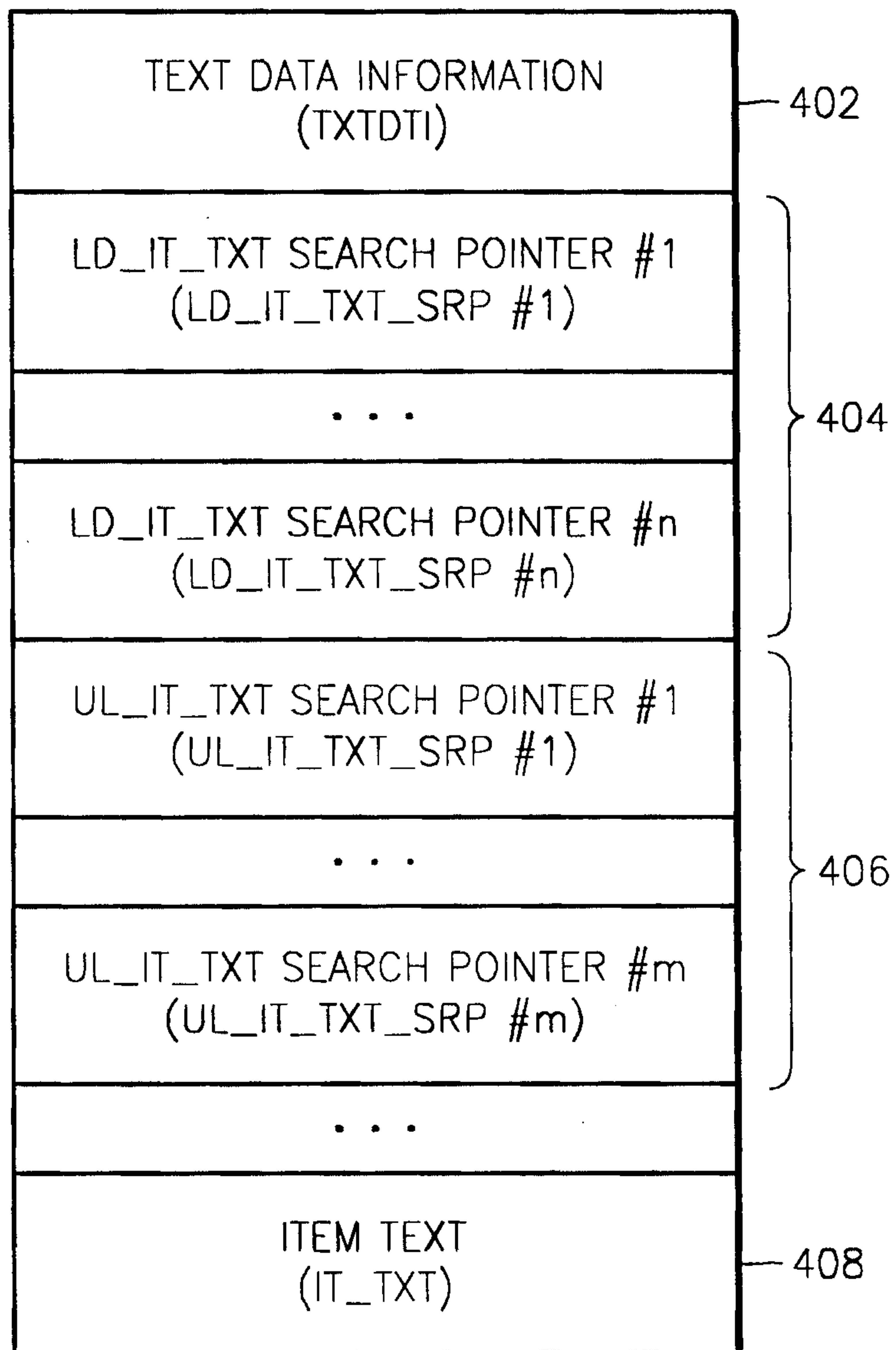


FIG. 5

	FIELD NAME	CONTENTS
502	CHRS	CHARACTER SET
504	RESERVED	RESERVED
506	LD_IT_TXT_SRP_Ns	NUMBER OF LD_IT_TXT_SRP
508	UL_IT_TXT_SRP_Ns	NUMBER OF UL_IT_TXT_SRP
510	TXTDT_MG_EA	END ADDRESS OF TXTDT_MG

