



УКРАЇНА

(19) **UA** (11) **108893** (13) **C2**  
(51) МПК**H04N 21/2343** (2011.01)**H04N 21/236** (2011.01)**H04N 21/845** (2011.01)**H04L 29/08** (2006.01)ДЕРЖАВНА СЛУЖБА  
ІНТЕЛЕКТУАЛЬНОЇ  
ВЛАСНОСТІ  
УКРАЇНИ**(12) ОПИС ДО ПАТЕНТУ НА ВІНАХІД****(21)** Номер заявки: **а 2013 02104****(22)** Дата подання заявки: **20.07.2011****(24)** Дата, з якої є чинними  
права на винахід: **25.06.2015****(31)** Номер попередньої  
заявки відповідно до  
Паризької конвенції: **12/840,146****(32)** Дата подання  
попередньої заявки  
відповідно до  
Паризької конвенції: **20.07.2010****(33)** Код держави-учасниці  
Паризької конвенції,  
до якої подано  
попередню заявку: **US****(41)** Публікація відомостей  
про заявку: **27.05.2013, Бюл.№ 10****(46)** Публікація відомостей  
про видачу патенту: **25.06.2015, Бюл.№ 12****(86)** Номер та дата  
подання міжнародної  
заявки, поданої  
відповідно до  
Договору РСТ **PCT/US2011/044745,  
20.07.2011****(72)** Винахідник(и):**Чень Ін (US),  
Карчевіч Марта (US)****(73)** Власник(и):**КВЕЛКОММ ІНКОРПОРЕЙТЕД,  
5775 Morehouse Drive, San Diego, California,  
92121-1714, United States of America (US)****(74)** Представник:**Мошинська Ніна Миколаївна, реєстр.  
№115****(56)** Перелік документів, взятих до уваги  
експертизою:

Steven McCanne "Low-complexity Video Coding for Receiver-driven Layered Multicast", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, 19970801 IEEE SERVICE CENTER, PISCATAWAY, US - ISSN 0733-8716; 01.08.1997

"Requirements on HTTP Streaming of MPEG Media"; 92. MPEG meeting; 19-4-2010 - 23-4-2010; Dresden; (Motion Picture Expert Group or ISO/IEC JTC1/SC29/WG11), 20100514 - ISSN 0000-0029; 14.05.2010

"Text of ISO/IEC 14496-12 3rd Edition"; 83. MPEG meeting; 14-1-2008 - 18-1-2008; Antalya; (Motion Picture Expert Group or ISO/IEC JTC1/SC29/WG11), 20080422 - ISSN 0000-0043; 22.04.2008

"Text of ISO/IEC 14496-12:2008/PDAM 2 Sub-track selection & switching"; 91. MPEG meeting; 18-1-2010 - 22-1-2010; Kyoto; (Motion Picture Expert Group or ISO/IEC JTC1/SC29/WG11), 20100122 - ISSN 0000-0030; 22.01.2010

Amon P; Rathgen T; Singer D: "File Format for Scalable Video Coding", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 20070901 IEEE SERVICE CENTER, PISCATAWAY, NJ, US - ISSN 1051-8215; 01.09.2007

**(54) РОЗМІЩЕННЯ ФРАГМЕНТІВ СУБТРЕКУ ДЛЯ ПОТОКОВОЇ ПЕРЕДАЧІ ВІДЕОДАНИХ****(57)** Реферат:

UA 108893 C2

Відеофайл може включати в себе фрагменти фільму, розділені на фрагменти субтреку, які зберігають всі зображення загальних ієрархічних рівнів для відповідних ієрархічних рівнів. В одному прикладі пристрій включає в себе інтерфейс, виконаний з можливістю виведення даних відповідно до протоколу потокової передачі; і блок керування, виконаний з можливістю збирання кодованих відеоданих в множині фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно зв'язаних відеозображень кодованих відеоданих, причому кожне з множини ієрархічно зв'язаних відеозображень відповідає загальному ієрархічному рівню; прийому запиту відповідно до протоколу потокової передачі, причому запит задає щонайменше один з множини фрагментів субтреку; і, у відповідь на запит, припис згаданому інтерфейсу виводити множину ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку.

Дані розкриття належать до зберігання та транспортування кодованих відеоданих.

#### РІВЕНЬ ТЕХНІКИ

Можливості цифрового відео можуть бути вбудовані в різноманітні пристрої, включаючи цифрові телевізори, системи безпосереднього цифрового мовлення, системи бездротового мовлення, персональні цифрові помічники (PDA), портативні або настільні комп'ютери, цифрові камери, цифрові записуючі пристрої, цифрові медіа програвачі, відеоігрові пристрої, відеоігрові приставки, стільникові або супутникові радіотелефони, пристрої для відеотелеконференцій тощо. Цифрові відеоприсрої здійснюють методи стиснення відео, такі як ті, які описані в стандартах, визначених в MPEG-2, MPEG-4, ITU-T H.263 або ITU-T H.264/MPEG-4, Part 10, вдосконалене відеокодування (AVC) і розширеннях таких стандартів, для більш ефективної передачі та прийому цифрової відеоінформації.

Методи стиснення відео виконують просторове прогнозування і/або часове прогнозування для зменшення або видалення надмірності, властивої відеопослідовностям. Для відеокодування на основі блоків відеокадр або слайс може розділятися на макроблоки. Кожний макроблок може додатково розділятися. Макроблоки у внутрішньо кодованому (I) кадрі або слайсі кодуються з використанням просторового прогнозування у відношенні сусідніх макроблоків. Макроблоки у зовні кодованому (P або B) кадрі або слайсі можуть використовувати просторове прогнозування у відношенні сусідніх макроблоків у цьому самому кадрі або слайсі або часове прогнозування відносно інших опорних кадрів.

Після того як відеодані будуть кодовані, відеодані можуть пакетуватися для передачі або зберігання. Відеодані можуть збиратися у відеофайл, що відповідає будь-якому з численних стандартів, такому як базовий формат медіафайлів Міжнародної організації зі стандартизації (ISO) і його розширень, такого як AVC.

Були зроблені спроби розробити нові стандарти відеокодування, основані на H.264/AVC. Одним таким стандартом є стандарт масштабованого відеокодування (SVC, який являє собою масштабоване розширення H.264/AVC. Іншим стандартом є багатовидове відеокодування (MVC), яке стає багатовидовим розширенням H.264/AVC. Спільний проєкт MVC описаний в JVT-AB204, "Joint Draft 8.0 on Multiview Video Coding, » 28<sup>th</sup> JVT meeting, Hannover, Germany, July 2008, доступному за адресою [http://wftp3.itu.int/av-arch/jvt-site/2008\\_07\\_Hannover/JVT-AB204.zip](http://wftp3.itu.int/av-arch/jvt-site/2008_07_Hannover/JVT-AB204.zip). Версія, вбудована в стандарт AVC, описана в JVT-AD007, "Editors' draft revision to ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding-in preparation for ITU-T SG 16 AAP Consent (in integrated form)», 30th JVT meeting, Geneva, CH, Feb. 2009, доступному за адресою [http://wftp3.itu.int/av-arch/jvt-site/2009\\_01\\_Geneva/JVT-AD007.zip](http://wftp3.itu.int/av-arch/jvt-site/2009_01_Geneva/JVT-AD007.zip).

#### СУТЬ ВИНАХОДУ

Загалом, дане розкриття описує методи для створення фрагментів субтреку відеофайлів для підтримки потокової передачі відеоданих. Замість організації кодованих відеозображень в фрагменті відео відеофайлу в порядку декодування, методи даного розкриття включають в себе розміщення кодованих відеозображень в порядку, що відповідає ієрархічному рівню або шару, до якого належать кодовані зображення. Кожний ієрархічний шар всередині фрагмента відео може відповідати відповідному фрагменту субтреку. Тобто кожний фрагмент субтреку може включати в себе всі кодовані відеозображення відповідного ієрархічного шару для конкретного фрагмента фільму в безперервному байтовому діапазоні фрагмента фільму. Відеозображення в фрагменті субтреку можуть як і раніше дотримуватися порядку декодування. Таким чином, пристрій призначення може представити єдиний запит на витягання всіх зображень фрагмента субтреку фрагмента фільму. У контексті відеофайлу і транспортування інкапсульовані кодовані відеозображення також можуть згадуватися як семпли відео.

В одному прикладі спосіб включає в себе збирання кодованих відеоданих у множині фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно зв'язаних відеозображень кодованих відеоданих, причому кожне з множини ієрархічно зв'язаних відеозображень відповідає загальному ієрархічному шару, прийом запиту відповідно до протоколу потокової передачі, причому запит задає щонайменше один з множини фрагментів субтреку, і, у відповідь на запит, виведення множини ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку.

В іншому прикладі пристрій включає в себе інтерфейс, виконаний з можливістю виведення даних відповідно до протоколу потокової передачі, і блок керування, виконаний з можливістю збирання кодованих відеоданих у множину фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно зв'язаних відеозображень кодованих відеоданих, причому кожне з множини ієрархічно зв'язаних відеозображень відповідає загальному ієрархічному шару, прийому запиту відповідно до протоколу потокової передачі, причому запит задає щонайменше один з множини фрагментів субтреку, і, у відповідь на запит, припис інтерфейсу

виводити множину ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку.

В іншому прикладі пристрій включає в себе засіб для збирання кодованих відеоданих у множині фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно зв'язаних відеозображень кодованих відеоданих, причому кожне з множини ієрархічно зв'язаних відеозображень відповідає загальному ієрархічному рівню, засіб для прийому запиту відповідно до протоколу потокової передачі, причому запит задає щонайменше один з множини фрагментів субтреку, і засіб для виведення множини ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку у відповідь на запит.

В іншому прикладі комп'ютерний програмний продукт включає в себе зчитуваним комп'ютером запам'ятовуючий носій, який містить інструкції, які, при виконанні, приписують процесору пристрою джерела збирати кодовані відеодані в множині фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно зв'язаних відеозображень кодованих відеоданих, причому кожне з множини ієрархічно зв'язаних відеозображень відповідає загальному ієрархічному шару, приймати запит відповідно до протоколу потокової передачі, причому запит задає щонайменше один з множини фрагментів субтреку, і, у відповідь на запит, виводити множину ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку.

В іншому прикладі спосіб включає в себе прийом інформації від пристрою джерела, яка описує ієрархічні рівні відеоданих для фрагмента фільму, і визначення піднабору ієрархічних рівнів відеоданих для запиту. Для кожного з ієрархічних рівнів піднабору спосіб включає в себе посилання не більше одного запиту на пристрій джерела на витягання всіх відеоданих фрагмента фільму на ієрархічному рівні. Спосіб додатково включає в себе прийом відеоданих визначеного піднабору ієрархічних рівнів і декодування і відображення прийнятих відеоданих.

В іншому прикладі пристрій включає в себе інтерфейс, виконаний з можливістю прийому інформації від пристрою джерела, яка описує ієрархічні рівні відеоданих для фрагмента фільму; і блок керування, виконаний з можливістю визначення піднабору ієрархічних рівнів відеоданих для запиту, причому для кожного з ієрархічних рівнів піднабору блок керування виконаний з можливістю посилання не більше одного запиту на пристрій джерела на витягання всіх відеоданих фрагмента фільму на ієрархічному рівні. Інтерфейс додатково виконаний з можливістю прийому відеоданих визначеного піднабору ієрархічних рівнів у відповідь на запити.

В іншому прикладі пристрій включає в себе засіб для прийому інформації від пристрою джерела, яка описує ієрархічні рівні відеоданих для фрагмента фільму, засіб для визначення піднабору ієрархічних рівнів відеоданих для запиту, засіб для посилання, для кожного з ієрархічних рівнів піднабору, не більше одного запиту на пристрій джерела для витягання всіх відеоданих фрагмента фільму на ієрархічному рівні, засіб для прийому відеоданих визначеного піднабору ієрархічних рівнів і засіб для декодування і відображення прийнятих відеоданих.

В іншому прикладі комп'ютерний програмний продукт включає в себе зчитуваний комп'ютером запам'ятовуючий носій, що містить інструкції, які приписують процесору пристрою призначення приймати інформацію від пристрою джерела, яка описує ієрархічні рівні відеоданих для фрагмента фільму, визначати піднабір ієрархічних рівнів відеоданих для запиту для кожного з ієрархічних рівнів піднабору, посилати не більше одного запиту на пристрій джерела на витягання всіх відеоданих фрагмента фільму на ієрархічному рівні, приймати відеодані визначеного піднабору ієрархічних рівнів, і декодувати і відображати прийняті відеодані.

Подробиці одного або більше прикладів викладаються на прикладених кресленнях і в описі нижче. Інші ознаки, задачі і переваги стануть зрозумілі з опису і креслень, і з формули винаходу.

#### КОРОТКИЙ ОПИС КРЕСЛЕНЬ

Фіг.1 являє собою блок-схему, що ілюструє зразкову систему, в якій пристрій джерела аудіо/відео (A/V) посилає аудіо- і відеодані на пристрій призначення A/V.

Фіг.2 являє собою блок-схему, що ілюструє компоненти зразкового блока інкапсуляції.

Фіг.3 являє собою блок-схему, що ілюструє елементи зразкового відеофайлу, який має фрагменти відео, причому кожний включає в себе фрагменти субтреку, що мають кодовані відеозображення загального ієрархічного рівня.

Фіг.4А являє собою блок-схему, що ілюструє зразковий фрагмент фільму.

Фіг.4В являє собою блок-схему, що ілюструє зразковий фрагмент фільму, який включає в себе об'єкти повторного збирача.

Фіг.5 являє собою блок-схему, що ілюструє зразковий фрагмент відео SVC, що включає в себе відеозображення, організовані відповідно до ієрархічних шарів.

Фіг.6 являє собою блок-схему, що ілюструє зразковий фрагмент відео MVC, що включає в себе відеозображення, організовані відповідно до ієрархічних шарів.

Фіг.7 являє собою блок-схему послідовності операцій, що ілюструє зразковий спосіб інкапсуляції відеоданих загальних ієрархічних рівнів у відповідних фрагментах субтреку фрагмента фільму у відеофайлі і передачі відеофайлу від пристрою джерела на пристрій призначення.

Фіг.8 являє собою блок-схему послідовності операцій, що ілюструє зразковий спосіб витягання фрагментів субтреку фрагмента фільму з використанням протоколу потокової передачі.

Фіг.9 являє собою схему концептуального представлення, що ілюструє зразковий шаблон прогнозування MVC.

#### ДОКЛАДНИЙ ОПИС

Загалом, дане розкриття описує методи для розміщення фрагментів субтреку (субдоріжки) відеофайлів для підтримки потокової передачі відеоданих. Зокрема, кодовані відеозображення фрагмента треку (доріжки) можуть розміщуватися відповідно до ієрархічного шару, до якого належать кодовані відеозображення. У даному розкритті, кодовані відеозображення також можуть згадуватися як кодовані семпли (вибірки) відео, або просто як "семпли" або "зображення". Таким чином, кодовані відеозображення загального шару можуть розміщуватися поряд у відеофайлі. Отже, пристрій призначення може витягувати кодовані зображення конкретного ієрархічного шару в фрагменті фільму, використовуючи єдиний запит. Для прикладу потокової передачі протоколу передачі гіпертексту (HTTP), єдиний запит може містити запит часткового GET HTTP, що задає байтовий діапазон кодованих відеозображень до необхідного ієрархічного шару.

Фрагмент треку може являти собою фрагмент представлення відео базового формату медіафайлів ISO або фрагмент системного потоку MPEG-2, який може бути будь-якого виду з наступних: пакетований елементарний потік (PES), програмний потік (PS) або транспортний потік (TS). У транспортному потоку (TS) MPEG-2 пакети, що відповідають блокам доступу, звичайно упорядковуються у порядку декодування. Блок доступу може сегментуватися на численні транспортні пакети в потоку TS. У випадку, якщо фрагмент треку визначається як безперервна частина системного потоку MPEG-2, фрагмент треку може бути представлений як файловий блок, наприклад, файл або сегмент файлу. Методи даного розкриття можуть включати в себе перевпорядкування блока доступу в фрагменті в декілька фрагментів субтреку, кожний з яких може відповідати відповідному ієрархічному шару блоків доступу (кодованих зображень), так що і кодовані зображення загального ієрархічного шару представляються безперервно в частині потоку. Фрагменти субтреку в фрагменті треку можуть розміщуватися відповідно до порядку декодування. Таким чином, кодовані відеозображення загального шару можуть розміщуватися поряд у відеофайлі. Отже, пристрій призначення може витягувати всі кодовані зображення до конкретного ієрархічного шару в фрагменті фільму, використовуючи єдиний запит, наприклад, запит часткового GET HTTP, що задає байтовий діапазон кодованих відеозображень до необхідного ієрархічного шару.

Як приклад, формат файлів вдосконаленого відеокодування (AVC) визначає, що кодовані відеозображення розміщуються в порядку декодування в будь-якому фрагменті треку або фрагменті фільму. Група зображень (GOP) може мати деяку кількість зображень, кодованих з використанням різних схем прогнозування, наприклад, внутрішнє прогнозування (I-зображення) і зовнішнє прогнозування (P-зображення і B-зображення). I-зображення можуть кодуватися без посилання на інші зображення, P-зображення можуть кодуватися відносно одного або більше опорних зображень в одному напрямку, і B-зображення можуть кодуватися відносно одного або більше зображень в обох напрямках (вперед і назад у відеопослідовності).

Зовні кодоване зображення може мати ієрархічний рівень, рівний або більше ієрархічного рівня опорного зображення для зовні кодованого зображення. Зразкова послідовність зображень в порядку відображення може бути  $I_0B_3B_2B_3B_1B_3B_2B_3P_0$ , де літера означає тип кодування для кожного зображення, і номер, в цьому випадку, означає ієрархічний рівень, якому відповідає зображення. Припустимо з метою ілюстрації, що кожне зображення асоціюється з числовим показником, що відповідає позиції зображення в порядку відображення. Як вказано вище, зразкова послідовність розставляється в порядку відображення. Щоб декодувати зображення, кодоване із зовнішнім прогнозуванням, спочатку може декодуватися опорне зображення для кодованого зображення. Таблиця 1 нижче наводить зразковий порядок декодування для даної зразкової послідовності, де число нижнього індексу посилається на порядок відображення зображення:

Таблиця 1

Зображення в порядку відображення	I <sub>0</sub>	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	P <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	P <sub>8</sub>
Часовий ієрархічний рівень	0	2	1	2	0	2	1	2	0
Порядок декодування	0	3	2	4	1	7	6	8	5

Отже, звичайний пристрій джерела може розміщувати дану зразкову послідовність кодованих зображень відповідно до їх порядку декодування. Звичайно, зображення всередині GOP (в прикладі таблиці 1 розмір GOP дорівнює 4) з однаковим часовим ієрархічним рівнем можуть відділятися від зображень в інших GOP цього самого ієрархічного рівня. Наприклад, B<sub>2</sub> та B<sub>6</sub> обидва являють собою зображення часового ієрархічного рівня 1 в прикладі таблиці 1, але розділяються зображеннями з іншими часовими рівнями, якщо будуть розміщені в порядку декодування. Навіть зображення з одним і тим самим часовим рівнем в одній GOP можуть розділятися зображеннями з різними часовими рівнями. Припустимо відносно фрагмента, який містить, наприклад, 10 GOP, що зображення з ідентичним часовим рівнем можуть розподілятися в цьому фрагменті як численні окремі екземпляри.

