

**УКРАЇНА****(19) UA****(11) 113752****(13) C2****(51) МПК****H04N 19/31 (2014.01)****H04N 19/46 (2014.01)****H04N 19/70 (2014.01)**

**ДЕРЖАВНА СЛУЖБА
ІНТЕЛЕКТУАЛЬНОЇ
ВЛАСНОСТІ
УКРАЇНИ**

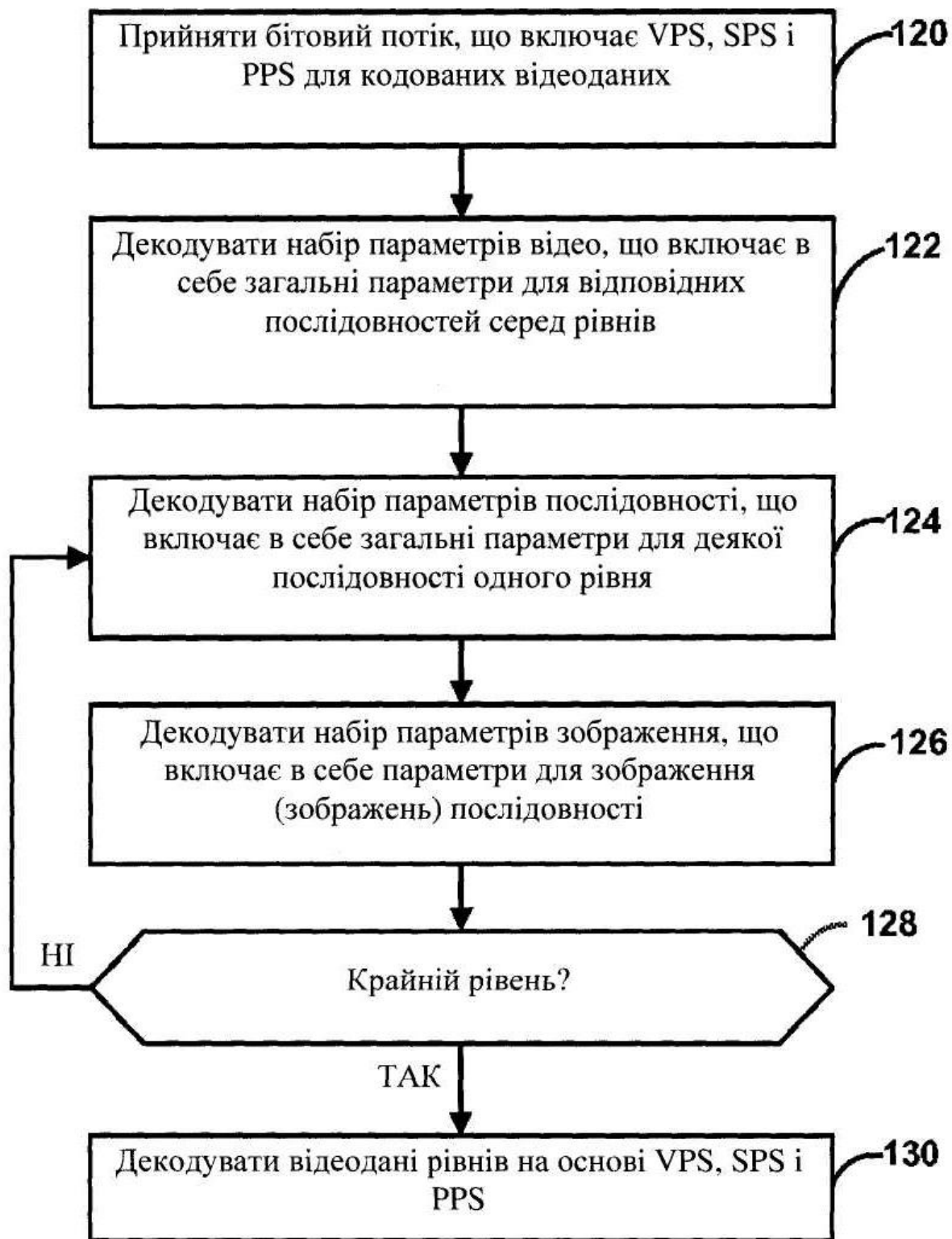
(12) ОПИС ДО ПАТЕНТУ НА ВІНАХІД

(21) Номер заявки:	а 2014 09119	(72) Винахідник(и):	Чень Ін (US), Ван Є-Куй (US)
(22) Дата подання заявки:	11.01.2013	(73) Власник(и):	КВЕЛКОММ ІНКОРПОРЕЙТЕД, Attn: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121-1714, United States of America (US)
(24) Дата, з якої є чинними права на винахід:	10.03.2017	(74) Представник:	Мошинська Ніна Миколаївна, реєстр. №115
(31) Номер попередньої заявки відповідно до Паризької конвенції:	61/586,777, 61/587,070, 61/588,629, 61/637,195, 61/637,774, 13/738,377	(56) Перелік документів, взятих до уваги експертизою:	Rickard Sjöberg. On num reorder frames and max dec frame buffering. 5th Meeting, Geneva, CH, 16.03.2011-23.03.2011. Joint Collaborative Team on Video Coding (JCT- VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-E339, WG11 Number: m19867, XP 030008845 US 2005254575 A1, 17.11.2005 BROSS B. et all. High Efficiency Video Coding (HEVC) text specification Working Draft 5., 7. JCT-VC MEETING. 98. MPEG MEETING, 21.11.2011 - 30.11.2011. GENEVA. JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16. JCTVC-G1103, 19.12.2011, XP030111032
(32) Дата подання попередньої заявки відповідно до Паризької конвенції:	14.01.2012, 16.01.2012, 19.01.2012, 23.04.2012, 24.04.2012, 10.01.2013		
(33) Код держави-учасниці Паризької конвенції, до якої подано попередню заявку:	US, US, US, US, US, US		
(41) Публікація відомостей про заявку:	10.11.2014, Бюл.№ 21		
(46) Публікація відомостей про видачу патенту:	10.03.2017, Бюл.№ 5		
(86) Номер та дата подання міжнародної заявки, поданої відповідно до Договору РСТ	PCT/US2013/021227, 11.01.2013		

(54) КОДУВАННЯ НАБОРІВ ПАРАМЕТРІВ І ЗАГОЛОВКІВ ОДИНИЦЬ NAL ДЛЯ КОДУВАННЯ ВІДЕО**(57) Реферат:**

У одному прикладі кодер відео, наприклад відеокодер або відеодекодер, конфігурується для кодування набору параметрів відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з одного або більше рівнів відеоданих посилається на VPS, і кодування одного або більше рівнів відеоданих, основуючись, щонайменше частково, на VPS. Кодер відео може кодувати VPS для відеоданих, відповідних Високоєфективному кодуванню відео, Багатовидовому кодуванню відео, Масштабованому кодуванню відео або іншим стандартам кодування відео, або розширенням стандартів кодування відео. VPS може включати в себе дані, що задають параметри для відповідних послідовностей відеоданих на різних рівнях (наприклад, видах, рівнях якості або т. п.). Параметри VPS можуть надавати указання про те, як кодуються відповідні відеодані.

UA 113752 C2



Фіг. 8

Дана заявка вимагає пріоритет:

попередньої заявки США з порядковим № 61/586,777, поданої 14 січня 2012 р.; попередньої заявки США з порядковим № 61/587,070, поданої 16 січня 2012 р.; попередньої заявки США з порядковим № 61/588,629, поданої 19 січня 2012 р.; попередньої заявки США з порядковим № 61/637,195, поданої 23 квітня 2012 р.; і попередньої заявки США з порядковим № 61/637,774, поданої 24 квітня 2012 р., кожна з яких повністю включена в цей документ шляхом посилання.

ГАЛУЗЬ ТЕХНІКИ

Дане розкриття винаходу стосується кодування відео.

РІВЕНЬ ТЕХНІКИ

Можливості цифрового відео можуть вбудовуватися в широкий діапазон пристроїв, включаючи цифрові телевізори, системи цифрового прямого мовлення, системи бездротового мовлення, персональні цифрові помічники (PDA), переносні або настільні комп'ютери, планшетні комп'ютери, електронні книги, цифрові камери, цифрові пристрої запису, цифрові мультимедійні програвачі, відеоігрові пристрої, ігрові приставки, стільникові або супутникові радіотелефони, так звані "смартфони", пристрої для відеоконференцзв'язку, пристрої для потокового відео і т. п. Цифрові відеопристрої реалізують методики кодування відео, наприклад, описані в стандартах, заданих MPEG-2, MPEG-4, H.263 ITU-T, H.264/MPEG-4 ITU-T, частина 10, Поліпшене кодування відео (AVC), стандартом Високоєфективного кодування відео (HEVC), що на даний час знаходиться в розробці, і розширеннях таких стандартів. Останній варіант майбутнього стандарту HEVC доступний за адресою http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G1103-v3.zip. Відеопристрої можуть ефективніше передавати, приймати, кодувати, декодувати і/або зберігати цифрову відеоінформацію за допомогою реалізації таких методик кодування відео.

Методики кодування відео включають в себе просторове (всередині (intra) зображення) прогнозування і/або часове (між (inter) зображеннями) прогнозування для зменшення або усунення надмірності, властивої відеопослідовностям. Для блокового кодування відео слайс (вирізка) відео (наприклад, відеокадр або частина відеокадру) може розбиватися на відеоблоки, які також можуть називатися блоками дерева, одиницями кодування (CU) і/або вузлами кодування. Відеоблоки в слайсі з внутрішнім кодуванням (I) зображення кодуються з використанням просторового прогнозування відносно опорних вибірок (відліків) в сусідніх блоках в тому ж зображенні. Відеоблоки в слайсі із зовнішнім кодуванням (P або B) зображення можуть використовувати просторове прогнозування відносно опорних вибірок в сусідніх блоках в тому ж зображенні або часове прогнозування відносно опорних вибірок в інших опорних зображеннях. Зображення можуть називатися кадрами, а опорні зображення можуть називатися опорними кадрами.

Просторове або часове прогнозування приводить до блока з прогнозуванням для блока, який повинен бути кодований. Залишкові дані являють собою різниці пікселів між вихідним блоком, який повинен бути кодований, і блоком з прогнозуванням. Блок із зовнішнім кодуванням кодується відповідно до вектора руху, який вказує на блок опорних вибірок, що утворюють блок з прогнозуванням, і залишкових даних, що вказують різницю між кодованим блоком і блоком з прогнозуванням. Блок з внутрішнім кодуванням кодується відповідно до режиму внутрішнього кодування і залишкових даних. Для додаткового стиснення залишкові дані можуть бути перетворені з області пікселів в область перетворення, що приводить до залишкових коефіцієнтів перетворення, які потім можна квантувати. Квантовані коефіцієнти перетворення, організовані спочатку в двовимірний масив, можна сканувати для створення одновимірного вектора коефіцієнтів перетворення, і може застосовуватися ентропійне кодування для досягнення ще більшого стиснення.

СУТЬ ВІНАХОДУ

Загалом дане розкриття винаходу описує методики для кодування наборів параметрів і одиниць рівня абстракції мережі (NAL) для кодування відео. Ці методики можуть застосовуватися до однорівневих кодованих даних, наприклад двовимірних відеоданих, а також до відеоданих масштабованого кодування відео (SVC) і відеоданих багатовидового кодування відео (MVC). Таким чином, набори параметрів і одиниці NAL можуть бути взаємно сумісні між різними типами відеоданих. Наприклад, кодер відео, такий як відеокодер або відеодекодер, може кодувати набір параметрів відео (VPS), який задає параметри для одного або більше рівнів відеоданих. Рівні можуть відповідати, наприклад, рівням SVC (що мають різні частоти кадрів, просторові розрізнення і/або рівні якості) і/або видам в даних MVC (наприклад, послідовностям зображень сцени, захоплених з різних ракурсів камер відносно горизонтальної осі).

У одному прикладі спосіб кодування відеоданих включає в себе кодування набору параметрів відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з одного або більше рівнів відеоданих посиляється на VPS, і кодування одного або більше рівнів відеоданих, основуючись щонайменше частково на VPS.

5 У іншому прикладі пристрій для кодування відеоданих включає в себе кодер відео, наприклад відеокодер або відеодекодер, який конфігурується для кодування набору параметрів відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з одного або більше рівнів відеоданих посиляється на VPS, і кодування одного або більше рівнів відеоданих, основуючись щонайменше частково на VPS.

10 У іншому прикладі пристрій для кодування відеоданих включає в себе засіб для кодування набору параметрів відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з одного або більше рівнів відеоданих посиляється на VPS, і засіб для кодування одного або більше рівнів відеоданих, основуючись щонайменше частково на VPS.

15 У іншому прикладі зчитуваний комп'ютером носій інформації містить збережені на ньому інструкції, які при виконанні спонукають процесор кодувати набір параметрів відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з одного або більше рівнів відеоданих посиляється на VPS, і кодувати один або більше рівнів відеоданих, основуючись щонайменше частково на VPS.

20 Подобиці одного або більше прикладів викладаються на прикладених кресленнях і в описі нижче. Інші ознаки, цілі і переваги стануть очевидні з опису, креслень і з формули винаходу.

КОРОТКИЙ ОПИС КРЕСЛЕНЬ

Фіг. 1 - блок-схема, що ілюструє зразкову систему кодування і декодування відео, яка може використовувати методики для кодування наборів параметрів і одиниць рівня абстракції мережі (NAL) для одного або більше рівнів відеоданих.

25 Фіг. 2 - блок-схема, що ілюструє приклад відеокодера 20, який може реалізувати методики для кодування наборів параметрів і одиниць NAL для одного або більше рівнів відеоданих.

Фіг. 3 - блок-схема, що ілюструє приклад відеодекодера 30, який може реалізувати методики для кодування наборів параметрів і одиниць NAL для одного або більше рівнів відеоданих.

Фіг. 4 - концептуальна схема, що ілюструє зразковий шаблон прогнозування MVC.

30 Фіг. 5 - концептуальна схема, що ілюструє набір параметрів відео (VPS) і різні набори параметрів рівнів (LPS).

Фіг. 6 - концептуальна схема, що ілюструє зразковий групувальний набір параметрів (GPS) і взаємозв'язку GPS з іншими наборами параметрів і заголовками слайсів.

35 Фіг. 7 - блок-схема послідовності операцій, що ілюструє зразковий спосіб для кодування відеоданих відповідно до методик з даного розкриття винаходу.

Фіг. 8 - блок-схема послідовності операцій, що ілюструє зразковий спосіб для декодування відеоданих відповідно до методик з даного розкриття винаходу.

40 Фіг. 9 - блок-схема послідовності операцій, що ілюструє зразковий спосіб кодування відеоданих, основуючись щонайменше частково на числі часових рівнів, які сигналізовані в VPS.

Фіг. 10 - блок-схема послідовності операцій, що ілюструє зразковий спосіб кодування відеоданих, основуючись щонайменше частково на числі зображень, які повинні бути переупорядковані на одному або більше рівнях, і зображень, які повинні бути збережені в буфер декодованих зображень.

45 Фіг. 11 - блок-схема послідовності операцій, що ілюструє зразковий спосіб кодування відеоданих, основуючись щонайменше частково на параметрах гіпотетичного еталонного декодера (HRD), сигналізованих в VPS.

Фіг. 12 - блок-схема послідовності операцій, що ілюструє зразковий спосіб кодування відеоданих, основуючись щонайменше частково на даних розширення, сигналізованих в VPS.

50 ДОКЛАДНИЙ ОПИС

Взагалі дане розкриття винаходу описує кодування відеоданих з використанням набору параметрів відео (VPS). Відеодані можна категоризувати ієрархічно як такі, що включають в себе множину рівнів, послідовність зображень на заданому рівні, зображення в послідовності, слайси в зображенні і блоки (наприклад, макроблоки або одиниці дерева кодування) в слайсі. Набори параметрів послідовності (SPS) можуть використовуватися для сигналізації параметрів, що нечасто змінюються, для послідовності зображень, а набори параметрів зображення (PPS) можуть використовуватися для сигналізації параметрів, що нечасто змінюються, для окремих зображень.

60 Відповідно до методик з даного розкриття винаходу VPS може сигналізувати параметри, що нечасто змінюються, для множини послідовностей між відповідними рівнями. Тобто VPS може

включати в себе параметри для набору суміщених у часі послідовностей різних рівнів. Різні рівні можуть включати в себе, наприклад, різні види для багатовидових відеоданих, різні рівні якості, різні рівні просторового розрізнення, масштабовані у часі рівні (тобто рівні, що допускають різні частоти кадрів) і т. п. Таким чином, один VPS може надаватися для множини різних рівнів, так

що VPS сигналізує параметри, які є загальними для кожного з відповідних рівнів (наприклад, відповідні послідовності на відповідних рівнях). Бітовий потік може вважатися таким, що включає в себе кожний з множини рівнів, і відповідні рівні можуть утворювати відповідні бітові субпотіки. Крім того, бітовий субпотік може відповідати поєднанню двох або більше рівнів.

Дане розкриття винаходу описує різні приклади даних, які можуть включатися в VPS. Такі дані в деяких прикладах можуть включати в себе указування числа підрівнів (наприклад, максимальне число підрівнів) на відповідних рівнях. Наприклад, VPS може включати в себе дані, які сигналізують число часових рівнів і/або максимальне число часових рівнів (наприклад, ідентифікатор найвищого часового рівня).