FIG. 6

	FIELD NAME	CONTENTS
602	IT_TXT_SA	START ADDRESS OF IT_TXT
604	IT_TXT_SZ	SIZE OF IT_TXT
606	IT_TXT_IDCD	ITEM TEXT IDENTIFIER CODE

FIG. 7

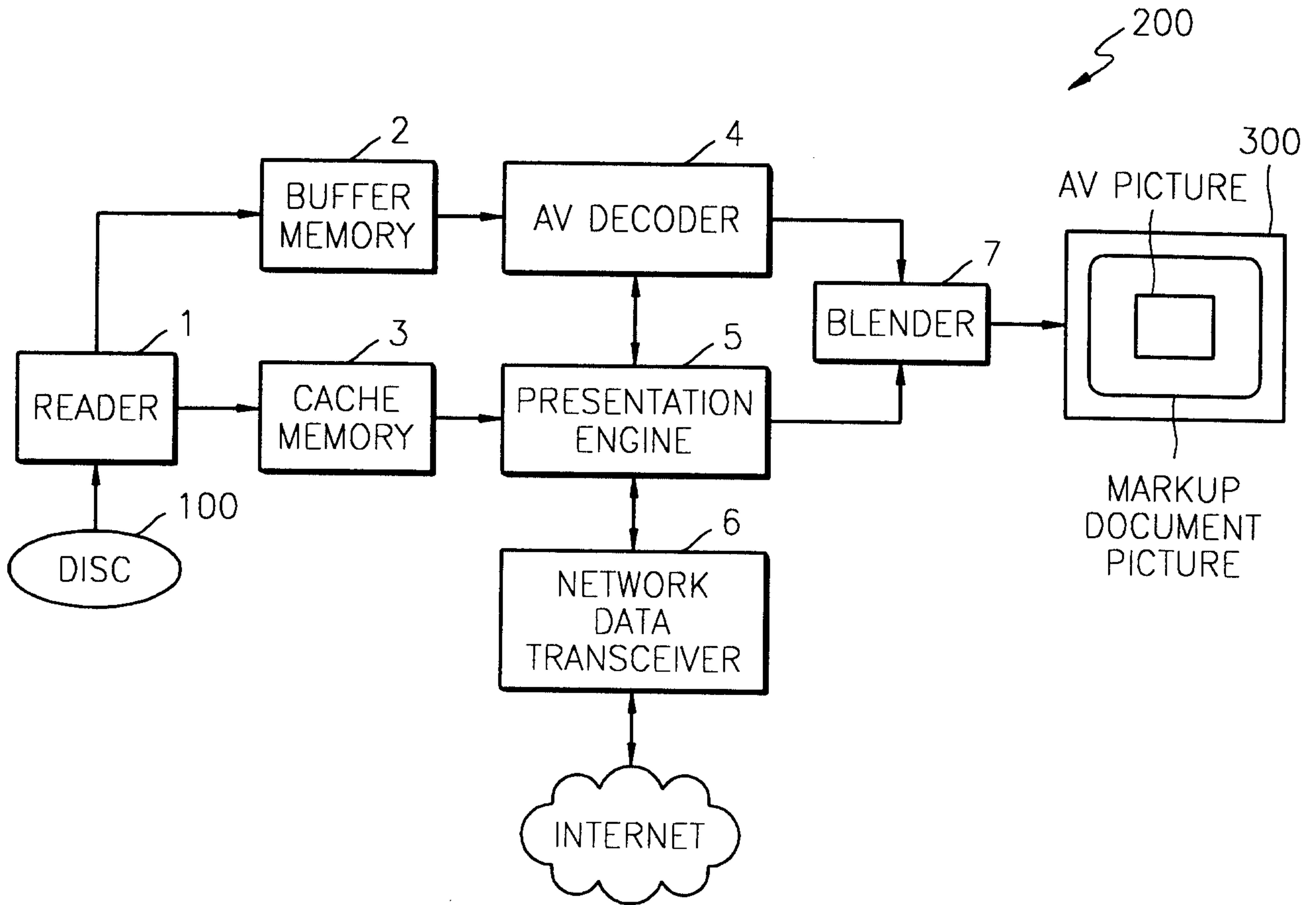


FIG. 8

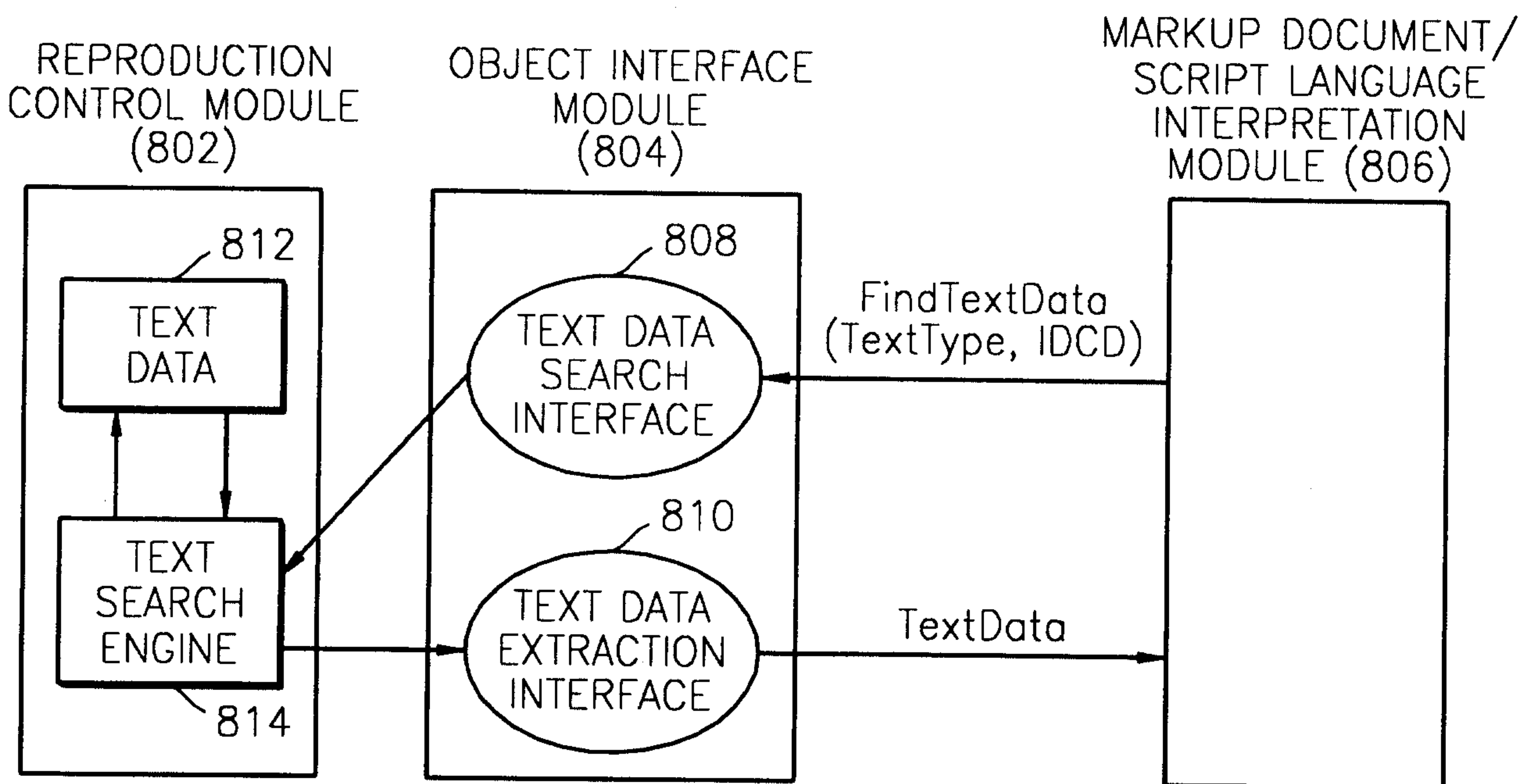