Методи даного розкриття, з іншого боку, забезпечують упорядкування відносно ієрархічного рівня послідовності кодованих зображень. Як приклад, пристрій джерела згідно з методами даного розкриття може розміщувати зразкову послідовність вище так, як показано в таблиці 2:

Таблиця 2

Зображення в порядку відображення	I <sub>0</sub>	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	P <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	P <sub>8</sub>
Часовий ієрархічний рівень	0	2	1	2	0	2	1	2	0
Порядок декодування	0	4	3	5	2	7	6	8	1
Порядок в файлі	0	5	3	6	1	7	4	8	2

Таким чином, кодовані відеозображення в послідовності можуть розміщуватися згідно з ієрархічним рівнем в фрагменті відеофайлу. Тобто зображення загального ієрархічного рівня у фрагменті можуть групуватися разом поряд у фрагменті. Кожний фрагмент субтреку (що відповідає конкретному ієрархічному рівню) може доставлятися на пристрій у відповідь на єдиний запит. Таким чином, пристрій призначення може видавати єдиний запит на витягання зображень до конкретного ієрархічного рівня. Байтовий діапазон кожного фрагмента субтреку може передаватися на пристрій призначення перед тим, як будуть запитані будь-які відеозображення, так що пристрій призначення може формувати запит для одного або більше ієрархічних рівнів.

Наприклад, пристрій призначення може бути виконаний з можливістю витягання зображень до ієрархічного рівня один, який може відповідати двом фрагментам субтреку: 0 та 1. Пристрій призначення може видавати запит, оснований на байтових діапазонах фрагментів субтреку 0 та 1. У відповідь на цей зразковий запит пристрій джерела може надати зображення у фрагменті 0 та 1 субтреку, що має порядок відображення 0, 8, 4, 2, 6 тощо.

За допомогою упорядкування зображень відповідно до ієрархічного рівня пристрій джерела може спростувати процес, за допомогою якого пристрій призначення може витягувати кодовані відеозображення загального ієрархічного рівня. Пристрою призначення немає необхідності, наприклад, визначати розташування кожного із зображень, що відповідають необхідному ієрархічному рівню, і індивідуально видавати численні запити на такі зображення, але, замість цього, може представити єдиний запит на витягання тільки зображень до необхідного ієрархічного рівня.

Під час прийому фрагментів субтреку, основуючись на сигналізації в фрагментах субтреку, пристрій призначення може перевпорядкувати прийняті відеозображення до ієрархічного рівня, утворюючи правильний порядок декодування, перед посиланням відеозображень на відеодекодер. Крім того, інформація, що описує ієрархію кожного фрагмента субтреку, може сигналізуватися, наприклад, часова масштабованість, частота кадрів і швидкість відтворення при використанні як прискореного перемотування вперед.

Пристрій призначення може бути виконаний з можливістю витягання зображень тільки до конкретного ієрархічного рівня з численних причин. Наприклад, пристрій призначення може підтримувати максимальну частоту кадрів, яка менше максимальної доступної частоти кадрів відеофайлу. Як інший приклад, пристрій призначення може підтримувати "режим спецефектів", такий як відтворення з прискореним перемотуванням вперед зі швидкістю в два рази більше, чотири рази більше, вісім разів більше або з іншою швидкістю, кратною нормальній швидкості відтворення. Таким чином, методи даного розкриття можуть підтримувати часову масштабованість.

Методи даного розкриття можуть застосовуватися до відеофайлів, що відповідають будь-якому з базового формату медіафайлів ISO, формату файлів вдосконаленого відеокодування (AVC), формату файлів Проекту партнерства зі створення системи третього покоління (3GPP), формату файлів масштабованого відеокодування (SVC) і/або формату файлів багатовидового відеокодування (MVC), або інших подібних форматів відеофайлів. Крім того, ієрархічний рівень може являти собою будь-який ієрархічний рівень відповідно до цих або інших форматів відеофайлів. Наприклад, що стосується SVC, ієрархічні рівні можуть відповідати різним шарам кодування, наприклад, базовому шару і одному або більше поліпшуючим шарам. Як інший приклад, що стосується MVC, ієрархічні рівні можуть відповідати різним видам.

Базовий формат медіафайлів ISO розроблений для того, щоб містити інформацію про часові медіа для представлення в гнучкому розширюваному форматі, який сприяє обміну, керуванню, редагуванню і представленню медіа. Базовий формат медіафайлів ISO (ISO/IEC 14496-12:2004) визначений в MPEG-4 Part-12, який визначає загальну структуру для основаних на часі медіафайлів. Він використовується як основа для інших форматів файлів у сімействі, таких як формат файлів AVC (ISO/IEC 14496-15), який визначає підтримку стиснення відео H.264/MPEG-4 AVC, формат файлів 3GPP, формат файлів SVC і формат файлів MVC. Формат файлів 3GPP і формат файлів MVC являють собою розширення формату файлів AVC. Базовий формат медіафайлів ISO містить часову інформацію, інформацію про структуру і медіа для часових послідовностей медіаданих, таких як аудіовізуальні представлення. Структура файлів може бути об'єктно-орієнтованою. Файл може бути розбитий на базові об'єкти дуже просто, і структура об'єктів передбачається з їх типу.

У прикладах MPEG-1 та MPEG-2 В-кодовані зображення забезпечують природну часову масштабованість. Трек відеофайлу, що відповідає MPEG-1 або MPEG-2, може включати в себе повний набір І-кодованих зображень, Р-кодованих зображень і В-кодованих зображень. Відкидаючи В-кодовані зображення, відеофайл може одержувати відповідні представлення відео з половинним розрізненням. MPEG-1 та MPEG-2 також забезпечують поняття базового шару і поліпшуючого шару для кодування двох часових шарів, причому зображення поліпшуючого шару можуть вибирати для кожного напрямку прогнозування зображення або з базового шару, або з поліпшуючого шару як опорне. Отже, пристрій призначення може запитувати, і пристрій джерела може забезпечувати, меншу кількість кодованих зображень, ніж повний набір І-, Р- та В-кодованих зображень, включених у відеофайл. Відеодані, що представляються пристроєм джерела на пристрій призначення, проте, можуть відповідати MPEG-1 та MPEG-2 і можуть мати половину (або менше) розрізнення крім вихідного, повного набору кодованих зображень.

Як інший приклад, H.264/AVC використовує ієрархічні В-кодовані зображення для підтримки часової масштабованості. Перше зображення відеопослідовності в H.264/AVC може згадуватися як зображення миттєвого оновлення декодера (IDR), також відоме як ключове зображення. Ключові зображення звичайно кодуються в регулярних або нерегулярних інтервалах, які є або внутрішньо кодованими, або зовні кодованими, з використанням попереднього ключового зображення як опорного для прогнозування з компенсацією руху. Група зображень (GOP) звичайно включає в себе ключове зображення і всі зображення, які часово розташовуються між ключовим зображенням і попереднім ключовим зображенням. GOP може бути розділена на дві частини, одна являє собою ключове зображення, і інша включає в себе неключові зображення. Неключові зображення ієрархічно прогнозуються за допомогою 2 опорних зображень, якими є найближчі зображення більш низького часового рівня з минулого і майбутнього часу.

Значення часового ідентифікатора може призначатися кожному зображенню для вказівки ієрархічного положення зображення, тобто ієрархічного шару, якому відповідає зображення. Таким чином, зображення зі значеннями часового ідентифікатора до N можуть утворювати сегмент відео з частотою кадрів, яка в два рази більше частоти сегмента відео, утвореного зображеннями зі значеннями часового ідентифікатора до N-1. Отже, методи даного розкриття також можуть використовуватися для досягнення часової масштабованості в H.264/AVC за допомогою розміщення кодованих відеозображень у фрагменти субтреку, так що пристрій призначення може запитувати один або більше фрагментів субтреку, але може запитувати меншу кількість, ніж повний набір фрагментів субтреку фрагмента фільму. Тобто пристрій призначення може запитувати фрагменти субтреку, що мають часові ідентифікатори, менші або рівні N.

Файли, що відповідають базовому формату медіафайлів ISO (і їх розширенням), можуть бути утворені у вигляді послідовностей об'єктів, які називаються "боксами". Дані в базовому форматі медіафайлів ISO містяться в боксах, так що немає необхідності іншим даним міститися

в файлі поза боксів. Він включає в себе будь-яку початкову сигнатуру, що вимагається конкретним форматом файлів. "Бокс" може являти собою об'єктно-орієнтований стандартний блок, визначений унікальним ідентифікатором типу і довжиною. Звичайно, представлення міститься в одному файлі, і представлення медіа є автономним. Контейнер фільму (бокс фільму) може містити метадані медіа, і кадри відео та аудіо можуть міститися в контейнері медіаданих і можуть бути в інших файлах.

Представлення (послідовність руху) може міститися в декількох файлах. Часова інформація та інформація про формування кадрів (положення і розмір), в основному, знаходяться в базовому медіафайлі ISO, і допоміжні файли можуть, по суті, використовувати будь-який формат. Це представлення може бути "локальним" для системи, що містить представлення, або може надаватися за допомогою мережі або іншого механізму доставки потоку.

Файли можуть мати логічну структуру, часову структуру і фізичну структуру, і не потрібно, щоб ці структури були зв'язані. Логічна структура файлу може бути фільму, який, в свою чергу, містить набір паралельних у часі треків. Часова структура файлу може бути такою, що треки містять послідовності зображень у часі, і ці послідовності відображаються на часову шкалу всього фільму необов'язковими монтажними аркушами. Фізична структура файлу може відділяти дані, необхідні для логічного, часового і структурного розбиття, від самих семплів медіаданих. Ця структурна інформація може концентруватися в боксі фільму, можливо розширеному у часі за допомогою боксів фрагментів фільму. Бокс фільму може документувати логічну і часову залежність семплів і також може містити покажчики на те, де вони розташовані. Ці покажчики можуть бути на цей самий файл або інший, наприклад, можуть посилатися за допомогою уніфікованого покажчика ресурсу (URL).

Кожний медіапотік може міститися в треку, спеціалізованому для цього типу медіа (аудіо, відео тощо) і може додатково параметризуватися елементом семпла. Елемент семпла може містити "ім'я" точного типу медіа (тип декодера, необхідний для декодування потоку) і будь-яку параметризацію цього необхідного декодера. Ім'я також може приймати вид чотирьохсимвольного коду, наприклад, "moov" або "trak". Визначаються формати елементів семпла не тільки для медіа MPEG-4, але також для типів медіа, які використовуються іншими організаціями, що використовують це сімейство форматів файлів.

Підтримка метаданих, в основному, приймає два види. По-перше, часові метадані можуть зберігатися у відповідному треку, синхронізуватися, коли потрібно, з медіаданими, які вони описують. По-друге, може бути загальна підтримка нечасових метаданих, що приєднуються до фільму або до індивідуального треку. Структурна підтримка є загальною і дозволяє, як в медіаданих, виконувати зберігання ресурсів метаданих десь в іншому місці в файлі або в іншому файлі. Крім того, ці ресурси можуть іменуватися і можуть захищатися.

У базовому форматі медіафайлів ISO групування семплів являє собою призначення кожного із семплів у треку членом однієї групи семплів. Не потрібно, щоб семпли в групі семплів були поряд. Наприклад, при представленні H.264/AVC в форматі файлів AVC семпли відео на одному часовому рівні можуть відбиратися в одну групу семплів. Групи семплів можуть представлятися двома структурами даних: боксом SampleToGroup (sbdp) і боксом SampleGroupDescription. Бокс SampleToGroup представляє призначення семплів групам семплів. Може бути один примірник боксу SampleGroupDescription для кожного елемента групи семплів для опису властивостей відповідної групи.

Необов'язковий трек метаданих може використовуватися для помітки кожного треку "характеристикою, що цікавить", яку він має, для якої його значення може відрізнитися від інших членів групи (наприклад, його швидкість передачі бітів, розмір екрана або мова). Деякі семпли в треку можуть мати спеціальні характеристики або можуть ідентифікуватися індивідуально. Одним прикладом характеристики є точка синхронізації (часто I-кадр відео). Ці точки можуть ідентифікуватися спеціальною таблицею в кожному треку. Звичайно, суть залежності між семплами треку також може документуватися з використанням метаданих. Метадані можуть структуруватися у вигляді послідовності семплів формату файлів, як трек відео. Такий трек може згадуватися як трек метаданих. Кожний семпл метаданих може структуруватися як оператор метаданих. Є різні види оператора, що відповідають різним питанням, які можуть бути задані про відповідний семпл формату файлу або його складові семпли.

Коли медіа доставляється за протоколом потокової передачі, може бути потрібне перетворення медіа з того виду, в якому воно представлене в файлі. Одним прикладом цього є те, коли медіа передається за протоколом реального часу (RTP). У файлі, наприклад, кожен кадр відео зберігається поряд у вигляді семпла формату файлу. У RTP правила пакетування, характерні для кодека, що використовується, повинні підкорятися розміщенню цих кадрів в пакетах RTP. Сервер потокової передачі може бути виконаний з можливістю обчислення



динамічно такого пакетування. Однак є підтримка сприянню серверів потокової передачі. Спеціальні треки, які називаються треками підказок, можуть розміщуватися у файлах.

Треки підказок містять загальні інструкції для серверів потокової передачі відносно того, як формувати потоки пакетів з треків медіа для конкретного протоколу. Оскільки форма цих інструкцій є незалежною від медіа, може не потрібним модифікування серверів, коли вводяться нові кодеки. Крім того, програмне забезпечення кодування і редагування може не мати відомостей про сервери потокової передачі. Якщо редагування закінчене над файлом, частина програмного забезпечення, яка називається підказувачем, може використовуватися для додавання треків підказок до файлу перед розміщенням його на сервері потокової передачі. Як приклад, є визначений формат треку підказок для потоків RTP в специфікації формату файлів MP4.

3GP (формат файлів 3GPP) являє собою формат мультимедійного контейнера, визначений Проектом партнерства зі створення системи 3-го покоління (3GPP) для мультимедійних послуг універсальної системи мобільного зв'язку (UMTS) третього покоління (3G). Він звичайно використовується на мобільних телефонах 3G та інших пристроях з можливостями 3G, але також може відтворюватися на деяких телефонах і пристроях 2G та 4G. Формат файлів 3GPP ґрунтується на базовому форматі медіафайлів ISO. Найостанніший 3GP визначається в 3GPP TS26.244 "Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)". Формат файлів 3GPP зберігає відеопотоки у вигляді MPEG-4 Part 2, або H.263, або MPEG-4 Part 10 (AVC/H.264). 3GPP дозволяє використовувати кодеки AMR та H.263 в базовому форматі медіафайлів ISO (MPEG-4 Part 12), оскільки 3GPP визначає використання полів Елемент семпла і полів шаблону в базовому форматі медіафайлів ISO, а також визначає нові бокси, на які посилаються кодеки. Для зберігання характерної для медіа MPEG-4 інформації в файлах 3GP, специфікація 3GP посилається на формат файлів MP4 та AVC, які також ґрунтуються на базовому форматі медіафайлів ISO. Специфікації форматів файлів MP4 та AVC описують використання контенту MPEG-4 в базовому форматі медіафайлів ISO.

Специфікація базового формату медіафайлів ISO визначає альтернативну групу треків. Альтернативна група включає в себе піднабір всіх доступних треків, і кожний трек може відповідати одній альтернативній групі. Загалом, пристрій призначення може вибирати один трек з кожної альтернативної групи за винятком інших треків в альтернативних групах. Специфікація формату файлів 3GPP визначає групу перемикання треків, яка подібна до альтернативної групи. Під час потокової передачі завантаження і відтворення пристрій призначення може перемикає між різними треками групи перемикання. Тобто треки в одній і тій самій групі перемикання доступні для перемикання під час сеансу, тоді як треки в різних групах перемикання звичайно недоступні для перемикання.

Формат файлів SVC, як розширення формату файлів AVC, забезпечує структури екстрактора і шару. Екстрактори являють собою покажчики, які надають інформацію про положення і розмір даних відеокодування в семплі з рівним часом декодування в іншому треку. Це дозволяє складати ієрархію треку безпосередньо в ділянці кодування. Трек екстрактора в SVC зв'язаний з одним або більше базовими треками, з яких він витягує дані під час виконання. Екстрактор являє собою покажчик з можливістю скасування посилання із заголовком блока NAL з розширеннями SVC. Якщо трек, що використовується для витягання, містить дані відеокодування з іншою частотою кадрів, тоді екстрактор також містить зміщення часу декодування, щоб забезпечити синхронність між треками. Під час виконання екстрактор повинен бути замінений даними, на які він вказує, перед тим як потік буде переданий на відеодекодер.

Оскільки треки екстрактора в SVC структуруються подібно до треків відеокодування, вони можуть представляти піднабір, який їм потрібен, різним чином. Трек екстрактора SVC містить тільки інструкції про те, як витягнути дані з іншого треку. У форматі файлів SVC є також агрегатори, які можуть агрегувати блок NAL в семплі разом як один блок NAL, включаючи агрегування блоків NAL в одному шарі в агрегатор. Екстрактор в SVC розроблений для витягання деякого діапазону байтів із семпла або агрегатора, або просто одного цілого блока NAL, але не численних блоків NAL, особливо тих, які не є послідовними в семплі. У форматі файлів SVC можуть бути численні робочі точки відео. Шари розроблені для групування семплів в одному або більше треках для робочої точки.

Формат файлів MVC також підтримує трек екстрактора, який витягує блоки NAL з різних видів, утворюючи робочу точку, яка являє собою піднабір видів з деякою частотою кадрів. Конструкція треку екстрактора MVC аналогічна екстрактору в форматі файлів SVC. Однак не підтримується використання треків екстрактора MVC для формування альтернативної групи. Для підтримки вибору треків пропонується наступна пропозиція MPEG для MPEG: P. Frojdh, A.

Norkin, and C. Priddle, "File format sub-track selection and switching, » ISO/IEC JTC1/SC29/WG11 MPEG M16665, London U. Ця пропозиція намагається зробити можливим принцип альтернативної групи / групи перемикавання на рівні субтреку.

Група семплів відображення являє собою розширення для групи семплів. У групі семплів відображення кожний елемент групи (семплів) має свій опис "groupID" (ідентифікатор групи), який фактично являє собою відображення на view\_id (ідентифікатор виду), після можливого агрегування блоків NAL у виді в один блок NAL. Іншими словами, кожний елемент групи семплів має свої види, що містяться в ньому, перелічені в значенні ScalableNALUMapEntry (масштабований елемент відображення блока NAL). grouping\_type (тип групування) цього елемента групи семплів є "scnm".

Термін "прогресивне завантаження" використовується для опису перенесення цифрових медіафайлів з сервера на клієнта, звичайно з використанням протоколу HTTP. Під час ініціювання з комп'ютера комп'ютер може почати відтворення медіа, до того як буде завершено завантаження. Одна відмінність між потоковим медіа і прогресивним завантаженням полягає в тому, як цифрові медіадані приймаються і зберігаються пристроєм кінцевого користувача, який виконує доступ до цифрових медіа. Медіапрогравач, який здатний виконувати відтворення з прогресивним завантаженням, ґрунтується на метаданих, розташованих у заголовку файлу, які є непошкодженими, і локальному буфері цифрового медіафайлу, коли він завантажується з веб-сервера. У момент часу, при якому задана кількість даних стає доступною для локального пристрою відтворення, пристрій може почати відтворення медіа. Це задана величина буфера може бути вбудована в файл виробником контенту в установках декодера і може бути підкріплена додатковими установками буфера, що призначаються медіапрогравачем клієнтського комп'ютера.

