

DOCUMENT MADE AVAILABLE UNDER THE PATENT COOPERATION TREATY (PCT)

International application number:	PCT/JP2015/000007
International filing date:	05 January 2015 (05.01.2015)
Document type:	Certified copy of priority document
Document details:	Country/Office: US
	Number: 61/944,280
	Filing date: 25 February 2014 (25.02.2014)
Date of receipt at the International Bureau:	07 February 2015 (07.02.2015)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a),(b) or (b-bis)

CERTIFICATE OF AVAILABILITY OF A CERTIFIED PATENT DOCUMENT IN A DIGITAL LIBRARY

The International Bureau certifies that a copy of the patent application indicated below has been available to the WIPO Digital Access Service since the date of availability indicated, and that the patent application has been available to the indicated Office(s) as of the date specified following the relevant Office code:

Document details: Country/Office: US

Filing date: 25 Feb 2014 (25.02.2014)

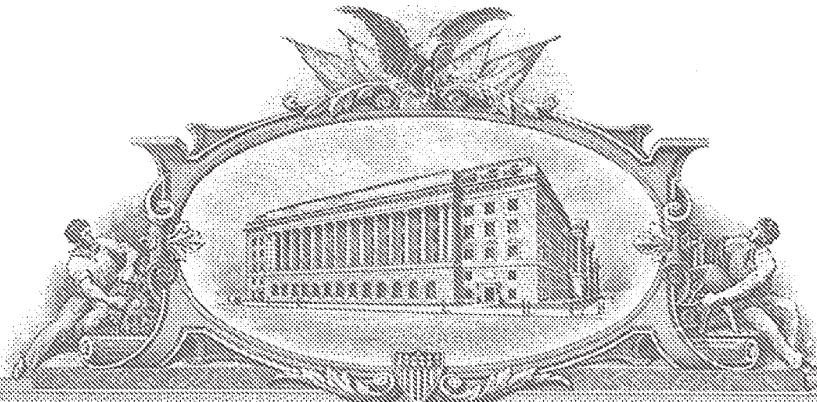
Application number: 61944280

Date of availability of document: 27 Mar 2014 (27.03.2014)

The following Offices can retrieve this document by using the access code:

AU, CN, ES, FI, GB, IB, JP, KR, SE, US, DK

Date of issue of this certificate: 09 Feb 2015 (09.02.2015)



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

February 09, 2015

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: 61/944,280
FILING DATE: February 25, 2014

THE COUNTRY CODE AND NUMBER OF YOUR PRIORITY APPLICATION, TO BE USED FOR FILING ABROAD UNDER THE PARIS CONVENTION, IS US61/944,280



Certified by

David J. Kyte

Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(e)

INVENTOR(S)		
Given Name (first and middle (if any))	Family Name or Surname	Residence (City and either State or Foreign Country)
Tadamas Tomoki Hiroshi	TOMA OGAWA YAHATA	Osaka, Japan Osaka, Japan Osaka, Japan
☐ <i>Additional inventors are being named on the separately numbered sheets attached hereto</i>		
TITLE OF THE INVENTION (280 characters max)		
DATA REPRODUCING METHOD AND DATA REPRODUCING APPARATUS		
CORRESPONDENCE ADDRESS		
<i>Direct all correspondence to:</i> ☐ Customer Number _____ - _____ Type Customer Number here Place Customer Number Bar Code Label here OR		
CORRESPONDENCE ADDRESS CUSTOMER NO. 52349		/Kenneth W. Fields/ 2014.02.25 14:57:33 -05'00' By: _____ Kenneth W. Fields Registration No. 52,430 WENDEROTH, LIND & PONACK, L.L.P. 1030 15th Street, N.W., Suite 400 East Washington, D.C. 20005-1503 Phone: (202) 721-8200 Fax: (202) 721-8250 February 25, 2014
ENCLOSED APPLICATION PARTS (<i>check all that apply</i>)		
☐ Specification ☒ Drawings	<i>Number of Pages</i> [] <i>Number of Sheets</i> [41]	☐ CD(s), Number _____ ☒ Other, (specify) <u>Application Data Sheet</u>
METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISION APPLICATION FOR PATENT (<i>check one</i>)		
☐ Applicant claims small entity status. See 37 CFR 1.27. ☐ A check or money order is enclosed to cover the filing fees of \$260 ☐ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account No. <u>23-0975</u> . Filing Fee Amount \$260 ☒ Please charge my Credit Card in the amount of \$260 to cover the above fees.		
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government. ☐ No. ☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____		

The USPTO is hereby authorized to charge any fees under 37 C.F.R. §§ 1.16, 1.17, and 1.492, which may be required by this paper and for the entire pendency of the present application to Deposit Account No. 23-0975, except fees for multiple dependent claims.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	PR14-0216A
		Application Number	
Title of Invention	DATA REPRODUCING METHOD AND DATA REPRODUCING APPARATUS		
The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76. This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.			

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1					Remove
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Tadamasa		TOMA		
Residence Information (Select One) ☐ US Residency ☒ Non US Residency ☐ Active US Military Service					
City	Osaka	Country of Residence i	JP		
Mailing Address of Inventor:					
Address 1		c/o Panasonic Corporation			
Address 2		1006, Oaza Kadoma, Kadoma-shi			
City	Osaka	State/Province			
Postal Code	571-8501	Country i	JP		
Inventor 2					Remove
Legal Name					
Prefix	Given Name	Middle Name	Family Name	Suffix	
	Tomoki		OGAWA		
Residence Information (Select One) ☐ US Residency ☒ Non US Residency ☐ Active US Military Service					
City	Osaka	Country of Residence i	JP		
Mailing Address of Inventor:					
Address 1		c/o Panasonic Corporation			
Address 2		1006, Oaza Kadoma, Kadoma-shi			
City	Osaka	State/Province			
Postal Code	571-8501	Country i	JP		
Inventor 3					Remove
Legal Name					

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	PR14-0216A
		Application Number	
Title of Invention	DATA REPRODUCING METHOD AND DATA REPRODUCING APPARATUS		

Prefix	Given Name	Middle Name	Family Name	Suffix
	Hiroshi		YAHATA	
Residence Information (Select One) ☐ US Residency ☒ Non US Residency ☐ Active US Military Service				
City	Osaka	Country of Residence i	JP	
Mailing Address of Inventor:				
Address 1	c/o Panasonic Corporation			
Address 2	1006, Oaza Kadoma, Kadoma-shi			
City	Osaka	State/Province		
Postal Code	571-8501	Country i	JP	
All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the Add button. Add				

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).			
☐ An Address is being provided for the correspondence information of this application.			
Customer Number	52349		
Email Address	wlp@wenderoth.com	Add Email	Remove Email

Application Information:

Title of the Invention	DATA REPRODUCING METHOD AND DATA REPRODUCING APPARATUS		
Attorney Docket Number	PR14-0216A	Small Entity Status Claimed	☐
Application Type	Provisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	41	Suggested Figure for Publication (if any)	

Filing By Reference :

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country i

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	PR14-0216A
		Application Number	
Title of Invention	DATA REPRODUCING METHOD AND DATA REPRODUCING APPARATUS		

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)
☐ Request Not to Publish. I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application has not and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.											
<table border="1"> <tr> <td>Please Select One:</td> <td>☒ Customer Number</td> <td>☐ US Patent Practitioner</td> <td>☐ Limited Recognition (37 CFR 11.9)</td> </tr> <tr> <td>Customer Number</td> <td colspan="3">52349</td> </tr> </table>				Please Select One:	☒ Customer Number	☐ US Patent Practitioner	☐ Limited Recognition (37 CFR 11.9)	Customer Number	52349		
Please Select One:	☒ Customer Number	☐ US Patent Practitioner	☐ Limited Recognition (37 CFR 11.9)								
Customer Number	52349										

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.

When referring to the current application, please leave the application number blank.

Prior Application Status		Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the Add button.			Add

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(d). When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX) the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(h)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

			Remove
Application Number	Country ⁱ	Filing Date (YYYY-MM-DD)	Access Code ^j (if applicable)

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	PR14-0216A
		Application Number	
Title of Invention	DATA REPRODUCING METHOD AND DATA REPRODUCING APPARATUS		
Additional Foreign Priority Data may be generated within this form by selecting the Add button. Add			

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

- ☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
- ☐ NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

Authorization to Permit Access:

☒ Authorization to Permit Access to the Instant Application by the Participating Offices
If checked, the undersigned hereby grants the USPTO authority to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application. In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application. In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date of filing this Authorization.

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	PR14-0216A
		Application Number	
Title of Invention	DATA REPRODUCING METHOD AND DATA REPRODUCING APPARATUS		

Applicant 1			Remove	
If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.				
Clear				
☒ Assignee		☐ Legal Representative under 35 U.S.C. 117		☐ Joint Inventor
☐ Person to whom the inventor is obligated to assign.		☐ Person who shows sufficient proprietary interest		
If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:				
Name of the Deceased or Legally Incapacitated Inventor :				
If the Applicant is an Organization check here. ☒				
Organization Name	Panasonic Corporation			
Mailing Address Information:				
Address 1	1006, Oaza Kadoma, Kadoma-shi			
Address 2				
City	Osaka	State/Province		
Country ⁱ	JP	Postal Code	571-8501	
Phone Number		Fax Number		
Email Address				
Additional Applicant Data may be generated within this form by selecting the Add button. Add				

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Assignee 1
Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.
Remove
If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	PR14-0216A
		Application Number	
Title of Invention	DATA REPRODUCING METHOD AND DATA REPRODUCING APPARATUS		

Prefix	Given Name	Middle Name	Family Name	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		State/Province	
Country i		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.

Add

Signature:

Remove

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications

Signature	/Kenneth W. Fields/		Date (YYYY-MM-DD)	2014-02-25
First Name	Kenneth W.	Last Name	Fields	Registration Number
				52430

Additional Signature may be generated within this form by selecting the Add button.

Add

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Acknowledgement Receipt	
EFS ID:	18295867
Application Number:	61944280
International Application Number:	
Confirmation Number:	7344
Title of Invention:	DATA REPRODUCING METHOD AND DATA REPRODUCING APPARATUS
First Named Inventor/Applicant Name:	Tadamasa TOMA
Customer Number:	52349
Filer:	Kenneth Wayne Fields/pam veazey
Filer Authorized By:	Kenneth Wayne Fields
Attorney Docket Number:	PR14-0216A
Receipt Date:	25-FEB-2014
Filing Date:	
Time Stamp:	15:35:51
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$ 260
RAM confirmation Number	2093
Deposit Account	230975
Authorized User	FIELDS, KENNETH
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees) Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)					
Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)					
Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)					
File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Provisional Cover Sheet (SB16)	AttachA_TRN_P66666901.pdf	325606	no	1
			a6fcb0b4b5f1fe8fc70e9a11a001a70e1a88101a		
Warnings:					
This is not a USPTO supplied Provisional Cover Sheet SB16 form.					
The PDF file has been signed with a digital signature and the legal effect of the document will be based on the contents of the file not the digital signature.					
Information:					
2	Application Data Sheet	AttachB_ADS_P66666901.pdf	1560861	no	7
			0b47a8b49b206776a6f1019de9b725ae1ef620c		
Warnings:					
Information:					
3	Drawings-only black and white line drawings	AttachC_DRW_P66666901.pdf	2668872	no	41
			e78fd7b26cc6759b932f44f4bb0d307c33240f3		
Warnings:					
The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing					
The PDF file has been signed with a digital signature and the legal effect of the document will be based on the contents of the file not the digital signature.					
Information:					
4	Fee Worksheet (SB06)	fee-info.pdf	29484	no	2
			7c9b0bde2c91180f83e630cfb3ca4c7049de6f1		
Warnings:					
Information:					
Total Files Size (in bytes):			4584823		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

- The present invention relates to a data multiplexing method, a data playback method, and a data conversion method for streams stored on an optical disc.

- Problems With the Prior Art

The file format commonly used on optical discs is the MPEG-2 TS format specified in ISO/IEC 13818-1, which multiplexes video, audio, and subtitle streams. In the MPEG-2 TS format, each stream, video, audio, and subtitles, is divided into TS packets of 188 bytes each before being multiplexed and recorded. The MPEG-2 TS format is optimized for cases in which data is retrieved and processed sequentially from the beginning, such as in broadcasting and on optical discs, and the format enables efficient stream retrieval, decoding, and playback even on consumer devices with relatively small buffer capacities.

In contrast, the file format increasingly used for content distribution over networks today is the MP4 format, as specified in ISO/IEC 14496-12. The MP4 format employs an extremely flexible data structure designed for use with random-access media such as hard disk drives (HDDs) and flash memory. In typical usage, video, audio, and subtitle streams are divided into fragments lasting a few seconds each, which are then concatenated to form a single file.

Looking ahead, it is expected that optical discs will continue to be widely used as the distribution medium for high-quality content such as 4K, due to cost considerations per bit. On the other hand, while smartphones and tablets do not have optical disc drives, they are used as devices for receiving and playing back content distributed over networks, thanks to their high portability and recent trends toward larger screens and higher resolution. For this reason, smartphones and tablets often support the MP4 format, while support for the MPEG-2 TS format has not progressed very far. When copying MPEG-2 TS content distributed on optical discs to smartphones or tablets and converting it to the MP4 format, it is necessary to first break down the MPEG-2 TS-multiplexed content back into its video, audio, and subtitle streams, and then convert them to the MP4 format. Generally, commercial content distributed on optical discs is encrypted. However, during conversion, it is necessary to first decrypt the content, convert it to a different format, and then re-encrypt it. Such decryption and encryption processing not only takes time to process on the conversion device but also raises security concerns because it involves creating plaintext content temporarily.

- Overview of the Invention

- In an MPEG-2 TS stream, a buffer is inserted after the PES_Header so that the TS payload containing the PES_Header and the TS payload containing the elementary streams are stored in separate TS_Packets.

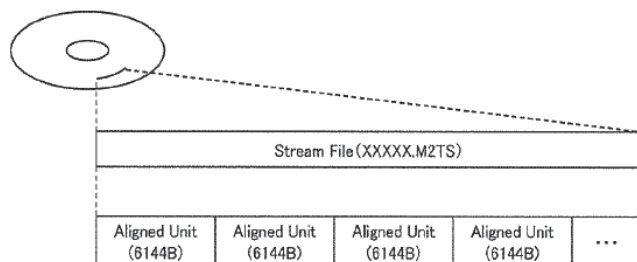
- Encryption is performed using AES-CTR mode, ensuring that just the portion of the TS_Payload containing the elementary streams is encrypted.