Як інший приклад VPS додатково або альтернативно може включати в себе дані, практично аналогічні будь-яким даним, раніше сигналізованим в SPS (тобто сигналізованим в традиційних SPS). Таким чином, коли послідовності двох або більше рівнів бітового потоку включають в себе практично аналогічні або ідентичні параметри, кодер відео може кодувати VPS для сигналізації параметрів для послідовностей цих рівнів замість надмірного кодування таких даних у відповідних SPS для різних послідовностей між різними рівнями.

VPS додатково або альтернативно може включати в себе дані, що задають інформацію про застосовність відео (VUI), наприклад інформацію про представлення відео, параметри гіпотетичного еталонного декодера (HRD) і/або інформацію про обмеження бітового потоку. Інформація про обмеження бітового потоку може включати в себе обмеження по діапазону векторів руху, розміру буфера декодованих зображень (DPB) (наприклад, в показниках числа зображень, які повинні бути збережені за допомогою DPB), числу переупорядковуваних кадрів (тобто указування числа кадрів, які повинні бути переупорядковані з порядку декодування в порядок відображення), кодованих розмірах блоків (наприклад, макроблоків (MB) або одиниць дерева кодування) і кодованих розмірах зображень. VPS може додатково надавати дані для одного або більше розширень VPS, так що VPS може розширятися майбутніми стандартами або розширеннями до майбутнього стандарту HEVC.

Фіг. 1 - блок-схема, що ілюструє зразкову систему 10 кодування і декодування відео, яка може використовувати методики для кодування наборів параметрів і одиниць рівня абстракції мережі (NAL) для одного або більше рівнів відеоданих. Як показано на фіг. 1, система 10 включає в себе пристрій-джерело 12, що надає кодовані відеодані, які пізніше повинні бути декодовані пристроєм-адресатом 14. Зокрема, пристрій-джерело 12 надає відеодані пристрою-адресату 14 за допомогою зчитуваного комп'ютером носія 16. Пристрій-джерело 12 і пристрій-адресат 14 можуть бути виконані у вигляді будь-якого з широкого діапазону пристроїв, що включають в себе настільні комп'ютери, блокнотні (тобто переносні) комп'ютери, планшетні комп'ютери, телевізійні приставки, телефонні трубки, наприклад так звані "інтелектуальні" телефони, так звані "інтелектуальні" планшети, телевізори, камери, пристрої відображення, цифрові мультимедійні програвачі, ігрові приставки, пристрій для потокового відео або т. п. В деяких випадках пристрій-джерело 12 і пристрій-адресат 14 можуть бути обладнані для бездротового зв'язку.

Пристрій-адресат 14 може приймати кодовані відеодані, які повинні бути декодовані, за допомогою зчитуваного комп'ютером носія 16. Зчитуваний комп'ютером носій 16 може бути виконаний у вигляді будь-якого типу носія або пристрою, що допускає переміщення кодованих відеоданих від пристрою-джерела 12 до пристрою-адресата 14. У одному прикладі зчитуваний комп'ютером носій 16 може бути виконаний у вигляді засобу зв'язку, щоб дати пристрою-джерелу 12 можливість передавати кодовані відеодані безпосередньо до пристрою-адресата 14 в реальному масштабі часу. Кодовані відеодані можуть модулюватися згідно зі стандартом зв'язку, наприклад протоколом бездротового зв'язку, і передаватися пристрою-адресату 14. Засіб зв'язку може бути виконаний у вигляді будь-якого засобу бездротового або дротового зв'язку, наприклад радіочастотного (RF) спектра або однієї або більше фізичних ліній передачі. Засіб зв'язку може утворювати частину пакетної мережі, наприклад локальної мережі, регіональної мережі або глобальної мережі, такої як Інтернет. Засіб зв'язку може включати в себе маршрутизатори, комутатори, базові станції або будь-яке інше обладнання, яке може бути корисне для спрощення зв'язку від пристрою-джерела 12 до пристрою-адресата 14.

У деяких прикладах кодовані дані можуть виводитися з інтерфейсу 22 виведення в запам'ятовуючий пристрій. Аналогічним чином до кодованих даних можна звертатися із запам'ятовуючого пристрою за допомогою інтерфейсу введення. Запам'ятовуючий пристрій

може включати в себе будь-який з ряду розподілених або локально доступних носіїв інформації, наприклад жорсткий диск, диски Blu-ray, DVD, CD-ROM, флеш-пам'ять, енергозалежний або енергонезалежний запам'ятовуючий пристрій або будь-які інші придатні цифрові носії інформації для зберігання кодованих відеоданих. У додатковому прикладі запам'ятовуючий пристрій може відповідати файловому серверу або іншому проміжному запам'ятовуючому пристрою, який може зберігати кодоване відео, сформоване пристроєм-джерелом 12. Пристрій-адресат 14 може звертатися до збережених відеоданих із запам'ятовуючого пристрою за допомогою потокової передачі або завантаження. Файловий сервер може бути будь-яким типом сервера, що допускає зберігання кодованих відеоданих і передачу цих кодованих відеоданих пристрою-адресату 14. Зразкові файлові сервери включають в себе веб-сервер (наприклад, для веб-сайта), FTP-сервер, мережеві пристрої зберігання (NAS) або локальний накопичувач на дисках. Пристрій-адресат 14 може звертатися до кодованих відеоданих за допомогою будь-якого стандартного інформаційного з'єднання, включаючи Інтернет-з'єднання. Це з'єднання може включати в себе радіоканал (наприклад, з'єднання Wi-Fi), дротове з'єднання (наприклад, DSL, кабельний модем і т. п.) або поєднання їх обох, яке придатне для звертання до кодованих відеоданих, збережених на файловому сервері. Передача кодованих відеоданих із запам'ятовуючого пристрою може бути потоковою передачею, завантаженням або їх поєднанням.

Методики з даного розкриття винаходу не обов'язково обмежуються бездротовими додатками або настройками. Методики можуть застосовуватися до кодування відео в підтримання будь-якого з ряду мультимедійних додатків, таких як ефірні телепередачі, кабельні телепередачі, супутникові телепередачі, передачі потокового відео по Інтернету, наприклад динамічна адаптивна потокова передача по HTTP (DASH), цифрове відео, яке кодується на носії інформації, декодування цифрового відео, збереженого на носії інформації, або інших додатків. У деяких прикладах система 10 може конфігуруватися для підтримання однонаправленої або двонаправленої передачі відео для підтримання таких додатків, як потокова передача відео, відтворення відео, телевізійне мовлення і/або відеотелефонія.

У прикладі з фіг. 1 пристрій-джерело 12 включає в себе джерело 18 відео, відеокодер 20 і інтерфейс 22 виведення. Пристрій-адресат 14 включає в себе інтерфейс 28 введення, відеокодер 30 і пристрій 32 відображення. Відповідно до даного розкриття винаходу відеокодер 20 в пристрої-джерелі 12 може конфігуруватися для застосування методик для кодування наборів параметрів і одиниць NAL для одного або більше рівнів відеоданих. У інших прикладах пристрій-джерело і пристрій-адресат можуть включати в себе інші компоненти або компонування. Наприклад, пристрій-джерело 12 може приймати відеодані із зовнішнього джерела 18 відео, наприклад зовнішньої камери. Також пристрій-адресат 14 може взаємодіяти із зовнішнім пристроєм відображення замість включення в себе вбудованого пристрою відображення.

Проілюстрована система 10 з фіг. 1 є усього лише одним прикладом. Методики для кодування наборів параметрів і одиниць NAL для одного або більше рівнів відеоданих можуть виконуватися будь-яким пристроєм кодування і/або декодування цифрового відео. Хоч методики з даного розкриття винаходу виконуються, як правило, пристроєм кодування відео, методики також можуть виконуватися відеокодером/декодером, звичайно називаним "кодеком". Крім того, методики з даного розкриття винаходу також можуть виконуватися препроцесором відео. Пристрій-джерело 12 і пристрій-адресат 14 є усього лише прикладами таких пристроїв кодування, в яких пристрій-джерело 12 формує кодовані відеодані для передачі пристрою-адресату 14. У деяких прикладах пристрої 12, 14 можуть працювати практично симетричним чином, так що кожен з пристроїв 12, 14 включає в себе компоненти кодування і декодування відео. Тому система 10 може підтримувати однонаправлену або двонаправлену передачу відео між відеопристроями 12, 14, наприклад, для потокової передачі відео, відтворення відео, телевізійного мовлення або відеотелефонії.

Джерело 18 відео в пристрої-джерелі 12 може включати в себе пристрій відеозахоплення, наприклад відеокамеру, відеоархів, що містить раніше захоплене відео, і/або інтерфейс джерела відеосигналу для прийому відео від постачальника відеоконтенту. Як додаткова альтернатива джерело 18 відео може формувати дані на основі комп'ютерної графіки як вихідне відео або поєднання "реального" відео, архівного відео і сформованого комп'ютером відео. У деяких випадках, якщо джерелом 18 відео є відеокамера, то пристрій-джерело 12 і пристрій-адресат 14 можуть утворювати так звані камерофони або відеотелефони. Однак, як згадувалося вище, описані в даному розкритті винаходу методики можуть бути застосовні до кодування відео загалом і можуть застосовуватися до бездротових і/або дротових додатків. У кожному випадку захоплене, попередньо захоплене або сформоване комп'ютером відео може

кодуватися відеокодером 20. Кодована відеоінформація потім може виводитися за допомогою інтерфейсу 22 виведення на зчитуваний комп'ютером носій 16.

Зчитуваний комп'ютером носій 16 може включати в себе тимчасові носії, наприклад бездротову широкомовну або дротову мережеву передачу, або носії інформації (тобто постійні носії інформації), наприклад жорсткий диск, флеш-накопичувач, компакт-диск, цифровий відеодиск, диск Blu-ray або інші зчитувані комп'ютером носії. У деяких прикладах мережевий сервер (не показаний) може приймати кодовані відеодані від пристрою-джерела 12 і надавати кодовані відеодані пристрою-адресату 14, наприклад, за допомогою мережевої передачі. Аналогічним чином обчислювальний пристрій з обладнання по виробництву носіїв, наприклад обладнання по штампуванню оптичних дисків, може прийняти кодовані відеодані від пристрою-джерела 12 і виробити диск, що містить кодовані відеодані. Тому в різних прикладах зчитуваний комп'ютером носій 16 може вважатися таким, що включає в себе один або більше зчитуваних комп'ютером носіїв різних видів.

Інтерфейс 28 введення в пристрої-адресаті 14 приймає інформацію зі зчитуваного комп'ютером носія 16. Інформація зі зчитуваного комп'ютером носія 16 може включати в себе синтаксичну інформацію, задану відеокодером 20, яка також використовується відеодекодером 30, що включає в себе синтаксичні елементи, які описують характеристики і/або обробку блоків і інших кодованих одиниць, наприклад GOP. Пристрій 32 відображення відображає користувачу декодовані відеодані і може бути виконаний у вигляді будь-якого з ряду пристроїв відображення, таких як електронно-променева трубка (CRT), рідкокристалічний дисплей (LCD), плазмовий дисплей, дисплей на органічних світлодіодах (OLED) або інший тип пристрою відображення.

Відеокодер 20 і відеодекодер 30 можуть працювати відповідно до деякого стандарту кодування відео, наприклад стандарту Високоєфективного кодування відео (HEVC), що на даний час знаходиться в розробці, і можуть відповідати Експериментальній моделі HEVC (HM). Як альтернатива відеокодер 20 і відеодекодер 30 можуть працювати відповідно до інших власних або промислових стандартів, наприклад стандарту H.264 ITU-T, як альтернатива названого MPEG-4, частина 10, Поліпшене кодування відео (AVC), або розширень таких стандартів. Однак методики в даному розкритті винаходу не обмежуються ніяким конкретним стандартом кодування. Інші приклади стандартів кодування відео включають в себе MPEG-2 і H.263 ITU-T. Хоч і не показано на фіг. 1, в деяких особливостях відеокодер 20 і відеодекодер 30 можуть бути об'єднані зі звуковим кодером і декодером і можуть включати в себе додатні модулі мультимплексування-демультиплексування або інші апаратні засоби і програмне забезпечення, щоб кодувати звук і відео в загальному потоці даних або в окремих потоках даних. Якщо застосовно, то модулі мультимплексування-демультиплексування можуть відповідати протоколу мультимплексора H.223 ITU або іншим протоколам, наприклад протоколу дейтаграм користувача (UDP).

Стандарт H.264/MPEG-4 (AVC) ITU-T був сформульований Експертною групою в галузі кодування відео (VCEG) ITU-T разом з Експертною групою по рухомих зображеннях (MPEG) ISO/IEC як результат колективної співпраці, відомої як Об'єднана команда по відео (JVT). У деяких особливостях описані в даному розкритті винаходу методики можуть застосовуватися до пристроїв, які загалом відповідають стандарту H.264. Стандарт H.264 описується Дослідницькою групою ITU-T в Рекомендації H.264 ITU-T, Поліпшене кодування відео для універсальних аудіовізуальних служб, датований березнем 2005 року і в цьому документі може називатися стандартом H.264 або специфікацією H.264 або стандартом або специфікацією H.264/AVC. Об'єднана команда по відео (JVT) продовжує працювати над розширеннями до H.264/MPEG-4 (AVC).

Відеокодер 20 і відеодекодер 30 можуть бути реалізовані у вигляді будь-якої з ряду додатних схем кодера, наприклад одного або більше мікропроцесорів, цифрових процесорів сигналів (DSP), спеціалізованих інтегральних схем (ASIC), програмованих користувачем вентильних матриць (FPGA), дискретної логіки, програмного забезпечення, апаратних засобів, мікропрограмного забезпечення або будь-яких їх поєднань. Коли методики реалізуються частково в програмному забезпеченні, пристрій може зберігати інструкції для програмного забезпечення на додатному, незмінному з часом зчитуваному комп'ютером носії і виконувати інструкції на апаратних засобах, що використовують один або більше процесорів, для виконання методик з даного розкриття винаходу. Кожний з відеокодера 20 і відеодекодера 30 може включатися в один або більше кодерів або декодерів, будь-який з яких може вбудовуватися як частина об'єднаного кодера/декодера (кодека) у відповідному пристрої.

JCT-VC працює над розвитком стандарту HEVC. Роботи по стандартизації HEVC основуються на моделі пристрою кодування відео, що розвивається, називаний

Експериментальною моделлю HEVC (HM). HM передбачає декілька додаткових можливостей у пристроїв кодування відео в порівнянні з існуючими пристроями, відповідними, наприклад, H.264/AVC ITU-T. Наприклад, тоді як H.264 надає дев'ять режимів кодування з внутрішнім прогнозуванням, HM може надати цілих тридцять три режими кодування з внутрішнім прогнозуванням.

Взагалі, робоча модель HM описує, що відеокадр або зображення можна розділити на послідовність блоків дерева або найбільших одиниць кодування (LCU), які включають в себе вибірки яскравості і кольоровості. Синтаксичні дані в бітовому потоці можуть задавати розмір для LCU, яка є найбільшою одиницею кодування в показниках числа пікселів. Слайс включає в себе деяке число послідовних блоків дерева в порядку кодування. Відеокадр або зображення може розбиватися на один або більше слайсів. Кожний блок дерева можна розділити на одиниці кодування (CU) згідно з квадродеревом. Звичайно структура даних квадродереву включає в себе один вузол на CU, причому кореневий вузол відповідає блоку дерева. Якщо CU розділяється на чотири суб-CU, то вузол, відповідний CU, включає в себе чотири листи, кожен з яких відповідає одній з суб-CU.

Кожний вузол в структурі даних квадродереву може надавати синтаксичні дані для відповідної CU. Наприклад, вузол в квадродереві може включати в себе прапор розділення, що вказує, чи розділяється відповідна вузлу CU на суб-CU. Синтаксичні елементи для CU можна задати рекурсивно, і вони можуть залежати від того, чи розділяється CU на суб-CU. Якщо CU більше не розділяється, то вона називається листовою CU. У даному розкритті винаходу чотири суб-CU в листовій CU також будуть називатися листовими CU, навіть якщо відсутнє явне розділення вихідної листової CU. Наприклад, якщо CU з розміром 16×16 більше не розділяється, то чотири суб-CU 8×8 також будуть називатися листовими CU, хоч CU 16×16 ніколи не розділялася.

CU має схожу з макроблоком зі стандарту H.264 ціль за винятком того, що CU не має відмінності в розмірі. Наприклад, блок дерева можна розділити на чотири дочірніх вузли (також звані суб-CU), і кожен дочірній вузол, в свою чергу, може бути батьківським вузлом і розділятися на чотири інших дочірніх вузли. Кінцевий, нерозділений дочірній вузол, називаний листом квадродереву, містить вузол кодування, також називаний листовою CU. Синтаксичні дані, асоційовані з кодованим бітовим потоком, можуть задавати максимальне число разів, яке можна ділити блок дерева, називане максимальною глибиною CU, а також можуть задавати мінімальний розмір вузлів кодування. Відповідно, бітовий потік також може задавати найменшу одиницю кодування (SCU). Дане розкриття винаходу використовує термін "блок", щоб посилатися на будь-яку з CU, PU або TU, застосовно до HEVC, або аналогічні структури даних, застосовно до інших стандартів (наприклад, макроблоки і їх субблоки в H.264/AVC).