AVC та 3GPP є розширеннями базового формату медіафайлів ISO, тоді як SVC та MVC є розширеннями формату файлів AVC. Отже, методи даного розкриття можуть бути застосовані у відношенні відеофайлів, що відповідають базовому формату медіафайлів ISO, формату файлів AVC і його розширенням, наприклад, SVC та MVC, і/або формату файлів 3GPP. Методи додатково можуть застосовуватися до цих та інших розширень цих форматів і додатково можуть застосовуватися для розширення інших форматів файлів для забезпечення фрагментів субтреку збиранням семплів відео в різних форматах файлів для потокової передачі HTTP.

У відношенні 3GPP як інший приклад, транспорт HTTP/TCP/IP підтримується для файлів 3GPP для завантаження і прогресивного завантаження. Крім того, використання HTTP для потокової передачі відео може забезпечувати деякі переваги, і стають популярними послуги потокової передачі відео, ґрунтовані на HTTP. Потокова передача HTTP може забезпечувати деякі переваги, включаючи те, що можуть використовуватися існуючі компоненти і протоколи Інтернету, так що не потрібні нові зусилля для розробки нових методів для транспортування відеоданих по мережі. Інші транспортні протоколи, наприклад, формат корисного навантаження RTP, вимагають, щоб проміжні мережні пристрої, наприклад, середні бокси, мали відомості про формат медіа і контекст сигналізації. Також, потокова передача HTTP може керуватися клієнтом, що може уникнути питань керування.

Наприклад, щоб використовувати ознаки для одержання оптимальних характеристик, сервер може відстежувати розмір і контент пакетів, прийом яких ще не підтверджений. Сервер також може аналізувати файлову структуру і відновлювати стан клієнтського буфера, щоб прийняти RD-оптимальні рішення про перемикавання/проріджування. Крім того, можуть виконуватися обмеження на зміни бітового потоку, щоб залишатися сумісним з узгодженими профілями. HTTP не обов'язково необхідні нові апаратні або програмні реалізації на веб-сервері, який реалізовував HTTP 1.1. Потокова передача HTTP також забезпечує дружність TCP і простежування брандмауером.

Під час потокової передачі HTTP операції, що часто використовуються, включають в себе GET і частковий GET. Операція GET витягує весь файл, асоційований з даним уніфікованим показником ресурсу (URL) або уніфікованим ім'ям ресурсу (URN). Операція частковий GET приймає байтовий діапазон як вхідний параметр і витягує безперервну кількість байтів файлу, що відповідає прийнятому байтовому діапазону. Таким чином, фрагменти фільму можуть забезпечуватися для потокової передачі HTTP, оскільки операція частковий GET може одержувати один або більше індивідуальних фрагментів фільму. Зазначте, що у фрагменті фільму може бути декілька фрагментів треку різних треків. Під час потокової передачі HTTP представленням медіа може бути структурована колекція даних, яка доступна для клієнта. Клієнт може запитати і завантажити інформацію про медіадані, щоб представити послугу потокової передачі користувачеві.

Фіг.1 являє собою блок-схему, що ілюструє зразкову систему 10, в якій пристрій 20 джерела аудіо/відео (A/V) посилає аудіо- і відеодані на пристрій 40 призначення A/V. Система 10 на фіг.1 може відповідати системі відеоконференції, системі сервер/клієнт, системі пристрій мовлення/приймач або будь-якій іншій системі, в якій відеодані посилаються з пристроєм джерела, такого як пристрій 20 джерела A/V, на пристрій призначення, такий як пристрій 40 призначення A/V. У деяких прикладах пристрій 20 джерела A/V і пристрій 40 призначення A/V можуть виконувати двонаправлений обмін інформацією. Тобто пристрій 20 джерела A/V і пристрій 40 призначення A/V можуть бути здатні як кодувати, так і декодувати (і передавати і приймати) аудіо- і відеодані. У деяких прикладах аудіокодер 26 може містити мовний кодер, що також згадується як вокодер.

Пристрій 20 джерела A/V в прикладі на фіг.1 містить джерело 22 аудіо і джерело 24 відео. Джерело 22 аудіо може містити, наприклад, мікрофон, який виробляє електричні сигнали, що представляють захоплені аудіодані, що підлягають кодуванню аудіокодером 26. Альтернативно, джерело 22 аудіо може містити запам'ятовуючий носій, що зберігає раніше записані аудіодані, генератор аудіоданих, такий як комп'ютеризований синтезатор, або будь-яке інше джерело аудіоданих. Джерело 24 відео може містити відеокамеру, яка виробляє відеодані, що підлягають кодуванню відеокодером 28, запам'ятовуючий носій, кодований раніше записаними відеоданими, блок генерування відеоданих або будь-яке інше джерело відеоданих.

Необроблені аудіо- і відеодані можуть містити аналогові або цифрові дані. Аналогові дані можуть відцифровуватися перед кодуванням аудіокодером 26 і/або відеокодером 28. Джерело 22 аудіо може одержувати аудіодані від учасника, що говорить, в той час як учасник, що говорить, говорить, і джерело 24 відео може одночасно одержувати відеодані учасника, що говорить. В інших прикладах джерело 22 аудіо може містити зчитуваний комп'ютером запам'ятовуючий носій, який містить аудіодані, що зберігаються, і джерело 24 відео може містити зчитуваний комп'ютером запам'ятовуючий носій, який містить відеодані, що зберігаються. Таким чином, методи, описані в даному розкритті, можуть застосовуватися до реальних, поточкових аудіо- і відеоданих або аудіо- і відеоданих в реальному масштабі часу або архівованих, заздалегідь записаних аудіо- і відеоданих.

Аудіокадри, які відповідають відеокадрам, являють собою, в основному, аудіокадри, що містять аудіодані, які були захоплені джерелом 22 аудіо одночасно з відеоданими, захопленими джерелом 24 відео, яке міститься у відеокадрах. Наприклад, в той час як учасник, що говорить, загалом, виробляє аудіодані за допомогою того, що він говорить, джерело 22 аудіо захоплює ці аудіодані, і джерело 24 відео одночасно захоплює відеодані учасника, що говорить, тобто в той час як джерело 22 аудіо захоплює аудіодані. Отже, аудіокадр може часово відповідати одному або більше конкретним відеокадрам. Отже, аудіокадр, що відповідає відеокадру, в основному, відповідає ситуації, в якій аудіодані та відеодані були захоплені одночасно, і для якої аудіокадр і відеокадр містять відповідно аудіодані і відеодані, які були захоплені одночасно.

У деяких прикладах аудіокодер 26 може кодувати часову позначку в кожному кодованому аудіокадрі, яка представляє момент часу, при якому були записані аудіодані для кодованого аудіокадру, і, аналогічно, відеокодер 28 може кодувати часову позначку в кожному кодованому відеокадрі, який представляє момент часу, при якому були записані відеодані для кодованого відеокадру. У таких прикладах аудіокадр, що відповідає відеокадру, може містити аудіокадр, що містить часову позначку, і відеокадр, що містить таку саму часову позначку. Пристрій 20 джерела A/V може включати в себе внутрішній тактовий сигнал, з якого аудіокодер 26 і/або відеокодер 28 можуть генерувати часові позначки, або яке джерело 22 аудіо і джерело 24 відео можуть використовувати, щоб асоціювати аудіо- і відеодані відповідно з часовою позначкою.

У деяких прикладах джерело 22 аудіо може посылати дані на аудіокодер 26, що відповідає моменту часу, при якому були записані аудіодані, і джерело 24 відео може посылати дані на відеокодер 28, що відповідає моменту часу, при якому були записані відеодані. У деяких прикладах аудіокодер 26 може кодувати ідентифікатор послідовності в кодованих аудіоданих для вказівки відносного часового упорядкування кодованих аудіоданих, але без необхідності вказівки абсолютного часу, при якому були записані аудіодані, і, аналогічно, відеокодер 28 також може використовувати ідентифікатори послідовності для вказівки відносного часового упорядкування кодованих відеоданих. Аналогічно, в деяких прикладах ідентифікатор послідовності може відображатися або іншим чином зіставлятися з часовою позначкою.

Методи даного розкриття, в основному, відносяться до зберігання і транспортування кодованих мультимедійних (наприклад, аудіо і відео) даних, і прийому та подальшої інтерпретації і декодування мультимедійних даних, що транспортуються. Як показано в прикладі на фіг.1, джерело 24 відео може подавати множину видів сцени на відеокодер 28.

Пристрій 20 джерела A/V може надавати "послугу" пристрою 40 призначення A/V. Послуга, в основному, відповідає піднабору доступних видів даних MVC. Наприклад, дані MVC можуть бути доступні для восьми видів, впорядкованих від нуля до семи. Одна послуга може відповідати стереовідео, що має два види, в той час як інша послуга може відповідати чотирьом видам, і ще інша послуга може відповідати всім восьми видам. Загалом, послуга відповідає будь-якій комбінації (тобто будь-якому піднабору) доступних видів. Послуга також може відповідати комбінації доступних видів, а також аудіоданих. Робоча точка може відповідати послугі, так що пристрій 20 джерела A/V може додатково забезпечувати дескриптор робочої точки для кожної послуги, що надається пристроєм 20 джерела A/V.

Кожний індивідуальний потік даних (або аудіо, або відео) згадується як елементарний потік. Елементарний потік являє собою єдиний, кодований цифровим чином (можливо стиснутий) компонент програми. Наприклад, частина кодованого відео або аудіо програми може являти собою елементарний потік. Елементарний потік може перетворюватися в пакетований елементарний потік (PES) перед інкапсуляцією у відеофайл. В одній і тій самій програмі ID потоку використовується для того, щоб відрізнити пакети PES, що належать одному елементарному потоку, від інших. Базовою одиницею даних елементарного потоку є пакет пакетованого елементарного потоку (PES). Таким чином, кожний вид відеоданих MVC відповідає відповідним елементарним потокам. Аналогічно, аудіодані відповідають одному або більше відповідним елементарним потокам.

Кодована відеопослідовність MVC може розділятися на декілька бітових підпотоків, кожний з яких є елементарним потоком. Кожний бітовий підпотік може ідентифікуватися з використанням піднабору view\_id MVC. Основуючись на принципі кожного піднабору view\_id MVC, визначається бітовий підпотік відео MVC. Бітовий підпотік відео MVC містить блоки NAL видів, перелічених в піднаборі view\_id MVC. Програмний потік, в основному, містить тільки блоки NAL, які являють собою ті, які з елементарних потоків. Також розроблено, що будь-які два елементарних потоки не можуть містити ідентичний вид, але, замість цього, можуть містити види, наприклад, різні ракурси сцени для створення трьохмірного ефекту.

У прикладі на фіг.1 блок 30 інкапсуляції приймає елементарні потоки, що містять відеодані від відеокодера 28, і елементарні потоки, що містять аудіодані від аудіокодера 26. У деяких прикладах кожний з відеокодера 28 і аудіокодера 26 може включати в себе формувачі пакетів для формування пакетів PES з кодованих даних. В інших прикладах кожний з відеокодера 28 і аудіокодера 26 може спрягатися з відповідними формувачами пакетів для формування пакетів PES з кодованих даних. У ще інших прикладах блок 30 інкапсуляції може включати в себе формувачі пакетів для формування пакетів PES з кодованих аудіо- і відеоданих.

"Програма", що використовується в даному розкритті, може містити комбінацію аудіоданих і відеоданих, наприклад, елементарний потік аудіо і піднабір доступних видів, що доставляються послугою пристрою 20 джерела A/V. Кожний пакет PES включає в себе stream\_id, який ідентифікує елементарний потік, до якого належить пакет PES. Блок 30 інкапсуляції відповідає за збирання елементарних потоків у відеофайл.

Блок 30 інкапсуляції приймає пакети PES для елементарних потоків програми від аудіокодера 26 і відеокодера 28 і формує відповідні блоки рівня мережної абстракції (NAL) з пакетів PES. У прикладі H.264/AVC (вдосконалене відеокодування) кодовані сегменти відео організуються в блоки NAL, які забезпечують відеопредставлення з "дружністю до мереж", що звертаються до додатків, таких як відеотелефонія, зберігання, широкомовлення або потокова передача. Блоки NAL можуть категоризуватися на блоки NAL рівня відеокодування (VCL) і блоки NAL не-VCL. Блоки VCL можуть містити механізм стиснення ядра і може включати в себе дані рівня блока, макроблока і/або слайса (вірізки). Іншими блоками NAL можуть бути блоки NAL не-VCL. У деяких прикладах кодоване зображення в одному моменті часу, звичайно представлене як головне кодоване зображення, може міститися в блоці доступу, який може включати в себе один або більше блоків NAL.

Блоки NAL не-VCL можуть включати в себе блоки NAL набору параметрів і блоки NAL SEI в числі інших. Набори параметрів можуть містити інформацію заголовка рівня послідовності (в наборах параметрів послідовності (SPS)) і інформацію заголовка рівня зображення, що рідко змінюється, (в наборах параметрів зображення (PPS)). З наборами параметрів (наприклад, PPS та SPS) інформацію, що рідко змінюється, немає необхідності повторювати для кожної послідовності або зображення, отже може бути підвищена ефективність кодування. Крім того, використання наборів параметрів може надати можливість виконувати позасмугову передачу важливої інформації заголовка, уникаючи необхідності надмірних передач для стійкості до помилок. У прикладах позасмугової передачі блоки NAL набору параметрів можуть передаватися по іншому каналу, ніж інші блоки NAL, такі як блоки NAL SEI.

Відповідно до методів даного розкриття блок 30 інкапсуляції може збирати семпли відео у фрагменти субтреку, кожний з яких може відповідати конкретному ієрархічному шару, наприклад, часовому шару. Блок 30 інкапсуляції також може представляти кожний фрагмент субтреку у відеофайлі у вигляді набору послідовних байтів. У деяких прикладах фрагмент субтреку може містити тільки нормальні, кодовані семпли. У деяких прикладах фрагмент субтреку може містити нормальні семпли, а також семпли повторного збирача, що вказують на семпли в одному або більше попередніх фрагментів субтреку в поточному фрагменті фільму. Крім того, в деяких прикладах фрагмент субтреку може містити тільки семпли повторного збирача.

Загалом, семпли фрагментів субтреку більш високого шару можуть кодуватися з посиланням на семпли фрагментів субтреку більш низького шару. Блок 30 інкапсуляції може забезпечувати, що семпли фрагментів субтреку більш низького шару не залежать від семплів більш високого шару, так що пристрій 40 призначення може витягувати семпли до необхідного шару без необхідності витягання більш високих шарів, ніж необхідний шар. Таким чином, пристрій 40 призначення може представляти один раз запит часткового GET HTTP для витягання одного або більше фрагментів субтреку. Наприклад, пристрій 40 призначення може представляти один запит для кожного необхідного шару. Коли шари розміщені поряд в файлі, пристрій 40 призначення може представляти запит на витягання даних для багатьох шарів.

У деяких прикладах блок 30 інкапсуляції може сигналізувати інформацію перевпорядкування у відеофайлі, яка вказує, як перевпорядкувати кодовані відеозображення більше одного фрагменту субтреку в порядок декодування. Наприклад, як описано вище, блок 30 інкапсуляції може включати в себе об'єкти повторного збирача в фрагменті субтреку. Загалом, об'єкт повторного збирача може діяти як покажчик на кодований семпл відео в попередньому, наприклад, в цьому самому фрагменті субтреку або фрагменті субтреку більш низького рівня. Пристрій 40 призначення може використовувати об'єкти повторного збирача для повторного розміщення семплів, після того як будуть прийняті фрагменти субтреку. Наприклад, після використання одного або більше запитів на витягання фрагментів субтреку відеофайлу блок 38 декапсуляції пристрою 40 призначення може використовувати об'єкти повторного збирача для збирання кодованого семплів відео в порядок декодування, перед тим як відеодекодер 48 декодує семпли. Блок 38 декапсуляції може використовувати найостанійший фрагмент субтреку в байтовому порядку як початкову точку для мультиплексування семплів за допомогою посилання на семпли в попередніх фрагментах субтреку. Об'єкт повторного збирача може включати в себе положення опорного семпла і індекс фрагмента субтреку, що включає в себе семпл, на який посилається об'єкт повторного збирача.

Крім того, якщо блок 30 інкапсуляції включає в себе об'єкти повторного збирача в фрагментах субтреку, блок 30 інкапсуляції може додатково включати в себе заголовки демультимплексування (які можуть, альтернативно, згадуватися як "заголовки повторного збирання"), які описують характеристики одного або більше фрагментів субтреку. Блок 30 інкапсуляції може включати в себе заголовки демультимплексування в різному розташуванні, такому як, наприклад, бокс фільму, заголовок фрагмента фільму і/або заголовок фрагмента треку. Заголовки демультимплексування можуть задавати унікальні ідентифікатори для кожного фрагмента субтреку, байтові діапазони для відповідних фрагментів субтреку, кількість зображень в кожному фрагменті субтреку і часову інформацію фрагментів субтреку. Часова інформація може описуватися як відносна часова інформація у вигляді семплів або всесвітній координований час (UTC). Блоку 30 інкапсуляції немає необхідності включати в себе таку часову інформацію, коли фрагменти субтреку не відповідають шарам з різними частотами кадрів або часовими рівнями.

У деяких прикладах, наприклад, у відношенні SVC та MVC, численні шари кодованого семплів відео можуть включатися в загальний трек. Наприклад, численні шари кодування (наприклад, в SVC) і численні види (наприклад, в MVC) можуть включатися в трек відеофайлу. Блок 30 інкапсуляції може розділяти зв'язані ієрархічні шари на відповідні фрагменти субтреку. Кожний фрагмент субтреку може відповідати загальному шару, наприклад, розмірність, часовий шар, шар відношення сигнал-шум, просторовий шар або вид. Як відмічено, дані для кожного фрагмента субтреку можуть бути включені у відеофайл як послідовні байти дані.

Блок 30 інкапсуляції може додатково визначати робочі точки як такі, що включають конкретні фрагменти субтреку. Зокрема, блок 30 інкапсуляції може визначати характеристики робочих точок, включаючи часовий рівень (temporal\_id (часовий ідентифікатор)), quality\_id (ідентифікатор якості), dependency\_id (ідентифікатор залежності) і/або view\_id. У прикладах, що відповідають SVC, характеристики можуть відповідати значенням в заголовку блока NAL блоків NAL SVC. У прикладах, що відповідають MVC, характеристики можуть відповідати значенням в заголовку

блока NAL блоків NAL MVC. У деяких прикладах тільки часовий рівень може бути присутнім як характеристика робочої точки. У контексті SVC можуть бути присутніми `temporal_id` (часовий рівень), `quality_id` та `dependency_id`. У контексті MVC можуть бути присутніми `temporal_id` та `view_id`.