- The counter for AES-CTR mode is managed separately for video and audio.

- Effect

This makes it possible to easily convert MPEG-2 TS content to MP4 without the need for decryption or re-encryption.

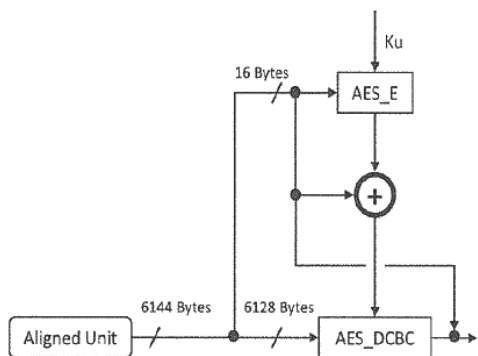
FIG. 1



· FIG. 1 shows the structure used when storing content on an optical disc using the conventional MPEG-2 TS format. Stream Files are stored on the optical disc as content. Although only one Stream File is shown in the figure, there may be multiple Stream Files. Furthermore, in this example, stream files are recorded with the file name XXXXX.M2TS. A number is assigned to XXXXX, allowing for individual management when multiple segments of content are stored.

· Each stream file is divided into units called "aligned units," each consisting of 6,144 bytes. The aligned unit serves as the unit of encoding. Note that content stored in a stream file does not have to be a multiple of 6,144 bytes. In such cases, the content should be made into a multiple of 6,144 bytes by, for example, appending NULL data to the end.

FIG. 2



· FIG. 2 shows the decryption method for each aligned unit. Content on an optical disc is encrypted using data called a unit key (labeled "Ku" in the figure). Here, each 6,144-byte block of data is separated into the first 16 bytes and the remaining 6,128 bytes. The first 16 bytes are processed using the unit key via AES_E, an AES decryption method. The data obtained in this way is then XORed with the initial 16 bytes, and this result is used as the key to decrypt the remaining 6,128 bytes of data using AES in CBC mode. This yields plaintext data, and by appending the first 16 bytes to it, 6144 bytes of plaintext are obtained.

FIG. 3

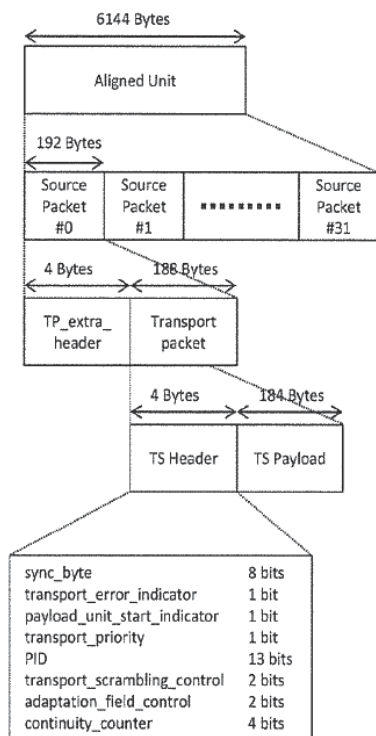


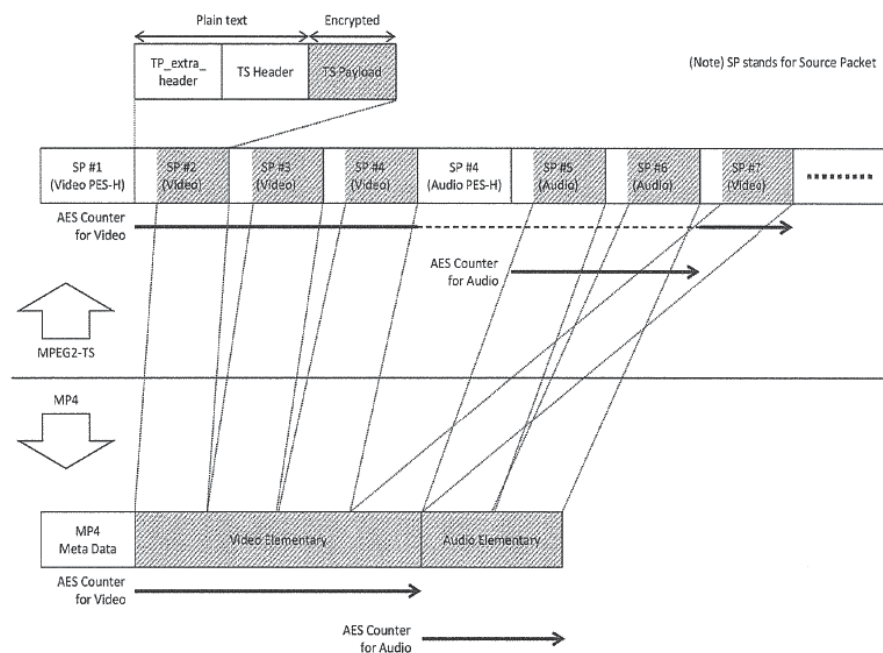
FIG. 3 shows the internal structure of an aligned unit in plaintext form. An aligned unit consists of 32 source packets, each 192 bytes in length. Each source packet consists of a 4-byte header (TP_extra_header) and a 188-byte transport packet. Meanwhile, the 188-byte transport packet consists of a 4-byte TS Header and a 184-byte TS payload. The TS Header contains information indicating the attributes of the TS payload, including sync_byte (8 bits), transport_error_indicator (1 bit), payload_unit_start_indicator (1 bit), transport_priority (1 bit), PID (13 bits), transport_scrambling_control (2 bits), adaptation_field_control (2 bits), and continuity_counter (4 bits). Here, the PID identifies the type of elementary stream stored in the TS payload, and this value allows for the distinction between video, audio, and other types of data. Furthermore, this PID is used to distinguish between multiple types of audio when they are recorded.

FIG. 4

TS Payload #1	TS Payload #2	*****	TS Payload #n
PES_Header	Elementary Stream		

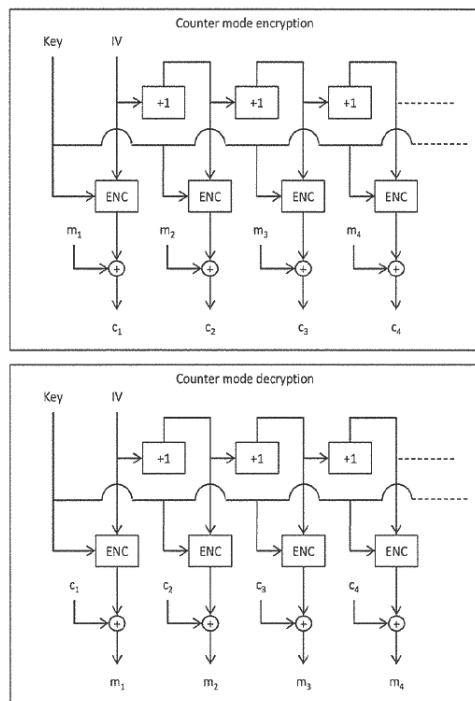
· FIG. 4 shows the method for creating an actual elementary stream from a TS payload. By concatenating multiple consecutive TS payloads with the same PID, a PES_Header and an elementary stream are formed. Here, the configuration is so that the initial TS payload serves as the PES_Header.

FIG. 5



· FIG. 5 shows the method for encrypting an MPEG-2 TS stream and converting it to MP4 according to the present invention. As previously mentioned, an MPEG-2 TS stream consists of multiple source packets. In the figure, source packets are abbreviated as "SP." Here, only the TS payload portion of each source packet is encrypted. For this encryption process, the AES-CTR mode is used. The processing in AES-CTR mode will be explained below. Additionally, in AES-CTR mode, encryption and decryption are performed using a counter value. In the figure, arrows indicate how this counter increments. Here, separate counters are managed for video and audio, the counter for video increments only in the video portion, and the counter for audio increments only in the audio portion. By adopting this configuration, it is possible to easily utilize an MP4 file simply by extracting the TS payload portion.

FIG. 6



· FIG. 6 shows the encryption and decryption of a message in AES-CTR mode. During encryption, the key and the initial value or initial vector (IV) are used. After encrypting the IV with the key, the result is XORed with the first 16 bytes of the message to produce the first 16 bytes of encrypted text. For the next 16-byte block, the IV is incremented by 1, and the same process is repeated. By performing this process consecutively, it is possible to generate encrypted text even for messages longer than 16 bytes. Note that if the message length is not a multiple of 16 bytes, the encrypted text is obtained by performing a bit-by-bit XOR operation on the final block.

· The lower half of FIG. 6 depicts the decryption process. This process is identical to that used for encryption. It is important to note that, even though this is decryption, ENC processing is performed.

· In MPEG-2 transport streams (TS) and MP4, the multiplexing methods for access unit data differ in part. Therefore, when embedding a TS into MP4, it is necessary to split some of the access unit and store them separately. As a basic concept, an access unit in a TS contains both the initialization information essential for decoding video pictures or audio frames, and the encoded data for each picture or frame. In contrast, an access unit in MP4 (referred to as a "sample" in MP4) consists solely of encoded data for pictures or frames, while the initialization information required for decoding is stored separately as part of the MP4 header information.

· The following is a description of a method for converting encoded data multiplexed using TS into MP4, utilizing support information and a multiplexing format that uses support information to reduce the computational load associated with the conversion. In particular, when the encoded data is encrypted, the computational load involved in decrypting and then re-encrypting the data is significant. Therefore, it is desirable to perform the conversion using only a copy of the data without decrypting the encoded data. Note that the TS does not have to be the TS specified in the MPEG-2 system but can also be a TS with a header of a predetermined number of bytes appended to the TS packet (as used in Blu-ray discs and video streaming, for example). Furthermore, MP4 may be MP4 that uses movie fragments or MP4 that does not. Additionally, it may be a format extended from MP4, such as the Common File Format (CFF) of the Digital Entertainment Content Ecosystem or MPEG-DASH (Dynamic Adaptive Streaming over HTTP).

· FIG. 7(a) shows the process of extracting MPEG-4 AAC access units stored in a transport stream and converting them to MP4. MPEG-4 AAC access units in a transport stream consist of three types of data: the LATM header, PayloadLengthInfo(), and PayloadMux(). The LATM header contains initialization information necessary for decoding AAC data, such as the number of channels and the sampling frequency. More specifically, this initialization information is stored in the AudioSpecificConfig() field in the LATM header. The PayloadLengthInfo() field stores the size of the PayloadMux(), and the PayloadMux() field stores the AAC encoded data. When storing this access unit in an MP4 file, the AudioSpecificConfig() in the LATM header is stored in the sample entry for stsd in the moov, while PayloadLengthInfo() and PayloadMux() are stored as sample data in the mdat. Note that the sample data in the mdat is referenced from the moov, or from the moov when using a movie fragment.

· In MPEG-2 AAC, the ADTS header is used instead of a standard LATM header, and the access unit consists of the ADTS header and the AAC data referred to as raw_data_block(). In this case as well, the ADTS header is separated from the access unit, and at least adts_fixed_header() is stored in the sample entry. Additionally, the AAC data is stored in mdat as sample data.

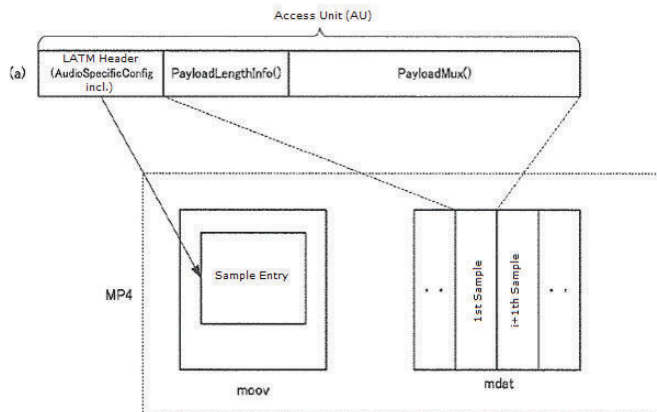


FIG. 7

· FIG. 8(a) shows an example of how MPEG-4 AVC (Advanced Video Coding) access units, which are stored in a TS, are embedded in an MP4 file. As with MPEG-4 AAC, in a TS, initialization information required for decoding, such as the Sequence Parameter Set (SPS) and Picture Parameter Set (PPS), is stored as part of the access unit. However, in MP4, this initialization information is stored separately as part of the MP4 header. The example in FIG. 8(a) shows the structure of an IDR (Instantaneous Decoder Refresh) picture. However, the SPS and PPS NAL (Network Adaptation Layer) units are separated from the access unit and provided as sample entries in the stsd in the moov. Other data is stored in the mdat as sample data.

· In MP4, it is also possible to select a mode that allows SPS or PPS initialization information to be included in the MPEG-4 AVC sample data. The mode is indicated solely by the sample entry. If the identifier is avc1 or avc2, inclusion of initialization information in the sample data is prohibited. However, if it is avc3 or avc4, initialization information may be included in the sample data. Therefore, when converting from TS to MP4, if the identifier in the MPEG-4 AVC data of the resulting MP4 file is set to avc1 or avc2, it is necessary to remove SPS, PPS, and NAL units containing filler data used as "stuffing" for the access units in the TS. However, if the identifier is set to avc3 or avc4, it is not necessary to remove SPS or PPS. Therefore, depending on the setting value of the identification information in the MP4 file, removal of the SPS and PPS is optional.

· HEVC (High Efficiency Video Coding), a next-generation video coding standard, has a NAL unit structure similar to MPEG-4 AVC and includes initialization information such as SPS and PPS. However, when embedding HEVC in MP4, this initialization information may be included in the sample data. Therefore, depending on the type of coding scheme, it is optional whether or not to remove the initialization information from the access unit when converting to MP4 and to switch the operation accordingly.