CU включає в себе вузол кодування і одиниці прогнозування (PU) і одиниці перетворення (TU), асоційовані з вузлом кодування. Розмір CU відповідає розміру вузла кодування і повинен мати квадратну форму. Розмір CU може варіюватися від 8×8 пікселів до розміру блока дерева з максимальним числом 64×64 пікселів або більше. Кожна CU може містити одну або більше PU і одну або більше TU. Синтаксичні дані, асоційовані з CU, можуть описувати, наприклад, розбиття CU на одну або більше PU. Режими розбиття можуть відрізнятися між тим, чи кодується CU в режимі пропуску або в прямому режимі, режимі внутрішнього прогнозування або режимі зовнішнього прогнозування. PU можуть розбиватися, щоб мати неквадратну форму. Синтаксичні дані, асоційовані з CU, також можуть описувати, наприклад, розбиття CU на одну або більше TU згідно з квадродеревом. TU може мати квадратну або неквадратну (наприклад, прямокутну) форму.

Стандарт HEVC допускає перетворення згідно з TU, які можуть відрізнятися для різних CU. TU звичайно мають розмір на основі розміру PU в даній CU, заданій для розділеної LCU, хоч це не завжди може бути так. TU звичайно мають такий же розмір або менше, ніж PU. У деяких прикладах залишкові вибірки, відповідні CU, можна розділити на менші одиниці, використовуючи структуру квадродереву, відому як "залишкове квадродревко" (RQT). Листи RQT можуть називатися одиницями перетворення (TU). Значення різниці пікселів, асоційовані з TU, можна перетворити для створення коефіцієнтів перетворення, які можна квантувати.

Листова CU може включати в себе одну або більше одиниць прогнозування (PU). Звичайно PU представляє просторову область, відповідну всій або частині відповідної CU, і може включати в себе дані для витягання опорної вибірки для PU. Крім того, PU включає в себе дані, пов'язані з прогнозуванням. Наприклад, коли PU кодується у внутрішньому режимі, дані для PU можуть включатися в залишкове квадродревко (RQT), яке може включати в себе дані, що описують режим внутрішнього прогнозування для TU, відповідної PU. Як інший приклад, коли PU кодується у зовнішньому режимі, PU може включати в себе дані, що задають один або

більше векторів руху для PU. Дані, що задають вектор руху для PU, можуть описувати, наприклад, горизонтальну складову вектора руху, вертикальну складову вектора руху, розрізнення для вектора руху (наприклад, точність в одну чверть пікселя або точність в одну восьму пікселя), опорне зображення, на яке вказує вектор руху, і/або список опорних зображень (наприклад, Список 0, Список 1 або Список C) для вектора руху.

Листова CU, що має одну або більше PU, також може включати в себе одну або більше одиниць перетворення (TU). Одиниці перетворення можуть бути точно визначені з використанням RQT (також званого структурою квадродерева TU), як обговорювалося вище. Наприклад, прапор розділення може вказувати, чи розділяється листова CU на чотири одиниці перетворення. Потім кожну одиницю перетворення можна далі розділити на додаткові суб-TU. Коли TU більше не розділяється, вона може називатися листовою TU. Як правило, для внутрішнього кодування всі листові TU, що належать листовій CU, спільно використовують один і той же режим внутрішнього прогнозування. Тобто один і той же режим внутрішнього прогнозування застосовується, як правило, для обчислення прогнозованих значень для всіх TU в листовій CU. Для внутрішнього кодування відеокoder може обчислювати залишкове значення для кожної листової TU, використовуючи режим внутрішнього прогнозування, як різницю між частиною CU, відповідною TU, і вихідним блоком. TU не обов'язково обмежується розміром PU. Таким чином, TU можуть бути більше або менше PU. Для внутрішнього кодування PU може співвідноситися з відповідною листовою TU для однієї і тієї ж CU. У деяких прикладах максимальний розмір листової TU може відповідати розміру відповідної листової CU.

Крім того, TU в листових CU також можуть асоціюватися з відповідними структурами даних квадродерева, званими залишковими квадродеревами (RQT). Тобто листова CU може включати в себе квадродерево, що вказує, як листова CU розбивається на TU. Кореневий вузол квадродерева TU, як правило, відповідає листовій CU, тоді як кореневий вузол квадродерева CU, як правило, відповідає блоку дерева (або LCU). TU в RQT, які не розділяються, називаються листовими TU. Взагалі, дане розкриття винаходу використовує терміни "CU" і "TU", щоб посилатися на листову CU і листову TU, відповідно, коли не вказане інше.

Відеопослідовність, як правило, включає в себе послідовність відеокadrів або зображень. Група зображень (GOP), як правило, містить послідовність з одного або більше відеозображень. GOP може включати в себе синтаксичні дані в заголовку GOP, заголовку одного або більше зображень або де-небудь в іншому місці, які описують число зображень, включених в GOP. Кожний слайс зображення може включати в себе синтаксичні дані слайсу, які описують режим кодування для відповідного слайсу. Відеокoder 20 звичайно діє на відеоблоки в окремих слайсах відео, щоб кодувати відеодані. Відеоблок може відповідати вузлу кодування в CU. Відеоблоки можуть мати фіксовані або змінні розміри і можуть відрізнятися за розміром відповідно до точно визначеного стандарту кодування.

Як приклад HM підтримує прогнозування в різних розмірах PU. Передбачаючи, що розмір конкретної CU дорівнює $2N \times 2N$, HM підтримує внутрішнє прогнозування в розмірах PU $2N \times 2N$ або $N \times N$ і зовнішнє прогнозування в симетричних розмірах PU $2N \times 2N$, $2N \times N$, $N \times 2N$ або $N \times N$. HM також підтримує асиметричне розбиття для зовнішнього прогнозування в розмірах PU $2N \times nU$, $2N \times nD$, $nL \times 2N$ і $nR \times 2N$. При асиметричному розбитті один напрямок CU не розбивається, тоді як інший напрямок розбивається на 25 % і 75 %. Частина CU, відповідна розбиттю 25 %, вказується за допомогою "n" з подальшим указанням "Зверху", "Знизу", "Зліва" або "Справа". Таким чином, наприклад, " $2N \times nU$ " посилається на CU $2N \times 2N$, яка розбивається горизонтально з PU $2N \times 0,5N$ зверху і PU $2N \times 1,5N$ знизу.

У даному розкритті винаходу " $N \times N$ " і " N на N " можуть використовуватися взаємозамінно, щоб посилатися на розміри пікселя відеоблока в показниках вертикального і горизонтального розмірів, наприклад 16×16 пікселів або 16 на 16 пікселів. Звичайно блок 16×16 буде мати 16 пікселів у вертикальному напрямку ($y=16$) і 16 пікселів в горизонтальному напрямку ($x=16$). Також блок $N \times N$, як правило, має N пікселів у вертикальному напрямку і N пікселів в горизонтальному напрямку, де N представляє ненегативне ціле значення. Пікселі в блоці можуть розташовуватися в рядках і стовпцях. Крім того, блокам не обов'язково мати таке ж число пікселів в горизонтальному напрямку, як і у вертикальному напрямку. Наприклад, блоки можуть містити $N \times M$ пікселів, де M не обов'язково дорівнює N .

Після кодування з внутрішнім прогнозуванням або зовнішнім прогнозуванням, що використовує PU в CU, відеокoder 20 може обчислити залишкові дані для TU в CU. PU можуть містити синтаксичні дані, що описують спосіб або режим формування даних пікселя з прогнозуванням в просторовій області (також називаній областю пікселів), а TU можуть містити коефіцієнти в області перетворення після застосування перетворення, наприклад дискретного косинусного перетворення (DCT), цілочислового перетворення, вейвлет-перетворення або

концептуально схожого перетворення, до залишкових відеоданих. Залишкові дані можуть відповідати різницям пікселів між пікселами некодованого зображення і значеннями прогнозування, відповідними PU. Відеокодер 20 може утворити TU, що включають в себе залишкові дані для CU, а потім перетворити TU для створення коефіцієнтів перетворення для CU.

Після будь-яких перетворень для створення коефіцієнтів перетворення відеокодер 20 може виконати квантування коефіцієнтів перетворення. Квантування загалом належить до процесу, в якому коефіцієнти перетворення квантуються, щоб зменшити по можливості об'єм даних, використовуваний для представлення коефіцієнтів, забезпечуючи додаткове стиснення. Процес квантування може зменшити бітову глибину, асоційовану з деякими або всіма коефіцієнтами. Наприклад, n -бітове значення під час квантування можна округлити в меншу сторону до m -бітового значення, де n більше m .

Після квантування відеокодер може сканувати коефіцієнти перетворення, утворюючи одновимірний вектор з двовимірної матриці, що включає в себе квантовані коефіцієнти перетворення. Сканування може бути призначене для розміщення коефіцієнтів з більшою енергією (і тому з меншою частотою) попереду масиву і розміщення коефіцієнтів з меншою енергією (і тому більшою частотою) позаду масиву. У деяких прикладах відеокодер 20 може використовувати попередньо визначений порядок сканування, щоб сканувати квантовані коефіцієнти перетворення для створення серіалізованого вектора, який можна ентропійно кодувати. У інших прикладах відеокодер 20 може виконувати адаптивне сканування. Після сканування квантованих коефіцієнтів перетворення для утворення одновимірного вектора відеокодер 20 може ентропійно кодувати одновимірний вектор, наприклад, відповідно до контекстно-адаптивного кодування із змінною довжиною (CAVLC), контекстно-адаптивного двійкового арифметичного кодування (CABAC), синтаксичного контекстно-адаптивного двійкового арифметичного кодування (SBAC), ентропійного кодування з розбиттям на інтервали імовірності (PIPE) або іншої методології ентропійного кодування. Відеокодер 20 також може ентропійно кодувати синтаксичні елементи, асоційовані з кодованими відеоданими, для використання відеодекодером 30 при декодуванні відеоданих.

Для виконання CABAC відеокодер 20 може призначити контекст в рамках контекстної моделі символу, який повинен бути переданий. Контекст може стосуватися, наприклад, того, чи є сусідні значення символу ненульовими. Для виконання CAVLC відеокодер 20 може вибрати код змінної довжини для символу, який повинен бути переданий. Кодові слова при VLC можуть бути побудовані так, що відносно більш короткі коди відповідають більш імовірним символам, тоді як більш довгі коди відповідають менш імовірним символам. Таким чином, за допомогою використання VLC може добитися економії бітів в порівнянні, наприклад, з використанням кодових слів рівної довжини для кожного символу, який повинен бути переданий. Визначення імовірності може ґрунтуватися на контексті, призначеному символу.

Відповідно до методик з даного розкриття винаходу кодер відео, наприклад відеокодер 20 або відеодекодер 30, може конфігуруватися для кодування набору параметрів відео (VPS) для одного або більше рівнів відеоданих і кодування одного або більше рівнів відеоданих, ґрунтуючись щонайменше частково на VPS. Таблиці 2 і 5, детальніше описані нижче, включають в себе зразкові набори синтаксичних елементів в VPS. Кожний з одного або більше рівнів відеоданих може посилатися на VPS, тобто на один і той же VPS. Іншими словами, VPS може застосовуватися до всіх рівнів загального набору відеоданих, наприклад всіх рівнів SVC і/або всіх видів у відеоданих MVC.

VPS може включати в себе різні категорії інформації. Наприклад, VPS може включати в себе опис лічильника вимірювань вибірок (SDCD). Тобто для кожного вимірювання кодер відео може сигналізувати набір індексів. Можливі вимірювання включають в себе snt_p : число рівнів пріоритету, що містяться в кодованій відеопослідовності; snt_d : скільки різних рівнів залежності в бітовому потоці, декілька рівнів з однаковими просторовим розрізненням і бітовою глибиною можуть належати різним рівням залежності; snt_t : скільки часових рівнів в бітовому потоці; snt_q : максимальне число рівнів якості для будь-якого рівня залежності в бітовому потоці; і snt_v : максимальне число видів. Налаштування бітової глибини можуть включати в себе 8-бітову або 12-бітову і можуть відрізнятися для різних складових кольору. Формати дискретизації кольоровості можуть включати в себе 4:0:0, 4:2:0 і 4:4:4.

VPS також може включати в себе відображення індексу вибірки на характеристики. Якщо для кожного вимірювання показчик характеристик не дорівнює індексу, що міняється від 0 до лічильника вимірювань вибірок мінус 1, то можна ввести цикл для точного визначення показчика характеристик для кожного індексу характеристик. Відображення для кожного індексу залежності може включати в себе специфічне просторове розрізнення зі специфічним

значенням бітової глибини і специфічним форматом вибірки кольоровості. Зазначимо, що це можна було б пропустити, якби на декодері завжди була фіксована довідкова таблиця, наприклад 0 може відповідати 4:2:0, 1 може відповідати 4:4:4 і 2 може відповідати 4:0:0. Відображення додатково або альтернативно може включати в себе: для кожного часового
 5 індексу/id - специфічну частоту кадрів або середню частоту кадрів; для кожного індексу виду - специфічний id виду; для кожного індексу бітової глибини - пару специфічних значень бітової глибини для яскравості і кольоровості; і для кожного формату дискретизації кольоровості - специфічний показник формату дискретизації кольоровості.

VPS також може включати в себе параметри керування і прапори дозволу/заборони інструментів, наприклад наступні: pcm_bit_depth_luma_minus1, pcm_bit_depth_chroma_minus1, loop_filter_across_slice_flag, pcm_loop_filter_disable_flag, temporal_id_nesting_flag, один або
 10 більше пов'язаних з фрагментом синтаксичних елементів, chroma_pred_from_luma_enabled_flag, sample_adaptive_offset_enabled_flag, adaptive_loop_filter_enabled_flag і inter_4x4_enabled_flag.

VPS також може включати в себе один або більше описів робочих точок. Робочі точки, як
 15 правило, описують підмножину загального числа видів у відеоданих, включених в бітовий потік. Робоча точка може включати в себе конкретне число видів, намічених для виведення, а також інші види, які можуть використовуватися для посилання при декодуванні, виведенні або тому і іншому. Бітовий потік може включати в себе одну або більше робочих точок, описаних за допомогою описів робочих точок. Описи робочих точок можуть включати в себе інформацію, що
 20 задає число максимальних робочих точок, залежність між різними рівнями або видами, профіль і рівень для кожної робочої точки, бітрейт (частота проходження бітів) для кожної робочої точки, залежність між робочими точками для кожної робочої точки, інші обмеження для кожної робочої точки, інформації про застосовність відео (VUI) або частини VUI, і/або VUI або частини VUI для кожного рівня або виду. Додатково або альтернативно описи робочих точок можуть включати в себе для кожної робочої точки представлення одиниці рівня абстракції мережі (NAL) рівня
 25 кодування відео (VCL) робочої точки. У деяких прикладах представлення одиниці NAL VCL робочої точки може включати в себе три можливих вибори для кожного вимірювання: (1) специфічне значення індексу: наприклад, для просторового розрізнення - бітова глибина для формату дискретизації кольоровості; (2) діапазон значення індексу: наприклад, для часових
 30 рівнів - від 0 до id найвищого часового рівня, для рівнів якості - від 0 до id найвищого рівня якості; або (3) список значень індексу, наприклад, для видів - список значень індексів виду.

У деяких прикладах VPS може включати в себе дані, що вказують максимальне число часових рівнів серед рівнів бітового потоку. Тобто відеокодер 20 і/або відеодекодер 30 можуть
 35 конфігуруватися для кодування VPS, що включає в себе дані, які вказують максимальне число часових рівнів для відповідного бітового потоку. Наприклад, відеокодер 20 може визначити максимальне число часових рівнів і кодувати VPS для включення в нього даних, що представляють визначене максимальне число часових рівнів, тоді як відеодекодер 30 може декодувати VPS для визначення максимального числа часових рівнів. Відеокодер 20 і відеодекодер 30 також можуть кодувати відеодані бітового потоку на основі визначеного
 40 максимального числа часових рівнів. Наприклад, максимальне число часових рівнів може впливати на число часових ідентифікаторів, які потрібні для представлення різних часових рівнів. Як інший приклад максимальне число часових рівнів може впливати на спосіб, яким відеокодер 20 і відеодекодер 30 кодують ідентифікатори опорних зображень, наприклад, з використанням значень лічильника послідовності зображення (POC).

Як ще один приклад відеокодер 20 і відеодекодер 30 можуть конфігуруватися для кодування
 45 даних конкретного часового рівня з використанням тільки опорних даних аж до і включаючи той же часовий рівень. Іншими словами, відеокодер 20 і відеодекодер 30 можуть конфігуруватися, щоб уникати кодування даних конкретного часового рівня з використанням опорних даних більш високого часового рівня. Таким чином, відеодекодер 30 може бути впевнений в точному
 50 декодуванні відеоданих даного набору часових рівнів навіть після витягання бітового субпотіку. Тобто, якщо виконується витягання бітового субпотіку, то деякі часові рівні вище найвищого рівня витягнутого бітового субпотіку не будуть доступні для посилання. За допомогою кодування даних кожного часового рівня тільки з посиланням на дані рівнів на поточному рівні або нижче можна уникнути помилок, які в іншому випадку могли б відбутися внаслідок
 55 залежності даних на конкретному рівні від даних з більш високого рівня, який був би втрачений внаслідок витягання бітового субпотіку.

У деяких прикладах VPS додатково або альтернативно включає в себе дані, що вказують
 60 будь-яке або обидва з числа зображень, які повинні бути переупорядковані на одному або більше рівнях бітового потоку, і/або числа зображень, які повинні бути збережені в буфер декодованих зображень (DPB). Як відмічалось вище, такі дані можуть називатися інформацією

про обмеження бітового потоку. Відповідно, пристрій-адресат 14 може визначити можливості відеодекодера 30 і використовувати інформацію про обмеження бітового потоку для визначення, чи придатний відповідний бітовий потік для декодування відеодекодером 30 або чи потрібно пристрою-адресату 14 вибрати альтернативний контент (наприклад, від постачальника мережевого контенту, передбачаючи, що доступно декілька версій контенту).