У деяких прикладах характеристики робочих точок можуть додатково включати в себе відображення вищезазначених характеристик на індекс фрагмента субтреку. Крім того, характеристики робочої точки можуть включати в себе інформацію про кодек, індикатор профілю (`profile_idc`), індикатор рівня (`level_idc`), частоту кадрів для робочої точки, середню швидкість передачі бітів для робочої точки, максимальну швидкість передачі бітів для робочої точки, просторове розрізнення для робочої точки, кількість видів для виведення для робочої точки і/або кількість видів для декодування для робочої точки. Ці характеристики можуть об'єднуватися з існуючими робочими точками, як визначено відповідним форматом файлів.

Блок 30 інкапсуляції може формувати блоки NAL, що містять заголовок, який ідентифікує програму, до якої належить NAL, а також корисне навантаження, наприклад, аудіодані, відеодані або дані, які описують транспортний або програмний потік, якому відповідає блок NAL. Наприклад, в H.264/AVC блок NAL включає в себе 1-байтовий заголовок і корисне навантаження розміру, що змінюється. В одному прикладі заголовок блока NAL містить елемент `priority_id`, елемент `temporal_id`, елемент `anchor_pic_flag` (прапор опорного зображення), елемент `view_id`, елемент `non_idr_flag` (прапор не-IDR) і елемент `inter_view_flag` (прапор міжвидового прогнозування). У звичайному MVC зберігається блок NAL, визначений в H.264, за винятком блоків NAL префікса і блоків NAL кодованого слайса MVC, які включають в себе 4-байтовий заголовок блока NAL MVC і корисне навантаження блока NAL.

Елемент `priority_id` (ідентифікатор пріоритету) заголовка NAL може використовуватися для процесу адаптації простого одномаршрутного бітового потоку. Елемент `temporal_id` може використовуватися для задавання часового рівня відповідного блока NAL, де різні часові рівні відповідають різним частотам кадрів.

Елемент `anchor_pic_flag` може означати, чи є зображення опорним зображенням або неопорним зображенням. Опорні зображення і всі зображення, що йдуть за ними в порядку виведення (тобто в порядку відображення), можуть правильно декодуватися без декодування попередніх зображень в порядку декодування (тобто в порядку бітового потоку), і, таким чином, можуть використовуватися як точки випадкового доступу. Опорні зображення і неопорні зображення можуть мати різну залежність, обидві з яких сигналізуються в наборі параметрів послідовності. Інші прапори будуть обговорюватися і використовуватися в наступних розділах даного розділу. Таке опорне зображення також може згадуватися як точка доступу відкритої групи зображень (GOP), в той самий час підтримується точка доступу закритої GOP, коли елемент `non_idr_flag` дорівнює нулю. Елемент `non_idr_flag` означає, чи є зображення миттєвим оновленням декодера (IDR) або зображенням IDR виду (V-IDR). Загалом, зображення IDR і всі зображення, що йдуть за ним в порядку виведення або порядку бітового потоку, можуть правильно декодуватися без декодування попередніх зображень або в порядку декодування, або в порядку відображення.

Елемент `view_id` може містити синтаксичну інформацію, яка може використовуватися для ідентифікації виду, який може використовуватися для інтерактивності даних всередині декодера MVC, наприклад, для міжвидового прогнозування, і поза декодером, наприклад, для візуалізації. Елемент `inter_view_flag` може задавати, чи використовується відповідний блок NAL іншими видами для міжвидового прогнозування. Щоб передати 4-байтову інформацію заголовка блока NAL для базового виду, який може бути сумісним з AVC, в MVC визначається блок NAL префікса. У контексті MVC блок доступу базового виду включає в себе блоки NAL VCL поточного моменту часу виду, а також його блок NAL префікса, який містить тільки заголовок блока NAL. Декодер H.264/AVC може ігнорувати блок NAL префікса.

Блок NAL, що включає в себе відеодані в його корисному навантаженні, може містити різні рівні гранулярності відеоданих. Наприклад, блок NAL може містити блок відеоданих, макроблок, множину макроблоків, слайс відеоданих або весь кадр відеоданих. Блок 30 інкапсуляції може приймати кодовані відеодані від відеокодера 28 у вигляді пакетів PES елементарних потоків. Блок 30 інкапсуляції може асоціювати кожний елементарний потік з відповідною програмою.

Блок 30 інкапсуляції також може збирати блоки доступу від множини блоків NAL. Загалом, блок доступу може містити один або більше блоків NAL для представлення кадру відеоданих, а також аудіодані, що відповідають кадру, коли такі аудіодані є доступними. Блок доступу, в основному, включає в себе всі блоки NAL для одного моменту часу виведення, наприклад, всі аудіо- і відеодані для одного моменту часу. Наприклад, якщо кожний вид має частоту кадрів 20 кадрів в секунду (fps), тоді кожний момент часу може відповідати інтервалу часу 0,05 секунд.

Під час цього інтервалу часу конкретні кадри для всіх видів одного і того самого блока доступу (одного і того самого моменту часу) можуть візуалізуватися одночасно. У прикладі, що відповідає H.264/AVC, блок доступу може містити кодоване зображення в одному моменті часу, який може бути представлений як головне кодоване зображення. Отже, блок доступу може містити всі аудіо- і відеокадри загального часового моменту, наприклад, всі види, що відповідають часу X. Дане розкриття також посиляється на кодоване зображення конкретного виду як "складова виду". Тобто складова виду може містити кодоване зображення (або кадр) для конкретного виду в конкретний час. Отже, блок доступу може визначатися як такий, що містить всі складові виду загального часового моменту. Порядку декодування блоків доступу немає необхідності обов'язково бути таким самим, що і порядок виведення або зображення.

H.264/AVC визначає синтаксис, семантику і процес декодування для бітових потоків без помилок, будь-який з яких узгоджується з деяким профілем або рівнем. H.264/AVC не задає кодер, але кодеру ставиться задача з гарантуванням, що згенеровані бітові потоки будуть сумісними зі стандартом для декодера. У контексті стандарту відеокодування, "профіль" відповідає піднабору алгоритмів, ознак або інструментальних засобів та обмежень, які застосовуються до них. Як визначено в стандарті H.264, наприклад, "профіль" являє собою піднабір всього синтаксису бітового потоку, який задається стандартом H.264. "Рівень" відповідає обмеженням споживання ресурсів декодера, таких як, наприклад, пам'ять декодера і обчислення, які відносяться до розрізнення зображень, швидкість передачі бітів і швидкість обробки макроблоків (MB). Профіль може сигналізуватися за допомогою значення `profile_idc` (індикатор профілю), тоді як рівень може сигналізуватися за допомогою значення `level_idc` (індикатор рівня).

Стандарт H.264, наприклад, визнає, що, в межах меж, що накладаються синтаксисом даного профілю, все ж можливо вимагати велику зміну в характеристиках кодерів і декодерів залежно від значень, що приймаються елементами синтаксису в бітовому потоку, такими як заданий розмір декодованих зображень. Стандарт H.264 додатково визнає, що в багатьох застосуваннях ні практично, ні економічно не реалізувати декодер, здатний мати справу з всіма гіпотетичними використаннями синтаксису в конкретному профілі. Отже, стандарт H.264 визначає "рівень" як заданий набір обмежень, що накладаються на значення елементів синтаксису в бітовому потоку. Цими обмеженнями можуть бути прості межі на значення. Альтернативно, ці обмеження можуть приймати вид обмежень на арифметичні комбінації значень (наприклад, ширина зображення, помножена на висоту зображення, помножена на кількість зображень, декодованих в секунду). Стандарт H.264 додатково забезпечує, що індивідуальні реалізації можуть підтримувати різний рівень для кожного профілю, що підтримується.

Декодер, що відповідає профілю, звичайно підтримує всі ознаки, визначені в профілі. Наприклад, як ознака кодування, кодування B-зображення не підтримується в базовому профілі H.264/AVC, але підтримується в інших профілях H.264/AVC. Декодер, що відповідає рівню, повинен бути здатний декодувати будь-який бітовий потік, який не вимагає ресурсів поза обмеженнями, визначеними в рівні. Визначення профілів і рівнів може бути корисним для інтерпретованості. Наприклад, під час передачі відео пари визначень профілю і рівня може обговорюватися і встановлюватися для всього сеансу передачі. Більш конкретно, в H.264/AVC рівень може визначати, наприклад, обмеження на кількість макроблоків, які необхідно обробити, розмір буфера декодованих зображень (DPB), розмір буфера кодованих зображень (CPB), діапазон вектора руху по вертикалі, максимальна кількість векторів руху на два послідовних MB і, чи може B-блок мати розділення на субмакроблоки менше 8×8 пікселів. Таким чином, декодер може визначати, чи є декодер здатним належним чином декодувати бітовий потік.

Стандарти стиснення відео, такі як ITU-T H.261, H.262, H.263, MPEG-1, MPEG-2 та H.264/MPEG-4 part 10 використовують часове прогнозування з компенсацією руху для зменшення часової надмірності. Кодер використовує прогнозування з компенсацією руху з деяких раніше кодованих зображень (що також згадуються в даному документі як кадри) для прогнозування поточних зображень, що кодуються відповідно до векторів руху. Є три головних типи зображень в типовому відеокодуванні. Ними є внутрішньо кодовані зображення ("I-зображення" або "I-кадри"), зображення з прогнозуванням ("P-зображення" або "P-кадри") і зображення з двонаправленим прогнозуванням ("B-зображення" або "B-кадри"). P-зображення використовують тільки опорне зображення перед поточним зображенням у часовому порядку. У B-зображенні кожний блок B-зображення може прогнозуватися з одного або двох опорних зображень. Ці опорні зображення можуть розташовуватися перед і після поточного зображення у часовому порядку.

Відповідно до стандарту кодування H.264, як приклад, B-зображення використовують два списки раніше кодованих опорних зображень, список 0 і список 1. Кожний з цих двох списків

може містити минулі і/або майбутні зображення, що кодуються у часовому порядку. Блоки в В-зображенні можуть прогнозуватися одним з декількома способами: прогнозування з компенсацією руху з опорного зображення списку 0, прогнозування з компенсацією руху з опорного зображення списку 1 або прогнозування з компенсацією руху з комбінації опорних зображень як списку 0, так і списку 1. Щоб одержати комбінацію опорних зображень як списку 0, так і списку 1, дві опорні ділянки з компенсацією руху одержуються з опорного зображення списку 0 і списку 1 відповідно. Їх комбінація використовується для прогнозування поточного блока.

Стандарт ITU-T H.264 підтримує внутрішнє прогнозування в різних розмірів блока, таких як 16 на 16, 8 на 8 або 4 на 4 для складової яскравості, і 8×8 для складових кольоровості, а також зовнішнє прогнозування в різних розмірах блока, таких як 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 та 4×4 для складових яскравості і відповідні масштабовані розміри для складових кольоровості. У даному розкритті "N×N" і "N на N" можуть використовуватися наперемінно для посилення на піксельні розміри блока в значенні розмірів по вертикалі і по горизонталі, наприклад, 16×16 пікселів або 16 на 16 пікселів. Звичайно, блок 16×16 має 16 пікселів у вертикальному напрямку (y=16) і 16 пікселів в горизонтальному напрямку (x=16). Аналогічно, блок N×N, як правило, має N пікселів у вертикальному напрямку і N пікселів в горизонтальному напрямку, де N представляє ненегативне ціле число. Пікселі в блоці можуть розміщуватися рядами і стовпцями. Блоки можуть мати різну кількість пікселів в розмірах по горизонталі і по вертикалі. Тобто блоки можуть включати в себе N×M пікселів, де N необов'язково дорівнює M.

Розміри блока, які менше 16 на 16, можуть згадуватися як розділення макроблока 16 на 16. Відеоблоки можуть містити блоки піксельних даних в піксельній ділянці або блоки коефіцієнтів перетворення в ділянці перетворення, наприклад, дотримуючись застосування перетворення, такого як дискретне косинусне перетворення (DCT), цілочисельне перетворення, вейвлет-перетворення або концептуально подібне перетворення в дані залишкового відеоблока, що представляють піксельні відмінності між кодованими відеоблоками і відеоблоками прогнозування. У деяких випадках відеоблок може містити блоки квантованих коефіцієнтів перетворення в ділянці перетворення.

Менші відеоблоки можуть забезпечувати краще розрізнення і можуть використовуватися для розташування відеокадру, яке включає в себе високі рівні деталізування. Загалом, макроблоки і різні розділення, що іноді згадуються як субблоки, можуть вважатися відеоблоками. Крім того, слайс може вважатися множиною відеоблоків, таких як макроблоки і/або субблоки. Кожний слайс може являти собою незалежно декодований блок відеокадру. Альтернативно, самі кадри можуть бути декодованими блоками, або інші частини кадру можуть визначатися як декодовані блоки. Термін "кодований блок" або "блок кодування" може посилатися на будь-який незалежно декодований блок відеокадру, такий як весь кадр, слайс кадру, групу зображень (GOP), що також згадується як послідовність, або інший незалежно декодований блок, визначений відповідно до застосованих методів кодування.

Термін "макроблок" посилається на структуру даних для кодування зображення і/або відеоданих відповідно до двовимірного масиву пікселів, який містить 16×16 пікселів. Кожний піксель містить складову кольоровості і складову яскравості. Отже, макроблок може визначати чотири блоки яскравості, причому кожний містить двовимірний масив 8×8 пікселів, два блоки кольоровості, причому кожний містить двовимірний масив 16×16 пікселів, і заголовок, що містить інформацію синтаксису, таку як структура кодованого блока (CBP), режим кодування (наприклад, режим внутрішнього (I) або зовнішнього (P або B) кодування), розмір розділення для розділень внутрішньо кодованого блока (наприклад, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 або 4×4), або один або більше векторів руху для зовні кодованого макроблока.

Кожний з відеокодера 28, відеодекодера 48, аудіокодера 26, аудіодекодера 46, блока 30 інкапсуляції і блока 38 декапсуляції може бути реалізований як будь-яка з численних підходящих схем обробки за відповідних умов, таких як один або більше мікропроцесорів, процесори цифрової обробки сигналів (DSP), спеціалізовані інтегральні схеми (ASIC), програмовані вентиляльні матриці (FPGA), дискретні логічні схеми, програмне забезпечення, апаратне забезпечення, програмно-апаратне забезпечення або будь-яка їх комбінація. Кожний з відеокодера 28 і відеодекодера 48 може бути включений в один або більше кодерів або декодерів, будь-який з яких може бути інтегрований як частина об'єднаного відео кодера/декодера (кодека). Аналогічно, кожний з аудіокодера 26 і аудіодекодера 46 може бути включений в один або більше кодерів або декодерів, будь-який з яких може бути інтегрований як частина об'єднаного кодека. Пристрій, що включає в себе відеокодер 28, відеодекодер 48, аудіокодер 26, аудіодекодер 46, блок 30 інкапсуляції і/або блок 38 декапсуляції, може містити



інтегральну схему, мікропроцесор і/або пристрій бездротового зв'язку, такий як стільниковий телефон.

Після того як блок 30 інкапсуляції збере блоки NAL і/або блоки доступу у відеофайл, оснований на прийнятих даних, блок 30 інкапсуляції передає відеофайл на інтерфейс 32 виводу для виведення. У деяких прикладах блок 30 інкапсуляції може зберігати відеофайл локально або посилати відеофайл на віддалений сервер за допомогою інтерфейсу 32 виводу, замість посилання відеофайлу безпосередньо на пристрій 40 призначення. Інтерфейс 32 виводу може містити, наприклад, передавач, приймач-передавач, пристрій для запису даних на зчитуваний комп'ютером носій, такий як, наприклад, накопичувач на оптичних дисках, накопичувач на магнітних носіях (наприклад, накопичувач на гнучких магнітних дисках), порт універсальної послідовної шини (USB), мережний інтерфейс або інший інтерфейс виводу. Інтерфейс 32 виводу виводить відеофайл на зчитуваний комп'ютером носій 34, такий як, наприклад, сигнал передачі, магнітний носій, оптичний носій, пам'ять, флеш-накопичувач або інший зчитуваний комп'ютером носій.

Зрештою, інтерфейс 36 вводу витягує дані із зчитуваного комп'ютером носія 34. Інтерфейс 36 вводу може містити, наприклад, накопичувач на оптичних дисках, накопичувач на магнітних носіях, порт USB, приймач, приймач-передавач або інший інтерфейс зчитуваного комп'ютером носія. Інтерфейс 36 вводу може подавати блок NAL або блок доступу на блок 38 декапсуляції. Блок 38 декапсуляції може декапсулювати елементи відеофайлу в складові потоків PES, депакетувати потоки PES для витягання кодованих даних, і посилати кодовані дані або на аудіодекодер 46, або на відеодекодер 48 залежно від того, чи є кодовані дані частиною аудіо-або відеопотоку, наприклад, як визначено заголовками пакету PES потоку. Аудіодекодер 46 декодує кодовані аудіодані і посилає декодовані аудіодані на вивід 42 аудіо, тоді як відеодекодер 48 декодує кодовані відеодані і посилає декодовані відеодані, які можуть включати в себе множину видів потоку, на вивід 44 відео.

Блок 38 декапсуляції може взаємодіяти з інтерфейсом 36 вводу для первинного запиту даних заголовка для відеофайлу, де дані заголовка можуть описувати характеристики відеофайлу. Наприклад, дані заголовка можуть описувати характеристики фрагментів субтреку, включених у фрагменти треку фрагментів фільму у відеофайлі. Дані заголовка можуть описувати, наприклад, байтові діапазони індивідуальних фрагментів субтреку фрагмента фільму. Дані заголовка також можуть описувати інші характеристики, які можуть сприяти блоку 38 декапсуляції при виборі піднабору доступних фрагментів субтреку відеофайлу. Після вибору конкретного набору доступних фрагментів субтреку блок 38 декапсуляції може представити один або більше запитів для вибраних фрагментів субтреку кожного фрагмента фільму відеофайлу.

Наприклад, блок 38 декапсуляції може вибирати конкретну робочу точку, яка може відповідати піднабору доступних ієрархічних шарів. Потім блок 38 декапсуляції може визначити для кожного фрагмента фільму, які фрагменти субтреку фрагмента фільму відповідають ієрархічним шарам робочої точки. Крім того, блок 38 декапсуляції може визначити байтові діапазони в кожному фрагменті фільму для відповідних фрагментів субтреку. Основуючись на цих визначених байтових діапазонах, блок 38 декапсуляції може генерувати запити часткового GET HTTP, які задають визначені байтові діапазони для фрагментів фільму для витягання фрагментів субтреку. У деяких прикладах блок 38 декапсуляції може генерувати індивідуальні запити для кожного потрібного шару. У деяких прикладах блок 38 декапсуляції може генерувати єдиний запит для фрагментів субтреку, що охоплює численні шари. Блок 38 декапсуляції потім може повторно розміщувати кодовані семпли відео фрагментів субтреку в порядку декодування, використовуючи об'єкти повторного збирача фрагментів субтреку і передати розміщені кодовані семпли відео на відеодекодер 48, який може декодувати семпли відео. Зрештою, вивід 44 відео може відображати декодовані семпли відео.

Фіг.2 являє собою блок-схему, що ілюструє компоненти зразкового блока інкапсуляції. У прикладі на фіг.2 блок 30 інкапсуляції включає в себе інтерфейс 80 вводу відео, інтерфейс 82 вводу аудіо, блок 60 створення відеофайлу та інтерфейс 84 виводу відеофайлу. Блок 60 створення відеофайлу в даному прикладі включає в себе блок 62 конструювання блока рівня мережної абстракції (NAL) і блок 64 створення фрагмента субтреку, який додатково включає в себе блок 66 керування шаром, блок 68 вставки семпла, блок 70 створення заголовка і блок 72 створення об'єкта повторного збирача.