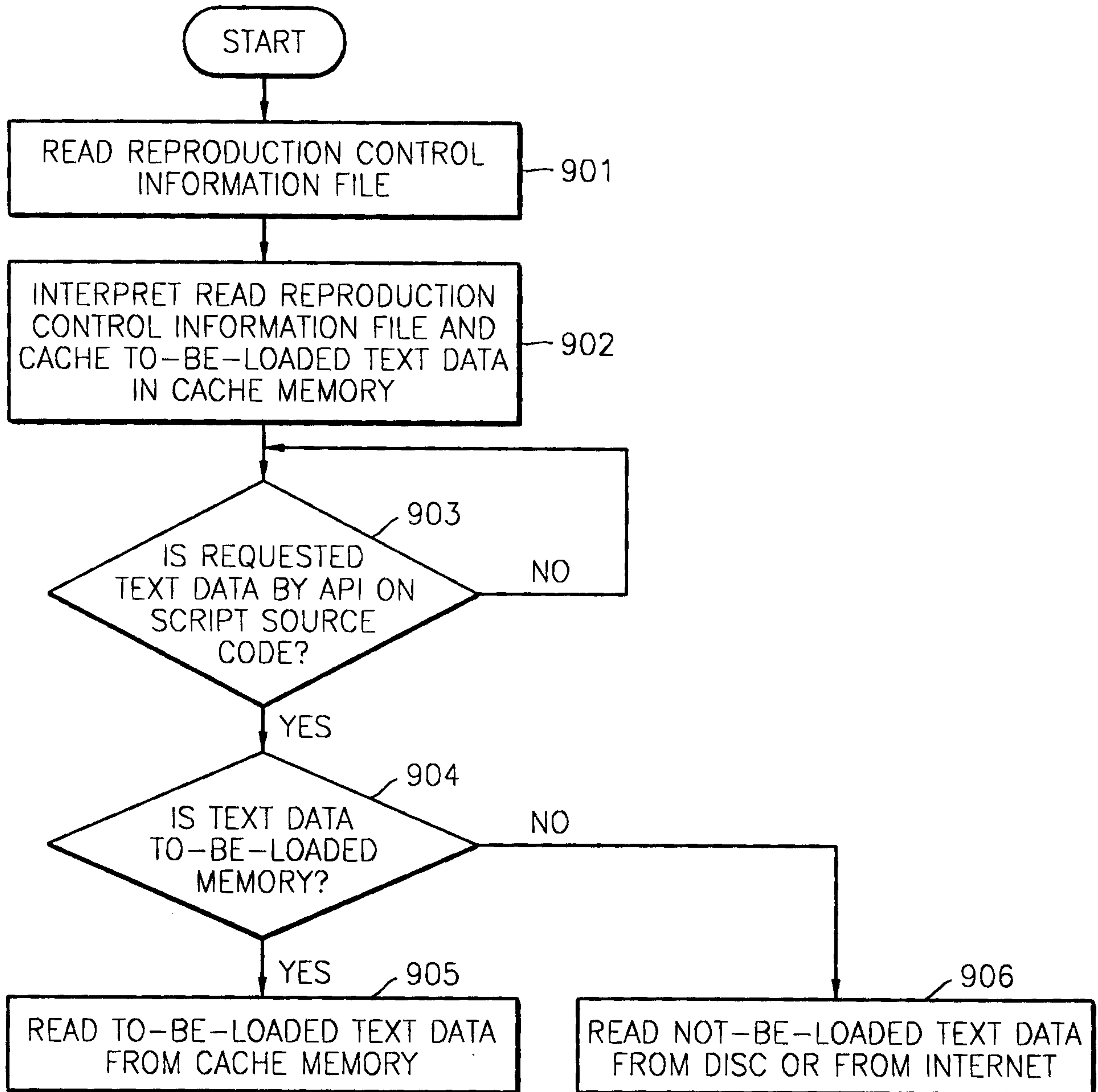
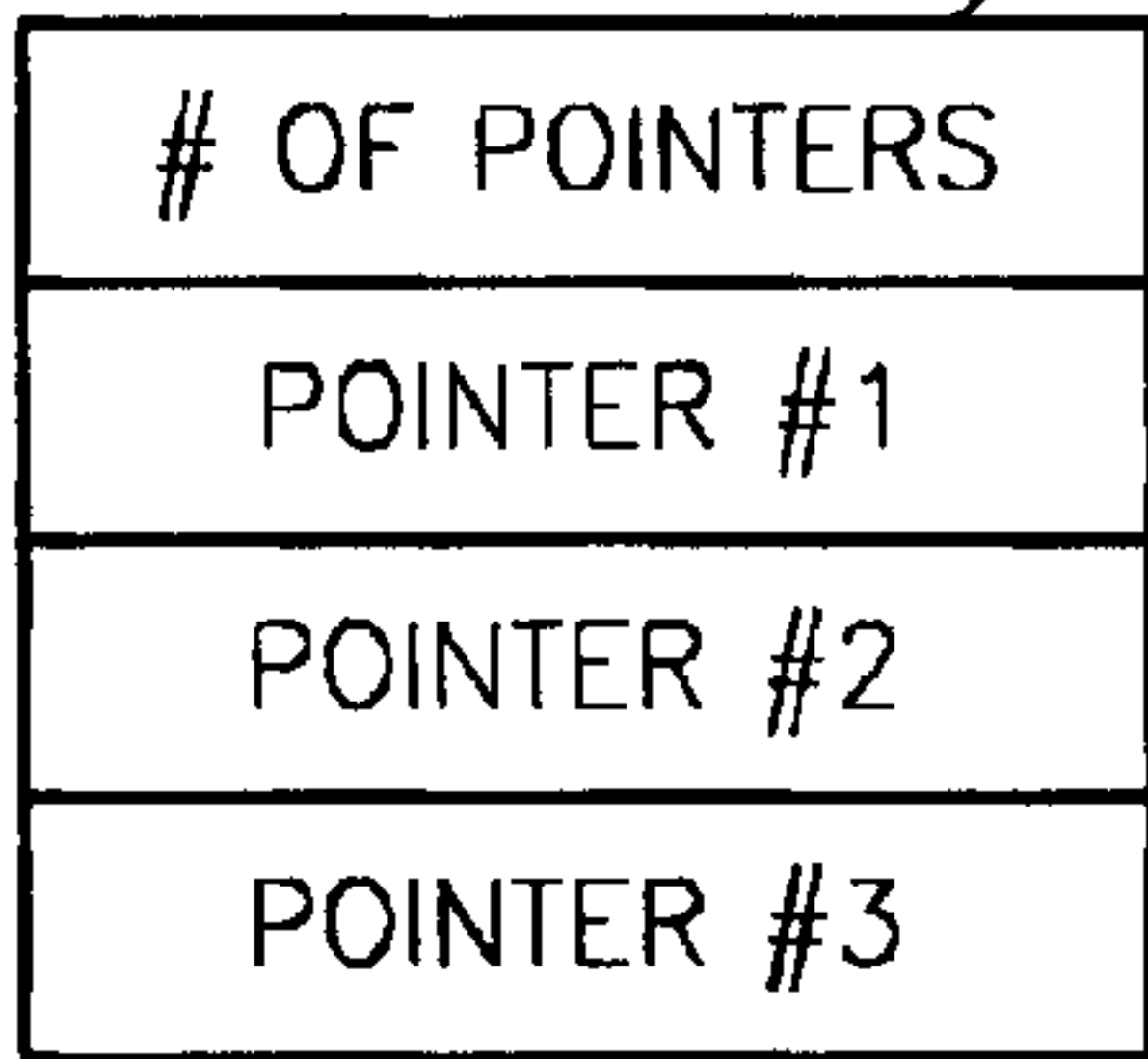