Крім того, відеокодер 20 і відеодекодер 30 можуть використовувати інформацію про обмеження бітового потоку під час кодування відеоданих. Наприклад, відеокодер 20 може гарантувати, що інформація про обмеження бітового потоку не порушується. Тобто, передбачаючи, що інформація про обмеження бітового потоку вказує, що потрібно зберегти не більше N зображень в DPB, відеокодер 20 може гарантувати, що не більше N зображень включається в будь-яке поєднання одного або більше списків опорних зображень в будь-який момент часу. Як інший приклад, передбачаючи, що інформація про переупорядкування зображень вказує, що зображення потрібно зсунути не більше ніж на M зображень, відеокодер 20 може гарантувати, що ніяке зображення не зсувається більше ніж на M зображень. Зсув зображень цим способом звичайно відповідає різниці між порядком декодування і порядком відображення зображення. Відеокодер 30 також може використовувати таку інформацію під час кодування, наприклад, для виконання керування DPB, наприклад очищення DPB. Відеокодер 20 і відеодекодер 30 також можуть використовувати інформацію про обмеження бітового потоку, наприклад максимальне число зображень, які повинні бути збережені в DPB, і/або число зображень, які повинні бути переупорядковані, при кодуванні значень ідентифікаторів опорних зображень.

У деяких прикладах VPS додатково або альтернативно включає в себе дані, що вказують параметри гіпотетичного еталонного декодера (HRD). Параметри HRD включають в себе, наприклад, дані, що описують моменти, в які дані потрібно видалити з буфера кодованих зображень (CPB). У декодерах, наприклад відеодекодері 30, CPB являє собою буфер, в якому кодовані відеодані зберігаються доти, поки дані не будуть готові для декодування. Декодери, наприклад відеодекодер 30, також можуть включати в себе буфер декодованих зображень (DPB), в якому зберігаються декодовані відеодані, наприклад, для використання як опорних даних для прогнозованих зовнішньо даних і для переупорядкування зображень з порядку декодування в порядок відображення.

Параметри HRD можуть включати в себе дані, що вказують, коли конкретні зображення потрібно видалити з CPB і декодувати. Таким чином, відеокодер 20 може кодувати параметри HRD у VPS для вказання, коли зображення можна видалити з CPB і декодувати, тоді як відеодекодер 30 може декодувати параметри HRD у VPS для визначення, коли видаляти зображення з CPB. Також відеокодер 20 і відеодекодер 30 можуть кодувати зображення відповідно до параметрів HRD, наприклад, в порядку кодування, вказаному параметрами HRD. Таким чином, відеокодер 20 і/або відеодекодер 30 можуть конфігуруватися для кодування VPS, що включає в себе параметри HRD, і кодування відеоданих, відповідних VPS, основуючись щонайменше частково на параметрах HRD.

VPS також може включати в себе дані розширення, що вказують, чи розширений VPS, наприклад, щоб надати дані для одного або більше додаткових інструментів кодування. Такі інструменти кодування можуть бути інструментами, які відрізняються від інструментів у відповідному стандарті кодування відео, наприклад H.264/AVC ITU-T або майбутньому стандарті HEVC. Крім того, такі інструменти кодування можуть вимагати конфігураційні дані. Ці конфігураційні дані можуть надаватися в даних розширення VPS. Таким чином, при кодуванні відеоданих з використанням таких інструментів кодування відеокодер 20 і/або відеодекодер 30 можуть кодувати VPS, що вказує, чи присутні дані розширення, і, якщо це так, дані розширення VPS. Крім того, коли присутні такі дані розширення, відеокодер 20 і/або відеодекодер 30 можуть використовувати відповідні інструменти кодування для кодування відеоданих з використанням даних розширення.

Різні стандарти кодування відео задають відповідний синтаксис, семантику і процес декодування для безпомилкових бітових потоків, будь-який з яких відповідає деякому профілю або рівню. Стандарти кодування відео, як правило, точно не визначають кодер, але в задачу кодера входить гарантування того, що сформовані бітові потоки відповідають стандарту для декодера. Застосовно до стандартів кодування відео "профіль" відповідає підмножині алгоритмів, властивостей або інструментів і обмежень, які до них застосовуються. Як задано, наприклад, стандартом H.264, "профіль" є підмножиною всього синтаксису бітового потоку, який точно визначається стандартом H.264. "Рівень" відповідає обмеженням споживання ресурсів декодера, наприклад, запам'ятовуючого пристрою і обчислення декодера, які належать до розрізнення зображень, бітрейту і швидкості обробки блоків. Профіль можна сигналізувати за

допомогою значення `profile_idc` (показник профілю), тоді як рівень можна сигналізувати за допомогою значення `level_idc` (показник рівня). Відповідно до методик з даного розкриття винаходу інформація про профіль і рівень може точно визначатися в описах робочих точок, які обговорювалися вище.

5 У деяких прикладах кожний рівень або вид в бітовому потоці посиляється на набір параметрів відео (VPS), і набір параметрів багаторівневої послідовності (LPS) може бути активним для кожного рівня. LPS можна підтримувати якомога більш легкодоступним шляхом посилання на VPS при проектуванні. LPS може включати в себе будь-яку або всю обговорювану нижче інформацію. LPS може включати в себе указання вимірювання вибірки, яке для кожного вимірювання вказує індекс до кожного вимірювання. Наприклад, якщо в VPS індекс до просторового розрізнення 0 призначається просторовій характеристиці 320×240, а індекс до просторового розрізнення 1 призначається 640×480, і поточному рівню потрібно призначити розрізнення 640×480, то відеокoder 20 і/або відеодекодер 30 можуть кодувати синтаксичний елемент значенням 1 для поточного рівня. Тобто відеокoder 20 може сигналізувати значення 1 для синтаксичного елемента, щоб точно визначити розрізнення 640×480, тоді як відеодекодер 30 на основі значення 1 для синтаксичного елемента може визначити, що поточний рівень з синтаксичним елементом, що має значення 1, має розрізнення 640×480.

LPS також може включати в себе параметри керування і прапори дозволу/заборони інструментів. Наприклад, параметри керування і прапори дозволу/заборони інструментів можуть включати в себе `pcm_bit_depth_luma_minus1`, `pcm_bit_depth_chroma_minus1`, `loop_filter_across_slice_flag`, `pcm_loop_filter_disable_flag`, один або більше пов'язаних з фрагментом синтаксичних елементів, `chroma_pred_from_luma_enabled_flag`, `sample_adaptive_offset_enabled_flag`, `adaptive_loop_filter_enabled_flag` і ієрархію одиниць кодування (CU).

25 LPS може додатково включати в себе інформацію про інші типи наборів параметрів, застосовуваних до слайсу, групи слайсів, зображення або декількох зображень. Кожний з цих наборів параметрів може посилатися на специфічний набір параметрів зображення (PPS).

Кодер відео, наприклад відеокoder 20 і відеодекодер 30, може конфігуруватися для підтвердження і/або визначення, що PPS не посиляється на LPS або VPS. Таким чином, кодер відео може гарантувати, що кожний PPS бітового потоку не посиляється на LPS або VPS. Синтаксичний аналіз PPS може бути незалежним. Коли PPS включає в себе один або більше таких же синтаксичних елементів, як і в VPS або LPS, синтаксичні елементи PPS можуть перезаписати елементи в VPS або LPS.

Кодер відео може додатково конфігуруватися для кодування груповального набору параметрів (GPS), який групує всі набори параметрів. Кодер відео може кодувати множину різних груп в GPS, причому кожна має окремі ідентифікатори GPS (id). Кожна з груп в GPS може включати в себе різне поєднання наборів параметрів. Таким чином, заголовку слайсу потрібно включати в себе тільки посилання на відповідний id GPS і не потрібно включати в себе указання типу набору параметрів. Попередня заявка США з порядковим № 61/590702, подана 25 січня 2012 р., також описує методики, в яких групуються різні типи наборів параметрів, і тільки ID у RBSP групування набору параметрів детально сигналізується в заголовку слайсу.

Як обговорювалося вище, кодер відео, наприклад відеокoder 20 або відеодекодер 30, може конфігуруватися для кодування набору параметрів відео і/або груповального набору параметрів. Приклади набору параметрів відео детально обговорюються відносно фіг. 5, тоді як приклади груповального набору параметрів детально обговорюються відносно фіг. 6.

Відеокoder 20 додатково може відправляти відеодекодеру 30 синтаксичні дані, наприклад блокові синтаксичні дані, кадрові синтаксичні дані і основані на GOP синтаксичні дані, наприклад, в заголовку кадру, заголовку блока, заголовку слайсу або заголовку GOP. Синтаксичні дані GOP можуть описувати число кадрів у відповідній GOP, а синтаксичні дані кадру можуть вказувати режим кодування/прогнозування, використовуваний для кодування відповідного кадру.

Відеокoder 20 і відеодекодер 30 можуть бути реалізовані у вигляді будь-якої з ряду придатних схем кодера або декодера у відповідних випадках, наприклад одного або більше мікропроцесорів, цифрових процесорів сигналів (DSP), спеціалізованих інтегральних схем (ASIC), програмованих користувачем вентиляльних матриць (FPGA), дискретних логічних схем, програмного забезпечення, апаратних засобів, мікропрограмного забезпечення або будь-яких їх поєднань. Кожний з відеокодера 20 і відеодекодера 30 може включатися в один або більше кодерів або декодерів, будь-який з яких може вбудовуватися як частина об'єднаного відеокодера/декодера (кодека). Пристрій, що включає в себе відеокoder 20 і/або відеодекодер

30, може містити інтегральну схему, мікропроцесор і/або пристрій бездротового зв'язку, наприклад стільниковий телефон.

Фіг. 2 - блок-схема, що ілюструє приклад відеокодера 20, який може реалізувати методики для кодування наборів параметрів і одиниць NAL для одного або більше рівнів відеоданих. Відеокодер 20 може виконувати внутрішнє і зовнішнє кодування відеоблоків в слайсах відео. Внутрішнє кодування спирається на просторове прогнозування, щоб зменшити або усунути просторову надмірність у відео в даному відеокадрі або зображенні. Зовнішнє кодування спирається на часове прогнозування, щоб зменшити або усунути часову надмірність у відео в сусідніх кадрах або зображеннях відеопослідовності. Внутрішній режим (I-режим) може належати до будь-якого з декількох режимів просторового кодування. Зовнішні режими, наприклад однонаправлене прогнозування (P-режим) або двонаправлене прогнозування (B-режим), можуть належати до будь-якого з декількох режимів часового кодування.

Як показано на фіг. 2, відеокодер 20 приймає поточний відеоблок у відеокадрі, який повинен бути кодований. У прикладі з фіг. 2 відеокодер 20 включає в себе модуль 40 вибору режиму, запам'ятовуючий пристрій 64 опорних зображень, суматор 50, модуль 52 обробки перетворення, модуль 54 квантування і модуль 56 ентропійного кодування. Модуль 40 вибору режиму в свою чергу включає в себе модуль 44 компенсації руху, модуль 42 оцінки руху, модуль 46 внутрішнього прогнозування і модуль 48 розбиття. Для відновлення відеоблока відеокодер 20 також включає в себе модуль 58 зворотного квантування, модуль 60 зворотного перетворення і суматор 62. Також може включатися фільтр зменшення блоковості (не показаний на фіг. 2) для фільтрації границь блока, щоб видалити артефакти блоковості з відновленого відео. При бажанні фільтр зменшення блоковості звичайно фільтрував би виведення суматора 62. Також можуть використовуватися додаткові фільтри (в циклі або після циклу) на доповнення до фільтра зменшення блоковості. Такі фільтри скорочено не показані, але при бажанні можуть фільтрувати виведення суматора 50 (як фільтр в циклі).

Під час процесу кодування відеокодер 20 приймає відеокадр або слайс, який повинен бути кодований. Кадр або слайс можна розділити на декілька відеоблоків. Модуль 42 оцінки руху і модуль 44 компенсації руху виконують кодування із зовнішнім прогнозуванням прийнятого відеоблока відносно одного або більше блоків в одному або більше опорних кадрах, щоб забезпечити часове прогнозування. Модуль 46 внутрішнього прогнозування альтернативно може виконувати кодування з внутрішнім прогнозуванням прийнятого відеоблока відносно одного або більше сусідніх блоків в тому ж кадрі або слайсі як блока, який повинен бути кодований, щоб забезпечити просторове прогнозування. Відеокодер 20 може виконувати декілька проходів кодування, наприклад, для вибору придатного режиму кодування для кожного блока відеоданих.

Крім того, модуль 48 розбиття може розбити блоки відеоданих на субблоки на основі оцінки попередніх схем розбиття в попередніх проходах кодування. Наприклад, модуль 48 розбиття може спочатку розбити кадр або слайс на LCU і розбити кожну з LCU на суб-CU на основі аналізу спотворення залежно від швидкості передачі (наприклад, оптимізації спотворення залежно від швидкості передачі). Модуль 40 вибору режиму додатково може створити структуру даних квадродерева, що вказує розбиття LCU на суб-CU. CU листа в квадродереві можуть включати в себе одну або більше PU і одну або більше TU.

Модуль 40 вибору режиму може вибрати один з режимів кодування, внутрішній або зовнішній, наприклад, на основі помилкових результатів і надає результуючий блок з внутрішнім або зовнішнім кодуванням суматору 50, щоб сформувати дані залишкового блока, і суматору 62, щоб відновити кодований блок для використання як опорного кадру. Модуль 40 вибору режиму також надає синтаксичні елементи, наприклад вектори руху, покажчики внутрішнього режиму, інформацію про розбиття і іншу таку синтаксичну інформацію, модулю 56 ентропійного кодування.

Модуль 42 оцінки руху і модуль 44 компенсації руху можуть бути спільно інтегровані, але ілюструються окремо з концептуальною метою. Оцінка руху, виконувана модулем 42 оцінки руху, є процесом формування векторів руху, які оцінюють руху для відеоблоків. Вектор руху може вказувати, наприклад, зміщення PU відеоблока в поточному відеокадрі або зображенні відносно блока з прогнозуванням в опорному кадрі (або іншій кодованій одиниці) відносно поточного блока, кодованого в поточному кадрі (або іншій кодованій одиниці). Блок з прогнозуванням є блоком, який визнаний точно співпадаючим з блоком, який повинен бути кодований, в показниках різниці пікселів, що може визначатися за допомогою суми абсолютних різниць (SAD), суми квадратів різниць (SSD) або інших показників різниці. У деяких прикладах відеокодер 20 може обчислювати значення для положень субцілого пікселя в опорних зображеннях, збережених в запам'ятовуючому пристрої 64 опорних зображень. Наприклад,

відеокодер 20 може інтерполювати значення положень однієї чверті пікселя, положень однієї восьмої пікселя або інших положень дробового пікселя в опорному зображенні. Тому модуль 42 оцінки руху може виконати пошук руху відносно положень повного пікселя і положень дробового пікселя і вивести вектор руху з точністю до дробового пікселя.

Модуль 42 оцінки руху обчислює вектор руху для PU відеоблока в слайсі із зовнішнім кодуванням шляхом порівняння положення PU з положенням блока з прогнозуванням в опорному зображенні. Опорне зображення може вибиратися з першого списку опорних зображень (Список 0) або другого списку опорних зображень (Список 1), кожний з яких ідентифікує одне або більше опорних зображень, збережених в пам'ятовуючому пристрої 64 опорних зображень. Модуль 42 оцінки руху відправляє обчислений вектор руху модулю 56 ентропійного кодування і модулю 44 компенсації руху.

Компенсація руху, виконувана модулем 44 компенсації руху, може включати в себе вибірку або формування блока з прогнозуванням на основі вектора руху, визначеного модулем 42 оцінки руху. Знов в деяких прикладах модуль 42 оцінки руху і модуль 44 компенсації руху можуть бути об'єднані функціонально. Після прийому вектора руху для PU поточного відеоблока модуль 44 компенсації руху може знайти блок з прогнозуванням, на який вказує вектор руху, в одному зі списків опорних зображень. Суматор 50 утворює залишковий відеоблок шляхом віднімання значень пікселів блока з прогнозуванням із значень пікселів поточного кодованого відеоблока, утворюючи значення різниці пікселів, які обговорюються нижче. Звичайно модуль 42 оцінки руху виконує оцінку руху відносно компонентів яскравості, а модуль 44 компенсації руху використовує вектори руху, обчислені на основі компонентів яскравості, для компонентів кольоровості і компонентів яскравості. Модуль 40 вибору режиму також може формувати синтаксичні елементи, асоційовані з відеоблоками і слайсом відео, для використання відеодекодером 30 при декодуванні відеоблоків в слайсі відео.

Модуль 46 внутрішнього прогнозування може внутрішньо прогнозувати поточний блок альтернативно зовнішньому прогнозуванню, виконуваному модулем 42 оцінки руху і модулем 44 компенсації руху, як описано вище. Зокрема, модуль 46 внутрішнього прогнозування може визначити режим внутрішнього прогнозування, щоб використовувати його для кодування поточного блока. У деяких прикладах модуль 46 внутрішнього прогнозування може кодувати поточний блок з використанням різних режимів внутрішнього прогнозування, наприклад, під час окремих проходів кодування, а модуль 46 внутрішнього прогнозування (або модуль 40 вибору режиму в деяких прикладах) може вибирати придатний режим внутрішнього прогнозування для використання з перевірених режимів.