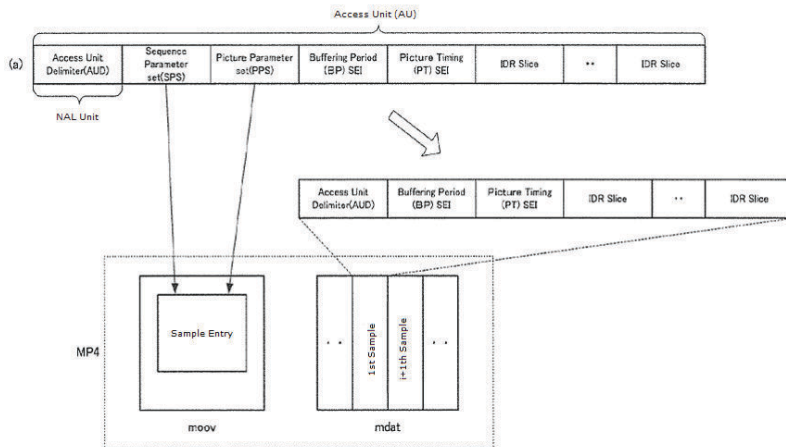


FIG. 8

- When storing MPEG-4 AAC encoded data in MP4, the header is separated from the LATM, and only PayloadLengthInfo() and PayloadMux() (hereinafter collectively referred to as the LATM payload) are stored as sample data. Therefore, when storing an MPEG-4 AAC access unit in a TS packet, the LATM header and LATM payload are stored in separate TS packets, making it easy to separate the LATM header. FIG. 9(a) shows an example of how the LATM header and LATM payload are stored in TS packets. Stuffing is performed if necessary to ensure that the LATM header and LATM payload are not present in the payload of the same TS packet. For example, the LATM header for Access Unit 1 is stored in the payload of the TS packet, but since the size of the LATM header is smaller than the size of the TS payload, the remaining area of the TS payload is stuffed. Note that although PES packets are not shown in this example, in practice, access unit data is placed in the PES packet payload, and the PES packet data is stored in the PES packet payload.

- Next, it will be described how to distinguish between packets containing an LATM header and TS packets containing an LATM payload. When storing one access unit as one PES packet, the payload_unit_start_indicator in the TS header is set to 1 in the packet containing the PES header data. By ensuring that the payload of a TS packet with the payload_unit_start_indicator set to 1 contains an LATM header, it is possible to determine whether an LATM header is included in a TS packet based on the value of the payload_unit_start_indicator. When storing multiple access units in a single PES packet, an AU_info table containing the number of access units included in the payload as well as the sizes of the LATM header and LATM payload for each access unit, may be placed at the beginning of the PES packet payload. FIG. 9(b) shows an example of the AU_info table syntax. The AU_info_identification_code is a bit string unique to the AU_info table, and by searching for this code, it is possible to determine whether the AU_info table is present. Alternatively, this may take the form of a data structure that combines the data size of a box with the box type, similar to the MP4 box structure. However, if the presence of the AU_info table is signaled in a descriptor in the TS or in support data for MP4 conversion prepared separately from the TS, this code may be omitted. Furthermore, the AU_info table can also be used when a single access unit is contained in a single PES packet.

- The method of specifying which data to separate when converting to MP4 by indicating the size of structural elements in an access unit and data offsets can also be applied when separating NAL units from SPS or PPS to store MPEG-4 AVC encoded data. FIG. 9(c) shows an example of the syntax for the AU_info table, which indicates whether each structural element in an access unit needs to be excluded from the MP4 sample data. Num_of_data_unit indicates the number of data units, and size_of_data_unit indicates the size of a data unit. The conversion_mode field specifies how data units are handled during MP4 conversion. Here, a value of 0 indicates that the data unit is copied to the MP4 sample data, while a value of 1 indicates that it is excluded from the sample data. If the value is 1, the field may also specify how the excluded data is handled afterward, such as storing it in a sample entry. Furthermore, information for each data unit is stored in ascending order based on decoding sequence. For example, when storing an MPEG-4 AVC access unit as an IPES packet, the data unit corresponds to a NAL unit, and the number of NAL units constituting the access unit is indicated by Num_of_data_unit. Then, in the NAL units of the SPS or PPS, conversion_mode is set to 1. Additionally, by treating both the LATM header and the LATM payload as data units and setting conversion_mode to 1 in the data unit corresponding to the LATM header, this method can also be applied to MPEG-4 AAC. When multiple access units are stored in a PES packet, number_of_data_unit indicates the total number of data units contained in all access units.

- The AU_info table may be stored in the adaptation_field of the TS packet header at the start of the PES packet, or it may be stored as part of the encoded data, such as in NAL units in MPEG-4 AVC or HEVC, or in SEI messages. When stored as part of the encoded data, it can be stored on a per-access-unit basis or on a per-random-access-unit basis. Furthermore, when stored in TS packets or PES packets, in addition to being stored for each PES packet, it may be stored only in the PES packet containing the access unit that marks the beginning of a random access unit in decoding order. When stored in each random access unit, the AU_info table contains information regarding all access units that constitute random access units.

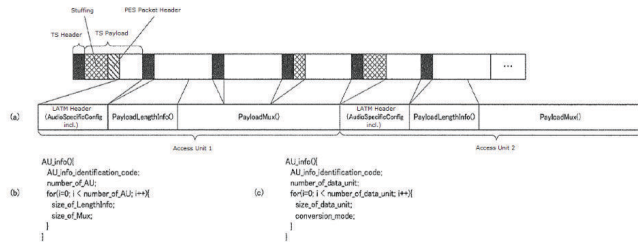


FIG. 9

· As previously noted, TS and MP4 differ in whether they include initialization information required for decoding in the NAL unit when storing encoded audio and video data. Furthermore, because the NAL unit storage formats in MPEG-4 AVC and HEVC differ between TS and MP4, converting from TS to MP4 requires a conversion of the storage format. FIG. 10(a) to FIG. 10(c) show the storage formats for NAL units in TS and MP4. FIG. 10(a) shows the structure of an NAL unit. An NAL unit consists of a header and a payload, and the header contains type information indicating the data type stored in the payload. FIG. 10(b) shows an example of how NAL units are stored in TS. In TS, a unique bit sequence called a start code is appended to mark the boundaries of NAL units (hereinafter referred to as the start code format). Decoders and other devices can isolate the desired NAL units by searching for the start code and the type information of the NAL unit data stored in the NAL header. FIG. 10(c) shows an example of how NAL units are stored in MP4. In MP4, a field indicating the data size of the NAL unit is appended to identify the boundaries of NAL units (hereinafter referred to as the "NAL size format"). The field length of this data size field is stored in the AVCDerConfigurationRecord, which is contained in the sample entry and stores initialization information for decoding. Decoders and other devices can separate NAL units based on the size of the NAL units. As described above, TS and MP4 differ in whether the identification information indicating the boundaries of NAL units takes the form of a start code or an NAL size format. Therefore, when converting from TS to MP4, the identification information indicating the boundaries of NAL units must be converted.

· Since the storage format for access units is specified for each encoding method, when converting to MP4, the operation is switched simply by referring to the audio or video encoding method.

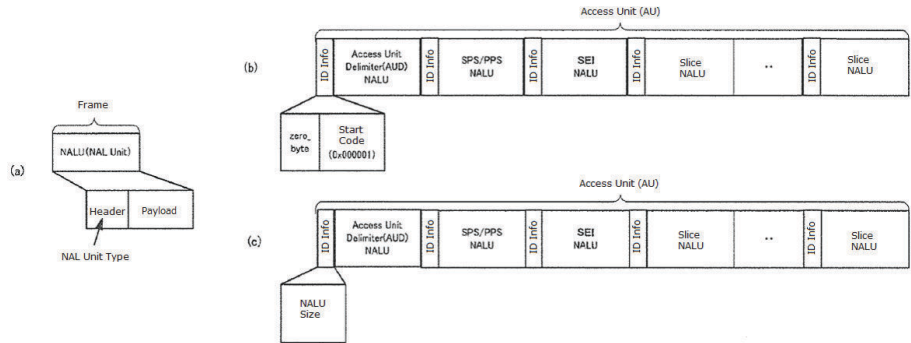


FIG. 10

- When handling data encoded in a format that uses NAL sizes during conversion to MP4, the size of the NAL units is required. Therefore, it is desirable to be able to obtain the size of the NAL units that constitute the access units in advance. Doing so reduces the processing load involved in determining the size of NAL units by searching for start codes in start code format access units and detecting NAL unit boundaries. FIG. 11(a) shows an example of storing the size of the NAL units that constitute an access unit multiplexed into a TS at the beginning of the access unit. Here, the size is stored by using a newly defined size information NAL unit. Since there are values available for NAL unit types that can be used for private data or user data, one of these types is selected and used as the size information NAL unit. Since size information does not require abbreviation, this NAL unit is stored in plaintext. FIG. 11(b) shows an example of the syntax for size information in a size information NAL unit. Here, number_of_nalu indicates the number of NAL units comprising an access unit, and nal_size indicates the size of each NAL unit. Since a size information NAL unit is not included in MP4, it is not necessary to specify the size in the size information NAL unit itself. Note that in this example, the size information NAL unit is placed before the NAL unit that signals the start of the access unit, called the Access Unit Delimiter (AUD). However, it may also be placed immediately after the AUD. When placed immediately after the AUD, the size information NAL unit indicates the size of the NAL units following the AUD. Although the size of the AUD itself is not indicated, since the AUD size is fixed, it is sufficient to specify that size in advance. Additionally, similar to the AUD_info table, it may be placed at the beginning of the PES packet payload. Furthermore, in a TS, NAL units may contain variable-length codes, such as zero_byte, in addition to the start code. Therefore, as shown in FIG. 11(c), the size of the identification information for units containing a start code may be stored separately. If the size of this additional information is fixed, it may be stored separately in the TS descriptor or as support data during MP4 conversion. Furthermore, the field length of the field indicating the size of the NAL unit in the NAL size format after MP4 conversion may also be specified separately.
- Additional information may also be treated as a data unit, and the contents of Nal_size_info may be indicated via the AU_info table. In this case, by setting conversion_mode to a multi-valued setting, a mode for converting the data structure of the data unit may be added in addition to the two existing actions, namely, deleting the data unit and leaving the data unit unaltered. Additionally, classification information for distinguishing between types of data units may be added to the AU_info table. For example, it is possible to distinguish NAL unit types, such as SPS and PPS in MPEG-4 AVC or HEVC, and slice NAL units, or to distinguish elements such as the LATM header and LATM payload in MPEG-4 AAC. Furthermore, a separate field may be provided to indicate whether a data unit is encrypted.
- Data units with different conversion_mode values may be stored in separate TS packets. Additionally, data units with a conversion_mode of 2 that are stored as header information (such as sample entries) may be stored in plaintext.
- Regarding encryption, only the NAL units of the slice data need to be encrypted, while the rest may be left in plaintext. Since encryption is performed on a per TS packet basis, plaintext and encrypted portions cannot be mixed in the payload of a single TS packet. However, this is because data such as AUD has a small size, and storing it as a separate packet would increase stuffing and reduce multiplexing efficiency. Furthermore, if the boundary of the NAL unit data in an encrypted slice is not aligned with the end of the TS packet payload, stuffing is required. When performing stuffing, one can use the adaptation_field in the TS packet header or insert a Filler Data NAL unit or an SEI for Filler Data into the encoded data. When using the adaptation_field in the TS packet header, the adaptation_field must be left in plaintext. However, switching between whether or not to encrypt a variable-length region imposes a high processing load. Therefore, when performing stuffing, it is preferable to use a data structure designated for stuffing in the encoded data to ensure that the boundaries of the NAL unit data are aligned with the end of the TS packet payload.
- Size information may be stored in the adaptation field of a TS packet or by inserting an SEI for storing user data (User unregistered SEI) in an access unit. When converting to MP4, the start code format can be converted to the NAL size format based on the NAL unit size information stored using one or more of these methods.
- Additionally, information indicating whether size information has been stored may be included using descriptors or similar elements in the TS.

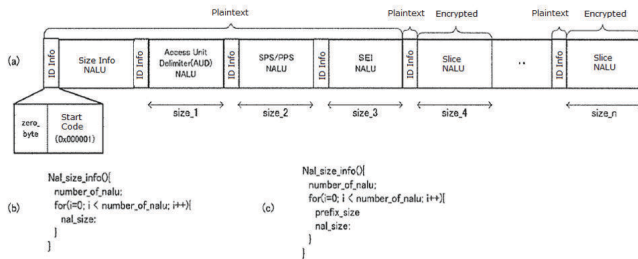


FIG. 11

· FIG. 12(a) is a flowchart showing an example of operations performed in the data conversion method of the present invention when realized by referencing the AU_info table. In step 101, the AU_info table, which is typically located at the beginning of the PES packet payload, is searched, and in step 102, a determination is made as to whether an AU_info table is present. If an AU_info table is present, the processing proceeds to step 103. If one is not present, the access unit separated from the PES packet is treated as a data unit, and the processing proceeds to step 105. When an access unit is stored in a PES packet, the payload of the PES packet may be separated by treating it as data corresponding to a single access unit. However, if multiple access units are stored in the PES packet, or if an access unit is fragmented and stored in the PES packet, the access unit boundaries in the encoded data are searched for to separate the access units. In step 103, the size of the data unit and the conversion mode are obtained, and in steps 104 to 108, conversion processing is performed according to the conversion mode. If the conversion mode is mode 0, the data unit is copied as raw data in step 105. If the conversion mode is mode 1, the data structure of the data unit is converted in step 107 and stored as raw data. For example, the type information at NAL unit boundaries is converted from start code format to NAL size format. If the conversion mode is 2, in step 108, at least a portion of the data in the data unit is stored in a sample entry. For example, NAL units from SPS or PPS in MPEG-4 AVC are stored in a sample entry, or AudioSpecificConfig() is extracted from the LATM header in MPEG-4 AAC and stored in a sample entry. Although the portion of the data in the data unit to be stored in a sample entry is predetermined by the encoding scheme, support information for identifying the portion to be stored in the sample entry may be separately indicated in the AU_info table.

· FIG. 12(b) is a block diagram showing the configuration of a data conversion device realizes the operations in the flowchart shown in FIG. 12(a).