Інтерфейс 80 виводу відео та інтерфейс 82 вводу аудіо приймають кодоване відео- і аудіодані відповідно. Інтерфейс 80 вводу відео та інтерфейс 82 вводу аудіо можуть приймати кодовані відео- і аудіодані, коли дані кодуються, або можуть витягувати кодовані відео- і аудіодані зі зчитуваного комп'ютером носія. Під час прийому кодованих відео- і аудіоданих

інтерфейс 80 вводу відео та інтерфейс 82 вводу аудіо передають кодовані відео- і аудіодані на блок 60 створення відеофайлу для збирання у відеофайл.

Блок 60 створення відеофайлу може відповідати блока керування, що включає в себе апаратне забезпечення, програмне забезпечення і/або програмно-апаратне забезпечення, виконане з можливістю виконання функцій і процедур, що приписуються їм. Кожний з субблоків блока 60 створення відеофайлу (блок 62 конструювання блока NAL, блок 64 створення фрагмента субтреку, блок 66 керування шаром, блок 68 вставки семпла, блок 70 створення заголовка і блок 72 створення об'єкта повторного збирача в даному прикладі) може бути реалізований у вигляді індивідуальних апаратних блоків і/або програмних модулів, і/або може бути функціонально інтегрований або додатково розділений на додаткові субблоки.

Блок 60 створення відеофайлу може відповідати будь-якому підходящому блоку обробки або схемі обробки, такий як, наприклад, один або більше мікропроцесорів, спеціалізовані інтегральні схеми (ASIC), програмовані вентильні матриці (FPGA), процесори цифрової обробки сигналів (DSP) або будь-яка їх комбінація. Блок 60 створення відеофайлу може додатково включати в себе зчитуваний комп'ютером носій, що містить інструкції для будь-якого або для всіх з блока 62 конструювання блока NAL, блока 64 створення фрагмента субтреку, блока 66 керування шаром, блока 68 вставки семпла, блока 70 створення заголовка і блока 72 створення об'єкта повторного збирача, а також процесор для виконання інструкцій.

Загалом, блок 60 створення відеофайлу може створювати відеофайл, що включає в себе прийняті аудіо- і відеодані. Блок 62 конструювання блока NAL може формувати блоки NAL, що включають в себе кодовані семпли відео та аудіо. Блок 60 створення відеофайлу може додатково бути виконаний з можливістю збирання фрагментів фільму, що включають в себе кодовані семпли відео, розміщені в порядку ієрархічного рівня. Тобто блок 60 створення відеофайлу може бути виконаний з можливістю організації кодованого семплів відео фрагмента фільму, так що кодовані семпли відео загального ієрархічного рівня фрагмента фільму зберігаються поряд у фрагменті фільму.

Блок 66 керування шаром може розрізняти різні ієрархічні шари даних для відеофайлу. Блок 66 керування шаром може додатково визначати відповідність між фрагментами субтреку та ієрархічними шарами, наприклад, основуючись на стандарті формату файлів, якому відповідає відеофайл. Наприклад, у відношенні H.264/AVC, блок 66 керування шаром може асоціювати часові шари кодування з фрагментами субтреку. Як інший приклад, у відношенні SVC, блок 66 керування шаром може асоціювати просторові шари (наприклад, базовий шар і один або більше поліпшувачих шарів) з фрагментами субтреку. Як інший приклад, у відношенні MVC, блок 66 керування шаром може асоціювати різні види з фрагментами субтреку.

Після визначення асоціювання між ієрархічними шарами і фрагментами субтреку блок 68 вставки семпла може вставляти кодовані семпли відео у підходящі фрагменти субтреку під час створення відеофайлу. Тобто блок 68 вставки семпла може приймати кодований семпл відео, визначати ієрархічний шар, якому відповідає семпл, визначати фрагмент субтреку фрагмента фільму, що відповідає ієрархічному шару, і вставляти семпл у визначений фрагмент субтреку. Таке розміщення може дозволяти виконувати витягання даних із загального ієрархічного шару, використовуючи єдиний запит, наприклад, єдиний запит часткового GET HTTP, який задає байтовий діапазон фрагмента субтреку, що відповідає ієрархічному шару.

Блок 70 створення заголовка може створювати заголовки для фрагментів фільму і/або фрагментів треку. У деяких прикладах блок 70 створення заголовка може зберігати дані заголовка в боксі фільму, який описує кількість фрагментів фільму створеного відеофайлу. Загалом, дані заголовка, створені блоком 70 створення заголовка, можуть описувати характеристики фрагментів субтреку, такі як, наприклад, байтові діапазони для фрагментів субтреку і/або кількість семплів у фрагменті субтреку. У деяких прикладах, наприклад, тих, для яких ієрархічні шари містять часові шари кодування, блок 70 створення заголовка може задавати часову інформацію для кожного фрагмента субтреку.

Блок 72 створення об'єкта мультіплексора може створювати і вставляти об'єкти повторного збирача у фрагменти субтреку. Об'єкт повторного збирача може служити як покажчик для ідентифікації семпла іншого фрагмента субтреку, який може бути вставлений в положення об'єкта повторного збирача у фрагменті субтреку, що включає в себе об'єкт повторного збирача. Наприклад, в AVC та SVC об'єкти повторного збирача можуть спрощувати задачу повторного розміщення і перевпорядкування кодованих семплів відео на відносно більш високих шарах (тобто шарах, що включають в себе відносно більшу кількість семплів). Блок 72 створення об'єкта повторного збирача може створювати об'єкти повторного збирача, які включають в себе індекс (або інший ідентифікатор) фрагмента субтреку, що включає в себе

опорний семпл, а також положення опорного семпла у фрагменті субтреку. Положення може бути виражене відносно положення повторного збирача в поточному фрагменті субтреку.

Отже, блок 60 створення відеофайлу може виробляти різні типи відеофайлів, що включають в себе фрагменти субтреку, відповідно до методів даного розкриття. Після того як блок 60 створення відеофайлу виробляє відеофайл, що включає в себе фрагменти фільму, які мають кодовані семпли відео, згруповані відповідно до їх відповідних ієрархічних рівнів, блок 60 створення відеофайлу може передати відеофайл на інтерфейс 84 виводу відеофайлу. Інтерфейс 84 виводу відеофайлу може виводити відеофайл, наприклад, на інтерфейс 32 виводу 20 джерела. У деяких прикладах інтерфейс 84 виводу відеофайлу може виводити відеофайл на запам'ятовуючий носій пристрою 20 джерела (не показаний). Відеофайл може зберігатися локально в пристрої 20 джерела, зберігатися на портативному запам'ятовуючому носії, такому як цифровий багатофункціональний диск (DVD), диск Blu-ray, флеш-накопичувач, гнучкий диск або інший портативний запам'ятовуючий носій, виводитися по мережі, наприклад, відповідно до протоколу потокової передачі, такої як потокова передача HTTP, або іншим чином виводитися таким чином, що відеофайл може прийматися клієнтським пристроєм, таким як пристрій 40 призначення.

Фіг.3 являє собою блок-схему, що ілюструє елементи зразкового відеофайлу 100, що має фрагменти 112 відео, причому кожний включає в себе фрагменти субтреку, які мають кодовані відеозображення загального ієрархічного рівня. Як описано вище, відеофайли відповідно до базового формату медіафайлів ISO і його розширень зберігають дані в послідовності об'єктів, що згадуються як "бокси". У прикладі на фіг.3, відеофайл 100 включає в себе бокс 102 типу файлу (FTYP), бокс 104 фільму (MOOV), бокси 112 фрагмента фільму (MOOF) і бокс 114 випадкового доступу фрагмента фільму (MFRA).

Бокс 102 типу файлу, загалом, описує тип файлу для відеофайлу 100. Бокс 102 типу файлу може включати в себе дані, які ідентифікують специфікацію, яка описує найкраще використання для відеофайлу 100. Бокс 102 типу файлу може розміщуватися перед боксом 104 MOOV, боксами 112 фрагмента фільму і боксом 114 MFRA.

Бокс 104 MOOV в прикладі на фіг.3 включає в себе бокс 106 заголовки фільму (MVHD), бокс 108 треку (TRAK) і один або більше боксів 110 розширення фільму (MVEX). Загалом, бокс 106 MVHD може описувати загальні характеристики відеофайлу 100. Наприклад, бокс 106 MVHD може включати в себе дані, які описують, коли відеофайл 100 був спочатку створений, коли в останній раз був модифікований відеофайл 100, часову шкалу для відеофайлу 100, тривалість відтворення для відеофайлу 100 або інші дані, які, загалом, описують відеофайл 100.

У деяких прикладах блок 30 інкапсуляції може визначати характеристики фрагментів субтреку, які відповідають робочим точкам в боксі 104 MOOV, або сегмента ініціалізації для потокової передачі HTTP. У деяких прикладах блок 30 інкапсуляції може генерувати бокс заголовка фрагмента субтреку, який може включатися як дані заголовка для відеофайлу 100, причому один з фрагментів 112 фільму включає в себе фрагмент субтреку, або в інших розташуваннях. Визначення фрагмента субтреку може включати в себе структуру даних, яка відображає кожний фрагмент субтреку робочої точки на описові характеристики для фрагмента субтреку, такі як, наприклад, значення часового рівня, значення *quality\_id*, значення *dependency\_id* і/або значення *view\_id*. Визначення робочої точки може додатково включати в себе описову інформацію, таку як, наприклад, інформація КОДЕК, інформація про профіль і рівень, частота кадрів для робочої точки, середня швидкість передачі бітів для робочої точки, максимальна швидкість передачі бітів для робочої точки, просторове розрізнення для робочої точки, кількість видів, що підлягають відображенню для робочої точки, і/або кількість видів, що підлягають декодуванню для робочої точки. Визначення робочої точки стандарту, що відноситься, можуть модифікуватися так, щоб включати в себе такі дані.

Бокс 108 TRAK може включати в себе дані для треку відеофайлу 100. Бокс 108 TRAK може включати в себе бокс заголовка треку (TKHD), який описує характеристики треку, що відповідає боксу 108 TRAK. У деяких прикладах бокс 108 TRAK може включати в себе кодовані семпли відео, тоді як в інших прикладах, кодовані семпли відео треку може бути включені у фрагменти 112 фільму, на які можуть посилатися дані боксу 108 TRAK.

У деяких прикладах відеофайл 100 може включати в себе більше одного треку. Отже, бокс 104 MOOV може включати в себе деяку кількість боксів TRAK, що дорівнює кількості треків у відеофайлі 100. Бокс 108 TRAK може описувати характеристики відповідного треку відеофайлу 100. Наприклад, бокс 108 TRAK може описувати часову і/або просторову інформацію для відповідного треку. Бокс TRAK, подібний до боксу 108 TRAK боксу 104 MOOV, може описувати характеристики треку набору параметрів, коли блок 30 інкапсуляції (фіг.1) включає в себе трек набору параметрів у відеофайлі, такому як відеофайл 100.

Бокси 110 MVEX можуть описувати характеристики відповідних фрагментів 112 фільму, наприклад, щоб сигналізувати, що відеофайл 100 включає в себе фрагменти 112 фільму в доповнення до відеоданих, включених в бокс 104 MOOV, якщо вони є. У контексті потокових відеоданих кодовані семпли відео можуть бути включені в фрагменти 112 фільму, а не в бокс 104 MOOV. Отже, всі кодовані семпли відео можуть бути включені в фрагменти 112 фільму, а не в бокс 104 MOOV.

Бокс 104 MOOV може включати в себе деяку кількість боксів 110 MVEX, кількість яких дорівнює кількості фрагментів 112 фільму у відеофайлі 100. Кожний з боксів 110 MVEX може описувати характеристики відповідного одного з фрагментів 112 фільму. Наприклад, кожний бокс MVEX може включати в себе бокс заголовка розширення фільму (MEHD), який описує часову тривалість для відповідного одного з фрагментів 112 фільму.

Фрагменти 112 фільму можуть включати в себе один або більше кодованих семплів відео. У деяких прикладах фрагменти 112 фільму можуть включати в себе одну або більше груп зображень (GOP), кожна з яких може включати в себе деяку кількість кодованих семплів відео, наприклад, кадрів або зображень. Крім того, як описано вище, фрагменти 112 фільму можуть включати в себе набори даних послідовності в деяких прикладах. Кожний з фрагментів 112 фільму може включати в себе бокс заголовка фрагмента фільму (MFHD). Бокс MVHD може описувати характеристики відповідного фрагмента фільму, такі як порядковий номер для фрагмента фільму. Фрагменти 112 фільму можуть бути включені в порядок порядкового номера у відеофайлі 100.

Як відмічено вище, блок 30 інкапсуляції може організовувати кодовані семпли відео кожного з фрагментів 112 фільму в порядку ієрархічних рівнів кодованих семплів відео. Тобто в кожному з фрагментів 112 фільму блок 30 інкапсуляції може організовувати кодовані семпли відео фрагмента фільму так, що кодовані семпли відео загального ієрархічного рівня зберігаються поряд в фрагменті фільму. Таким чином, пристрій 40 призначення (фіг.1) може витягувати всі кодовані семпли відео до конкретного ієрархічного шару з одного з фрагментів 112 фільму за допомогою представлення єдиного запиту, наприклад, часткового GET HTTP, який задає байтовий діапазон, що включає в себе необхідний діапазон ієрархічних рівнів. Аналогічно, пристрій 40 призначення може витягувати кодовані семпли відео загального ієрархічного шару, використовуючи єдиний запит, і може представляти один запит для кожного необхідного ієрархічного шару.

Бокс 104 MOOV і/або фрагменти 112 фільму можуть включати в себе дані заголовка, які описують фрагменти субтреку фрагментів 112 фільму, такі як, наприклад, байтові діапазони фрагментів 112 фільму, що включають в себе конкретні фрагменти субтреку. Таким чином, пристрій 40 призначення може витягувати бокс 104 MOOV і/або заголовки фрагментів 112 фільму для визначення, яку частину(-и) фрагментів 112 фільму запитувати, основуючись на необхідних фрагментах субтреку.

Бокс 114 MFRA може описувати точки випадкового доступу у фрагментах 112 фільму відеофайлу 100. Це може сприяти виконанню пошуку конкретного часового розташування у відеофайлі 100. Бокс 114 MFRA, звичайно, є необов'язковим і немає необхідності включати його у відеофайли. Аналогічно, клієнтському пристрою, такому як пристрій 40 призначення, немає необхідності обов'язково посилатися на бокс 114 MFRA для правильного декодування і відображення відеоданих відеофайлу 100. Бокс 114 MFRA може включати в себе деяку кількість боксів випадкового доступу фрагмента треку (TFRA), що дорівнює кількості треків відеофайлу 100, або в деяких прикладах дорівнює кількості треків медіа (наприклад, треків без підказок) відеофайлу 100.

Фіг.4А являє собою блок-схему, що ілюструє зразковий фрагмент 180А фільму. Фрагмент 180А фільму може відповідати одному з фрагментів 112 фільму (фіг.3). У прикладі на фіг.4А фрагмент 180А фільму включає в себе різні фрагменти субтреку. Зокрема, в даному прикладі фрагмент 180А фільму включає в себе фрагмент 182 субтреку шару 0, фрагмент 188 субтреку шару 1 і фрагмент 192 субтреку шару 2.

Фрагмент 182 субтреку шару 0 може включати в себе кодовані семпли відео, що мають часовий ієрархічний шар кодування нуля. У даному прикладі цей шар включає в себе І-кадр 184 та Р-кадри 186А-186N (Р-кадри 186). Р-кадри 186 можуть кодуватися відносно попередніх р-кадрів 186 і/або І-кадру 184. Наприклад, макроблоки Р-кадру 186А можуть кодуватися відносно І-кадру 184, тоді як макроблоки Р-кадру 186В можуть кодуватися відносно І-кадру 184 або Р-кадру 186А.

Фрагмент 188 субтреку шару 1 в даному прикладі включає в себе В-кадри 190А-190N (В-кадри 190). Кожний з В-кадрів 190 має часову ієрархію кодування шару 1. Отже, В-кадри 190

можуть кодуватися відносно одного або більше кадрів фрагмента 182 субтреку шару 0, тобто I-кадру 184 і/або P-кадрів 186.

Фрагмент 192 субтреку шару 2 в даному прикладі включає в себе B-кадри 194A-194N (B-кадри 194). Кожний з B-кадрів 194 має часову ієрархію кодування шару 2. Отже, B-кадри 194 можуть кодуватися відносно одного або більше кадрів фрагмента 188 субтреку шару 1, тобто B-кадрів 190. Крім того, фрагмент 180 відео може включати в себе додаткові фрагменти субтреку, що відповідають більш високим часовим шарам кодування, як вказано багатокрапкою, що йде за фрагментом 192 субтреку шару 2.

Хоча кардинальне число кожних з P-кадрів 186, B-кадрів 190 і B-кадрів 194 виражено змінною "N", необхідно розуміти, що N є змінною в кожному випадку. Тобто кількість P-кадрів 186 необов'язково дорівнює кількості B-кадрів 190, яка додатково необов'язково дорівнює кількості B-кадрів 194.

Пристрій 40 призначення може визначати витягання фрагментів субтреку до конкретного ієрархічного шару. Отже, пристрій 40 призначення може представляти один або більше запитів на витягання фрагментів субтреку, що відповідають ієрархічним шарам менше і/або рівних визначеному шару. Наприклад, передбачаючи, що пристрій 40 призначення визначив витягання фрагментів субтреку до шару один, пристрій 40 призначення може представити запити часткового GET HTTP для витягання фрагмента 182 субтреку шару 0 і фрагмента 188 субтреку шару 1. В деяких прикладах пристрій 40 призначення може представляти максимально два запити часткового GET HTTP на витягання фрагмента 182 субтреку шару 0 і фрагмента 188 субтреку шару 1. В прикладі на фіг.4A пристрій 40 призначення може альтернативно представити єдиний запит часткового GET HTTP на витягання як фрагмента 182 субтреку шару 0, так і фрагмента 188 субтреку шару 1, оскільки фрагмент 182 субтреку шару 0 і фрагмент 188 субтреку шару 1 розміщені поряд в фрагменті 180 відео, в даному прикладі.

Фіг.4B являє собою блок-схему, що ілюструє зразковий фрагмент 180B фільму. Фрагмент 180B фільму аналогічний фрагменту 180A фільму на фіг.4A, за винятком того, що в прикладі на фіг.4B фрагменти субтреку більш високого шару можуть включати в себе об'єкти повторного збирача. Наприклад, фрагмент 196 субтреку шару 1 в даному прикладі включає в себе об'єкти 198A, 198B та 198C повторного збирача. Об'єкт 198A повторного збирача ідентифікує I-кадр 184 фрагмента 182 субтреку шару 0, об'єкт 198B повторного збирача ідентифікує P-кадр 186A фрагмента 182 субтреку шару 0, і об'єкт 198C повторного збирача ідентифікує P-кадр 186B фрагмента 182 субтреку шару 0 в даному прикладі. Фрагменти субтреку більш високих шарів можуть включати в себе повторні збирачі, які ідентифікують кадри фрагмента 182 субтреку шару 0, і повторні збирачі, які ідентифікують B-кадри 199 фрагмента 196 субтреку шару 1.