Наприклад, модуль 46 внутрішнього прогнозування може обчислити значення спотворення залежно від швидкості передачі, використовуючи аналіз спотворення залежно від швидкості передачі для різних перевірених режимів внутрішнього прогнозування, і вибрати режим внутрішнього прогнозування, що має найкращі характеристики спотворення залежно від швидкості передачі серед перевірених режимів. Аналіз спотворення залежно від швидкості передачі, як правило, визначає величину спотворення (або помилки) між кодованим блоком і вихідним, некодованим блоком, який кодувався для створення кодованого блока, а також бітрейт (тобто число бітів), використовуваний для створення кодованого блока. Модуль 46 внутрішнього прогнозування може обчислити відношення зі спотворень і швидкостей для різних кодованих блоків, щоб визначити, який режим внутрішнього прогнозування демонструє найкраще значення спотворення залежно від швидкості передачі для блока.

Після вибору режиму внутрішнього прогнозування для блока модуль 46 внутрішнього прогнозування може надати модулю 56 ентропійного кодування інформацію, що вказує вибраний режим внутрішнього прогнозування для блока. Модуль 56 ентропійного кодування може кодувати інформацію, що вказує вибраний режим внутрішнього прогнозування. Відеокодер 20 може включити в переданий бітовий потік конфігураційний дані, які можуть включати в себе множину таблиць індексів режимів внутрішнього прогнозування і множину таблиць індексів змінених режимів внутрішнього прогнозування (також званих таблицями відображення кодових слів), визначення контекстів кодування для різних блоків і указання найбільш імовірного режиму внутрішнього прогнозування, таблиці індексів режимів внутрішнього прогнозування і таблиці індексів зміненого режиму внутрішнього прогнозування для використання для кожного з контекстів.

Відеокодер 20 утворює залишковий відеоблок шляхом віднімання даних прогнозування від модуля 40 вибору режиму з вихідного кодованого відеоблока. Суматор 50 являє собою компонент або компоненти, які виконують цю операцію віднімання. Модуль 52 обробки перетворення застосовує до залишкового блока перетворення, наприклад дискретне косинусне перетворення (DCT) або концептуально схоже перетворення, створюючи відеоблок, що містить

значення залишкових коефіцієнтів перетворення. Модуль 52 обробки перетворення може виконувати і інші перетворення, які концептуально схожі з DCT. Також могли б використовуватися вейвлет-перетворення, цілочислові перетворення, перетворення піддіапазонів або інші типи перетворень.

У будь-якому випадку модуль 52 обробки перетворення застосовує перетворення до залишкового блока, створюючи блок залишкових коефіцієнтів перетворення. Перетворення може перетворювати залишкову інформацію з області значень пікселів в область перетворення, наприклад частотну область. Модуль 52 обробки перетворення може відправити результуючі коефіцієнти перетворення модулю 54 квантування. Модуль 54 квантування квантує коефіцієнти перетворення для додаткового зменшення бітрейту. Процес квантування може зменшити бітову глибину, асоційовану з деякими або всіма коефіцієнтами. Міру квантування можна змінювати шляхом регулювання параметра квантування. У деяких прикладах модуль 54 квантування потім може виконати сканування матриці, що включає в себе квантовані коефіцієнти перетворення. Альтернативно сканування може виконати модуль 56 ентропійного кодування.

Після квантування модуль 56 ентропійного кодування ентропійно кодує квантовані коефіцієнти перетворення. Наприклад, модуль 56 ентропійного кодування може виконувати контекстно-адаптивне кодування із змінною довжиною (CAVLC), контекстно-адаптивне двійкове арифметичне кодування (CABAC), синтаксичне контекстно-адаптивне двійкове арифметичне кодування (SBAC), ентропійне кодування з розбиттям на інтервали імовірності (PIPE) або іншу методику ентропійного кодування. У випадку контекстного ентропійного кодування контекст може ґрунтуватися на сусідніх блоках. Після ентропійного кодування за допомогою модуля 56 ентропійного кодування кодований бітовий потік може передаватися іншому пристрою (наприклад, відеодекодеру 30) або архівуватися для подальшої передачі або витягання.

Модуль 58 зворотного квантування і модуль 60 зворотного перетворення застосовують зворотне квантування і зворотне перетворення відповідно, щоб відновити залишковий блок в області пікселів, наприклад, для подальшого використання як опорного блока. Модуль 44 компенсації руху може обчислити опорний блок шляхом додавання залишкового блока до блока з прогнозуванням в одному з кадрів в запам'ятовуючому пристрої 64 опорних зображень. Модуль 44 компенсації руху також може застосувати один або більше інтерполюючих фільтрів до відновленого залишкового блока, щоб обчислити значення субцілих пікселів для використання при оцінці руху. Суматор 62 додає відновлений залишковий блок до блока прогнозування з компенсованим рухом, створеного модулем 44 компенсації руху, щоб створити відновлений відеоблок для збереження в запам'ятовуючому пристрої 64 опорних зображень. Відновлений відеоблок може використовуватися модулем 42 оцінки руху і модулем 44 компенсації руху як опорний блок, щоб зовнішньо кодувати блок в подальшому відеокадрі.

Відеокодер 20 може додатково конфігуруватися для кодування набору параметрів відео (VPS), набору параметрів рівня (LPS) і/або групувального набору параметрів відповідно до методик з даного розкриття винаходу, а також набору параметрів послідовності (SPS), набору параметрів зображення (PPS), набору параметрів адаптації (APS) або інших таких структур сигнальних даних. Конкретніше, модуль 56 ентропійного кодування може конфігуруватися для кодування будь-якої або всіх цих структур даних. У випадку, якщо параметри цих різних структур даних можуть впливати на продуктивність кодування, модуль 40 вибору режиму може вибрати придатні параметри і передати ці параметри модулю 56 ентропійного кодування для включення, наприклад, в VPS. Інші параметри, наприклад число часових рівнів, число зображень, які повинні бути переупорядковані, і число зображень, які повинні бути збережені в буфер декодованих зображень, можуть вибиратися користувачем, наприклад адміністратором. У інших прикладах деякі параметри, наприклад параметри HRD, можуть виникати протягом процесу кодування.

Модуль 56 ентропійного кодування може кодувати VPS для включення будь-якого або всіх різних типів даних, описаних даним розкриттям винаходу. Відеокодер 20 також може кодувати дані відповідно до параметрів VPS. Конкретніше, відеокодер 20 може кодувати послідовності зображень з числа одного або більше рівнів відеоданих, яким відповідає VPS, відповідно до параметрів VPS.

Таким чином, відеокодер 20 з фіг. 2 представляє приклад відеокодера, сконфігурованого для кодування набору параметрів відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з одного або більше рівнів відеоданих посилається на VPS, і кодування одного або більше рівнів відеоданих, ґрунтуючись щонайменше частково на VPS.

Кодування VPS, хоч загалом і описане відносно відеокодера, може виконуватися іншими пристроями, наприклад інформованим про середовище мережевим елементом (MANE). MANE може відповідати мережевому елементу між пристроєм-джерелом (наприклад, пристроєм-

джерелом 12 з фіг. 1) і пристроєм-адресатом (наприклад, пристроєм-адресатом 14). MANE може конфігуруватися для кодування VPS відповідно до методик з даного розкриття винаходу. MANE може формувати VPS, використовуючи дані з іншими структурами даних, прийнятими MANE, наприклад набори параметрів послідовності.

Фіг. 3 - блок-схема, що ілюструє приклад відеодекодера 30, який може реалізувати методики для кодування наборів параметрів і одиниць NAL для одного або більше рівнів відеоданих. У прикладі з фіг. 3 відеодекодер 30 включає в себе модуль 70 ентропійного декодування, модуль 72 компенсації руху, модуль 74 внутрішнього прогнозування, модуль 76 зворотного квантування, модуль 78 зворотного перетворення, запам'ятовуючий пристрій 82 опорних зображень і суматор 80. Запам'ятовуючий пристрій 82 опорних зображень також може називатися "буфером декодованих зображень" або DPB. Відеодекодер 30 в деяких прикладах може виконувати прохід декодування, загалом зворотний відносно проходу кодування, описаного відносно відеокодера 20 (фіг. 2). Модуль 72 компенсації руху може формувати дані прогнозування на основі векторів руху, прийнятих від модуля 70 ентропійного декодування, тоді як модуль 74 внутрішнього прогнозування може формувати дані прогнозування на основі покажчиків режиму внутрішнього прогнозування, прийнятих від модуля 70 ентропійного декодування.

Під час процесу декодування відеодекодер 30 приймає від відеокодера 20 бітовий потік кодованого відео, який являє собою відеоблоки кодованого слайсу відео і асоційовані синтаксичні елементи. Модуль 70 ентропійного декодування у відеодекодері 30 ентропійно декодує бітовий потік, щоб сформувати квантовані коефіцієнти, вектори руху або покажчики режиму внутрішнього прогнозування і інші синтаксичні елементи. Модуль 70 ентропійного декодування перенаправляє вектори руху і інші синтаксичні елементи в модуль 72 компенсації руху. Відеодекодер 30 може прийняти синтаксичні елементи на рівні слайсу відео і/або рівні відеоблока.

Коли слайс відео кодується у вигляді слайсу з внутрішнім кодуванням (I), модуль 74 внутрішнього прогнозування може сформувати дані прогнозування для відеоблока поточного слайсу відео на основі сигналізованого режиму внутрішнього прогнозування і даних з раніше декодованих блоків поточного кадру або зображення. Коли відеокадр кодується у вигляді слайсу із зовнішнім кодуванням (тобто B, P або GPB), модуль 72 компенсації руху створює блоки з прогнозуванням для відеоблока поточного слайсу відео на основі векторів руху і інших синтаксичних елементів, прийнятих від модуля 70 ентропійного декодування. Блоки з прогнозуванням можуть створюватися з одного з опорних зображень в одному зі списків опорних зображень. Відеодекодер 30 може побудувати списки опорних кадрів, Список 0 і Список 1, використовуючи методики побудови за замовченням на основі опорних зображень, збережених в запам'ятовуючому пристрої 82 опорних зображень.

Модуль 72 компенсації руху визначає інформацію прогнозування для відеоблока поточного слайсу відео шляхом синтаксичного аналізу векторів руху і інших синтаксичних елементів і використовує інформацію прогнозування для створення блоків з прогнозуванням для поточного декодованого відеоблока. Наприклад, модуль 72 компенсації руху використовує деякі з прийнятих синтаксичних елементів для визначення режиму прогнозування (наприклад, внутрішнє або зовнішнє прогнозування), використовуюваного для кодування відеоблоків в слайсі відео, типу слайсу зовнішнього прогнозування (наприклад, B-слайс, P-слайс або GPB-слайс), інформації побудови для одного або більше списків опорних зображень для слайсу, векторів руху для кожного відеоблока із зовнішнім кодуванням в слайсі, стану зовнішнього прогнозування для кожного відеоблока із зовнішнім кодуванням в слайсі і іншої інформації для декодування відеоблоків в поточному слайсі відео.

Модуль 72 компенсації руху також може виконати інтерполяцію на основі інтерполуючих фільтрів. Модуль 72 компенсації руху може використовувати інтерполуючі фільтри, які використовувалися відеокодером 20 під час кодування відеоблоків, щоб обчислити інтерпольовані значення для субцілих пікселів опорних блоків. У цьому випадку модуль 72 компенсації руху може визначити інтерполуючі фільтри, використовувані відеокодером 20, з прийнятих синтаксичних елементів і використовувати інтерполуючі фільтри для створення блоків з прогнозуванням.

Модуль 76 зворотного квантування зворотно квантує, тобто деквантує, квантовані коефіцієнти перетворення, надані в бітовому потоці і декодовані модулем 80 ентропійного декодування. Процес зворотного квантування може включати в себе використання параметра QP_Y квантування, обчисленого відеокодером 30 для кожного відеоблока в слайсі відео, щоб визначити міру квантування, а також міру зворотного квантування, що потрібно застосувати. Модуль 78 зворотного перетворення застосовує до коефіцієнтів перетворення зворотне

перетворення, наприклад зворотне DCT, зворотне цілочислове перетворення або концептуально схожий процес зворотного перетворення, щоб створити залишкові блоки в області пікселів.

Після того, як модуль 72 компенсації руху сформує блок з прогнозуванням для поточного відеоблока на основі векторів руху і інших синтаксичних елементів, відеодекодер 30 утворює декодований відеоблок шляхом підсумовування залишкових блоків від модуля 78 зворотного перетворення з відповідними блоками з прогнозуванням, сформованими модулем 72 компенсації руху. Суматор 90 являє собою компонент або компоненти, які виконують цю операцію підсумовування. При бажанні також можна застосувати фільтр зменшення блоковості для фільтрації декодованих блоків, щоб видалити артефакти блоковості. Інші фільтри циклу (або в циклі кодування, або після циклу кодування) також можуть використовуватися для згладжування переходів пікселів або іншого підвищення якості відео. Декодовані відеоблоки в заданому кадрі або зображенні потім зберігаються в запам'ятовуючому пристрої 82 опорних зображень, який зберігає опорні зображення, використовувані для подальшої компенсації руху. Запам'ятовуючий пристрій 82 опорних зображень також зберігає декодоване відео для подальшого представлення на пристрої відображення, наприклад пристрої 32 відображення з фіг. 1.

Відповідно до методик з даного розкриття винаходу відеодекодер 30 може декодувати набір параметрів відео (VPS), набір параметрів рівня (LPS) і/або групувальний набір параметрів відповідно до методик з даного розкриття винаходу, а також набір параметрів послідовності (SPS), набір параметрів зображення (PPS), набір параметрів адаптації (APS) або інші такі структури сигнальних даних. Конкретніше, модуль 70 ентропійного декодування може конфігуруватися для декодування будь-якої або всіх цих структур даних. Внаслідок декодування цих різних структур даних модуль 70 ентропійного декодування може визначити параметри, які повинні бути використані для декодування відповідних відеоданих. Наприклад, відеодекодер 30 може декодувати відповідні послідовності відеоданих одного або більше рівнів, використовуючи параметри декодованого VPS.

Хоч і не показано на фіг. 3, відеодекодер 30 додатково може включати в себе буфер кодованих зображень (CPB). CPB звичайно надавався б перед модулем 70 ентропійного декодування. Як альтернатива CPB може сполучатися з модулем 70 ентропійного декодування для тимчасового зберігання або з виведенням модуля 70 ентропійного декодування для зберігання ентропійно декодованих даних, поки такі дані потрібно декодувати. Звичайно CPB зберігає кодовані відеодані, поки кодовані відеодані потрібно декодувати, наприклад, як указано параметрами HRD, які відеодекодер 30 може витягнути з декодованого VPS. Також інші елементи відеодекодера 30 можуть конфігуруватися для декодування відеоданих з використанням, наприклад, VPS. Наприклад, відеодекодер 30 може декодувати часові ідентифікатори для зображень різних часових рівнів, дані, що вказують число зображень, які повинні бути переупорядковані і/або збережені в запам'ятовуючому пристрої 82 опорних зображень (що представляє DPB).

Крім того, відеодекодер 30 може включати в себе додаткові модулі обробки для обробки відеоданих відповідно до різних інструментів кодування, наданих розширеннями стандарту кодування відео. Альтернативно існуючі елементи відеодекодера 30, показані на фіг. 3, можуть конфігуруватися для використання інструментів кодування з таких розширень. Модуль 70 ентропійного декодування може конфігуруватися для декодування даних розширення VPS і надання таких даних розширення модулям, сконфігурованим для виконання інструментів кодування, наданих розширеннями.

Таким чином, відеодекодер 30 з фіг. 3 представляє приклад відеодекодера, сконфігурованого для кодування набору параметрів відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з одного або більше рівнів відеоданих посиляється на VPS, і кодування одного або більше рівнів відеоданих, основуючись щонайменше частково на VPS.

Декодування VPS, хоч загалом і описане відносно відеодекодера, може виконуватися іншими пристроями, наприклад інформованим про середовище мережевим елементом (MANE). MANE може конфігуруватися для декодування VPS відповідно до методик з даного розкриття винаходу. MANE додатково може формувати дані інших наборів параметрів, наприклад один або більше наборів параметрів послідовності, використовуючи дані в VPS. Таким чином, MANE може передбачати зворотну сумісність з попередніми стандартами, наприклад H.264/AVC ITU-T.

Фіг. 4 - концептуальна схема, що ілюструє зразковий шаблон прогнозування MVC. Багатовидове кодування відео (MVC) є розширенням H.264/AVC ITU-T. Аналогічна методика може застосовуватися до HEVC. У прикладі з фіг. 4 ілюструється вісім видів (що мають ID видів

з "S0" по "S7"), і для кожного виду ілюструється дванадцять положень у часі (з "T0" по "T11"). Тобто кожний рядок на фіг. 4 відповідає виду, тоді як кожний стовпець вказує положення у часі.

На фіг. 4 показана типова структура прогнозування MVC (що включає зовнішнє прогнозування зображення в кожному виді і міжвидове прогнозування) для багатовидового кодування відео, де прогнозування вказуються стрілками, причому вказуваний об'єкт використовує вказуючий об'єкт для посилання для прогнозування. У MVC міжвидове прогнозування спирається на компенсацію диспаратності/руху, яка може використовувати синтаксис компенсації руху H.264/AVC, але дозволяє використовувати зображення в іншому виді як опорне зображення.

Кодування двох видів також могло б підтримуватися MVC, і одна з переваг MVC полягає в тому, що кодер MVC міг би приймати більше двох видів як вхід тривимірного відео, а декодер MVC міг би декодувати таке багатовидове представлення. Тому будь-який засіб візуалізації з декодером MVC може конфігуруватися для прийому тривимірного відеоконтенту більше ніж з двома видами.