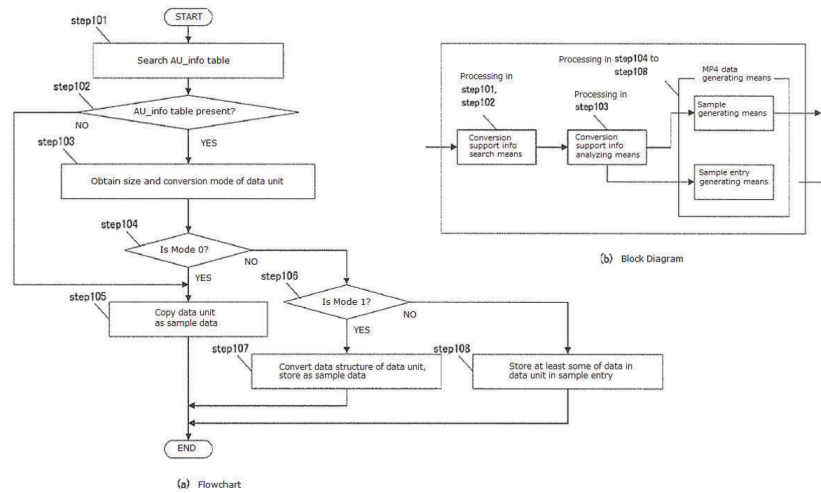


FIG. 12

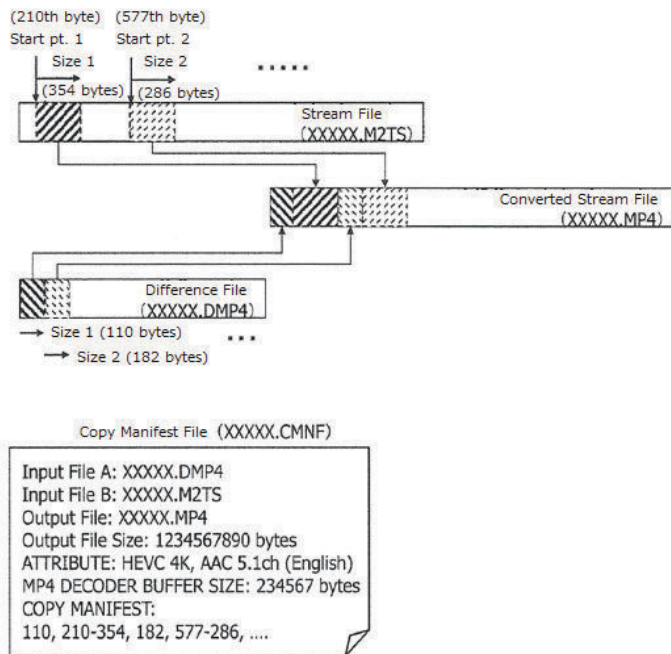


FIG. 13

When converting a stream file (XXXXX.M2TS) to another stream file (XXXXX.MP4), it is desirable to be able to easily convert the stream file to a different format without decoding the audio or video elementary streams (such as HE-AAC or MPEG-4 AVC) or, if encryption is applied, without decrypting it. However, this requires a great deal of complex processing, and until now, the only method available was to first demultiplex the file to extract the elementary streams, select only the necessary elementary streams, and then multiplex them to obtain a different system stream. One example disclosed herein will be described with reference to FIG. 13. In this example, a stream file (XXXXX.M2TS), a copy manifest file (XXXXX.CMNF) describing the conversion rules for processing the M2TS file into an MP4 file, and a difference file (XXXXX.DMP4) storing the difference data required for conversion to an MP4 file are recorded on a Blu-ray disc. When converting this stream file (XXXXX.M2TS), the stream file (XXXXX.M2TS) and the difference file (XXXXX.DMP4) are alternately combined in accordance with the instructions in the corresponding copy manifest file (XXXXX.CMNF) to generate the converted stream file (XXXXX.MP4). This copy manifest file (XXXXX.CMNF) contains the following information. Two source files are given the names "Input file A" and "Input file B," and the actual conversion process begins with the copy at the start of Input File A.

The output file is given the name "Output File." The data size of the output file is specified in terms of the number of bytes as "Output File Size," and by checking this value, one can verify before the conversion process begins whether there is sufficient free space on the destination medium or device. If there is insufficient free space on the destination storage medium or device, the system may choose not to output the data, or it may output only the amount of data that can be accommodated. Alternatively, the system may first output only the amount of data that can be accommodated, allow the user to view that portion, and then delete the data before outputting the remaining portion. In this case, the device performing the conversion, or the target storage medium or device, stores the size of the data already outputted. Furthermore, the output destination may store the data size of the unoutputted portion so that, after erasing the data or taking similar actions, it can determine whether there is sufficient free space to output the unoutputted portion.

The ATTRIBUTE element specifies which elementary streams are converted and what types of files are generated. Here, it is specified that the converted MP4 file will contain 4K video compressed using HEVC and 5.1-channel English audio in AAC. This allows one to verify in advance, before the conversion process begins, what the resulting file will look like when converted according to the copy manifest file. Additionally, one may specify the data structure of the MP4 file and specify the MP4 brand stored in ftyp of the MP4 file. It is also possible to specify the decoder resources (such as memory capacity and data transfer rate) required to play back the converted stream file. Here, the minimum buffer size required to play back the converted stream file (XXXXX.MP4) is specified as the MP4 DECODER BUFFER SIZE. This value depends on the multiplexing rules used to multiplex the video, audio, and other data in the converted MP4 file.

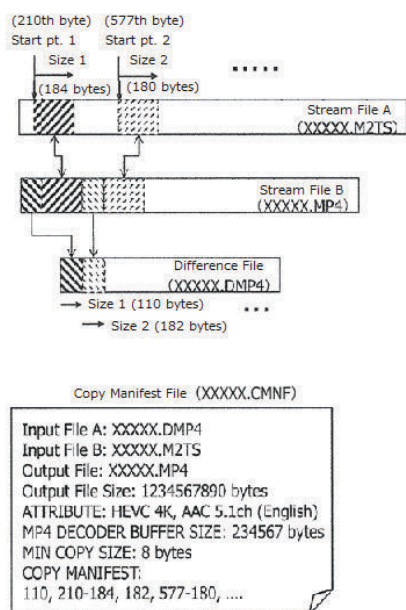
Finally, when generating the Output File by alternately copying portions of Input File A, Input File B, and Input File A, the copy start position and copy size are specified in sequence in the COPY MANIFEST. Since the difference file (XXXXX.DMP4) copies data in the specified-size units starting from the beginning of the file, there is no need to specify a copy start position. Here, a single size value represents one conversion process (copy operation). Since the stream file (XXXXX.M2TS) contains data that is not needed in the converted stream file (XXXXX.MP4), it is described using the pattern "copy start position (byte position at the start of the file) minus copy size (byte size)" to specify only a specific region. By copying specified regions alternately from the two input files according to the copy manifest file, it is possible to generate the same converted stream file (XXXXX.MP4) regardless of the device used for conversion. This makes it easy to play back the converted stream file without issues with any device.

Note that while this example assumes the difference file (XXXXX.DMP4) and the copy manifest file (XXXXX.CMNF) are recorded on a Blu-ray disc, either or both of these files may be obtained via the internet. In this case, it becomes possible to present users with various options, such as converting to the latest file formats or to MP4 files containing different video and audio content, without requiring any special modifications to the conversion device.

The section above described the process for handling stream files while they remain encrypted. However, it is also possible to decrypt the M2TS file, convert it to MP4, and then re-encrypt the resulting MP4 file. In this case, XXXXX.M2TS may be in plaintext. Note that since the difference file consists of header information, encryption is not required. However, if XXXXX.M2TS or XXXXX.MP4 is in plaintext, operations such as loading the entire file into memory or temporarily saving it to the hard disk may not be permitted for security reasons. Therefore, regarding the plaintext data in XXXXX.M2TS, it may be deleted from the area where the conversion to MP4 has been completed, and regarding XXXXX.MP4, it may be encrypted sequentially each time a movie fragment or a predetermined number of MP4 samples are generated. If an encryption method is used that does not change the data size before and after encryption, the area of data copied will remain unchanged regardless of whether encryption is applied. Additionally, information indicating whether XXXXX.M2TS is encrypted, or whether the encoded data multiplexed into the M2TS file is processed while still encrypted, or converted to plaintext, converted to MP4, and then re-encrypted, may be specified in the copy manifest file.

In addition, while the BD-ROM standard does not support MPEG-2, MPEG-4 AAC, or HE-AAC as audio encoding formats, AAC and HE-AAC are widely used on mobile devices such as smartphones and tablets that play converted files. Therefore, to enable the selection of an audio encoding method in the converted MP4 file, an AAC encoding stream corresponding to the audio data stored in the original M2TS file may be stored in a differential file. In this case, the converted MP4 file can be generated from the data in the differential file and the video encoding stream multiplexed into the M2TS1 file. Alternatively, the AAC encoding stream may be prepared as a separate file, and the manifest file may reference and copy the video encoding data from the M2TS file and the audio encoding data from the AAC encoding stream.

FIG. 14



The difference file (XXXXX.DMP4) and the copy manifest file (XXXXX.CMNF) can be generated as follows. During the authoring stage, both stream file A (XXXXX.M2TS) and stream file B (XXXXX.MP4), which is a format-converted version of stream file A, are generated. Overlapping segments (matching segments) are searched for and identified in both files so that a segment of stream file A matches the data in a segment of stream file B, and the data size of that matching segment is maximized. Once the matching segments have been identified, all segments of stream file B excluding the matching segments are concatenated as a difference file in the order of their occurrence, and the correlation results of this file are recorded in the copy manifest file in the COPY MANIFEST format.

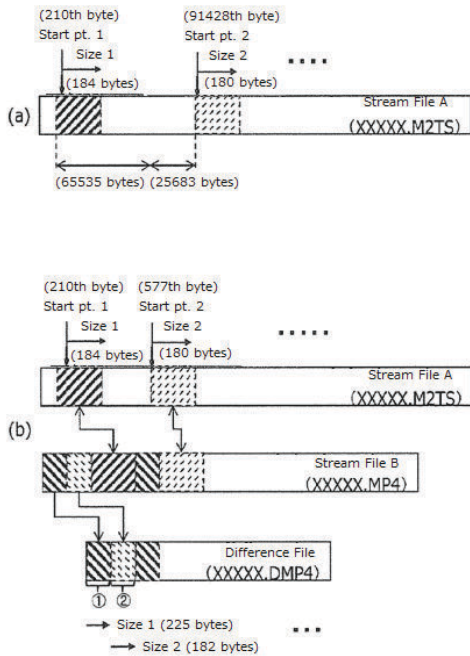
Also, the matching segments may be searched for in stream file A and stream file B without backtracking from the beginning of the file. Doing so enables the conversion process (copying process) to be performed sequentially in a single continuous operation, thereby achieving faster processing and reduced memory usage.

Moreover, when multiplexing video into a TS stream, the video frame data at a given PTS may be multiplexed before the corresponding audio frame at the same PTS to prevent underflow of data for frames with high bit rates, such as I-frames. Meanwhile, in MP4, when generating a movie fragment, for example, it is common for the data to be multiplexed so that the start time of the first picture in the movie fragment matches or is close to that of the video and audio. Thus, if the multiplexing units for AV or text differ between stream file A and stream file B, the conversion process cannot be performed sequentially, and the read or write pointer may need to be moved back and forth sequentially during conversion. In this case, information on whether or not the conversion process can be performed continuously and, if not, the maximum size requiring backtracking, may be registered in the manifest file.

Note that specifying matching segments in units that are too small can increase the data size of the copy manifest file or complicate the conversion process. Therefore, a specific threshold—for example, 8 bytes or more—may be set so that only matching segments of 8 bytes or more are registered in the copy manifest file, while segments smaller than this are treated as mismatches and stored in a difference file. Furthermore, this threshold value may be specified in the copy manifest file as "MINI COPY SIZE: 8 bytes."

In a copy manifest file, the source and destination of the data copy are specified. However, it is also possible to use a conversion manifest file that indicates the multiplexing units in the outputted MP4 file without using a difference file. For example, audio, video, and text can be multiplexed into separate movie fragments. At this time, the movie fragment header and the DTS and PTS values of the last audio frames or video pictures or the byte offset values from the beginning of the input TS, may be registered in the decoding order along with the media identification information. The registration order of the movie fragments match the order in which they appear in the MP4 file.

FIG. 15



In the copy manifest file, the address of the copy start position in the stream file does not have to be an absolute value from the beginning of the file. It may instead be a difference value relative to the start address of the immediately preceding entry. For example, in FIG. 15, the second entry corresponding to the stream file can be described as (367, 180) rather than (577, 180).

There are also three possible ways to specify the start address and copy size, as shown below.

Mode 1: Alternatingly copy data from the stream file and the difference file.

Notation example: (address, size)

Examples of bit lengths for each field: (int32, int8)

Mode 2: If the copy size is 0, the copying is skipped. Since data can be copied consecutively from the same file, the data length of the address can be shorter compared to Mode 1.

Notation example: (address, size)

Example of bit lengths for each field: (int16, int8)

Mode 3: In Mode 3, instead of explicitly specifying the copy size as in Mode 2, data is copied up to the end of the payload in the TS packet indicated by the address. The copy size field is omitted. Additionally, instead of specifying an address, the difference in index numbers of TS packets in the stream file may be used.

Notation example: (packet number, copy mode)

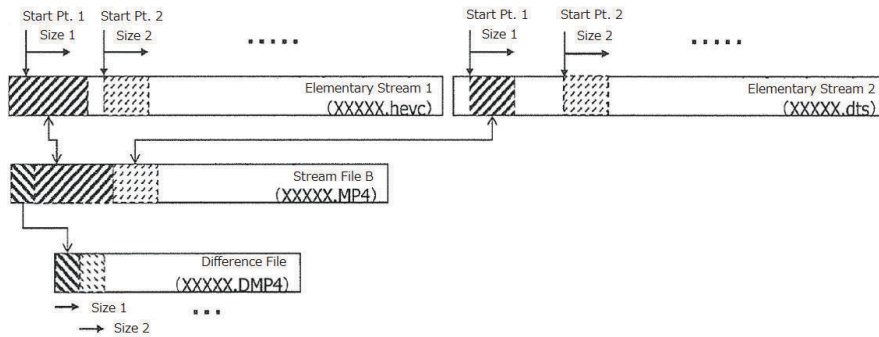
Examples of bit lengths for each field: (int7, int1)

Copy mode indicates whether to skip copying.

A concrete example of address specification when using Mode 2 is shown below. In FIG. 16(a), start position 1 and start position 2 are at the 210th byte and 91,428th byte, respectively. If the field length of the address difference is 16 bits, the maximum field value is 65,535, which cannot be represented by a single entry. Therefore, when jumping from start position 1 to start position 2, two entries, namely, (65,535, 0) and (25,683, 180), are used. This allows the system to jump to start position 2 and then retrieve 180 bytes of data. FIG. 16(b) shows an example of reading a contiguous region that exceeds the maximum copy size limit. In this case, by specifying the entries as (255), (0, 0), and (182), regions (1) and (2) in the difference file can be copied contiguously.

Note that the start address and copy size may be specified in 4-byte units rather than 1-byte units. Also, by including separate identification information in each entry to indicate whether to copy from the difference file or the stream file, it is possible to describe entries for a single file consecutively, rather than alternating between entries from both files.