Пристрій 40 призначення може використовувати повторні збирачі 198, щоб сприяти перевпорядкуванню кадрів у порядок декодування. Наприклад, блок 38 декапсуляції може перевпорядковувати кадри фрагмента 182 субтреку шару 0 і фрагмента 196 субтреку шару 1 для вироблення набору кадрів у порядку декодування I-кадру 184, P-кадру 186A, B-кадру 199A, P-кадру 186B, B-кадру 199B тощо. Блок 38 декапсуляції потім може направляти кадри в порядку декодування на відеодекодер 48. Відеодекодер 48 потім може декодувати кадри, і відеодисплей 44 може, зрештою, відображати кадри в порядку відображення, який може бути відрізнитися від порядку декодування і порядку кадрів, розміщених у фрагменті 180B відео.

Фіг.5 являє собою блок-схему, що ілюструє зразковий фрагмент 200 відео SVC. У даному прикладі фрагмент 200 відео SVC включає в себе фрагмент 202 субтреку базового шару, фрагмент 206 субтреку поліпшуючого шару 1 і фрагмент 210 субтреку поліпшуючого шару 2. Фрагмент 202 субтреку базового шару включає в себе кадри 204A-204N базового шару (кадри 204 базового шару). Фрагмент 206 субтреку поліпшуючого шару 1 включає в себе поліпшуючі кадри 208A-208N (поліпшуючі кадри 208). Фрагмент 210 субтреку поліпшуючого шару 2 включає в себе поліпшуючі кадри 212A-212N (поліпшуючі кадри 212). Знов-таки, необхідно розуміти, що N є потенційно різним для будь-якого з кадрів 204 базового шару, поліпшуючих кадрів 208 і поліпшуючих кадрів 212.

Кадри 204 базового шару можуть відповідати кадрам формату "чверть загального проміжного формату" (QCIF). Поліпшуючі кадри 208 можуть відповідати кадрам просторового поліпшуючого шару CIF. Поліпшуючі кадри 212 можуть відповідати додатковим кадрам просторового поліпшуючого шару.

Таким чином, методи даного розкриття можуть застосовуватися в контексті SVC. Крім того, поліпшуючі шари в SVC також можуть включати в себе об'єкти повторного збирача, які посилаються на кадри базового шару і/або більш низьких поліпшуючих шарів. Отже, пристрій 40 призначення може вибирати максимальний необхідний шар і представляти один або більше

запитів (наприклад, запити часткового GET HTTP) на витягання даних для шарів до вибраного шару.

Фіг.6 являє собою блок-схему, що ілюструє зразковий фрагмент 220 відео MVC. У даному прикладі фрагмент 220 відео MVC включає в себе фрагмент 222 субтреку виду 0, фрагмент 226 субтреку виду 1 і фрагмент 230 субтреку виду 2. Кожний вид може включати в себе деяка кількість складових виду. Наприклад, фрагмент 222 субтреку виду 0 включає в себе кадри 224A-224N виду 0 (кадри 224 виду 0), фрагмент 226 субтреку виду 1 включає в себе кадри 228A-228N м 1 (кадри 228 виду 1), і фрагмент 230 субтреку виду 2 включає в себе кадри 232A-232N виду 2 (кадри 232 виду 2).

У контексті MVC складові виду кожного виду можуть розміщуватися в різних фрагментах субтреку, як показано на фіг.6. Крім того, як описано вище, фрагменти субтреку виду можуть включати в себе повторні збирачі, які вказують на складові виду попередніх фрагментів субтреку, які можуть містити кодовані семпли відео складових виду.

Пристрій 40 призначення може витягувати складові виду конкретного виду за допомогою видачі запиту часткового GET HTTP, який задає байтовий діапазон для фрагмента субтреку виду, що відповідають виду. Наприклад, щоб витягнути складові виду у виду 0, пристрій 40 призначення може представити запит часткового GET HTTP, що задає байтовий діапазон фрагмента 222 субтреку виду 0 в фрагменті 220 відео MVC. Аналогічно, пристрій 40 призначення може видати індивідуальні запити на витягання будь-якого або всіх з інших видів фрагмента 220 відео MVC. При прийомі запитаних видів пристрій 40 призначення може впорядкувати складові виду в порядку декодування, декодувати складові виду і відобразити декодовані відеодані.

Фіг.7 являє собою блок-схему послідовності операцій, що ілюструє зразковий спосіб інкапсуляції відеоданих загальних ієрархічних рівнів у відповідних фрагментах субтреку фрагмента фільму у відеофайлі і подачі відеофайлу від пристрою джерела на пристрій призначення. Хоча він описаний відносно компонентів пристрою 20 джерела і пристрою 40 призначення (фіг.1) для цілей прикладу і пояснення, необхідно розуміти, що будь-який підходящий пристрій може реалізувати методи фіг.7.

Пристрій 20 джерела може спочатку скласти відеофайл. Щоб це зробити, пристрій 20 джерела може приймати набір кодованих семплів відео (210). Наприклад, пристрій 20 джерела може витягувати кодовані семпли відео із запам'ятовуючого носія або приймати кодовані семпли відео в реальному часі, як семпли кодуються, наприклад, відеокoderом 28. Набір семплів відео може відповідати фрагменту фільму в більшому відеофайлі. Тобто пристрій 20 джерела може визначати, що прийнятий набір семплів відео повинен розміщуватися в загальному фрагменті відео.

Пристрій 20 джерела потім може розділяти семпли для фрагмента відео на відповідні шари (212). Наприклад, для AVC пристрій 20 джерела може розділяти семпли відео на шари часового кодування. Як інший приклад, для SVC пристрій 20 джерела може розділяти семпли відео на базовий шар і один або більше поліпшувачих шарів. Як ще інший приклад, для MVC пристрій 20 джерела може розділяти семпли на відповідні види. У будь-якому випадку, пристрій 20 джерела може виробляти фрагменти субтреку для кожного відповідного шару, так що фрагменти субтреку включають в себе кодовані семпли відео для відповідного шару (214). Пристрій 20 джерела потім може виводити фрагмент фільму (216). Тобто пристрій 20 джерела може включати в себе фрагмент фільму у відеофайлі, що зберігається на зчитуванні комп'ютером носії.

Наприклад, пристрій 20 джерела може служити як мережний сервер для подачі даних на пристрої призначення у відповідь на запити потокової передачі HTTP. Альтернативно, пристрій 20 джерела може посилати фрагмент фільму на окремий мережний сервер. У деяких прикладах пристрій 20 джерела може виводити фрагмент фільму за допомогою посилання фрагмента фільму безпосередньо на клієнтський пристрій.

Пристрій 20 джерела може виробляти кожний фрагмент фільму відеофайлу. Крім того, пристрій 20 джерела може зберігати дані заголовка для відеофайлу, які ідентифікують байтові діапазони кожного фрагмента субтреку кожного фрагмента фільму. Аналогічно, пристрій 20 джерела може включати в себе об'єкти повторного збирача у фрагментах субтреку, які посилаються на кодовані семпли попередніх фрагментів субтреку. Пристрій 20 джерела також може включати в себе заголовки демультимплексування у фрагментах фільму, які задають, наприклад, байтові діапазони кожного фрагмента субтреку фрагмента фільму, кількість семплів у кожному фрагменті субтреку і/або часову інформацію для кодованих семплів відео.

Є випадки, коли об'єктам повторного збирача необов'язково перевпорядковувати блоки доступу в різних фрагментах субтреку для підтримки правильного порядку декодування.

Наприклад, в TS MPEG-2 пакети, що містять відеодані, можуть включати в себе часову позначку декодування. Таким чином, може визначатися час декодування кожного блока доступу, і такий процес перевпорядкування не потребує додаткової сигналізації. Також, в деяких прикладах чергування ієрархічного шару з індексом  $i$  та ієрархічного шару з індексом  $i+1$  може дотримуватися фіксованого шаблону і, таким чином, може сигналізуватися дуже незначна сигналізація, наприклад, деяка кількість семплів відео в ієрархічному шарі  $i$  та інша кількість семплів відео, що йдуть за семплами відео в ієрархічному шарі  $i+1$  в періоді. Наприклад, якщо зображеннями часового шару  $0 \in I, P_4, P_8$  тощо, і зображеннями часового шару  $1 \in B_2, B_6$  тощо, проста сигналізація (1, 1) може бути достатньою для семплів відео в двох часових шарах, що підлягають правильному перевпорядкуванню. Інформація про перевпорядкування, що сигналізується, для кожного фрагмента субтреку, тому, може відповідати ідентифікатору фрагмента субтреку і кількості зображень у фрагменті субтреку.

Пристрій 40 призначення потім може визначати один або більше шарів відеофайлу для запиту (218). Пристрій 40 призначення може ґрунтувати це рішення на різних факторах, таких як, наприклад, можливість візуалізації виведення 44 відео, можливість декодування відеодекодера 48, переваги користувача, стани мережі (наприклад, доступна смуга частот), рівні потужності, використання пам'яті, обчислювальна потужність/використання або інші такі фактори. Пристрій 40 призначення потім може запитувати фрагменти субтреку, що відповідають визначеним шарам (220). У деяких прикладах пристрій 40 призначення може використовувати єдиний запит часткового GET HTTP для кожного фрагмента субтреку. Таким чином, пристрій 40 призначення може уникнути витягання необов'язкових відеоданих і може уникнути визначення розташування деякої кількості кодованих семплів у фрагменті фільму, кожний з яких ієрархічно зв'язаний, тобто загального ієрархічного шару.

Пристрій 20 джерела може подавати фрагменти субтреку запиту(-ів) на пристрій 40 призначення (222). Після прийому фрагментів субтреку фрагмента фільму пристрій 40 призначення може перевпорядковувати семпли відео фрагмента фільму, так що семпли відео розміщуються в порядку декодування (224). Потім пристрій 40 призначення може декодувати і відображати прийняті семпли (226).

Фіг.8 являє собою блок-схему послідовності операцій, що ілюструє зразковий спосіб витягання фрагментів субтреку фрагмента фільму. Хоча він описаний відносно компонентів пристрою 40 призначення (фіг.1) для цілей прикладу і пояснення, необхідно розуміти, що будь-який підходящий пристрій може реалізувати методи за фіг.8.

Спочатку пристрій 40 призначення може приймати запит на доступ до відеофайлу (230). Наприклад, користувач може запустити веб-браузер, використовуючи пристрій 40 призначення для запиту URL або URN відеофайлу. У відповідь на цей запит пристрій 40 призначення може завантажити дані заголовка відеофайлу (232). Дані заголовка можуть описувати, як організований відеофайл, і можуть сигналізувати, що відеофайл розміщений відповідно до методів даного розкриття, так що кодовані семпли відео фрагментів фільму розміщуються відповідно до ієрархічних шарів кодованих семплів відео. Дані заголовка додатково можуть описувати кожний з ієрархічних шарів відеофайлу, наприклад, байтові діапазони для фрагментів субтреку у фрагменті фільму. Дані заголовка також можуть вказувати, що фрагменти субтреку фрагментів фільму відеофайлу включають в себе кодовані семпли відео загального ієрархічного шару, як описано в даному розкритті.

Пристрій 40 призначення потім може визначити, який з ієрархічних шарів витягувати (234). Основуючись на цьому визначенні, пристрій 40 призначення може визначити байтові діапазони кожного з фрагментів субтреку, що відповідають ієрархічним шарам, що підлягають витягання (236). Пристрій 40 призначення може продовжувати видавати індивідуальні запити, які задають байтовий діапазон відповідного фрагмента субтреку, що підлягає витягання (238), доки не будуть прийняті всі потрібні фрагменти субтреку (240).

Після прийому всіх потрібних фрагментів субтреку блок 38 демультимплексування пристрою 40 призначення може перевпорядковувати прийняті семпли, так що семпли знаходяться в порядку декодування (242). Блок 38 демультимплексування потім може направити семпли на відеодекодер 48 для декодування, який може направити декодовані семпли відео на вивід 44 відео для відображення (244).

Спосіб за фіг.8 зображає приклад способу, який включає в себе прийом клієнтським пристроєм інформації від пристрою джерела, яка описує ієрархічні рівні відеоданих для фрагмента фільму, визначення піднабору ієрархічних рівнів відеоданих для запиту для кожного ієрархічного рівня піднабору, посилення не більше одного запиту на пристрій джерела для витягання всіх відеоданих фрагмента фільму на ієрархічному рівні, прийом відеоданих визначеного піднабору ієрархічних рівнів і декодування і відображення прийнятих відеоданих.

За допомогою посилання не більше одного запиту на пристрій джерела пристрій призначення може посилати єдиний запит на витягання даних з деякої кількості необхідних ієрархічних шарів або може посилати до одного запиту на потрібний ієрархічний шар.

Фіг.9 являє собою схему концептуального представлення, що ілюструє зразковий шаблон прогнозування MVC. У прикладі на фіг.9 зображено вісім видів (що мають ID виду "S0" - "S7"), і дванадцять часових розташувань ("T0" - "T11") зображені для кожного виду. Тобто кожний ряд на фіг.9 відповідає виду, тоді як кожний стовпець вказує часове розташування.

Хоча MVC має так званий базовий вид, який є декодованим декодерами H.264/AVC, і пара стерео видів може підтримуватися також MVC, перевага MVC полягає в тому, що він може підтримувати приклад, який використовує більше двох видів як ввід 3D-відео і декодує це 3D-відео, представлене багатьма видами. Візуалізатор клієнта, що має декодер MVC, може чекати контент 3D-відео з багатьма видами. Складова опорного виду і складова неопорного виду у виді можуть мати різну залежність виду. Наприклад, складові опорного виду на виді S2 залежать від складових виду на виді S0. Однак складові неопорного виду на виді S2 не залежать від складових виду на інших видах.

Кадри на фіг.9 вказані для кожного ряду і кожного стовпця на фіг.9 з використанням заштрихованих блоків, що включають в себе літеру, що вказує, чи є відповідний кадр внутрішньо кодованим (тобто I-кадром), або зовні кодованим в одному напрямку (тобто P-кадром), або в численних напрямках (тобто B-кадром). Звичайно, прогнозування вказуються стрілками, при цьому кадр, на який виконується вказівка, використовує об'єкт, від якого виконується вказівка, для опорного зображення для прогнозування. Наприклад, P-кадр виду S2 у часовому розташуванні T0 прогнозується з I-кадру виду S0 у часовому розташуванні T0.

Як і з відеокодуванням єдиного виду, кадри відеопослідовності багатовидового відеокодування можуть кодуватися з прогнозуванням відносно кадрів у різних часових розташуваннях. Наприклад, b-кадр виду S0 у часовому розташуванні T1 має стрілку, що вказується на нього від I-кадру виду S0 у часовому розташуванні T0, вказуючи, що b-кадр прогнозується з I-кадру. Додатково, однак в контексті багатовидового відеокодування кадри можуть прогнозуватися міжвидовим чином. Тобто складова виду може використовувати складові виду в інших видах для опорного зображення. У MVC, наприклад, міжвидове прогнозування реалізовується, як якби складова виду в іншому вигляді була опорним зображенням зовнішнього прогнозування. Потенційні міжвидові опорні зображення можуть сигналізуватися в розширенні MVC набору параметрів послідовності (SPS) і можуть модифікуватися процесом складання списку опорних зображень, який дозволяє виконувати гнучке упорядкування опорних зображень зовнішнього прогнозування або міжвидового прогнозування. Таблиця 3 нижче забезпечує зразкове визначення для набору параметрів послідовності розширення MVC.



Таблиця 3

seq_parameter_set_mvc_extension( ) {	C	Дескриптор
num_views_minus1	0	ue(v)
for( i = 0; i <= num_views_minus1; i++ )		
view_id[ i ]	0	ue(v)
for( i = 1; i <= num_views_minus1; i++ ) {		
num_anchor_refs_l0[ i ]	0	ue(v)
for( j = 0; j < num_anchor_refs_l0[ i ]; j++ )		
anchor_ref_l0[ i ][ j ]	0	ue(v)
num_anchor_refs_l1[ i ]	0	ue(v)
for( j = 0; j < num_anchor_refs_l1[ i ]; j++ )		
anchor_ref_l1[ i ][ j ]	0	ue(v)
}		
for( i = 1; i <= num_views_minus1; i++ ) {		
num_non_anchor_refs_l0[ i ]	0	ue(v)
for( j = 0; j < num_non_anchor_refs_l0[ i ]; j++ )		
non_anchor_ref_l0[ i ][ j ]	0	ue(v)
num_non_anchor_refs_l1[ i ]	0	ue(v)
for( j = 0; j < num_non_anchor_refs_l1[ i ]; j++ )		
non_anchor_ref_l1[ i ][ j ]	0	ue(v)
}		
num_level_values_signalled_minus1	0	ue(v)
for( i = 0; i <= num_level_values_signalled_minus1; i++ ) {		
level_idc[ i ]	0	u(8)
num_applicable_ops_minus1[ i ]	0	ue(v)
for( j = 0; j <= num_applicable_ops_minus1[ i ]; j++ ) {		
applicable_op_temporal_id[ i ][ j ]	0	u(3)
applicable_op_num_target_views_minus1[ i ][ j ]	0	ue(v)
for( k = 0; k <=		
applicable_op_num_target_views_minus1[ i ][ j ]; k++ )		
applicable_op_target_view_id[ i ][ j ][ k ]	0	ue(v)
applicable_op_num_views_minus1[ i ][ j ]	0	ue(v)
}		
}		
}		
}		

Фіг.9 забезпечує різні приклади міжвидового прогнозування. Кадри виду S1 в прикладі на фіг.9 зображені як такі, що прогнозуються з кадрів в різних часових розташуваннях виду S1, а також такі, що одержують за допомогою міжвидового прогнозування з кадрів видів S0 та S2 в цих самих часових розташуваннях. Наприклад, b-кадр виду S1 у часовому розташуванні T1 прогнозується з кожного з B-кадрів виду S1 у часових розташуваннях T0 та T2, а також з b-кадрів видів S0 та S2 у часовому розташуванні T1.

У прикладі на фіг.9 велика літера "B" і мала літера "b" призначені, щоб вказувати різну ієрархічну залежність між кадрами, а не різні методології кодування. Загалом, кадри з великою літерою "B" є відносно більш високими в ієрархії прогнозування, ніж кадри з малою літерою "b". Тобто в прикладі на фіг.9 "b"-кадри кодуються з посиланням на "B"-кадри. Можуть бути додані додаткові ієрархічні рівні, що мають додаткові двонаправлено кодовані кадри, які можуть посилатися на "b"-кадри фіг.9. Фіг.9 також ілюструє варіанти в ієрархії прогнозування з використанням різних рівнів штрихування, при цьому кадри з більшою мірою штрихування (тобто відносно темніші) є більш високими в ієрархії прогнозування, ніж ті кадри, які мають менше штрихування (тобто відносно світліші). Наприклад, всі I-кадри на фіг.9 зображені з повним штрихуванням, тоді як P-кадри мають трохи світліше штрихування, і B-кадри (і кадри з малою літерою b) мають різні рівні штрихування відносно один одного, але завжди світліші, ніж штрихування P-кадрів та I-кадрів.