Хоч MVC має так званий основний вид, який є декодованим декодерами H.264/AVC, і стереопара також могла б підтримуватися MVC, одна перевага MVC полягає в тому, що воно могло б підтримувати приклад, який використовує більше двох видів як вхід тривимірного відео і декодує це тривимірне відео, представлене декількома видами. Засіб візуалізації клієнта, що має декодер MVC, може очікувати тривимірний відеоконтент з декількома видами.

Типовий порядок декодування MVC називається кодуванням переважно за часом. Одиниця доступу може включати в себе кодовані зображення всіх видів для одного вихідного моменту часу. Наприклад, кожне із зображень моменту T0 може включатися в загальну одиницю доступу, кожне із зображень моменту T1 може включатися у другу, загальну одиницю доступу і так далі. Порядок декодування не обов'язково ідентичний порядку виведення або відображення.

Кадри на фіг. 4 вказуються на перетині кожного рядка і кожного стовпця на фіг. 4, використовуючи заштрихований блок, що включає в себе букву, яка вказує, чи має відповідний кадр внутрішнє кодування (тобто I-кадр) або зовнішнє кодування в одному напрямку (тобто P-кадр) або в декількох напрямках (тобто B-кадр). Взагалі, прогнозування вказуються стрілками, де вказуваний кадр використовує вказуючий об'єкт для посилання для прогнозування. Наприклад, P-кадр у виді S2 в положенні T0 у часі прогнозується з I-кадру у виді S0 в положенні T0 у часі.

Як і у випадку кодування відео з одиночним видом кадри у відеопослідовності багатовидового кодування відео можуть кодуватися з прогнозуванням відносно кадрів в інших положеннях у часі. Наприклад, b-кадр у виді S0 в положенні T1 у часі має стрілку, що вказує на нього від I-кадру у виді S0 в положенні T0 у часі, вказуючи, що b-кадр прогнозується з I-кадру. Однак застосовно до багатовидового кодування відео кадри можуть прогнозуватися між видами. Тобто деякий компонент виду може використовувати компоненти виду в інших видах для посилання. У MVC, наприклад, міжвидове прогнозування реалізовується, як якби компонент виду в іншому виді був посиланням для зовнішнього прогнозування. Можливі міжвидові посилання сигналізуються в розширенні набору параметрів послідовності (SPS) MVC і можуть змінюватися процесом побудови списку опорних зображень, який дає можливість гнучкого упорядкування посилань для зовнішнього прогнозування або міжвидового прогнозування.

У розширенні MVC в H.264/AVC як приклад міжвидове прогнозування спирається на компенсацію диспаратності/руху, яка використовує синтаксис компенсації руху H.264/AVC, але дозволяє використовувати зображення в іншому виді як опорне зображення. Кодування двох видів може підтримуватися MVC, що звичайно називається стереоскопічними видами. Одна з переваг MVC полягає в тому, що кодер MVC міг би приймати більше двох видів як вхід тривимірного відео, а декодер MVC міг би декодувати таке багатовидове представлення. Тому пристрій візуалізації з декодером MVC може очікувати тривимірний відеоконтент більше ніж з двома видами.

У MVC міжвидове прогнозування (IVP) дозволене серед зображень в одній і тій же одиниці доступу (тобто з однаковим моментом часу). Одиниця доступу, як правило, є одиницею даних, що включає в себе всі компоненти виду (наприклад, всі одиниці NAL) для загального моменту часу. Таким чином, в MVC міжвидове прогнозування дозволене серед зображень в одній і тій же одиниці доступу. При кодуванні зображення в одному з неосновних видів зображення може додаватися в список опорних зображень, якщо воно знаходиться в іншому виді, але з таким же моментом часу (наприклад, таким же значенням POC і, відповідно, в тій же одиниці доступу). Опорне зображення міжвидового прогнозування можна помістити в будь-яке положення списку опорних зображень, як і будь-яке опорне зображення зовнішнього прогнозування.

Застосовно до багатовидового кодування відео існує два види векторів руху. Один є звичайними векторами руху, що вказують на часові опорні зображення, і відповідний режим зовнішнього прогнозування називається прогнозуванням з компенсованим рухом (MCP). Інший є векторами диспаратності/руху, що вказують на зображення в іншому виді, і відповідний режим міжвидового прогнозування називається прогнозуванням з компенсованою диспаратністю (DCP).

У традиційному HEVC є два режими для прогнозування параметрів руху: один є режимом злиття, а інший є поліпшеним прогнозуванням вектора руху (AMVP). У режимі злиття будується список кандидатів в параметри руху (опорні зображення і вектори руху), де кандидат може бути з просторових або часових сусідніх блоків. Сусідні в просторі і у часі блоки можуть утворювати список кандидатів, тобто набір кандидатів, з якого може вибиратися інформація прогнозування руху. Відповідно, відекодер 20 може кодувати параметри руху, вибрані як інформація прогнозування руху, шляхом кодування індексу в списку кандидатів. Після того, як відеодекодер 30 декодував індекс, в режимі злиття можуть успадковуватися всі параметри руху відповідного блока, куди вказує індекс.

У AMVP, відповідно і в традиційному HEVC, список кандидатів в предиктори векторів руху для кожної гіпотези руху виводиться на основі кодованого опорного індексу. Цей список включає в себе вектори руху сусідніх блоків, які асоціюються з однаковим опорним індексом, а також предиктор часового вектора руху, який виводиться на основі параметрів руху сусіднього блока в суміщеному блоці у часовому опорному зображенні. Вибрані вектори руху сигналізуються шляхом передачі індексу в списку кандидатів. До того ж значення опорного індексу і різниці векторів руху також сигналізуються.

Фіг. 4 надає різні приклади міжвидового прогнозування. У прикладі з фіг. 4 кадри у виді S1 ілюструються як прогнозовані з кадрів в інших положеннях у часі у виді S1, а також прогнозовані між видами з кадрів у видах S0 і S2 в тих же положеннях у часі. Наприклад, b-кадр у виді S1 в положенні T1 у часі прогнозується з кожного з B-кадрів у виді S1 в положеннях T0 і T2 у часі, а також b-кадрів у видах S0 і S2 в положенні T1 у часі.

У прикладі з фіг. 4 заголовна буква "B" і мала буква "b" призначені для вказання різних ієрархічних взаємозв'язків між кадрами, а не різних методологій кодування. Взагалі, кадри з заголовною "B" знаходяться відносно вище по ієрархії прогнозування, ніж кадри з рядковою "b". Фіг. 4 також ілюструє зміни в ієрархії прогнозування, використовуючи різні рівні штриховки, де кадри з більшою величиною штриховки (тобто відносно більш темні) знаходяться вище по ієрархії прогнозування, ніж кадри, що мають менше штриховки (тобто відносно більш світлі). Наприклад, всі I-кадри на фіг. 4 ілюструються з повною штриховкою, тоді як P-кадри мають частково більш світлу штриховку, а B-кадри (і b-кадри з малою буквою) мають різні рівні штриховки один відносно одного, але завжди світліше штриховки P-кадрів і I-кадрів.

Звичайно ієрархія прогнозування стосується індексів порядку видів в тому, що кадри відносно вище по ієрархії прогнозування потрібно декодувати перед декодуванням кадрів, які знаходяться відносно нижче по ієрархії, так що кадри відносно вище по ієрархії можуть використовуватися як опорні кадри під час декодування кадрів відносно нижче по ієрархії. Індекс порядку видів є індексом, який вказує порядок декодування компонентів виду в одиниці доступу. Індеси порядку видів мають на увазі в розширенні SPS MVC, яке точно визначене Додатком H до H.264/AVC (поправка MVC). У SPS для кожного індексу і сигналізується відповідний view_id. У деяких прикладах декодування компонентів виду повинно дотримуватися зростаючого порядку індексу порядку видів. Якщо представляються всі види, то індекси порядку видів знаходяться в послідовному порядку від 0 до num_views_minus_1.

Таким чином, кадри, використовувані як опорні кадри, можуть декодуватися перед декодуванням кадрів, які кодуються з посиланням на опорні кадри. Індекс порядку видів є індексом, який вказує порядок декодування компонентів виду в одиниці доступу. Для кожного індексу і порядку видів сигналізується відповідний view_id. Декодування компонентів виду дотримується зростаючого порядку індексів порядку видів. Якщо представляються всі види, то набір індексів порядку видів може містити послідовно впорядкований набір від нуля до повного числа видів, меншого на одиницю.

Для деяких кадрів на рівних рівнях ієрархії порядок декодування один відносно одного може не мати значення. Наприклад, I-кадр у виді S0 в положенні T0 у часі використовується як опорний кадр для P-кадру у виді S2 в положенні T0 у часі, який, в свою чергу, використовується як опорний кадр для P-кадру у виді S4 в положенні T0 у часі. Відповідно, I-кадр у виді S0 в положенні T0 у часі потрібно декодувати перед P-кадром у виді S2 в положенні T0 у часі, який потрібно декодувати перед P-кадром у виді S4 в положенні T0 у часі. Однак між видами S1 і S3 порядок декодування не має значення, оскільки види S1 і S3 не спіраються один на одного для

прогнозування, а замість цього прогнозуються тільки з видів, які вище по ієрархії прогнозування. Крім того, вид S1 можна декодувати перед видом S4 при умові, що вид S1 декодується після видів S0 і S2.

Таким чином, ієрархічне упорядкування може використовуватися для опису видів S0-S7. Нехай нотація $SA \succ SB$ означає, що вид SA потрібно декодувати перед видом SB. Використовуючи цю нотацію, в прикладі з фіг. 4 $S0 \succ S2 \succ S4 \succ S6 \succ S7$. Також відносно прикладу фіг. 4 $S0 \succ S1$, $S2 \succ S1$, $S2 \succ S3$, $S4 \succ S3$, $S4 \succ S5$ і $S6 \succ S5$. Можливий будь-який порядок декодування для видів, який не порушує ці вимоги. Відповідно, можливі багато які різні порядки декодування, тільки з деякими обмеженнями.

Відповідно до методик з даного розкриття винаходу кожний з видів S0-S7 може вважатися відповідним рівнем відповідного бітового потоку. Таким чином, VPS може описувати параметри бітового потоку, застосовні до будь-якого або всіх видів S0-S7, хоч окремі набори параметрів рівнів можуть надаватися для будь-якого або всіх видів S0-S7. До того ж групувальний набір параметрів може надаватися для групи наборів параметрів, так що слайси в окремих зображеннях у видах S0-S7 можуть просто посилатися на ідентифікатор групувального набору параметрів.

Як показано на фіг. 4, компонент виду може використовувати компоненти виду в інших видах для посилання. Це називається міжвидовим прогнозуванням. У MVC міжвидове прогнозування реалізовується, як яби компонент виду в іншому виді був посиланням для зовнішнього прогнозування. Відеокодер 20 і відеодекодер 30 можуть кодувати можливі міжвидові посилання в розширенні набору параметрів послідовності (SPS) MVC (як показано в прикладі Таблиці 1). Відеокодер 20 і відеодекодер 30 додатково можуть змінювати можливі міжвидові посилання шляхом виконання процесу побудови списку опорних зображень, який може зробити можливим гнучке упорядкування посилань для зовнішнього прогнозування або міжвидового прогнозування.

Таблиця 1

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_signaled_minus1	0	ue(v)
for(i = 0; i <= num_level_values_signaled_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)
}		
}		
}		

У розширенні SPS MVC, показаному в Таблиці 1, для кожного виду сигналізується число видів, яке може використовуватися для утворення Списку 0 опорних зображень і Списку 1 опорних зображень. Взаємозв'язок прогнозування для опорного зображення, яке сигналізується в розширенні SPS MVC, може відрізнятися від взаємозв'язку прогнозування для неопорного зображення (сигналізованого в розширенні SPS MVC) в тому ж виді.

Стандарти кодування відео включають в себе H.261 ITU-T, MPEG-1 Visual ISO/IEC, H.262 ITU-T або MPEG-2 Visual ISO/IEC, H.263 ITU-T, MPEG-4 Visual ISO/IEC і H.264 ITU-T (також відомий як MPEG-4 AVC ISO/IEC), включаючи його розширення Масштабованого кодування відео (SVC) і Багатовидового кодування відео (MVC).

До того ж є новий стандарт кодування відео, а саме Високоєфективне кодування відео (HEVC), розроблюваний Об'єднаною командою по кодуванню відео (JCT-VC) з Експертної групи в галузі кодування відео (VCEG) ITU-T і Експертної групи по рухомих зображеннях (MPEG) ISO/IEC. Останній робочий варіант (WD) HEVC, в подальшому званий WD4 HEVC, доступний за адресою http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F803-v3.zip, позначений як WD4d1 HEVC.

Механізм набору параметрів послідовності і зображення відділяє передачу інформації, що нечасто змінюється, від передачі даних кодованого блока. Набори параметрів послідовності і зображення в деяких застосуваннях можуть передаватися "поза смугою", використовуючи надійний транспортний механізм. Корисне навантаження необробленої послідовності байтів (RBSP) набору параметрів зображення може включати в себе параметри, на які можуть посилатися одиниці рівня абстракції мережі (NAL) кодованого слайсу в одному або більше кодованих зображеннях. RBSP набору параметрів послідовності може включати в себе параметри, на які можуть посилатися одне або більше RBSP набору параметрів зображення або одна або більше одиниць NAL додаткової інформації розширення (SEI), що містять повідомлення SEI про період буферизації. RBSP набору параметрів послідовності може включати в себе параметри, на які можуть посилатися одне або більше RBSP набору параметрів зображення або одна або більше одиниць NAL SEI, що містять повідомлення SEI про період буферизації.

Набір параметрів послідовності може включати в себе необов'язковий набір параметрів, називаний інформацією про застосовність відео (VUI). VUI може включати в себе наступні три категорії необов'язкової інформації: інформація про представлення відео, інформація про гіпотетичний еталонний декодер (HRD) і інформація про обмеження бітового потоку. Інформація про представлення відео включає в себе співвідношення сторін, пов'язану з перетворенням кольорного простору інформацію, фазові зсуви кольоровості відносно яскравості і частоту кадрів. HRD включає в себе параметри буферизації відео для кодованих відеопослідовностей. Інформація про обмеження бітового потоку включає в себе обмеження по діапазону векторів руху, розміру буфера декодованих зображень (DPB), числу переупорядковуваних кадрів і кодованих розмірах блоків (наприклад, макроблоків або одиниць кодування (CU)) і зображень.

WD5 HEVC включає в себе підтримання набору параметрів адаптації (APS). Поняття набору параметрів адаптації (APS) також можна знайти в JCTVC-F747, доступному за адресою http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F747-v4.zip.

Уніфікований заголовок одиниці NAL може використовуватися для немасштабованих бітових потоків HEVC, а також для масштабованих бітових потоків, відповідних можливим масштабованим або багатовидовим розширенням HEVC. Уніфікований заголовок одиниці NAL може відрізнятися від діючого заголовка одиниці NAL HEVC в наступних особливостях: може мати місце фіксована довжина заголовка одиниці NAL для однієї повністю кодованої відеопослідовності, тоді як довжина може мінятися між різними кодованими відеопослідовностями, і ефективне кодування синтаксичних елементів масштабованості в заголовку одиниці NAL, і коли конкретний синтаксичний елемент не потрібен, йому не потрібно бути присутнім. При такому виконанні різний тип одиниці NAL або набір параметрів може використовуватися для усього бітового потоку.

Фіг. 5 - концептуальна схема, що ілюструє набір параметрів відео (VPS) і різні набори параметрів рівнів (LPS). Три крапки після другого LPS на фіг. 5 призначені для вказання, що може мати місце будь-яке число N VPS, де N - ціле число. Наприклад, кожний рівень (наприклад, кожний рівень SVC або кожний вид MVC) може мати відповідний LPS. Кодер відео, наприклад відеокoder 20 або відеодекoder 30, може конфігуруватися для кодування VPS і одного або більше LPS, наприклад, проілюстрованих на фіг. 5.

Нижче Таблиця 2 надає зразковий синтаксис корисного навантаження необробленої послідовності байтів (RBPS) для VPS.

Таблиця 2

video_parameter_set_rbsp() {	Дескриптор
video_para_set_id	u(8)
// sample dimension counter description	
cnt_p	u(3)
cnt_d	u(3)
cnt_t	u(3)
cnt_q	u(3)
cnt_v	u(4)
cnt_f	u(4)
// sample index to characteristic mapping	
for(i = 0; i < cnt_d; i++) {	
pic_width_in_luma_samples[i]	ue(v)
pic_height_in_luma_samples[i]	ue(v)
bit_depth_luma_minus8[i]	ue(v)
bit_depth_chroma_minus8[i]	ue(v)
chroma_format_idc[i]	u(2)
}	
for(i = 0; i < cnt_t; i++)	
average_frame_rate[i]	u(16)
if(cnt_v > 1)	
for(i=0; i < cnt_v; i++)	
view_id[i]	ue(v)
// control parameters and tool enabling/disabling flags	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
chroma_pred_from_luma_enabled_flag	u(1)
loop_filter_across_slice_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
pcm_loop_filter_disable_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
temporal_id_nesting_flag	u(1)
inter_4x4_enabled_flag	u(1)
operation_point_description()	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbbsp_trailing_bits()	
}	

Кодери відео можуть конфігуруватися так, що кодована відеопослідовність (наприклад, бітовий потік, що включає в себе один або більше рівнів) може мати тільки один активний набір параметрів відео (VPS). VPS можна помістити в одиницю NAL конкретного типу. Наприклад,

5

nal_unit_type для RBSP VPS може дорівнювати 10. Зразкова семантика для VPS з Таблиці 2 описується нижче.