FIG. 16



Thus far, the process of copying data from an M2TS file to generate an MP4 file has been explained. However, data may also be copied based on elementary streams separated from the payload of PES packets multiplexed into TS packets. In TS files, copy information is required for every 188-byte packet, or every 192-byte packet in timestamped TS used in formats such as BD, and for content with high bitrates, the number of TS packets increases, leading to a larger manifest file size. By using elementary streams separated from the TS, data can be copied on a per-frame basis or in units such as NAL units in HEVC or AVC, thereby significantly reducing the number of entries included in the COPY MANIFEST portion of the manifest file. For example, if a TS file containing multiplexed video is 10 GB in size and each TS packet is 192 bytes, the number of packets would be 55,924,053, requiring the same number of entries. However, if the playback duration of this video is 2 hours and the frame rate is 30 Hz, the total number of frames would be 216,000, allowing for a significant reduction in the number of entries compared to generating an entry for each packet. Furthermore, even in cases in which the M2TS file is doubly encrypted, such as with AACs and BD+, and the data is interleaved and rearranged by BD+, the order of the data constituting the encoded streams can be uniquely determined after decrypting and separating the encoded streams. This makes copy operations using this method possible.

When copying from an elementary stream, since there are multiple elementary streams, identifying information is required. For the COPY MANIFEST, the following examples are possible, corresponding to Mode 1 and Mode 2 shown in FIG. 16. Here, the start address indicates the difference value from the previous entry.

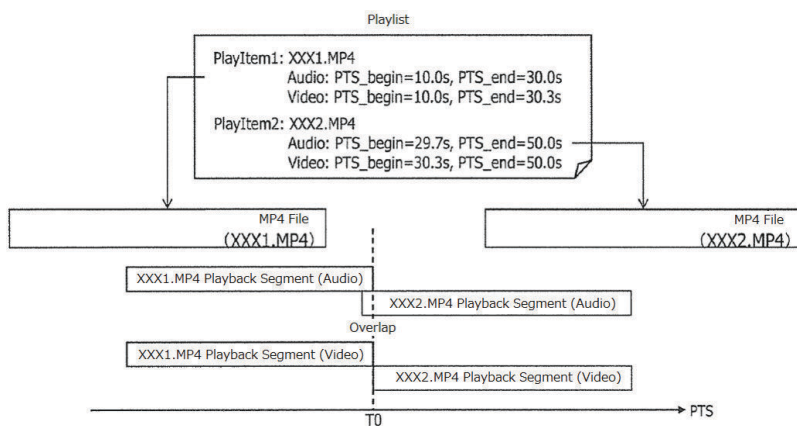
Mode 1: This consists of (file ID, address, size). Example bit lengths for each field: (int4, int22 in 1-byte units, int22)

The file ID identifies the file from which data is copied. For example, 0 is assigned to the difference file, and higher values are assigned to elementary stream files. For HEVC or AVC encoded data with a resolution of 4k, if there is a minimum compression ratio of 4, the maximum size of a single frame is approximately 4 MB. When copying data from an encoded stream, the maximum value of the difference in start addresses can be set to the maximum size of a single frame, and 4 MB can be represented with 22 bits. In this case, 22 bits are used to represent the maximum size of a single frame as the copy size.

Mode 2: This consists of (file ID, address, size). Examples of bit lengths for each field: (int4, int12 in 1-byte units, int16)

Note that in AVC and HEVC, the NAL unit header must be converted from start code format to NAL size format, and since data must be copied in NAL unit payload units, a start address field is provided. However, if the elementary streams in the TS can be copied without alteration, the start address field may be omitted. Alternatively, a single entry may represent an entire NAL unit, and the NAL unit header may be converted to NAL size format after copying. In this case as well, since the data can be read continuously, a start address is not required.

FIG. 17



The BD-ROM standard allows for the continuous playback of multiple M2TS files referenced from a playlist and specifies constraints on streams at file boundaries. For example, when multiple files or playback segments are seamlessly connected, the playback segments of the two audio tracks to be connected may overlap.

In MP4 as well, it is possible to specify multiple MP4 files to be played consecutively using a playlist, and similar constraints can be expected to be applied. FIG. 18 shows an example of audio and video playback segments for two MP4 files played consecutively, in which the audio playback segments overlap. However, DTS and PTS in MP4 are expressed as relative times based on the DTS of the first sample in the file (=0), and cannot represent absolute times. Therefore, when two MP4 files (XXX1.MP4 and XXX2.MP4) with overlapping playback segments are played back consecutively, the first sample of XXX2.MP4 is played immediately after the last sample of XXX1.MP4, resulting in the overlapping portion being played twice. For this reason, information indicating whether the playback segments of XXX1.MP4 and XXX2.MP4 overlap, or specifying the overlapping playback segments, may be stored in the playlist or in the MP4 files. In this example, the playlist describes the PTS of the first sample and the end time of the last sample of the audio and video referenced by the play items in the MP4 files. Additionally, the absolute time of the DTS or PTS of the first sample of the MP4 file may be stored in the moov of the MP4 file, or the absolute times of the DTS or PTS of the first sample of the movie fragment or the first sample of each track in the movie fragment may be stored in the moov or traf. For example, if in XXX1.MP4 and XXX2.MP4, the video playback segments have no overlap or gaps and the PTS values are guaranteed to be continuous, information regarding overlap only has to be provided for the audio or text. If T0 is the PTS of the first sample of the video in XXX2.MP4, the system may store the offset value between the PTS of the first audio crossfade in XXX2.MP4 and T0, or the offset value between the playback end time of the last audio sample in XXX1.MP4 and T0. The offset value between the end time of the final audio sample in XXX1.MP4 and the PTS of the first sample in XXX2.MP4 may also be stored in XXX2.MP4. The playback device uses this overlap information to select and synthesize the audio samples to be output.

Note the video playback segments may overlap, or a gap may be provided instead of overlapping the audio and video playback segments.

FIG. 18

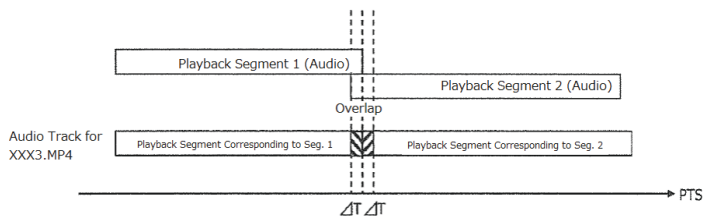


FIG. 18 describes a case in which multiple files are played back consecutively using a playlist, but the playback segments that are played back consecutively can also be combined into a single MP4 file. For example, in FIG. 19, of playback segment 1 and playback segment 2 are combined to generate XXX3.MP4, in which the audio for playback segment 1 consists of 100 samples ranging from sample 1-1 to sample 1-100, and the audio for playback segment 2 consists of 100 samples ranging from sample 2-1 to sample 2-100. The following explanation uses audio as an example. First, if there is neither overlap nor a gap between playback segment 1 and playback segment 2, the audio track of XXX3.MP4 will contain samples 1-1 through 1-100 followed by samples 2-1 through 2-100, and during playback, each sample will be played in sequence. Next, consider the case in which the playback segments for samples 1-100 and samples 2-1 overlap. If the playback segments for samples 1-100 and sample 2-1 (start time: PTS, end time: PTS + playback duration) are identical, the overlap between samples can be resolved by deleting sample 2-1 when constructing the audio track of XXX3.MP4. The same applies when the playback segments of multiple samples overlap. On the other hand, if the playback segments of sample 1-100 and sample 2-1 overlap but are not identical, the issue cannot be resolved by deleting samples. In this case, both sample 1-100 and sample 2-1 are stored in the audio track of XXX3.MP4, and information indicating the overlapping playback segment is stored separately. Here, ΔT is the duration of the overlapping portion. For example, treating the last sample of playback segment 1 and the first sample of playback segment 2 as separate movie fragments, ΔT can be stored in traf of moof to indicate that the segment from the start of the audio track in the movie fragment containing playback segment 2 overlaps with the playback segment of the audio track in the preceding movie fragment. If a gap exists, this can be handled by generating a non-playback segment corresponding to the gap using the MP4 edit list functionality or by setting the "duration-is-empty" flag in the movie fragment. Furthermore, overlapping and gap segments may be indicated separately in the metadata in the MP4 header or in an external file. It should be noted that MP4 files containing gap segments are more compatible with conventional MP4 files than those containing overlapping segments. Therefore, when merging playback segments that contain overlapping segments to generate an MP4 file, for example, in playback segment 2 shown in FIG. 19, samples containing overlapping segments are removed. If a gap occurs as a result, it is addressed by generating a non-playback segment. This method can be applied not only to audio but also to video and subtitles. Furthermore, information indicating whether deletion is necessary, the samples that need to be deleted, and the duration of any gaps resulting from the deletion may be stored separately as support information, such as in a copy manifest.

When playing back these MP4 files, based on information contained in the MP4 file, such as overlapping segments, the system can output the decoded audio samples by combining them in the overlapping segments. Furthermore, when merging playback segments that are played consecutively, it is necessary to identify whether overlapping segments or gap segments exist in the segments to be merged. However, information indicating the presence or absence of overlapping segments, etc., as well as their durations, may be obtained separately as support information from an optical disc or from the internet.

FIG. 19

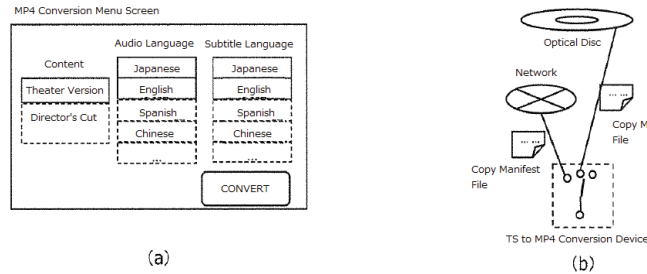
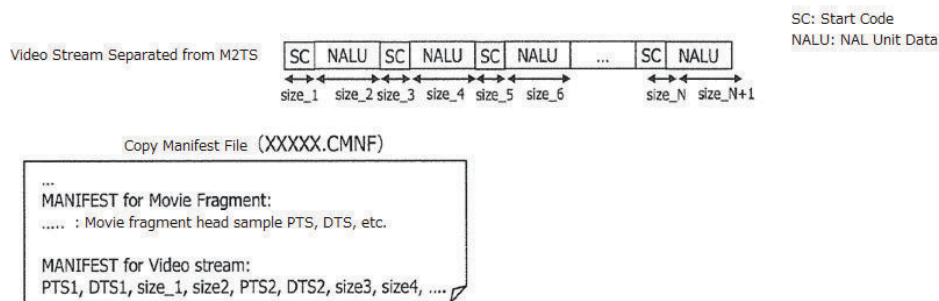


FIG. 20(a) shows an example of a menu screen for generating an MP4 file from content stored on an optical disc. The disc contains audio and subtitles in multiple languages, and when generating the MP4 file, the user can select the languages to be included. In this example, from the Japanese, English, Spanish, and Chinese language content stored on the disc, both audio and subtitles are selected in Japanese and English. Here, if the sample size for audio or subtitles differs by language, the contents of the copy manifest file depend on the sample size; consequently, the number of copy manifest file types increases in proportion to the combinations of selected languages for audio and subtitles. Therefore, the MP4 file may always store audio and subtitle data for all languages, while storing information indicating the user's selected language separately, allowing the user to select their preferred language during playback. By doing so, the copy manifest file can remain identical regardless of the selected language. Alternatively, two copy manifest files could be prepared: one that stores only one language and another that stores all languages. Furthermore, regarding audio encoding formats such as AAC and AC3, users could be allowed to select the format based on the support capabilities of the device playing the MP4 file, or all audio data in every encoding format could be stored in the MP4 file. When audio data for all encoding formats is included, the encoding format is selected during playback based on user choice or preset information in the playback device.

Alternatively, audio and subtitles for all languages may be included in the MP4 file during generation without requiring language selection, leaving the choice to the user during playback. Furthermore, if the copy manifest file can be obtained via a network, there is no need to store it on an optical disc. This is particularly effective when the number of copy manifest files increases, such as when any language can be selected. Only the copy manifest file corresponding to the default language combination may be stored on the optical disc, while copy manifest files corresponding to other combinations may be downloaded from a server. Also, the languages of the audio and subtitles available on the optical disc or via the network, as well as the languages of the audio and subtitles contained in the MP4 file, can be retrieved in advance. Users can then select and retrieve languages not included in the MP4 file but available externally. Alternatively, a list of audio tracks available from an external source may be stored in the MP4 file or elsewhere, and when playing the MP4 file, if the playback device cannot decode the audio encoding format in the MP4 file, the device may select audio in an encoding format it supports and retrieve it from an external source. In this case, the data retrieved from an external source may consist solely of encoded data containing subtitles or audio, or it may be an MP4 file itself. In such cases, during playback, the video and other content contained in the original MP4 file are synchronized with the newly retrieved data. Alternatively, a complete MP4 file containing video, audio, and subtitles may be retrieved from an external source.

Furthermore, the content stored on an optical disc may be in MP4 format rather than M2TS. In this case, without performing any conversion, the MP4 data stored on the optical disc may be copied or exported without alteration to device-bound or media-bound recording media or devices. During copying or exporting, the encryption key for the content may be changed. Devices that generate MP4 files, such as BD players, may determine whether the content on the optical disc is in M2TS or MP4 format based on data format identification information, assess whether conversion to MP4 is necessary, and generate MP4 data accordingly. Alternatively, support information, such as a copy manifest file, may indicate whether conversion to MP4 is required. Furthermore, if the content stored on the optical disc is in MP4 format, the user may select specific languages for audio and subtitles in the content, or select a version such as a theatrical release or director's cut, and based on the selection results, the selected data alone may be extracted from the MP4 data on the optical disc to generate an MP4 file.