Загалом, ієрархія прогнозування зв'язана з індексами порядку виду так, що кадри відносно більш високі в ієрархії прогнозування повинні декодуватися перед декодуванням кадрів, які є відносно більш низькими в ієрархії, так що ті кадри, які є відносно більш високими в ієрархії, можуть використовуватися як опорні кадри при декодуванні кадрів, які є відносно більш низькими в ієрархії. Індекс порядку виду являє собою індекс, який вказує порядок декодування складових виду в блоці доступу. Індеси порядку виду передбачаються в розширенні MVC SPS, як визначено в Доповненні H до стандарту H.264/AVC (виправлення MVC). У SPS для кожного індексу  $i$  сигналізується відповідний `view_id`. Декодування складових виду повинно дотримуватися в зростаючому порядку індексу порядку виду. Якщо представлені всі види, тоді індеси порядку виду знаходяться в послідовному порядку від 0 до `num_views_minus_1`.

Таким чином, кадри, що використовуються як опорні кадри, можуть декодуватися перед декодуванням кадрів, які кодуються з посиланням на опорні кадри. Індекс порядку виду являє собою індекс, який вказує порядок декодування складових виду в блоці доступу. Для кожного індексу  $i$  порядку виду сигналізується відповідний `view_id`. Декодування складових виду йде в зростаючому порядку індексів порядку виду. Якщо представлені всі види, тоді набір індексів порядку виду містить послідовно впорядкований набір від нуля до повної кількості видів мінус один.

Для деяких кадрів на рівних рівнях ієрархії порядок декодування може не мати значення відносно один одного. Наприклад, I-кадр виду S0 у часовому розташуванні T0 використовується як опорний кадр для P-кадру виду S2 у часовому розташуванні T0, який, в свою чергу, використовується як опорний кадр для P-кадру виду S4 у часовому розташуванні T0. Отже, I-кадр виду S0 у часовому розташуванні T0 повинен декодуватися перед P-кадром виду S2 у часовому розташуванні T0, який повинен декодуватися перед P-кадром виду S4 у часовому розташуванні T0. Однак між видами S1 та S3 не має значення порядок декодування, оскільки види S1 та S3 не ґрунтуються один на одному для прогнозування, але, замість цього, прогнозуються тільки з видів, які є більш високими в ієрархії прогнозування. Крім того, вид S1 може декодуватися перед видом S4, оскільки вид S1 декодується після видів S0 та S2.

Зрозуміло, що може бути ієрархічна залежність між кадрами кожного виду, а також часовими розташуваннями кадрів кожного виду. Що стосується прикладу на фіг.9, кадри у часовому розташуванні T0 є або з внутрішнім прогнозуванням, або з міжвидовим прогнозуванням з кадрів інших видів у часовому розташуванні T0. Аналогічно, кадри у часовому розташуванні T8 є або з внутрішнім прогнозуванням або з міжвидовим прогнозуванням з кадрів інших видів у часовому розташуванні T8. Отже, що стосується часової ієрархії, часові розташування T0 та T8 знаходяться у верху часової ієрархії.

Кадри у часовому розташуванні T4 в прикладі на фіг.9 знаходяться нижче у часовій ієрархії, ніж кадри часових розташувань T0 та T8, оскільки кадри часового розташування T4 кодуються двонаправлено з посиланням на кадри часових розташувань T0 та T8. Кадри у часових розташуваннях T2 та T6 знаходяться нижче у часовій ієрархії, ніж кадри у часовому розташуванні T4. Нарешті, кадри у часових розташуваннях T1, T3, T5 та T7 знаходяться нижче у часовій ієрархії, ніж кадри часових розташувань T2 та T6.

Відповідно до методів даного розкриття кожний з видів, зображений на фіг.9, може розглядатися відповідним відповідному ієрархічному рівню. Методи даного розкриття можуть використовуватися для розділення семплів відео для кожного виду на відповідні фрагменти субтреку. Тобто фрагмент фільму, що включає в себе семпли відео видів 1-N, може складатися так, що семпли виду  $X$  (де  $1 \leq X \leq N$ ) зберігаються в фрагменті субтреку, і семпли можуть зберігатися поряд в фрагменті фільму.

В одному або більше прикладах описані функції можуть бути реалізовані апаратним, програмним, програмно-апаратним забезпеченням або будь-якою їх комбінацією. Якщо вони реалізовані програмним забезпеченням, функції можуть зберігатися або передаватися у вигляді однієї або більше інструкцій або коду на зчитуваному комп'ютером носії, і інструкції можуть виконуватися блоком обробки. Зчитувані комп'ютером носії можуть включати в себе запам'ятовуючі зчитувані комп'ютером носії, які відповідають матеріальному носію, такому як носії для зберігання даних, або середовище зв'язку, що включає в себе будь-яке середовище, яке сприяє перенесенню комп'ютерної програми з одного місця на інше, наприклад, згідно з протоколом зв'язку. Таким чином, зчитувані комп'ютером носії, як правило, можуть відповідати (1) матеріальним зчитуваним комп'ютером запам'ятовуючим носіям, які не є короточасними, або (2) середовищі зв'язку, такому як сигнал або несуча хвиля. Носії для зберігання даних можуть являти собою будь-які доступні носії, до яких можуть звертатися один або більше комп'ютерів або один або більше процесорів для витягання інструкцій, коду і/або структур даних

для реалізації методів, описаних в даному розкритті. Комп'ютерний програмний продукт може включати в себе зчитуваний комп'ютером носій.

Як приклад, але не обмеження, такі зчитувані комп'ютером запам'ятовуючі носії можуть містити RAM, ROM, EEPROM, CD-ROM або інший запам'ятовуючий пристрій на оптичному диску, запам'ятовуючий пристрій на магнітних дисках або інші магнітні запам'ятовуючі пристрої, флеш-пам'ять або будь-який інший носій, який може використовуватися для зберігання необхідного програмного коду у вигляді інструкцій або структур даних, і до якого може звертатися комп'ютер. Також, будь-яке з'єднання потрібно називати зчитуванням комп'ютером носієм. Наприклад, якщо інструкції передаються з веб-сайта, сервера або іншого віддаленого джерела з використанням коаксіального кабелю, волоконно-оптичного кабелю, виті пари, цифрової абонентської лінії (DSL) або бездротових технологій, таких як інфрачервоні, радіочастотні або мікрохвильові, тоді коаксіальний кабель, волоконно-оптичний кабель, вита пара, DSL або бездротові технології, такі як інфрачервоні, радіочастотні і мікрохвильові, включаються у визначення носія. Необхідно розуміти, однак, що зчитувані комп'ютером запам'ятовуючі носії і носії для зберігання даних не включають в себе з'єднання, несучі хвилі, сигнали або інших тимчасові носіїв, але, замість цього, відносяться до не короточасних, матеріальних запам'ятовуючих носіїв. Термін диск, що використовується в даному документі, включає в себе компакт-диск (CD), лазерний диск, оптичний диск, цифровий багатофункціональний диск (DVD), дискету і диск Blu-ray, при цьому магнітні диски звичайно відтворюють дані магнітним чином, тоді як оптичні диски відтворюють дані оптичним чином за допомогою лазерів. Комбінації вищезазначеного також повинні бути включені в обсяг зчитуваних комп'ютером носіїв.

Інструкції можуть виконуватися одним або більше процесорами, такими як один або більше процесорів цифрової обробки сигналів (DSP), мікропроцесорами загального призначення, спеціалізованими інтегральними схемами (ASIC), програмованими вентилями матрицями (FPGA) або іншими еквівалентними інтегральними або дискретними логічними схемами. Отже, термін "процесор", як він використовується в даному документі, може посилатися на будь-яку з вищеописаних конструкцій або будь-яку іншу конструкцію, підходящу для реалізації методів, описаних в даному документі. Крім того, в деяких аспектах функціональна можливість, описана в даному документі, може забезпечуватися в спеціалізованих апаратних і/або програмних модулях, виконаних з можливістю кодування і декодування, або вбудованих в об'єднаний кодек. Також, методи можуть бути повністю реалізовані в одній або більше схемах або логічних елементах.

Методи даного розкриття можуть бути реалізовані в численних апаратах або пристроях, що включають в себе бездротовий переносний телефон, інтегральну схему (IC) або набір IC (наприклад, набір мікросхем). Різні компоненти, модулі або блоки описані в даному розкритті для підкреслення функціональних аспектів пристроїв, виконаних з можливістю виконання описаних методів, але необов'язково вимагають реалізації різними апаратними блоками. Швидше, як описано вище, різні блоки можуть бути об'єднані в апаратний блок кодека або можуть забезпечуватися сукупністю взаємодіючих апаратних блоків, включаючи один або більше процесорів, як описано вище, разом з підходящим програмним і/або апаратним забезпеченням.

Були описані різні приклади. Ці та інші приклади знаходяться в межах обсягу нижченаведеної формули винаходу.

- Посилальні позиції
- 20 пристрій джерела A/V
- 22 джерело аудіо
- 24 джерело відео
- 26 аудіокодер
- 28 відеокодер
- 30 блок інкапсуляції
- 32 інтерфейс виводу для виведення
- 34 зчитуваний комп'ютером носій
- 36 інтерфейс вводу
- 38 блок декапсуляції
- 40 пристрій призначення A/V
- 42 вивід аудіо
- 44 вивід відео
- 46 аудіодекодер
- 48 відеодекодер

- 60 блок створення відеофайлу
- 62 блок конструювання блока рівня мережної абстракції
- 64 блок створення фрагмента субтреку
- 66 блок керування шаром
- 5 68 блок вставки семпла
- 70 блок створення заголовка
- 72 блок створення об'єкта повторного збирача
- 80 інтерфейс вводу відео
- 82 інтерфейс вводу аудіо
- 10 84 інтерфейс виводу відеофайлу
- 100 зразковий відеофайл
- 102 бокс типу файлу
- 104 бокс фільму
- 106 бокс заголовки фільму
- 15 108 бокс треку
- 110 бокс розширення фільму
- 112 бокси фрагмента фільму
- 114 бокс MFRA
- 114 бокс випадкового доступу фрагмента фільму
- 20

## ФОРМУЛА ВИНАХОДУ

1. Спосіб виведення кодованих відеоданих, причому спосіб включає етапи, на яких:  
збирають кодовані відеодані у фрагмент відеофайла, причому фрагмент містить множину  
25 фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно  
зв'язаних кодованих відеозображень кодованих відеоданих, розміщених безперервно у порядку  
декодування в межах відповідного фрагмента субтреку, причому ієрархічно зв'язані кодовані  
відеозображення кожного з фрагментів субтреку відповідають загальному ієрархічному шару  
для відповідного фрагмента субтреку;
- 30 приймають запит відповідно до протоколу потокової передачі, причому запит задає  
щонайменше один з фрагментів субтреку; і  
у відповідь на запит виводять ієрархічно зв'язані кодовані відеозображення щонайменше одного  
з множини фрагментів субтреку.
2. Спосіб за п. 1, який додатково включає етап, на якому сигналізують інформацію  
35 перевпорядкування, яка вказує як перевпорядкувати кодовані відеозображення більше одного з  
фрагментів субтреку у порядок декодування.
3. Спосіб за п. 2, в якому сигналізація інформації перевпорядкування включає етапи, на яких:  
виробляють об'єкт повторного збирача, який посилається на кодоване відеозображення  
40 першого з множини фрагментів субтреку; і  
зберігають об'єкт повторного збирача у другому з множини фрагментів субтреку.
4. Спосіб за п. 3, в якому вироблення об'єкта повторного збирача містить етап, на якому  
виробляють об'єкт повторного збирача для включення значення індексу для першого з множини  
фрагментів субтреку і значення положення, яке вказує положення семпла в першому з множини  
фрагментів субтреку.
- 45 5. Спосіб за п. 2, в якому сигналізація інформації перевпорядкування включає етап, на якому  
сигналізують для кожного фрагмента субтреку ідентифікатор фрагмента субтреку і кількість  
зображень у фрагменті субтреку, причому спосіб додатково включає етап, на якому:  
розміщують фрагменти субтреку, так що фрагменти субтреку дотримуються фіксованого  
шаблону.
- 50 6. Спосіб за п. 1, в якому відеофайл зв'язаний з окремим уніфікованим покажчиком ресурсу  
(URL).
7. Спосіб за п. 6, в якому прийом запиту містить прийом запиту часткового GET протоколу  
передачі гіпертексту (HTTP), який задає URL відеофайла і байтовий діапазон, що відповідає  
щонайменше одному з множини фрагментів субтреку.
- 55 8. Спосіб за п. 6, в якому множина фрагментів субтреку містить перший фрагмент субтреку,  
другий фрагмент субтреку і третій фрагмент субтреку, причому перший фрагмент субтреку  
включає в себе перший набір ієрархічно зв'язаних кодованих відеозображень в першому шарі  
ієрархії, причому другий фрагмент субтреку включає в себе другий набір ієрархічно зв'язаних  
кодованих відеозображень у другому шарі ієрархії, більшому, ніж перший шар, і при цьому  
60 третій фрагмент субтреку включає в себе третій набір ієрархічно зв'язаних кодованих

відеозображень в третьому шарі ієрархії, більшому, ніж перший шар і другий шар, причому прийом запиту містить прийом запиту часткового GET протоколу передачі гіпертексту (HTTP), який задає URL відеофайла і байтовий діапазон, що відповідає щонайменше першому фрагменту субтреку і другому фрагменту субтреку, і при цьому виведення містить виведення першого набору ієрархічно зв'язаних кодованих відеозображень в першому шарі ієрархії і виведення другого набору ієрархічно зв'язаних кодованих відеозображень у другому шарі ієрархії без виведення третього набору ієрархічно зв'язаних кодованих відеозображень в третьому шарі ієрархії.

9. Спосіб за п. 1, в якому відеофайл містить перший відеофайл з множини відеофайлів медіаконтенту, причому перший файл відповідає першому часовому сегменту медіаконтенту, причому спосіб додатково включає етап, на якому збирають кодовані відеодані для медіаконтенту у множину відеофайлів, причому кожний з відеофайлів відповідає відповідному часовому сегменту медіаконтенту.

10. Спосіб за п. 9, в якому кожний з відеофайлів зв'язаний з окремим уніфікованим покажчиком ресурсу (URL) так що, URL для першого відеофайла відрізняється від URL другого відеофайла з множини відеофайлів, причому другий відеофайл відрізняється від першого відеофайла.

11. Спосіб за п. 9, в якому фрагмент першого відеофайла відповідає часовому підсегменту першого часового сегмента, якому відповідає перший відеофайл.

12. Спосіб за п. 1, який додатково включає етап, на якому виробляють заголовок фрагмента субтреку, який задає байтовий діапазон для щонайменше одного з множини фрагментів субтреку.

13. Спосіб за п. 12, який додатково включає етап, на якому виводять заголовок фрагмента субтреку перед виведенням ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку.

14. Спосіб за п. 1, в якому запит містить запит часткового GET протоколу передачі гіпертексту (HTTP), який задає байтовий діапазон, що відповідає щонайменше одному з множини фрагментів субтреку.

15. Спосіб за п. 1, який додатково включає етап, на якому визначають, у фрагменті ініціалізації медіафайла, що включає в себе фрагменти субтреку, щонайменше одну робочу точку, яка відповідає піднабору з множини фрагментів субтреку.

16. Пристрій для виведення кодованих відеоданих, причому пристрій містить: інтерфейс, виконаний з можливістю виведення даних відповідно до протоколу потокової передачі; і

блок керування, виконаний з можливістю збирання кодованих відеоданих у фрагмент відеофайла, причому фрагмент містить множину фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно зв'язаних відеозображень кодованих відеоданих, розміщених безперервно у порядку декодування в межах відповідного фрагмента субтреку, причому ієрархічно зв'язані відеозображення кожного з фрагментів субтреку відповідають загальному ієрархічному шару для відповідного фрагмента субтреку, прийому запиту відповідно до протоколу потокової передачі, причому запит задає щонайменше один з множини фрагментів субтреку, і, у відповідь на запит, розпорядження інтерфейсу виводити ієрархічно зв'язані відеозображення щонайменше одного з фрагментів субтреку.

17. Пристрій за п. 16, в якому блок керування додатково виконаний з можливістю вироблення об'єкта повторного збирача, який посилається на семпл першого з множини фрагментів субтреку, і збереження об'єкта повторного збирача у другому з множини фрагментів субтреку.

18. Пристрій за п. 17, в якому блок керування виконаний з можливістю вироблення об'єкта повторного збирача для включення значення індексу для першого з множини фрагментів субтреку і значення положення, яке вказує положення семпла в першому з множини фрагментів субтреку.

19. Пристрій за п. 16, в якому блок керування виконаний з можливістю вироблення заголовка фрагмента субтреку, який задає байтовий діапазон для щонайменше одного з множини фрагментів субтреку.

20. Пристрій за п. 19, в якому блок керування виконаний з можливістю виведення заголовка фрагмента субтреку перед виведенням множини ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку.

21. Пристрій за п. 16, в якому запит містить запит часткового GET протоколу передачі гіпертексту (HTTP), який задає байтовий діапазон, що відповідає щонайменше одному з множини фрагментів субтреку.

22. Пристрій за п. 16, в якому блок керування виконаний з можливістю визначення, в сегменті ініціалізації медіафайла, що включає в себе фрагменти субтреку, щонайменше однієї робочої точки, яка відповідає піднабору з множини фрагментів субтреку.

23. Пристрій за п. 16, причому пристрій містить щонайменше одне з:

5 інтегральної схеми;

мікропроцесора; і

пристрою бездротового зв'язку, який включає в себе згаданий блок керування.

24. Пристрій для виведення кодованих відеоданих, причому пристрій містить:

10 засіб для збирання кодованих відеоданих у фрагмент відеофайла, причому фрагмент містить множини фрагментів субтреку, причому кожний з фрагментів субтреку містить множини ієрархічно зв'язаних відеозображень кодованих відеоданих, розміщених безперервно у порядку декодування у відповідному

фрагменті субтреку, причому множина ієрархічно зв'язаних відеозображень кожного з фрагментів субтреку відповідає загальному ієрархічному шару для відповідного фрагмента субтреку;

15 засіб для прийому запиту відповідно до протоколу потокової передачі, причому запит задає щонайменше один з множини фрагментів субтреку; і

засіб для виведення множини ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку у відповідь на запит.

20 25. Пристрій за п. 24, який додатково містить:

засіб для вироблення об'єкта повторного збирача, який посилається на семпл першого з множини фрагментів субтреку; і

засіб для збереження об'єкта повторного збирача у другому з множини фрагментів субтреку.

25 26. Пристрій за п. 25, в якому засіб для вироблення об'єкта повторного збирача містить засіб для вироблення об'єкта повторного збирача для включення значення індексу для першого з множини фрагментів субтреку і значення положення, яке вказує положення семпла в першому з множини фрагментів субтреку.

27. Пристрій за п. 24, який додатково містить засіб для вироблення заголовка фрагмента субтреку, який задає байтовий діапазон для щонайменше одного з множини фрагментів субтреку.

30 28. Пристрій за п. 27, який додатково містить засіб для виведення заголовка фрагмента субтреку перед виведенням множини ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку.

35 29. Пристрій за п. 24, в якому запит містить запит часткового GET протоколу передачі гіпертексту (HTTP), який задає байтовий діапазон, що відповідає щонайменше одному з множини фрагментів субтреку.