У цьому прикладі video_para_set_id ідентифікує відповідний набір параметрів відео (VPS).

У цьому прикладі cnt_p точно визначає максимальне число значень priority_id, присутніх у відповідній кодованій відеопослідовності.

У цьому прикладі cnt_d точно визначає максимальне число рівнів залежності, присутніх у відповідній кодованій відеопослідовності. Декілька видів з однаковим розрізненням можуть вважатися такими, що належать до одного рівня залежності. Два рівні залежності можуть мати однакове просторове розрізнення.

У цьому прикладі cnt_t точно визначає максимальне число часових рівнів, присутніх в кодованій відеопослідовності.

У цьому прикладі cnt_q точно визначає максимальне число рівнів якості, присутніх на деякому рівні залежності в кодованій відеопослідовності.

У цьому прикладі cnt_v точно визначає максимальне число видів, присутніх в кодованій відеопослідовності.

У цьому прикладі cnt_f точно визначає число бітів, використовуване для представлення синтаксичного елемента reserved_flags в заголовку одиниці NAL.

У цьому прикладі pic_width_in_luma_samples[i] і pic_height_in_luma_samples[i] точно визначають відповідно ширину і висоту розрізнення i-го рівня залежності в одиницях вибірок яскравості.

У цьому прикладі bit_depth_luma_minus8[i] плюс 8 і bit_depth_chroma_minus8[i] плюс 8 точно визначають бітову глибину компонентів яскравості і кольоровості у i-го представлення бітової глибини.

У цьому прикладі chroma_format_idc[i] точно визначає формат вибірки кольоровості у i-го представлення формату вибірки кольоровості. Наприклад, значення, що дорівнює 0, може вказувати 4:2:0; значення, що дорівнює 1, може вказувати 4:4:4, значення, що дорівнює 2, може вказувати 4:2:2, і значення, що дорівнює 3, може вказувати 4:0:0.

У цьому прикладі average_frame_rate[i] точно визначає середню частоту кадрів у i-го представлення часового рівня в одиницях кадрів в 256 секунд.

У цьому прикладі view_id[i] точно визначає ідентифікатор виду у i-го виду, який має індекс порядку видів, що дорівнює i. Коли відсутній, значення view_id[0] можна передбачити рівним 0. vps_extension_flag, що дорівнює 0, точно визначає, що ніякі синтаксичні елементи vps_extension_data_flag не присутні в структурі синтаксису RBSP набору параметрів відео. vps_extension_flag може дорівнювати 0 в бітових потоках, відповідних майбутньому стандарту HEVC. Значення 1 для vps_extension_flag може бути зарезервоване, наприклад, для майбутнього використання ITU-T|ISO/IEC. Декодер, наприклад відеодекодер 30, можуть ігнорувати всі дані, які ідуть за значенням 1 для vps_extension_flag в одиниці NAL набору параметрів відео.

У цьому прикладі vps_extension_data_flag може мати будь-яке значення. Це не впливає на відповідність профілям, точно визначеним в майбутньому стандарті HEVC, а передбачає подальший розвиток майбутнього стандарту.

Інші синтаксичні елементи в VPS можуть мати таку ж семантику, як і синтаксичні елементи з такими ж назвами в SPS поточного робочого варіанта HEVC. Ці синтаксичні елементи можуть застосовуватися до кодованої відеопослідовності, яка посилається на цей VPS, поки не перезаписані наборами параметрів більш низького рівня.

У деяких прикладах в VPS може додатково сигналізуватися 3DV_flag для вказання, чи присутня глибина в кодованій відеопослідовності.

У деяких прикладах параметри VUI сигналізуються в LPS.

У деяких прикладах синтаксичні елементи cnt_p, cnt_t, cnt_d, cnt_q і cnt_v точно визначають числа бітів, використовувані для кодування priority_id, temporal_id, dependency_id, quality_id і view_idx, відповідно, і максимальні числа значень priority_id, часових рівнів, рівнів залежності, рівнів якості і видів, присутніх в кодованих відеопослідовностях, також можуть сигналізуватися в VPS.

У деяких прикладах можна ввести інший тип одиниці NAL, щоб він містив синтаксичні елементи cnt_p, cnt_t, cnt_d, cnt_q, cnt_v і cnt_f. Цей новий тип одиниці NAL також може включати в себе ідентифікатор (ID), і на цей ID можуть посилатися в VPS.

У деяких прикладах синтаксичні елементи від log2_max_pic_order_cnt_lsb_minus4 до inter_4x4_enabled_flag в Таблиці 2 не сигналізуються в VPS, а замість цього відеокодер 20 і відеодекодер 30 можуть кодувати ці синтаксичні елементи в LPS.

У деяких прикладах структура синтаксису `operation_point_description()` з Таблиці 2 не включається в VPS; замість цього відеокoder 20 і відеокoder 30 або інші елементи (наприклад, інтерфейс 22 виведення і/або інтерфейс 28 введення) можуть кодувати контент в структурі синтаксису `operation_point_description()` в повідомленні з додатковою інформацією розширення (SEI).

У деяких прикладах відеокoder 20 і/або відеокoder 30 можуть кодувати параметри інформації про застосовність відео (VUI) в VPS. Наприклад, VPS може включати в себе дані, що точно визначають інформацію про обмеження бітового потоку, наприклад обмеження по діапазону векторів руху, розміру DPB, числу переупорядковуваних кадрів і кодованих розмірах блоків (наприклад, макроблоків або CU) і зображень. Таким чином, VPS може точно визначати інформацію, що вказує необхідний розмір DPB, щоб відеокoder (наприклад, відеокoder 30) належно декодував відповідний бітовий потік, тобто бітовий потік, що включає в себе VPS. Також VPS може точно визначати інформацію про переупорядковування зображень, тобто число зображень, яке може передувати даному зображенню в порядку декодування і яке іде за даним зображенням в порядку виведення (тобто порядку відображення).

Додатково або альтернативно VPS може включати в себе дані, що точно визначають інформацію про гіпотетичний еталонний декодер (HRD). Як відмічалось вище, відеокoder 20 і/або відеокoder 30 можуть кодувати (тобто сигналізувати) в VPS параметри VUI, які можуть включати в себе інформацію про HRD. Таким чином, VPS може включати в себе дані, які описують, наприклад, робочі точки відповідного бітового потоку. Наприклад, VPS може включати в себе дані, які описують одне або більше з числа максимальних робочих точок, залежності між різними рівнями або видами, інформації про профіль і рівень для кожної робочої точки, представлення одиниці NAL VCL робочої точки для кожної робочої точки, бітрейту для кожної робочої точки, залежності між робочими точками, обмежень для кожної робочої точки, VUI або часткової VUI для кожної робочої точки і/або VUI або часткової VUI для кожного рівня або виду.

VPS також може включати в себе для кожного вимірювання: специфічне значення індексу, діапазон значень індексу або список значень індексу. Наприклад, коли VPS включає в себе дані, які описують специфічне значення індексу, значення індексу може, для просторового розрізнення, відповідати бітній глибині для формату дискретизації кольоровості. Як інший приклад, коли VPS включає в себе діапазон значень індексу, для часових рівнів цей діапазон може містити від нуля (0) до ID найвищого часового рівня, а для рівнів якості діапазон може містити від нуля (0) до ID найвищого рівня якості. Як ще один приклад, коли VPS включає в себе дані, які описують список значень індексу, цей список може містити список значень індексів виду для декількох видів.

У деяких прикладах відеокoder 20 може кодувати (тобто сигналізувати), а відеокoder може декодувати один або більше параметрів формату представлення (ширина, висота, бітова глибина і т. п.), і можуть мати місце різні набори параметрів формату представлення. Рівень або робоча точка тоді можуть посилатися на індекс такого набору параметрів формату представлення. Приклад проекту синтаксису для такого набору показаний в Таблиці 3 нижче.

Таблиця 3

num_rep_formats_minus1	ue(v)
for(i = 0; i <= num_rep_formats_minus1; i++) {	
pic_width_in_luma_samples[i]	ue(v)
pic_height_in_luma_samples[i]	ue(v)
bit_depth_luma_minus8[i]	ue(v)
bit_depth_chroma_minus8[i]	ue(v)
chroma_format_idc[i]	u(2)
}	
for(i = 0; i < cnt_d; i++) {	
rep_format_idx[i]	ue(v)
}	

У деяких прикладах замість цього може сигналізуватися `ref_format_idx` в наборі параметрів рівня.

Нижче Таблиця 4 надає зразковий синтаксис для описів робочих точок.

5

Таблиця 4

<code>operation_points_description() {</code>	Descriptor
<code>num_operation_point_minus1</code>	<code>ue(v)</code>
<code>for(i = 0; i <= num_operation_points_minus1; i++) {</code>	
<code>op_profile_level_idc[i]</code>	<code>u(24)</code>
<code>operation_point_id[i]</code>	<code>ue(v)</code>
<code>priority_id[i]</code>	<code>ue(v)</code>
<code>temporal_id[i]</code>	<code>ue(v)</code>
<code>quality_id[i]</code>	<code>ue(v)</code>
<code>dependency_id[i]</code>	<code>ue(v)</code>
<code>if(cnt_v > 1) {</code>	
<code>num_target_output_views_minus1[i]</code>	<code>ue(v)</code>
<code>for(j = 0; j <= num_target_output_views_minus1[i]; j++)</code>	
<code>view_idx[i][j]</code>	<code>ue(v)</code>
<code>}</code>	
<code>frm_rate_info_present_flag[i]</code>	<code>u(1)</code>
<code>avg_bitrate[i]</code>	<code>u(16)</code>
<code>max_bitrate[i]</code>	<code>u(16)</code>
<code>max_bitrate_calc_window[i]</code>	<code>u(16)</code>
<code>constant_frm_rate_idc[i]</code>	<code>u(2)</code>
<code>if(cnt_v > 1) {</code>	
<code>num_directly_dependent_views[i]</code>	
<code>for(j = 0; j < num_directly_dependent_views[i]; j++) {</code>	
<code>directly_dependent_view_idx[i][j]</code>	
<code>}</code>	
<code>}</code>	
<code>if(cnt_v > 1)</code>	
<code>for(i = 1; i < cnt_v; i++) {</code>	
<code>num_ref_views[i]</code>	<code>ue(v)</code>
<code>for(j = 0; j < num_ref_views[i]; j++)</code>	
<code>ref_view_idx[i][j]</code>	<code>ue(v)</code>
<code>}</code>	
<code>}</code>	

Приклади семантики для синтаксичних елементів з Таблиці 4 обговорюються нижче.

У цьому прикладі `num_operation_point_minus1` плюс 1 точно визначає максимальне число робочих точок, які присутні в кодованій відеопослідовності і для яких інфоДескриптор робочі точки сигналізується наступними синтаксичними елементами.

10

У цьому прикладі `op_profile_level_idc[i]`, `operation_point_id[i]`, `priority_id[i]`, `num_target_output_views_minus1[i]`, `frm_rate_info_present_flag[i]`, `avg_bitrate[i]`, `max_bitrate[i]`, `max_bitrate_calc_window[i]`, `constant_frm_rate_idc[i]` і `num_directly_dependent_views[i]` можуть мати таку ж семантику, як і синтаксичні елементи з такими ж назвами в повідомленні SEI з інформацією про масштабованість виду з H.264.

15

У цьому прикладі `quality_id[i]` і `dependency_id[i]` можуть мати таку ж семантику, як і синтаксичні елементи з такими ж назвами в повідомленні SEI з інформацією про масштабованість з H.264.

5 У цьому прикладі `directly_dependent_view_idx[i][j]` точно визначає індекс виду у *j*-го виду, від якого безпосередньо залежить цільовий вихідний вид поточної робочої точки в рамках представлення поточної робочої точки.

10 У цьому прикладі `num_ref_views[i]` точно визначає число компонентів виду для міжвидового прогнозування в початковому списку `RefPicList0` і `RefPicList1` опорних зображень при декодуванні компонентів виду з індексом порядку видів, що дорівнює *i*. В цьому прикладі значення `num_ref_views[i]` не повинно бути більше `Min(15, num_views_minus1)`. У деяких прикладах значення `num_ref_views[0]` дорівнює 0.

15 У цьому прикладі `ref_view_idx[i][j]` точно визначає індекс порядку видів у *j*-го компонента виду для міжвидового прогнозування в початковому списку `RefPicList0` і `RefPicList1` опорних зображень при декодуванні компонента виду з індексом порядку видів, що дорівнює *i*. В цьому прикладі значення `ref_view_idx[i][j]` повинно бути в діапазоні від 0 до 31, включно.

У деяких прикладах як альтернатива деякі з синтаксичних елементів в повідомленні SEI з інформацією про масштабованість (наприклад, яке описане в H.264), наприклад синтаксичні елементи, пов'язані з інформацією про залежність рівнів, можуть включатися в структуру синтаксису `operation_points_description()` з Таблиці 4.

20 У деяких прикладах відеокідер 20 і/або відеокідер 30 можуть кодувати (тобто сигналізувати) деякі параметри VUI в структурі синтаксису `operation_points_description()` з Таблиці 4.

Нижче Таблиця 5 надає альтернативний синтаксис для набору параметрів відео.

Таблица 5

video_parameter_set_rbsp() {	Дескриптор
video_parameter_set_id	ue(v)
num_temporal_layers_minus1	u(3)
for (i = 0; i <= num_temporal_layers_minus1; i++) {	
profile_idc[i]	u(8)
reserved_zero_8bits[i] /* equal to 0 */	u(8)
level_idc[i]	u(8)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
chroma_format_idc	u(2)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pic_cropping_flag	u(1)
if(pic_cropping_flag) {	
pic_crop_left_offset	ue(v)
pic_crop_right_offset	ue(v)
pic_crop_top_offset	ue(v)
pic_crop_bottom_offset	ue(v)
}	
temporal_id_nesting_flag	u(1)
bit_equal_to_one /* equal to 1 */	u(1)
extension_type /* equal to 0 for 3DV */	ue(v)
num_layers_minus2	ue(v)
num_rep_formats_minus1	ue(v)
for(i = 1; i <= num_rep_formats_minus1; i++) {	
bit_depth_luma_minus8[i]	ue(v)
bit_depth_chroma_minus8[i]	ue(v)
chroma_format_idc[i]	u(2)
pic_width_in_luma_samples[i]	ue(v)
pic_height_in_luma_samples[i]	ue(v)
pic_cropping_flag[i]	u(1)
if(pic_cropping_flag[i]) {	
pic_crop_left_offset[i]	ue(v)
pic_crop_right_offset[i]	ue(v)
pic_crop_top_offset[i]	ue(v)
pic_crop_bottom_offset[i]	ue(v)
}	
}	
for(i = 1; i <= num_layers_minus1; i++) {	
rep_format_idx[i]	ue(v)
if(extension_type == 1) {	
dependency_id[i]	ue(v)
quality_id[i]	ue(v)
num_directly_dependent_layers[i]	ue(v)

for(j = 0; j < num_directly_dependency_layers[i]; j++)	
delta_reference_layer_id_minus1[i][j]	ue(v)
}	
}	
num_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i)	
if(extension_type == 0)	
view_dependency()	
num_additional_profiles_levels_minus1	ue(v)
for(i = 0; i <= num_additional_profiles_levels_minus1; i++) {	
additional_profile_idc[i]	u(8)
additional_reserved_zero_8bits[i] /* equal to 0 */	u(8)
additional_level_idc[i]	u(8)
num_applicable_operation_points_minus1[i]	ue(v)
for(j = 0; j <= num_applicable_operation_points[i]; j++) {	
temporal_id[i][j]	ue(v)
layer_id[i][j]	ue(v)
if(extension_type == 0) { /* Always true for 3DV */	
depth_included_flag	u(1)
num_target_output_views_minus1[i][j]	ue(v)
for(k = 0; k < num_target_output_views_minus1[i][j]; k++)	
layer_id[i][j][k]	ue(v)
}	
else(extension_type == 1)	
layer_id[i][j]	ue(v)
}	
}	
vps_vui_parameters_present_flag	u(1)
if(vps_vui_parameters_present_flag)	
vps_vui_parameters()	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Приклади семантики для синтаксису набору параметрів відео з Таблиці 5 обговорюються нижче. Взагалі, аналогічно названі синтаксичні елементи, які не обговорюються нижче, можуть мати таку ж семантику, як обговорювалася вище відносно Таблиці 2. Семантика для інших синтаксичних елементів може бути наступною.

5

У цьому прикладі bit_equal_to_one дорівнює 1 (тобто значенню двійкової "1").

У цьому прикладі extension_type, що дорівнює 0, вказує, що декілька рівнів виду може бути присутньо в бітовому потоці. У цьому прикладі extension_type, що дорівнює 1, точно визначає, що декілька рівнів залежності і/або якості може бути присутньо в бітовому потоці.

10

У цьому прикладі num_rep_formats_minus1 плюс 1 точно визначає максимальне число різних наборів форматів представлення, підтримуваних цим набором параметрів відео, причому формат представлення включає в себе бітову глибину і формат кольоровості (тобто набори із значень bit_depth_luma_minus8, bit_depth_chroma_minus8 і chroma_format_idc), розрізнення зображення і інформацію про вікно обрізання в кодованій відеопослідовності. Значення num_rep_formats_minus1 може бути в діапазоні від 0 до X, включно. Відекодер 20 і

15

відеодекодер 30 можуть кодувати набір з бітової глибини і формату кольоровості для основного рівня за допомогою `bit_depth_luma_minus8`, `bit_depth_chroma_minus8` і `chroma_format_idc`, і набори з бітової глибини і формату кольоровості сигналізуються для рівнів розширення за допомогою наступного набору синтаксичних елементів: `bit_depth_luma_minus8[i]`, `bit_depth_chroma_minus8[i]` і `chroma_format_idc[i]`.