FIG. 20

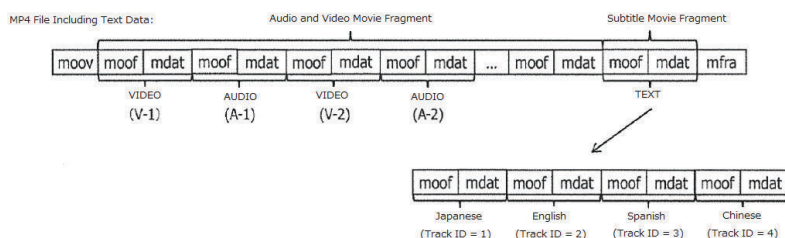


As explained in FIG. 15, specifying information in the copy manifest file to identify the multiplexing units in the output MP4 file can also reduce file size, which is effective when there are a large number of copy manifest files. In this case, the copy manifest information, such as the PTS or DTS of the sample at the beginning of a movie fragment, or other information from the M2TS file from which the sample data is obtained, can be used to uniquely determine the movie fragment units in the MP4 file, and language-independent information can be used. However, in the MP4 file header information, elements such as the Box contained in the stbl in moov and trun in moof vary depending on the sample size. Therefore, they cannot be included in a copy manifest file intended for use across multiple languages. Therefore, when converting to MP4, the unit of the movie fragment is determined based on the copy manifest file, and the MP4 header information is generated separately based on the PTS and DTS for each sample, as well as the sample size.

When obtaining PTS, DTS, or sample sizes for each sample, it is necessary to search for access unit boundaries in data such as audio, video, and subtitles, whether multiplexed into the M2TS or stored separately in a different area, or to parse PES packet headers, and this results in a high processing load for high-bitrate video. Therefore, at least for video, the size of the NAL units constituting an access unit (equivalent to an MP4 sample) and information indicating PTS and DTS may be stored in the copy manifest. This information not only significantly reduces the processing required to search the video stream but also offers the advantage of enabling conversion from the start code format in M2TS to the NAL size format in MP4 using the size information. Size information may be specified separately for the start code portion and the NAL unit portion, and by aligning the byte length of the field indicating the NAL unit size with the byte length of the size portion in the NAL size format, the start code portion data can be converted to the NAL size format by replacing it with data indicating the size of the NAL unit portion. Furthermore, if the size of the start code portion is identical throughout, only a default value needs to be set. Regarding PTS and DTS, the data format used in the MP4 header may be employed instead of the PTS and DTS values in the M2TS PES header (for example, for DTS, the difference between the DTS values of two consecutive samples in decoding order may be indicated, and for PTS, the difference between the DTS and PTS values may be indicated). The time scale of this information may be converted to the MP4 time scale. Furthermore, information indicating the absolute values of the PTS and DTS of the first sample may be stored separately. Additionally, information for identifying NAL units to be deleted during MP4 conversion may be added. Also, when storing similar support information for the AAC-encoded stream encoding stream, header information such as ADTS or LATM is removed from the sample data. However, if the size of this header information is fixed, it is permissible to indicate only the sum of the header information and payload data sizes, or the payload data size alone. Additionally, in the case of audio, since the frame rate is fixed, the DTS information may consist solely of default values.

Note that if the audio encoding method differs, the playback duration per sample may also differ. As a result, since the PTS and DTS per sample also differ, a separate copy manifest file may be prepared for each audio encoding method.

FIG. 21



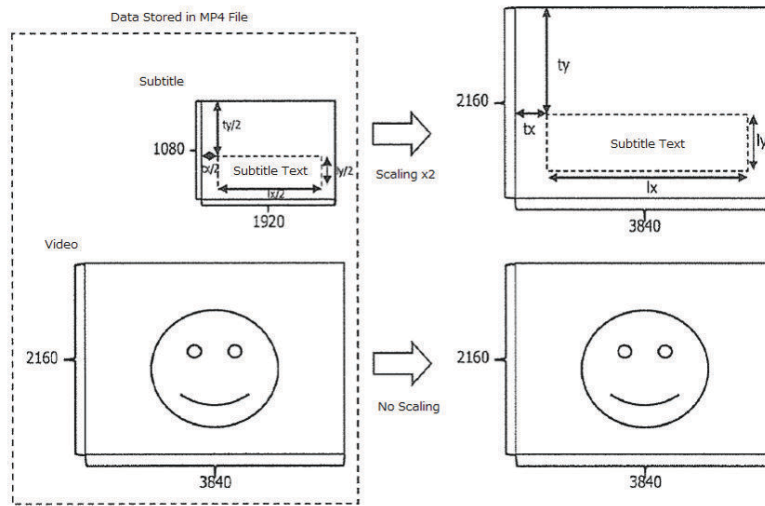
When storing subtitle data in an MP4 file, grouping the subtitle data at the end of the file ensures that the AV data movie fragment remains the same even if the subtitle language is changed. On the other hand, if the video and audio movie fragments are interleaved with the subtitle movie fragments, a change in the size of the subtitle data will cause the positions of the AV data movie fragments to shift, resulting in the need to modify the content of the moov. Also, since subtitle data is smaller in size than AV data, it is possible to load the subtitle data for the entire content, or in units such as chapters into which the content is divided, into bulk memory for playback. In this case, this also has the advantage of making data easier to retrieve when it is stored collectively at the end of the file. Here, the subtitle data may be based on a text font or may be image data such as PNG format. In the case of image data, since the file size is larger than that of text data, it is possible to generate blocks for each unit (such as chapters) to improve accessibility to the subtitle data contained in that unit. Alternatively, depending on the buffer size used to hold text data during playback, blocks can be generated so that the size of the subtitle data comprising each block is less than or equal to the buffer size.

Also, when storing subtitle data in multiple languages, storing each language in a separate movie fragment makes it easy to retrieve subtitle data for a specific language. In this case, information is required to identify the language stored in each movie fragment. This can be achieved, for example, by treating each language as a separate track and associating a track ID with the languages. The track ID is indicated by the Box element in the traf element. The information mapping track IDs to languages may be stored in the Box used for metadata storage in MP4, or as management information separate from the MP4 file. In addition, the mapping between languages and movie fragments can also be applied to audio.

Randomly accessible samples in video, audio, or subtitles are indicated by mfra. Here, for video and audio, when the playback ranges of consecutive video and audio movie fragments overlap, it is sufficient to specify only the random access point of the video. Audio samples with the same PTS, or those immediately preceding or following it, can be retrieved from the subsequent movie fragment. For example, in FIG. 22, the PTS of the first sample of V-1 aligns with that of the first sample of A-1. On the other hand, when text is stored at the end of the file, separate random access points must be specified for the text.

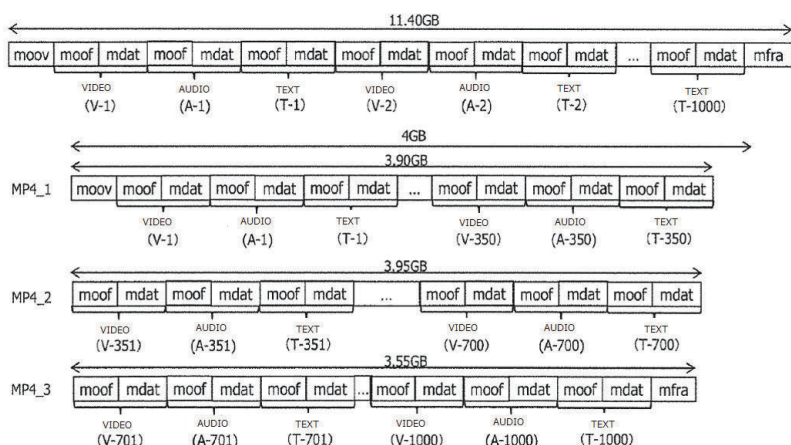
Also, while audio and subtitle data for all languages in optical disc content are stored in the MP4 file, if the user selects a desired language when an MP4 file is generated, random access points in the mfra may be specified only for the tracks corresponding to the selected language.

FIG. 22



Here, it is assumed that the resolutions of the video and graphics differ in the content stored on an optical disc. For example, the video resolution might be 4K, while the resolution of graphics such as subtitles might be set to 2K to reduce the processing load. FIG. 23 shows a case in which subtitles with a resolution of 2K are scaled up to 4K for display. To display subtitles, information is required not only for the subtitle data and its resolution but also to specify the display area for the subtitles. The display area is specified, for example, using the size of the display area rectangle and the rectangle display position. For instance, information indicating the track resolution could specify that the subtitle track is 2K and the video track is 4K. Also, regarding the display area, in standards such as SMPTE or W3C Timed Text, it may be described as part of the XML data that constitutes the Timed Text, or it may be stored in the Box that indicates metadata in an MP4 file. When playing an MP4 file, the resolutions of the video and subtitles are retrieved, and if their resolutions differ, the subtitles are scaled to match the video resolution before being displayed. In this case, if the subtitles are image data, the image data is enlarged, and if they are text data, a size consistent with the video resolution is selected. The display area is also calculated and determined based on the scaling factor and other parameters. Note that the information indicating the display area may be stored after it has been scaled to match the video resolution.

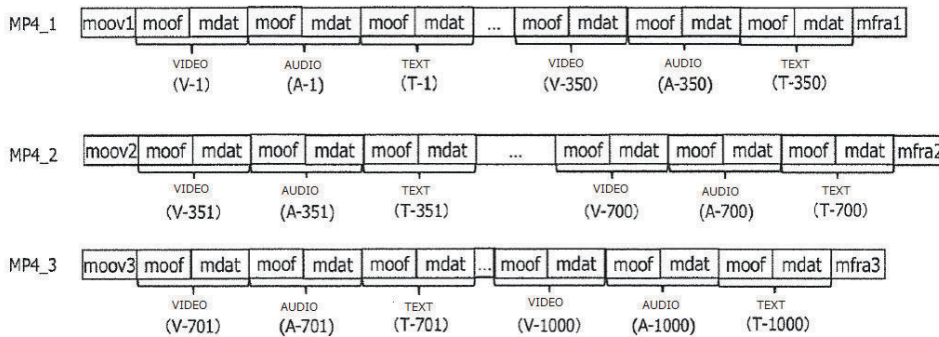
FIG. 23



MP4 files generated from data such as M2TS are saved to storage devices, such as removable media (e.g., SD cards) or the internal storage (e.g., hard disk drives) of smartphones or tablets, where the MP4 files will be stored. In this case, the maximum file size that can be handled is limited by the file system of the storage device. For example, the limit is 4 GB for an SD card and 32 GB for an SDHC card. Therefore, if the size of the MP4 file containing all of the content exceeds the file system limit of the storage device, the MP4 file must be split into multiple parts. FIG. 23 shows an example of splitting an MP4 file, whose total size is 11.40 GB, into three files, each 4 GB or less, in order to store it on an SD card. In MP4 files using movie fragments, it is desirable to store movie fragments with identical or nearly identical playback segments, such as for audio, video, or subtitles, in the same file. This is because when audio or video data with the same playback range is stored in different files, the system must access data from multiple MP4 files during playback. Depending on the data retrieval speed and seek performance of the recording device, this can lead to issues such as the need to buffer additional data before playback compared to playing from a single file. Therefore, MP4 files are generated so that audio, video, and subtitle movie fragments with the same or adjacent playback ranges are included in the same MP4 file, while ensuring the file size does not exceed 4 GB. By keeping the file size at 4 GB or less and storing as many movie fragments as possible in a single file, the number of generated files can be minimized.

Furthermore, since the data size of audio and subtitles is smaller than that of video, data for a longer duration can be buffered. Therefore, rather than storing all movie fragments of different media with the same playback range in a single file, it is acceptable to split them, at least video movie fragments, so that they are contained in the same file. For media other than video, splitting movie fragment data into different files may be prohibited.

FIG. 24



As shown in the example in FIG. 24, if the data from the original MP4 file is split without alteration, the files starting from the second one will not contain a moov file. Therefore, to obtain the initialization information required for decoding each media stream stored in a sample entry, it is necessary to refer to the first file. Also, since the mfra is included only in the final file, random access to MP4 files other than the final one requires referencing the mfra stored in the final MP4 file. Consequently, each MP4 file cannot support self-contained playback. Therefore, a moov or an mfra can be added to each file. Regarding the moov, since the sample entry of the original MP4 file contains all the initialization information required for the entire content, its contents do have to be modified. Similarly, while other Box elements generally do not require modification, the tkhd or mvhd of each track stores the total playback duration for the entire file. Therefore, these playback durations may be modified to correspond to the playback durations of the split MP4 files. Alternatively, if support information indicates that the files have been split, modifying the playback durations may not be necessary. On the other hand, the content of mfra may also be modified to indicate random access points in each MP4 file. The random access information in mfra specifies the offset from the beginning of the moov file to the sample that serves as the random access point. If the content of the mfra is not modified, the offset values will correspond to the original MP4 file and will not match the offset values of the split files. Therefore, when locating random access points in the split files, for the second and subsequent files, the sample data serving as the random access point is obtained by adding the total size of the MP4 files that come before them in playback order to reproduce the file offset values of the original MP4 file.

If modifying the content of moov or mfra in any of the split files results in an update to the size of moov or mfra in the original MP4 file, the files must be split so that the updated file size does not exceed the maximum size allowed by the file system.

Note that the playback order of the split files is identified by the file names or by content management information that stores information indicating the playback order of the files.

Also, as shown in the example in FIG. 22, when storing subtitle data collectively at the end of the MP4 file without interleaving, information for identifying the segment files containing the subtitle data is included in the content management information. During playback, the subtitle data is retrieved by accessing the segment files in which it is stored. Since subtitle data is usually small in size, it is desirable for all subtitle data to be contained in a single file. Here, the initialization information required to decode the subtitle data is stored in the MOOV header, just as it is for audio and video.

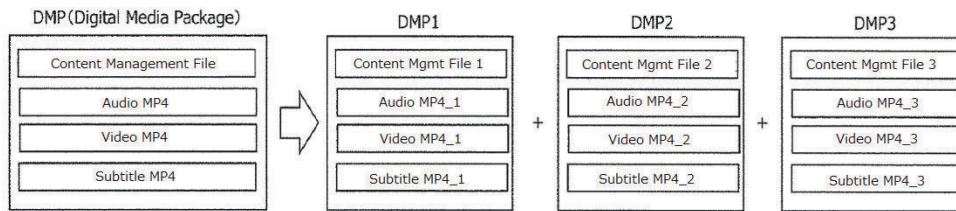
FIG. 25

When generating an MP4 file, it is necessary to obtain in advance the maximum file size limit on the target file system for the original MP4 file. Therefore, if the output destination is another device such as a tablet, communication is performed via some means with the connected device, and if the output destination is a device equipped with an optical disc (such as a BD player) and the SD card connected to that device is a storage medium such as an HDD, communication is performed in the device to obtain information indicating the maximum file size of the file system at the output destination or information indicating whether the original MP4 file needs to be split. For example, prior to generating the MP4 file, the MP4 generating module may take the initiative to obtain this information, such as by requesting information indicating the maximum file size in the file system of the target storage medium. Alternatively, the target device can be notified of the original MP4 file, and if splitting is necessary, have the target device send information indicating the maximum file size to the MP4 generating module.