30. Пристрій за п. 24, який додатково містить засіб для визначення, в сегменті ініціалізації медіафайла, що включає в себе фрагменти субтреку, щонайменше однієї робочої точки, яка відповідає піднабору з множини фрагментів субтреку.

40 31. Зчитуваний комп'ютером запам'ятовуючий носій, що містить інструкції, які зберігаються на ньому, які, при виконанні, приписують процесору пристрою

джерела для виведення кодованих відеоданих:

45 збирати кодовані відеодані у фрагмент відеофайла, причому фрагмент містить множини фрагментів субтреку, причому кожний з фрагментів субтреку містить множини ієрархічно зв'язаних відеозображень кодованих відеоданих, розміщених безперервно у порядку декодування у відповідному фрагменті субтреку, причому множина ієрархічно зв'язаних відеозображень кожного з фрагментів субтреку відповідає загальному ієрархічному шару для відповідного фрагмента субтреку;

50 приймати запит відповідно до протоколу потокової передачі, причому запит задає щонайменше один з множини фрагментів субтреку; і

у відповідь на запит виводити множини ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку.

32. Зчитуваний комп'ютером запам'ятовуючий носій за п. 31, який додатково містить інструкції для:

55 вироблення об'єкта повторного збирача, який посилається на семпл першого з множини фрагментів субтреку; і

збереження об'єкта повторного збирача у другому з множини фрагментів субтреку.

33. Зчитуваний комп'ютером запам'ятовуючий носій за п. 32, в якому інструкції для вироблення об'єкта повторного збирача містять інструкції для вироблення об'єкта повторного збирача для

включення значення індексу для першого з множини фрагментів субтреку і значення положення, яке вказує положення семпла в першому з множини фрагментів субтреку.

34. Зчитуваний комп'ютером запам'ятовуючий носій за п. 31, який додатково містить інструкції для вироблення заголовка фрагмента субтреку, який задає байтовий діапазон для щонайменше одного з множини фрагментів субтреку.

35. Зчитуваний комп'ютером запам'ятовуючий носій за п. 34, який додатково містить інструкції для виведення заголовка фрагмента субтреку перед виведенням множини ієрархічно зв'язаних відеозображень щонайменше одного з множини фрагментів субтреку.

36. Зчитуваний комп'ютером запам'ятовуючий носій за п. 31, в якому запит містить запит часткового GET протоколу передачі гіпертексту (HTTP), який задає байтовий діапазон, що відповідає щонайменше одному з множини фрагментів субтреку.

37. Зчитуваний комп'ютером запам'ятовуючий носій за п. 31, який додатково містить інструкції для визначення, в сегменті ініціалізації медіафайла, що включає в себе фрагменти субтреку, щонайменше однієї робочої точки, яка відповідає піднабору з множини фрагментів субтреку.

38. Спосіб прийому кодованих відеоданих, причому спосіб включає етапи, на яких: приймають інформацію від пристрою джерела, яка описує ієрархічні шари відеоданих для фрагмента фільму відеофайла, причому фрагмент містить множину фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно зв'язаних кодованих відеозображень кодованих відеоданих, розміщених безперервно у порядку декодування в межах відповідного фрагмента субтреку, причому ієрархічно зв'язані кодовані відеозображення кожного з фрагментів субтреку відповідають загальному ієрархічному шару для відповідного фрагмента субтреку; визначають піднабір ієрархічних шарів відеоданих для запиту; для кожного ієрархічного шару піднабору, відповідно до протоколу потокової передачі посилають не більше одного запиту на пристрій джерела на витягування всіх відеоданих згаданого фрагмента фільму на ієрархічних шарах піднабору; приймають відеодані згаданого визначеного піднабору ієрархічних шарів; і декодують і відображають прийняті відеодані.

39. Спосіб за п. 38 в якому інформація, яка описує ієрархічні шари, містить заголовок фрагмента субтреку, який задає для кожного ієрархічного шару байтовий діапазон у фрагменті фільму відеоданих, що відповідає ієрархічному шару.

40. Спосіб за п. 38, в якому прийняті відеодані містять перший фрагмент субтреку для першого ієрархічного шару і другий фрагмент субтреку для другого ієрархічного шару, причому перший фрагмент субтреку включає в себе множину відеозображень, і причому другий фрагмент субтреку включає в себе об'єкт повторного збирача, який посилається на одне з множини відеозображень першого фрагмента субтреку, причому спосіб додатково включає етап, на якому:

розміщують послідовність відеозображень з першого фрагмента субтреку і другого фрагмента субтреку у порядку декодування, використовуючи об'єкт повторного збирача.

41. Спосіб за п. 38, в якому інформація, яка описує ієрархічні шари, містить сегмент ініціалізації медіафайла, що включає в себе відеодані, причому бокс фільму визначає робочу точку, яка відповідає піднабору ієрархічних шарів, і причому визначення піднабору включає етап, на якому вибирають робочу точку.

42. Пристрій для прийому кодованих відеоданих, причому пристрій містить: інтерфейс, виконаний з можливістю прийому інформації від пристрою джерела, яка описує ієрархічні шари відеоданих для фрагмента фільму відеофайла, причому фрагмент містить множину фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно зв'язаних кодованих відеозображень кодованих відеоданих, розміщених безперервно у порядку декодування в межах відповідного фрагмента субтреку, причому ієрархічно зв'язані кодовані відеозображення кожного з фрагментів субтреку відповідають загальному ієрархічному шару для відповідного фрагмента субтреку; і

блок керування, виконаний з можливістю визначення піднабору ієрархічних шарів відеоданих для запиту, причому, для кожного ієрархічного шару піднабору, блок керування виконаний з можливістю посилання не більше одного запиту відповідно до протоколу потокової передачі на пристрій джерела на витягування всіх відеоданих згаданого фрагмента фільму на ієрархічному шарі,

причому інтерфейс виконаний з можливістю прийому відеоданих згаданого визначеного піднабору ієрархічних шарів у відповідь на запити.

43. Пристрій за п. 42, в якому інформація, яка описує ієрархічні шари, містить заголовок фрагмента субтреку, який задає для кожного ієрархічного шару

байтовий діапазон у фрагменті фільму відеоданих, що відповідає ієрархічному шару.

44. Пристрій за п. 42, в якому прийняті відеодані містять перший фрагмент субтреку для першого ієрархічного шару і другий фрагмент субтреку для другого ієрархічного шару, причому перший фрагмент субтреку включає в себе множину відеозображень, і причому другий

фрагмент субтреку включає в себе об'єкт повторного збирача, який посилається на одне з множини відеозображень першого фрагмента субтреку, і причому блок керування виконаний з можливістю розміщення послідовності відеозображень з першого фрагмента субтреку і другого

фрагмента субтреку у порядку декодування з використанням об'єкта повторного збирача.

45. Пристрій за п. 42, в якому інформація, яка описує ієрархічні шари, містить сегмент ініціалізації медіафайла, що включає в себе відеодані, причому бокс фільму визначає робочу точку, яка відповідає піднабору ієрархічних шарів, і причому для визначення піднабору блок керування виконаний з можливістю вибору робочої точки.

46. Пристрій за п. 42, причому пристрій містить щонайменше одне з: інтегральної схеми;

мікропроцесора; і

пристрою бездротового зв'язку, який включає в себе згаданий блок керування.

47. Пристрій для прийому кодованих відеоданих, причому пристрій містить:

засіб для прийому інформації від пристрою джерела, яка описує ієрархічні шари відеоданих для фрагмента фільму відеофайла, причому фрагмент містить множину фрагментів субтреку, причому кожний з фрагментів субтреку містить множину ієрархічно зв'язаних кодованих відеозображень кодованих відеоданих, розміщених безперервно у порядку декодування в межах відповідного фрагмента субтреку, причому ієрархічно зв'язані кодовані відеозображення кожного з фрагментів субтреку відповідають загальному ієрархічному шару для відповідного фрагмента субтреку;

засіб для визначення піднабору ієрархічних шарів відеоданих для запиту;

засіб для посилення для кожного ієрархічного шару піднабору відповідно до протоколу потокової передачі не більше одного запиту на пристрій джерела на витягування всіх відеоданих згаданого фрагмента фільму на ієрархічному шарі;

засіб для прийому відеоданих згаданого визначеного піднабору ієрархічних шарів; і

засіб для декодування і відображення прийнятих відеоданих.

48. Пристрій за п. 47, в якому інформація, яка описує ієрархічні шари, містить заголовок фрагмента субтреку, який задає для кожного ієрархічного шару

байтовий діапазон у фрагменті фільму відеоданих, що відповідає ієрархічному шару.

49. Пристрій за п. 47, в якому прийняті відеодані містять перший фрагмент субтреку для першого ієрархічного шару і другий фрагмент субтреку для другого ієрархічного шару, причому перший фрагмент субтреку включає в себе множину відеозображень, і причому другий фрагмент субтреку включає в себе об'єкт повторного збирача, який посилається на одне з множини відеозображень першого фрагмента субтреку, причому пристрій додатково містить:

засіб для розміщення послідовності відеозображень з першого фрагмента субтреку і другого фрагмента субтреку у порядку декодування з використанням об'єкта повторного збирача.

50. Пристрій за п. 47, в якому інформація, яка описує ієрархічні шари, містить сегмент ініціалізації медіафайла, що включає в себе відеодані, причому бокс фільму визначає робочу точку, яка відповідає піднабору ієрархічних шарів, і причому засіб для визначення піднабору містить засіб для вибору робочої точки.

51. Зчитуваний комп'ютером запам'ятовуючий носій, що містить інструкції, що зберігаються на ньому, які, при виконанні, приписують процесору пристрою для прийому кодованих відеоданих: приймати інформацію від пристрою джерела, яка описує ієрархічні шари

відеоданих для фрагмента фільму відеофайла, причому фрагмент містить множину

фрагментів субтреку, причому кожний з фрагментів субтреку містить множину

ієрархічно зв'язаних кодованих відеозображень кодованих відеоданих, розміщених

безперервно у порядку декодування в межах відповідного фрагмента субтреку,

причому ієрархічно зв'язані кодовані відеозображення кожного з фрагментів

субтреку відповідають загальному ієрархічному шару для відповідного фрагмента субтреку;

визначати піднабір ієрархічних шарів відеоданих для запиту;

для кожного ієрархічного шару піднабору, посилати відповідно до протоколу потокової передачі не більше одного запиту на пристрій джерела на витягування всіх відеоданих згаданого фрагмента фільму на ієрархічному шарі,

приймати відеодані згаданого визначеного піднабору ієрархічних шарів; і



декодувати і відобразити прийняті відеодані.

52. Зчитуваний комп'ютером запам'ятовуючий носій за п. 51, в якому інформація, яка описує ієрархічні шари, містить заголовок фрагмента субтреку, який задає для кожного ієрархічного шару байтовий діапазон у фрагменті фільму відеоданих, що відповідає ієрархічному шару.

53. Зчитуваний комп'ютером запам'ятовуючий носій за п. 51, в якому прийняті відеодані містять перший фрагмент субтреку для першого ієрархічного шару і другий фрагмент субтреку для другого ієрархічного шару, причому перший фрагмент субтреку включає в себе множину відеозображень, і причому другий фрагмент субтреку включає в себе об'єкт повторного збирача, який посилається на одне з множини відеозображень першого фрагмента субтреку, причому зчитуваний комп'ютером запам'ятовуючий носій додатково містить інструкції для: розміщення послідовності відеозображень з першого фрагмента субтреку і другого фрагмента субтреку у порядку декодування з використанням об'єкта повторного збирача.

54. Зчитуваний комп'ютером запам'ятовуючий носій за п. 51, в якому інформація, яка описує ієрархічні шари, містить сегмент ініціалізації медіафайла, що включає в себе відеодані, причому бокс фільму визначає робочу точку, яка відповідає піднабору ієрархічних шарів, і причому інструкції для визначення піднабору містять інструкції для вибору робочої точки.

20

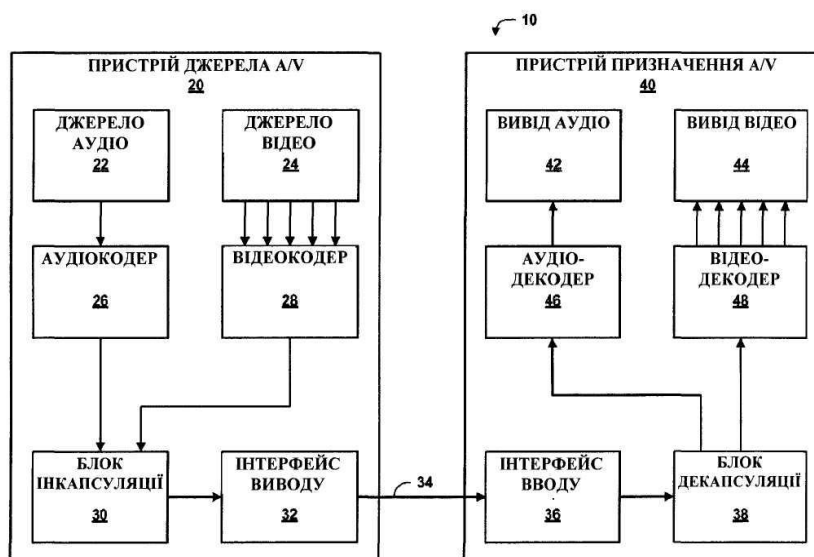
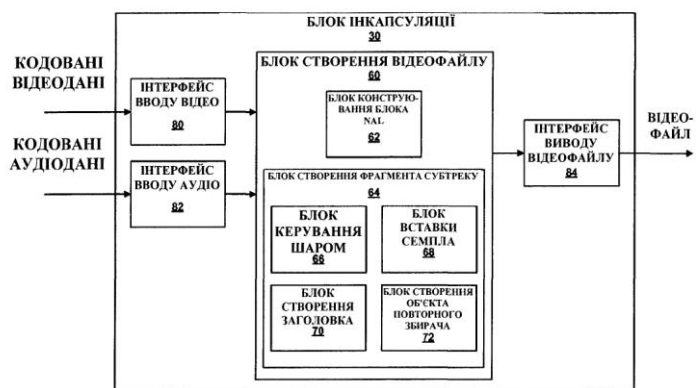
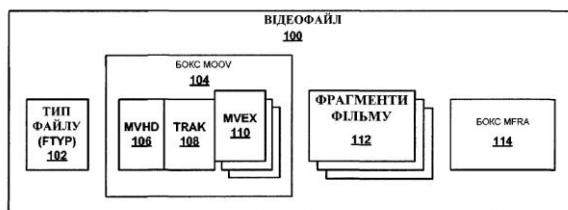


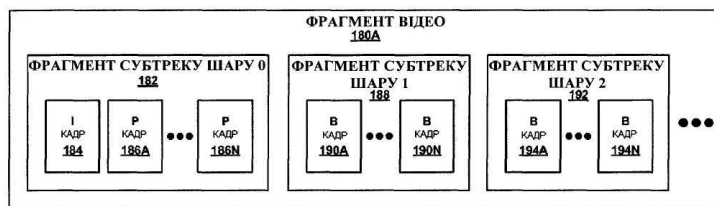
Fig. 1



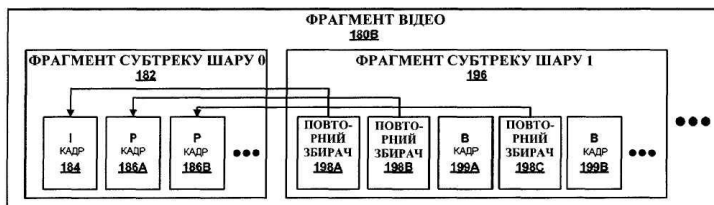
Фіг. 2



Фіг. 3



Фіг. 4A



Фіг. 4B



Fig. 5

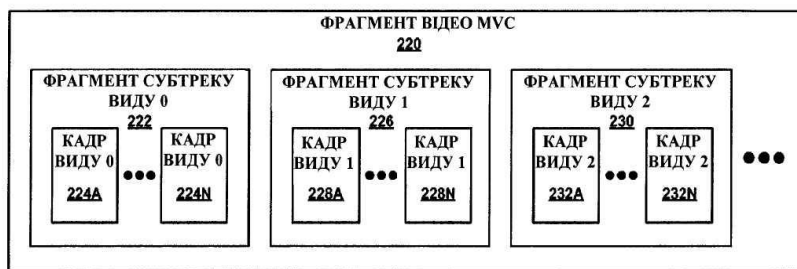


Fig. 6

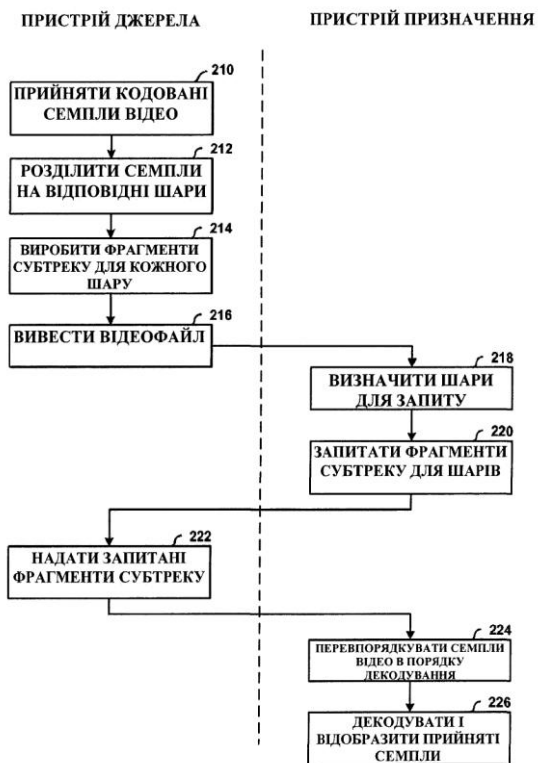
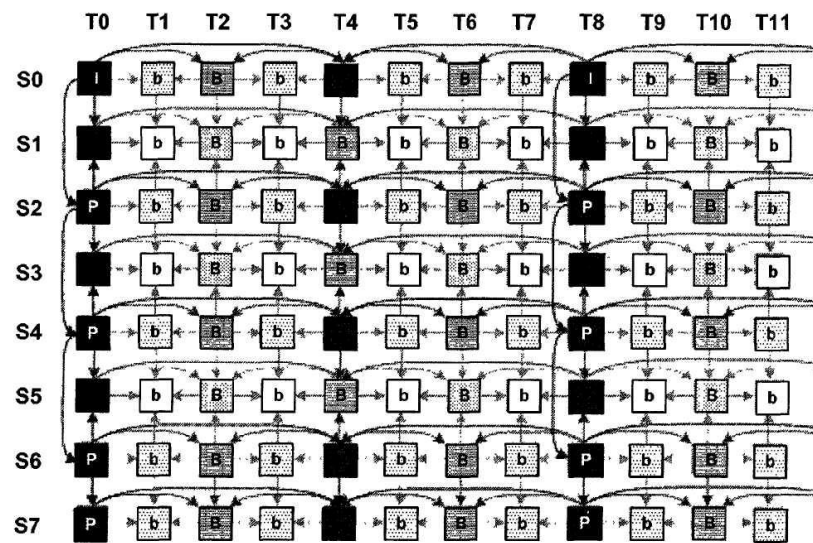


Fig. 7



Фіг. 8



Фиг. 9