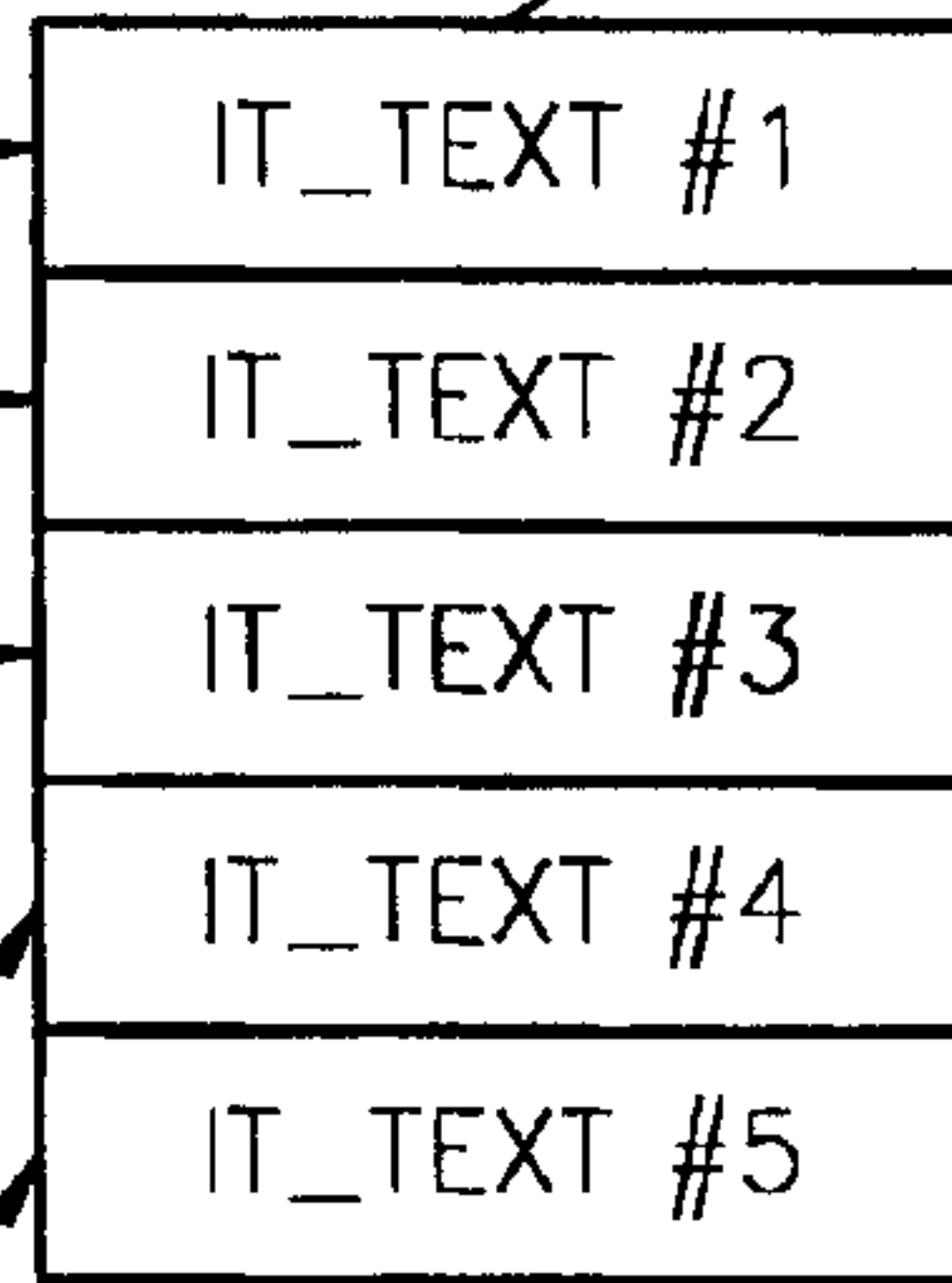
FIG. 9



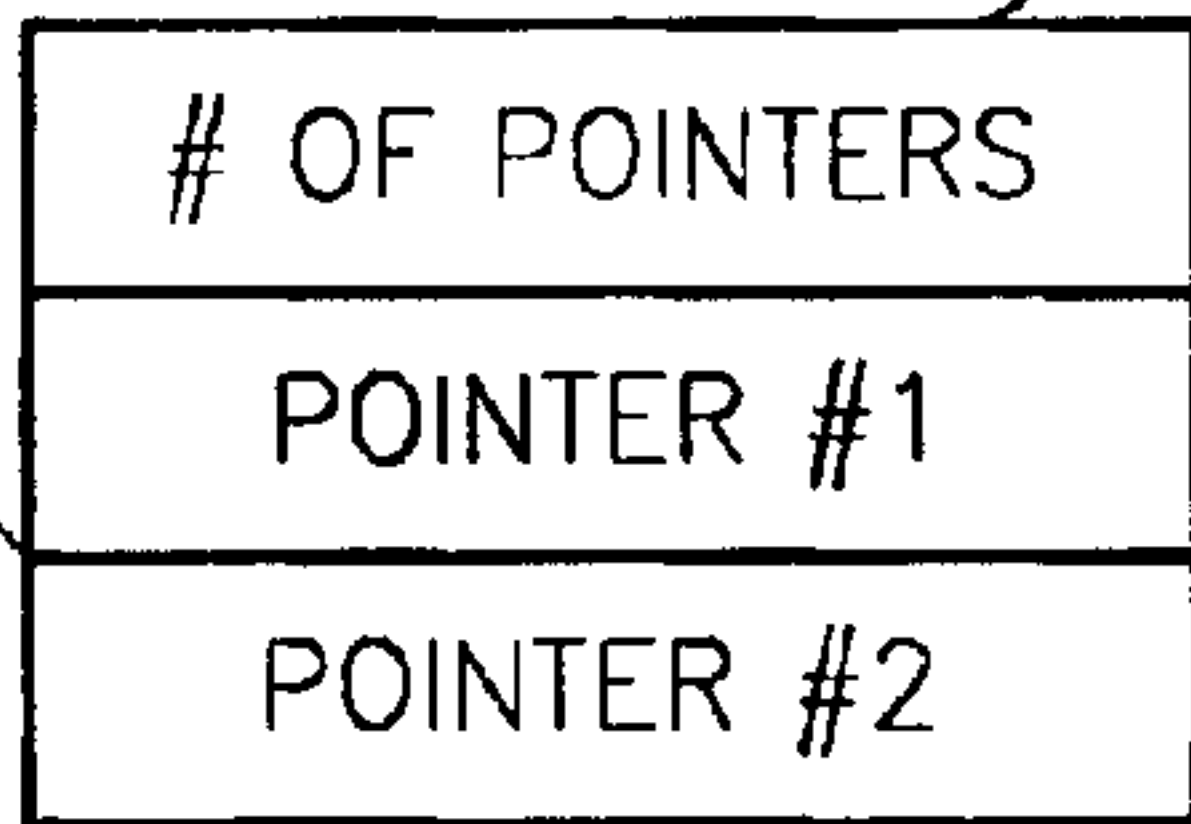
TO-BE-LOADED  
POINTER LIST 31



35



NOT-TO-BE-LOADED  
POINTER LIST 32



33