When other devices retrieve MP4 files via a communication network, they can first retrieve a metadata file containing information such as the available languages, the source of the MP4 files, and the URL of the DRM (Digital Rights Management) management server, and then issue a request to retrieve the MP4 file based on the results of analyzing that metadata file. In such cases, information indicating the maximum file size may be included in the MP4 file retrieval request message or command.

In addition, the system may be designed to accommodate updates to these metadata files. The default metadata file is retrieved from the optical disc, and if an update is available, the updated metadata file is retrieved via the network. By specifying the URL of the management server that the playback device accesses first in the metadata file, the management server can determine whether the metadata file is up-to-date based on the version number of the metadata file on the optical disc, and if it is not up-to-date, the management server can notify the playback device of the source for the latest metadata file. Also, depending on the country or region, the DRM, content source, or content details described in the default metadata file may differ. As in the case of metadata file updates, the management server can process these differences based on identification information for the country or region where the playback device is located. The same applies when the content of metadata file updates differs by country or region.

FIG. 26



When generating MP4 files, it is common to multiplex data from different media types, such as audio, video, or subtitles, into a single file. However, there is also a format that creates separate MP4 files for each media type. This format is called DMP (Digital Media Package), and during playback, the media data obtained from multiple MP4 files is played back in synchronization. In DMP, MP4 files and content management files are archived using the ZIP format or some other similar format. However, if the archived file exceeds the maximum file size of the target file system, it must be split. When splitting the file, the process can be performed in the same manner as the splitting methods described in FIG. 24 to FIG. 26. In other words, the MP4 files for each media type comprising the same DMP are split so that their playback segments are identical or adjacent. Alternatively, one can ensure that a single movie fragment in an MP4 file for a given media item is not stored across multiple DMPs. The same applies to the handling of moov and mfra files. Also, since audio and subtitles have smaller data sizes compared to video, only the video needs to be split into multiple files. All audio and subtitle data may be stored in the first split file.

Regarding the content of the content management file, attribute information pertaining to all of the content may be stored in the content management file contained in the first DMP after splitting. Also, the content management file stored in the first DMP may also contain identification information for the subsequent split DMP files, as well as information indicating the playback order. In the content management files in subsequent DMPs, one may describe only the archive information for the files in the DMP or the DMP's identification information, without describing information pertaining to the content as a whole. Note that instead of generating multiple DMPs, the data of a single DMP may be split into multiple files.

In addition, as explained in FIG. 18 and FIG. 19, it is possible to play back MP4 files comprising multiple playback segments consecutively in a DMP by referencing them via a playlist or similar mechanism. When multiplexing multiple media streams into a single file, the destination and source are each a single file. However, in a DMP, the destination and source each consist of multiple files, and the playlist contains the identification information for those multiple files. Also, audio gaps and overlaps at the connection points between files played consecutively can be handled in the same manner using the processing described above.

FIG. 27

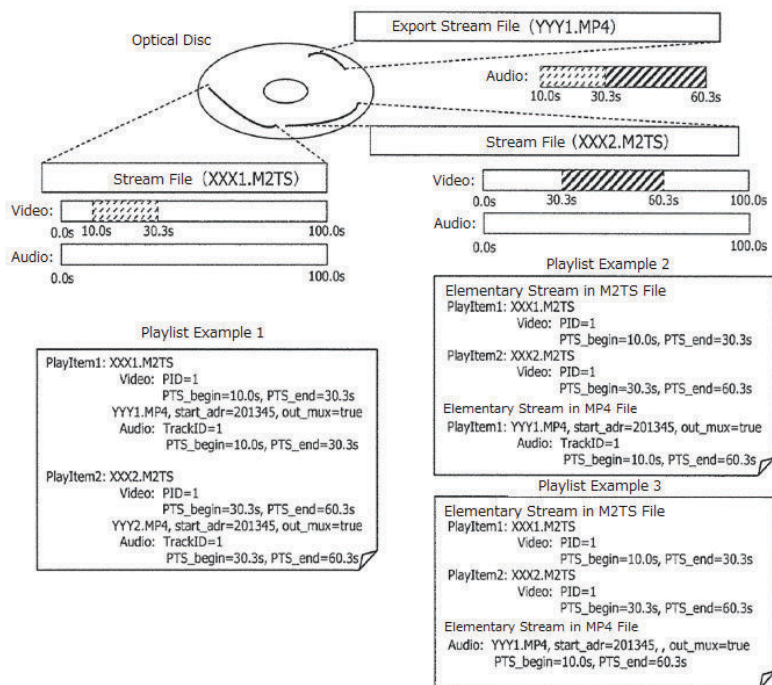


FIG. 28

When generating MP4 files, data such as audio, video, and subtitles can be extracted from content stored on an optical disc and multiplexed into a single MP4 file, or multiple MP4 files can be generated in DMP format. For example, the video elementary stream can be separated from M2TS data stored on an optical disc to generate a video MP4 file, and for the audio and subtitles, MP4 files containing audio and subtitles stored separately from the M2TS can be obtained. These MP4 files can then be combined into DMP format. In this case, the MP4 files for audio and subtitles may be independent of each other, or they may be multiplexed together. If multiple languages are present, it is also possible to multiplex the audio and subtitles for the same language into a single MP4 file. During export, the items to be exported can be determined by referencing an export playlist, and this playlist may be stored on the disc in advance or downloaded via a network.

FIG. 28 shows the method for storing export data and provides an example of an export playlist. Here, two M2TS files with multiplexed AV (XXX1.M2TS and XXX2.M2TS) and an MP4 file with multiplexed audio for export (YYY1.MP4) are stored on the disk. Subtitles can be handled in the same way as audio. Playlist example 1 consists of two play items, and for each play item, it is indicated that the video is taken from XXX1.M2TS and XXX2.M2TS, and the audio is taken from YYY1.MP4 and YYY2.MP4. For the YYY1.MP4 and YYY2.MP4 files, the starting position of the file can be indicated by its logical address on the disk. Alternatively, the files may be distinguished solely by their file names without explicitly specifying the logical address. Also, out_mux may indicate whether the relevant data is multiplexed in an M2TS file or stored as a separate file from the M2TS. Additionally, while the audio data is indicated by the MP4 track on which it is stored and the TrackID, in cases in which the MP4 file consists of a single track (hereinafter referred to as a "single-track MP4"), such as with DMP, the TrackID field may be omitted. In a single-track MP4, it is permissible to identify the type of media stored in the file, such as audio, video, or subtitles, based on the file name, and it is also permissible to indicate the language or encoding method. Regarding the encoding method, rather than specifying the encoding method itself, it is permissible, in the case of audio, for example, to distinguish the audio quality. For example, lossless-compressed audio might be classified as high quality, while audio compressed using MPEG-2 or MPEG-4 AAC might be classified as medium quality. Playlist example 2 shows a case in which the audio is stored in an MP4 file, but the video represented by play item 1 and play item 2 is provided as a continuous playback segment in YYY.MP4. Play item example 3 is similar to play item example 2, except that YYY.MP4 is a single-track MP4 file and the TrackID field is omitted.

Playlists for export, such as those shown in playlist example 1 through playlist example 3, may be distinguished from playlists used to play back content on the disc by means such as file names. Also, playlists for export and playlists for playing back content on the disc may be stored in different directories in the file system.

Also, in the exported MP4, features such as PinP (Picture-in-Picture), multi-angle, 3D, or subpaths and browsable slideshows may be prohibited. However, whether or not such functional restrictions are imposed may be indicated by the file name or by metadata used to manage the export playlists. If a playlist for export contains features that are prohibited, the playlist may be deemed invalid, and error handling, such as not generating an MP4 file for export, may be performed.

In addition, the export playlist may be obtained by downloading it over a network, and the M2TS or MP4 data referenced by the export playlist may be downloaded over a network. When downloading M2TS or MP4 files over a network, the IP address or URL of the download destination may be specified separately in the playlist. Alternatively, if the content used for export is stored collectively in a single directory on the server, the URL of that directory may be specified in the metadata associated with the export playlist, and only the file names may be listed in the playlist itself. When retrieving content, the complete URL is constructed by concatenating the URL specified in the metadata with the file name before downloading the file.

FIG. 29

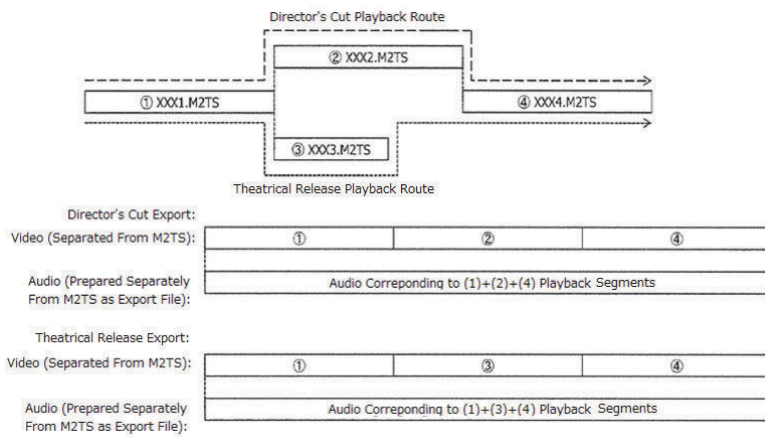


FIG. 30

Optical discs such as BDs may contain multiple versions of the main content, such as a director's cut and a theatrical release version. For example, in the figure above, the director's cut plays in the order of playback segments (1), (2), and (4), while the theatrical release version plays in the order of playback segments (1), (3), and (4). The audio for export is stored as single track MP4. When exporting, the video is generated by concatenating the elementary streams separated from the three M2TS files, while the audio elementary stream is pre-stored in a single MP4 file. In other words, in the MP4 file for the director's cut version, the elementary streams corresponding to playback segments (1), (2), and (4) are stored consecutively, while in the MP4 file for the theatrical release version, the elementary streams corresponding to playback segments (1), (3), and (4) are stored consecutively. In this way, by consolidating the audio elementary streams for export into a single elementary stream in advance, an advantage is conferred in that no audio overlap or gaps occur at the junctions between playback segments (2) and (4), or between playback segments (3) and (4), for example.

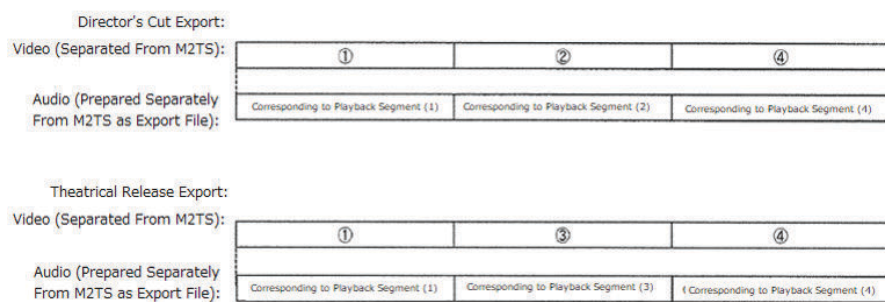
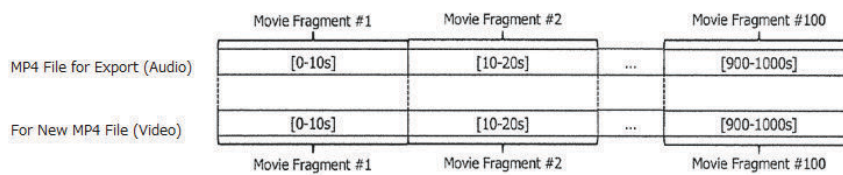


FIG. 30 shows the case in which the audio for export is contained in a single MP4 file. However, it is also possible to store multiple MP4 files corresponding to the video playback segments. In this case, audio overlaps or gaps will occur at the junctions between playback segments, but these should be handled using the method described above.

FIG. 31



When creating DMP-format content, it is desirable for the movie fragment units across different media, such as audio, video, and subtitles, to be aligned. Generally, when randomly accessing and playing back content from a point midway through the content, the PTS of the random access point is determined based on the random access point of the video. For audio and subtitles, playback begins from the frame with a PTS that is identical to, or immediately before or after, the PTS of the random access point in the video. For audio and subtitles, when using pre-stored single-track MP4 files intended for export, the data in these MP4 files is exported without modification. Therefore, for MP4 video files generated based on the elementary streams extracted from an M2TS file, they should be generated so that the movie fragment units align with those of the audio and subtitles intended for export. By doing so, during random access, playback can begin from the movie fragment with the same index number across all media. Note that the sample at the beginning of a movie fragment that is DTS is a randomly accessible picture, such as an I-picture or IDR picture in MPEG-4 AVC or HEVC.

FIG. 32

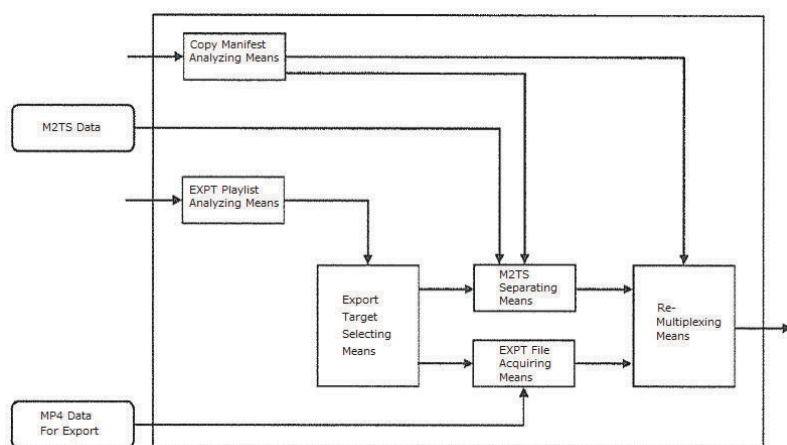


FIG. 33 is a block diagram showing the configuration of a multiplexer that generates MP4 files for export. This multiplexer generates and outputs MP4 files containing audio, video, and/or subtitles based on the elementary streams extracted from M2TS and single-track MP4 files intended for export. The output MP4 may be in a format that stores multiple media elementary streams in a single file, or it may be in DMP format.

The EXPT playlist analyzing means analyzes one or more export playlists obtained from a disk or via a network. The export target selecting means, based on the analysis results from the EXPT playlist analyzing means, generates a menu for the user to select the content provided by the playlist to be exported, as well as whether or not to export language settings or subtitle data in the content, and determines the playlist to be exported based on the user's selection. Next, the system determines the data in which the elementary streams constituting the user-selected content are stored. If the elementary streams are stored in an M2TS file, the M2TS separating means separates the elementary streams based on the copy manifest information provided by the copy manifest analyzing means. If the elementary streams are provided in an MP4 file for export, the EXPT file acquiring means acquires the relevant MP4 file. Finally, the remultiplexing means generates an MP4 file for output using the elementary streams acquired from the M2TS separating means and the MP4 file acquired by the EXPT file acquiring means. When storing multiple media streams in a single MP4 file, the elementary streams are first separated from the export MP4 data, and then the audio, video, or subtitle elementary streams are multiplexed into a single MP4 file. When outputting in DMP format, a single-track MP4 is generated from the elementary streams outputted by the M2TS separating means, and DMP content is generated by combining this with the export MP4 data outputted by the EXPT file acquiring means. In this case, metadata for managing the multiple MP4 files contained in the DMP are generated separately.

FIG. 33

Arrangement of Data For Export Stream File

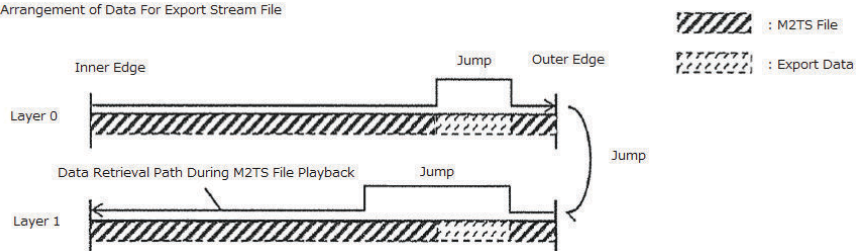


FIG. 34 shows the storage locations on an optical disc for data used during export, such as the MP4 file for export. Since it is desirable for export data to be read or copied quickly, it is stored in a contiguous area on the outer periphery of the optical disc. Meanwhile, for M2TS files, it is essential to place enough data on the outer periphery to ensure that audio, video, or graphics data does not underflow when jumping between layers of the optical disc. For example, if the time required to jump from Layer 0 to Layer 1 is T_1 , the time from jumping to Layer 1 until video data can be acquired is T_2 , and the video bitrate is R , then at least $R \cdot (T_1 + T_2)$ worth of video data must be buffered immediately before jumping from Layer 0. The bitrate R may be the maximum bitrate specified by the profile or level used to encode the video, or it may be the average bitrate of the actual encoded data. Here, the average rate refers to the average rate of the video over the segment from T_{dec1} , the time corresponding to the moment immediately before the jump, to $(T_{dec1} + T_1 + T_2)$. Furthermore, jumps occur during M2TS playback even in the area where export data is stored. Therefore, the storage area for export data must be determined so that the M2TS data does not underflow when jumping through that area. Thus, while ensuring that the M2TS data does not underflow during jump playback, the export data is placed as close to the outermost edge as possible. Also, for MP4 files intended for export, the same file is preferably stored in the same layer.

FIG. 34

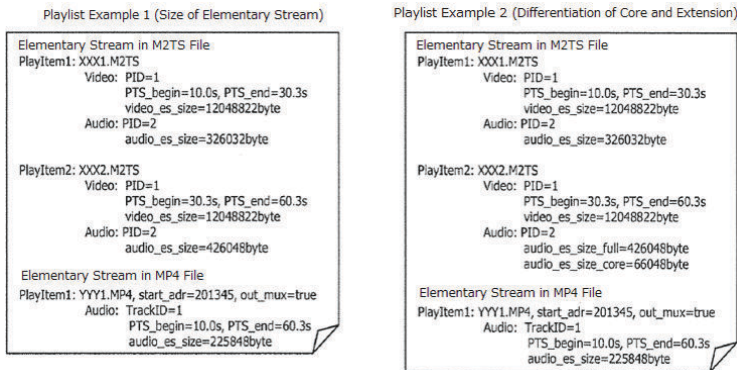


FIG. 35 shows an example of the sizes of elementary streams in a playlist for export. In playlist example 1, the sizes of the elementary streams multiplexed into M2TS and MP4 files are shown. During export, the data size of the exported MP4 file can be estimated based on these sizes, allowing the system to determine whether there is sufficient free space on the export destination storage medium to store the exported MP4 file. Alternatively, if the copy manifest contains data from the header section of the MP4 to be exported, the size of the MP4 header can be determined. When combined with the size of the elementary streams, this allows the data size of the exported MP4 to be calculated. Playlist example 2 shows the sum of the sizes of the base portion and the extended portion (such as extensions for lossless encoding), as well as the size of the base portion alone, when the audio consists of a base portion and an extended portion. This allows the system to determine the size of the output data when exporting only the base portion, only the extended portion, or both the base portion and the extended portion together. This applies not only to audio but also to video encoded using a layered structure.

FIG. 35

Playlist Example (Playlist Connection Conditions)

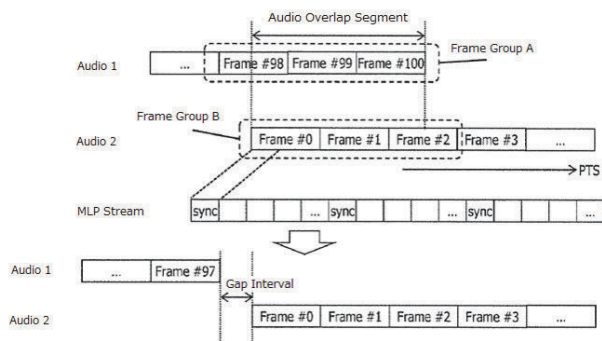
```
Elementary Stream in M2TS File
PlayItem1: XXX1.M2TS
  Video: PID=1
    PTS_begin=10.0s, PTS_end=30.3s
    video_es_size=12048822byte
  Audio: PID=2
    audio_es_size=326032byte

PlayItem2: XXX2.M2TS
  Video: PID=1
    PTS_begin=30.3s, PTS_end=60.3s
    video_es_size=12048822byte
  Audio: PID=2
    audio_es_size=426048byte
    connection_condition=seamless_w_gap
    gap_duration=20msec

Elementary Stream in MP4 File
PlayItem1: YYY1.MP4, start_addr=201345, out_mux=true
  Audio: TrackID=1
    PTS_begin=10.0s, PTS_end=60.3s
    audio_es_size=225848byte
```

FIG. 36 shows an example of how information on gaps that occur at the junctions between play items is indicated in a playlist. If connection_condition is set to seamless_w_gap, this indicates that while video is seamlessly connected between play items, gaps occur in the audio. If a gap exists, its duration can be specified using gap_duration. In MP4, it is necessary to include audio gaps when concatenating multiple files. However, using this information, it is possible to determine whether a gap exists and, if so, obtain the duration of the gap, thereby setting the duration of the gap at the concatenation point of the MP4 file as part of the MP4 header information. For content on optical discs, only audio overlap is permitted between playback items. Therefore, the mode corresponding to seamless_w_gap is for export purposes only.

FIG. 36



Playlist Example (Specifies Export Range)

Elementary Stream in M2TS File

PlayItem1: XXX1.M2TS

Video: PID=1

PTS_begin=10.0s, PTS_end=30.3s

video_es_size=12048822byte

Audio: PID=2

audio_es_size=326032byte

last_frame_pts=30.1s

PlayItem2: XXX2.M2TS

Video: PID=1

PTS_begin=30.3s, PTS_end=60.3s

video_es_size=12048822byte

Audio: PID=2

audio_es_size=426048byte

FIG. 37 shows an example of a method used to generate audio gaps during export.

In streams in an M2TS file, if the playback segments of audio frames are continuous at a seamless connection point, the audio playback segments between play items are made to overlap. When concatenating two audio streams that overlap with each other into an MP4 file, it is necessary to delete the audio frames at the connection point to create a gap. In the figure above, when concatenating audio 1 and audio 2, the three frames from frame 98 to frame 100 of audio 1 are deleted to create a gap between frame 97 of audio 1 and frame 0 of audio 2. Here, during export, it is desirable to be able to identify the frames that need to be deleted to create the gap.

The playlist example shown in the figure on the left shows the frames that must be deleted to create a gap. When concatenating play item 1 and play item 2, the PTS of the final frame of the audio referenced by play item 1 is indicated by last_frame_pts. For example, if the PTS of frame 97 in audio 1 in the figure above is 30.1s, then last_frame_pts is set to 30.1s. Also, frames to be deleted may be identified based on information other than PTS, such as the frame index number or storage address. Additionally, as shown in the playlist example in FIG. 36, the duration of the gap may also be specified. In the explanation above, the support information used during export was stored in the playlist. However, it may also be stored in playback management metadata separate from the playlist.

FIG. 37

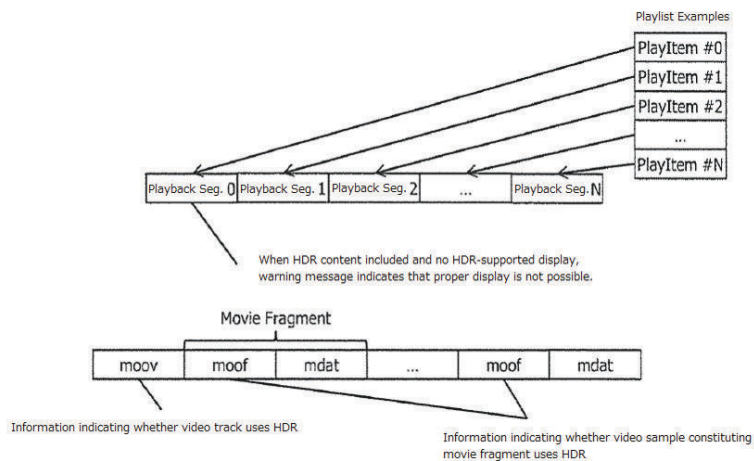


FIG. 38 shows an example of storing video data with luminance levels beyond the range of the BT.709 standard (SDR: Standard Dynamic Range), known as HDR (High Dynamic Range), in an MP4 file. Playing HDR content requires an HDR-compatible display. When playing on an SDR display, processing such as mapping the HDR luminance range to the SDR luminance range is necessary. In this example, a zero-indexed play item, the first item to be played in the export playlist, contains HDR content and includes a warning message indicating that the content cannot be displayed correctly on a non-HDR display. When generating the MP4 file during export, the playback segment corresponding to the 0th play item is stored as the initial data in the MP4 file. Therefore, this warning message is displayed first when the MP4 is played. Note that the warning message may consist solely of subtitles. Alternatively, the presence of HDR content in the MP4 file may be signaled in the MP4 header information. For example, in the moov, it is possible to indicate whether at least some playback segments in a video track use HDR, and in the moof, it is possible to indicate whether the video samples comprising a movie fragment use HDR. When playing back HDR content, luminance information, such as luminance levels and white levels, which can be switched on a per-stream or per-scene basis, is used to adjust the gamma curve according to the maximum luminance of the output display. Therefore, luminance information may also be stored as part of the MP4 header. If the luminance information is fixed in the MP4 file, it should be set as information in the moov field, and if it varies in the MP4 file, it may be set as information in the moof field. In this case, the luminance information is assumed to be the same in the movie fragment.

Note that since switching between HDR and non-HDR modes, or changes in luminance information, are visually noticeable and may cause discomfort to the user, HDR usage and luminance information may be kept fixed during segments that are played back seamlessly and continuously. For example, in a playlist for playing M2TS files, information indicating that HDR-related information is fixed between play items may be stored. Similarly, in MP4 files, information indicating that HDR-related information is fixed in the video track may be stored.

Also, HDR-related information, etc. may be stored as metadata used to manage the MP4 file, rather than being stored directly in the MP4 file.

FIG. 38

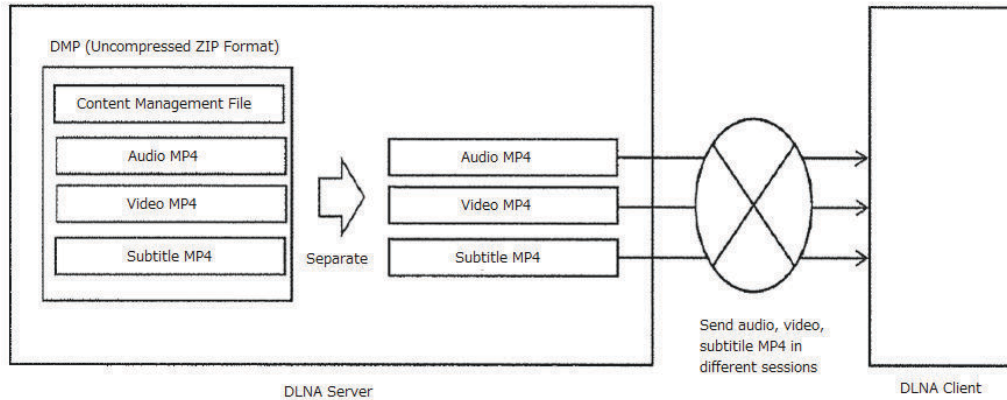


FIG. 39 is a diagram used to explain the process of viewing DMP data, exported from an optical disc to an SD card, portable hard drive, or similar storage device, on mobile devices such as smartphones and tablets in accordance with standards such as DLNA (Digital Living Network Alliance). For example, in the case of DLNA, the storage medium containing the DMP data acts as a DLNA server, while smartphones and tablets act as DLNA clients. When viewing content on a DLNA client, it is desirable to be able to play the content while it is being downloaded, rather than downloading all of the DMP data before playback. However, since DMP data is stored in an uncompressed ZIP format containing multiple MP4 files, and the data in these MP4 files is not interleaved, it is difficult to synchronize the playback of audio, video, or subtitle data while downloading the DMP data. Therefore, the DLNA server separates the audio, video, and subtitle MP4 files from the DMP, establishes independent sessions for each of these MP4 files, and then transmits them to the DLNA client. The DLNA client separates the elementary streams from the multiple media MP4 files received via these independent sessions and plays them back in sync.

FIG. 39

TRANSLATOR CERTIFICATION

Date: May 21, 2026

To whom it may concern:

I, Frank McGee, a translator fluent in the Japanese and English languages, on behalf of Morningside Translations, do solemnly and sincerely declare that the following is, to the best of my knowledge and belief, a true and correct translation of the documents listed below in a form that reflects the intention and meaning of the original text.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

The document is designated as:

- Toma-446 provisional 2 - 61944280_scope pages



Frank McGee