Відеокодер 20 і відеокодер 30 можуть кодувати перший набір формату представлення за допомогою `bit_depth_luma_minus8`, `bit_depth_chroma_minus8`, `chroma_format_idc`, `pic_width_in_luma_samples`, `pic_height_in_luma_samples`, `pic_cropping_flag`, `pic_crop_left_offset`, `pic_crop_right_offset`, `pic_crop_top_offset` і `pic_crop_bottom_offset`.

У цьому прикладі `bit_depth_luma_minus8[i]`, `bit_depth_chroma_minus8[i]` і `chroma_format_idc[i]` точно визначають відповідно *i*-ий набір значень `bit_depth_luma_minus8`, `bit_depth_chroma_minus8` і `chroma_format_idc` в кодованій відеопослідовності.

У цьому прикладі `pic_width_in_luma_samples[i]` і `pic_height_in_luma_samples[i]` точно визначають відповідно ширину і висоту кожного декодованого зображення в одиницях вибірок яскравості, використовуючи *i*-ий формат представлення.

У цьому прикладі `pic_cropping_flag[i]`, `pic_crop_left_offset[i]`, `pic_crop_right_offset[i]`, `pic_crop_top_offset[i]` і `pic_crop_bottom_offset[i]` точно визначають для *i*-го набору формату представлення вибірки зображень в кодованій відеопослідовності, які виводяться з процесу декодування, у вигляді прямокутної області, точно визначеної координатами зображення для виведення.

У цьому прикладі `rep_format_idx[i]` точно визначає індекс значень до набору з додаткової бітової глибини і формату кольоровості, який застосовується до рівня з `layer_id`, що дорівнює *i*. Значення `bit_depth_luma_minus8`, `bit_depth_chroma_minus8` і `chroma_format_idc` для рівня з `layer_id`, що дорівнює *i*, можуть дорівнювати `bit_depth_luma_minus8[rep_format_idx[i]]`, `bit_depth_chroma_minus8[rep_format_idx[i]]` і `chroma_format_idc[rep_format_idx[i]]`, відповідно. Значення `rep_format_idx[i]` повинно бути в діапазоні від 0 до *X*, включно.

У цьому прикладі `dependency_id[i]` точно визначає ідентифікатор залежності для рівня з `layer_id`, що дорівнює *i*. `dependency_id[i]` може бути в діапазоні від 0 до *X*, включно. Коли відсутній, `dependency_id[i]` можна передбачити рівним 0. Коли `num_directly_dependent_layers[i]` більше 0, `dependency_id[i]` може бути більше або дорівнювати ідентифікатору залежності будь-якого рівня, від якого залежить рівень з `layer_id`, що дорівнює *i*.

У цьому прикладі `quality_id[i]` точно визначає ідентифікатор якості для рівня з `layer_id`, що дорівнює *i*. `quality_id[i]` може бути в діапазоні від 0 до *X*, включно. Коли відсутній, `quality_id[i]` можна передбачити рівним 0. Коли `num_directly_dependent_layers[i]` більше 0, `quality_id[i]` може бути більше або дорівнювати ідентифікатору якості будь-якого рівня, від якого залежить рівень з `layer_id`, що дорівнює *i*, і який має ідентифікатор залежності, що дорівнює `dependency_id[i]`.

У цьому прикладі `num_short_term_ref_pic_sets` точно визначає число короткочасних наборів опорних зображень, які точно визначаються в наборі параметрів відео. Значення `num_short_term_ref_pic_sets` може бути в діапазоні від 0 до 64, включно.

У цьому прикладі `depth_included_flag`, що дорівнює 1, вказує, що поточна робоча точка 3DV містить глибину. У цьому прикладі `depth_included_flag`, що дорівнює 0, вказує, що поточна робоча точка 3DV не містить глибину.

Зразковий синтаксис для елемента залежності видів з Таблиці 5 надається нижче в Таблиці 6.

Таблиця 6

view_dependency() {	
num_views_minus1	ue(v)
for(i = 0; i <= num_views_minus1; i++)	
view_id[i]	ue(v)
for(i = 1; i <= num_views_minus1; i++) {	
num_ref_views[i]	
for(j = 0; j < num_ref_views[i]; j++)	
ref_view_idx[i][j]	ue(v)
inter_view_texture_flag[i][j]	u(1)
}	
}	
}	

Нижче Таблиця 7 задає зразковий набір даних, в якому залежність видів у кожного неосновного виду безпосередньо сигналізується на рівні послідовності.

5

Таблиця 7

for(i = 1; i <= num_views_minus1; i++) {	
num_ref_views[i]	ue(v)
for(j = 0; j < num_ref_views[i]; j++)	
ref_view_idx[i][j]	ue(v)
}	

У цьому прикладі num_ref_views[i] точно визначає число компонентів виду для міжвидового прогнозування в початковому списку RefPicList0 і RefPicList1 опорних зображень при декодуванні компонентів виду з індексом порядку видів, що дорівнює i. В цьому прикладі значення num_ref_views[i] не більше Min(15, num_views_minus1). У цьому прикладі значення num_ref_views[0] дорівнює 0.

10

У цьому прикладі ref_view_idx[i][j] точно визначає індекс порядку видів у j-го компонента виду для міжвидового прогнозування в початковому списку RefPicList0 і RefPicList1 опорних зображень при декодуванні компонента виду з індексом порядку видів, що дорівнює i. В цьому прикладі значення ref_view_idx[i][j] знаходиться в діапазоні від 0 до 31, включно.

15

Як відмічалось вище, конкретний тип одиниці NAL (наприклад, тип 10 одиниці NAL) може використовуватися для поміщення в неї набору параметрів відео. Синтаксис одиниці NAL можна змінити, як показано нижче в прикладі з Таблиці 8.

Таблиця 8

nal_unit(NumBytesInNALunit) {	Дескриптор
forbidden_zero_bit	f(1)
nal_ref_flag	u(1)
nal_unit_type	u(6)
NumBytesInRBSP = 0	
nalUnitHeaderBytes = 1	
if(nal_unit_type != 10) { // not VPS NAL unit	
if(cnt_p > 1)	
priority_id	u(v)
if(cnt_t > 1)	
temporal_id	u(v)
reserved_one_bit	u(1)
if(cnt_d > 1)	
dependency_id	u(v)
if(cnt_q > 1)	
quality_id	u(v)
reserved_one_bit	u(1)
if(cnt_v > 1)	
view_idx	u(v)
if(cnt_f)	
reserved_flags	u(v)
m = Ceil(log2(cnt_p)) + Ceil(log2(cnt_t)) + Ceil(log2(cnt_d)) + Ceil(log2(cnt_q)) + Ceil(log2(cnt_v)) + cnt_f + 2	
if(((m + 7 >> 3) << 3) - m)	
reserved_bits	u(v)
nalUnitHeaderBytes += ((m + 7) >> 3)	
}	
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {	
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte[NumBytesInRBSP++]	b(8)
rbsp_byte[NumBytesInRBSP++]	b(8)
i += 2	
emulation_prevention_three_byte /* equal to 0x03 */	f(8)
} else	
rbsp_byte[NumBytesInRBSP++]	b(8)
}	
}	

У цьому прикладі додаються елементи всередині оператора "if(nal_unit_type!= 10)" відносно традиційного синтаксису одиниці NAL. У цьому прикладі число бітів, використовуване для

сигналізації синтаксичних елементів `priority_id`, `temporal_id`, `dependency_id`, `quality_id` і `view_idx`, дорівнює $\text{Ceil}(\log_2(\text{cnt}_p))$, $\text{Ceil}(\log_2(\text{cnt}_t))$, $\text{Ceil}(\log_2(\text{cnt}_d))$, $\text{Ceil}(\log_2(\text{cnt}_q))$ і $\text{Ceil}(\log_2(\text{cnt}_v))$, відповідно. Також в цьому прикладі, коли відсутній будь-який з синтаксичних елементів `priority_id`, `temporal_id`, `dependency_id`, `quality_id` і `view_idx`, значення цього синтаксичного

5 елемента передбачається рівним 0.

За винятком заданого вище відносно числа бітів і висновків, які можна зробити, семантику для синтаксичних елементів з Таблиці 8 можна задати наступним чином. Семантика `priority_id`, `dependency_id` і `quality_id` може бути такою, як задана в розширенні SVC в H.264/AVC ITU-T. Семантика `temporal_id` може бути такою, як задана в WD4 в HEVC. У цьому прикладі

10 `reserved_one_bit` дорівнює 1. Значення 0 для `reserved_one_bit` може бути точно визначене майбутніми розширеннями стандарту HEVC. Декодер, наприклад відеодекодер 30, можуть конфігуруватися для ігнорування значення `reserved_one_bit`.

У цьому прикладі `view_idx` точно визначає індекс порядку видів для деякого виду. Семантика `view_idx` може бути такою ж, як синтаксичний елемент "індекс порядку видів", який точно

15 визначений в розширенні MVC в H.264/AVC ITU-T.

У цьому прикладі кожний біт у `reserved_flags` дорівнює 1. Інші значення для `reserved_flags` можуть бути точно визначені майбутніми розширеннями майбутнього стандарту HEVC. Декодер, наприклад відеодекодер 30, можуть конфігуруватися для ігнорування значення `reserved_flags`, поки не сконфігуровані для роботи відповідно до розширення, яке призначає

20 семантику бітам у `reserved_flags`. У цьому прикладі число бітів, використовуване для представлення `reserved_flags`, дорівнює `reserved_flags_len`.

У цьому прикладі кожний біт у `reserved_bits` дорівнює 1. Інші значення для `reserved_bits` можуть бути точно визначені майбутнім розширенням майбутнього стандарту HEVC. Декодер, наприклад відеодекодер 30, можуть конфігуруватися для ігнорування значення `reserved_bits`, знову поки не сконфігуровані відповідно до такого майбутнього розширення. Число бітів,

25 використовуване для представлення `reserved_bits`, в цьому прикладі дорівнює $((m+7 \gg 3) \ll 3) - m$.

Нижче Таблиця 9 надає зразковий синтаксис для набору параметрів рівня. У деяких прикладах може використовуватися однаковий синтаксис для кожного з LPS з фіг. 5.

Таблиця 9

layer_para_set() {	Дескриптор
depth_flag	u(1)
layer_para_set_id	ue(v)
vps_id	ue(v)
// cu hierarchy {{	
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
log2_min_pcm_coding_block_size_minus3	ue(v)
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
// cu hierarchy }}	
pcm_bit_depth_luma_minus1	u(4)
pcm_bit_depth_chroma_minus1	u(4)
loop_filter_across_slice_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
pcm_loop_filter_disable_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
//tiles	

num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
if (num_tile_columns_minus1 != 0 num_tile_rows_minus1 != 0) {	
tile_boundary_independence_idc	u(1)
uniform_spacing_idc	u(1)
if (uniform_spacing_idc != 1) {	
for (i=0; i<num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for (i=0; i<num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
}	
lps_extension_flag	u(1)
if(lps_extension_flag)	
while(more_rbsp_data())	
lps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Приклади семантики для синтаксису LPS з Таблиці 9 описуються нижче. Різні рівні (наприклад, різні види в MVC або різні рівні в SVC) можуть посилатися на різні LPS. Різні рівні якості на одному і тому ж рівні залежності можуть спільно використовувати один і той же LPS. Різні часові рівні на одному і тому ж рівні залежності можуть спільно використовувати один і той же LPS. Як альтернатива різні види можуть посилатися на один і той же LPS, і різні рівні залежності можуть посилатися на один і той же LPS.

У цьому прикладі **depth_flag**, що дорівнює 1, точно визначає, що LPS застосовується до представлень глибини, ідентифікованих значеннями **temporal_id**, **dependency_id**, **quality_id** і **view_idx** в одиниці NAL LPS. **depth_flag**, що дорівнює 0, точно визначає, що LPS застосовується до представлень текстури, ідентифікованих значеннями **temporal_id**, **dependency_id**, **quality_id** і **view_idx** в одиниці NAL LPS.

У цьому прикладі **layer_para_set_id** точно визначає id поточного набору параметрів рівня (LPS). Різні набори параметрів рівнів з однаковими значеннями **dependency_id** і **view_idx** відповідно спільно використовують один простір значень для **layer_para_set_id**, отже різні LPS з різним поєднанням **depencey_id** і **view_idx** можуть мати однакове значення **layer_para_set_id**.

Як альтернатива всі LPS можуть спільно використовувати один простір значень, отже кожний LPS має індивідуальне значення **layer_para_set_id**.

У цьому прикладі **vps_id** ідентифікує набір параметрів відео, на який посилається цей набір параметрів рівня.

У цьому прикладі **lps_extension_flag**, що дорівнює 0, точно визначає, що ніякі синтаксичні елементи **lps_extension_data_flag** не присутні в структурі синтаксису RBSP набору параметрів рівня. У цьому прикладі **lps_extension_flag** може дорівнювати 0 в бітових потоках, відповідних майбутньому стандарту HEVC. Значення 1 для **lps_extension_flag** зарезервоване для майбутнього використання ITU-T/ISO/IEC. Декодер, наприклад відеодекодер 30, можуть ігнорувати всі дані, які ідуть за значенням 1 для **lps_extension_flag** в одиниці NAL набору параметрів рівня.

У цьому прикладі **lps_extension_data_flag** може мати будь-яке значення і не впливає на відповідність профілям, точно визначеним в майбутньому стандарті HEVC.

Інші синтаксичні елементи можуть мати таку ж семантику, як і синтаксичні елементи з такими ж назвами в SPS з WD HEVC, але при застосуванні тільки до зображень, що посилаються на цей LPS.

LPS може міститися в одиниці NAL, заголовок якої можна задати відповідно до Таблиці 8 вище. Нижченаведені синтаксичні елементи мають наступну трохи змінену семантику, коли вони асоціюються з LPS.

У цьому прикладі `priority_id` дорівнює мінімальному значенню із значень `priority_id` всіх одиниць NAL, що посилаються на цей LPS.

У цьому прикладі `temporal_id` дорівнює мінімальному значенню з `temporal_id` всіх одиниць NAL, що посилаються на цей LPS.

5 У цьому прикладі `dependency_id` дорівнює `dependency_id` всіх одиниць NAL, що посилаються на цей LPS.

У цьому прикладі `quality_id` дорівнює мінімальному значенню з `quality_id` всіх одиниць NAL, що посилаються на цей LPS.

10 У цьому прикладі `v_idx` є індексом виду у поточного LPS. Всі зображення, що посилаються на цей LPS, можуть мати `id` виду, що дорівнює `view_id[v_idx]`.

Як альтернатива вищезазначені синтаксичні елементи можуть безпосередньо сигналізуватися в таблиці синтаксису набору параметрів рівня, як показано в прикладі з Таблиці 10. Більш докладну таблицю синтаксису можна розробити відповідно до Таблиці 9 вище. У цьому випадку ці синтаксичні елементи не знаходяться в заголовку одиниці NAL в LPS, і
15 синтаксичний аналіз LPS може залежати від VPS з `ID`, що дорівнює `vps_id`.

Таблиця 10

<code>layer_para_set() {</code>	Дескриптор
<code>vps_id</code>	<code>u(8)</code>
<code>if(cnt_p > 1)</code>	
<code>priority_id</code>	<code>u(v)</code>
<code>if(cnt_t > 1)</code>	
<code>temporal_id</code>	<code>u(v)</code>
<code>reserved_one_bit</code>	<code>u(1)</code>
<code>if(cnt_d > 1)</code>	
<code>dependency_id</code>	<code>u(v)</code>
<code>if(cnt_q > 1)</code>	
<code>quality_id</code>	<code>u(v)</code>
<code>if(cnt_v > 1)</code>	
<code>view_idx</code>	<code>u(v)</code>
<code>depth_flag</code>	<code>u(1)</code>
<code>layer_para_set_id</code>	<code>ue(v)</code>
<code>// cu hierarchy { {</code>	
<code>...</code>	
<code>rbsp_trailing_bits()</code>	
<code>}</code>	

20 LPS в цьому випадку не повинен мати заголовок одиниці NAL, що дублює вищезазначені синтаксичні елементи. Передбачаючи, що типом одиниці NAL у одиниці NAL, що вміщує в себе LPS, є, наприклад, 5, синтаксис заголовка одиниці NAL можна трохи змінити, як показано в Таблиці 11, яка додає виключення `"&& nal_unit_type!= 5"` до оператора `"if"` з Таблиці 8.

Таблица 11

nal_unit(NumBytesInNALunit) {	Дескриптор
forbidden_zero_bit	f(1)
nal_ref_flag	u(1)
nal_unit_type	u(6)
NumBytesInRBSP = 0	
nalUnitHeaderBytes = 1	
if(nal_unit_type != 10 && nal_unit_type != 5) { // not VPS NAL unit	
if(cnt_p > 1)	
priority_id	u(v)
if(cnt_t > 1)	
temporal_id	u(v)
reserved_one_bit	u(1)
if(cnt_d > 1)	
dependency_id	u(v)
if(cnt_q > 1)	
quality_id	u(v)
reserved_one_bit	u(1)
if(cnt_v > 1)	
view_idx	u(v)
if(cnt_f)	
reserved_flags	u(v)
m = Ceil(log2(cnt_p)) + Ceil(log2(cnt_t)) + Ceil(log2(cnt_d)) + Ceil(log2(cnt_q)) + Ceil(log2(cnt_v)) + cnt_f + 2	
if(((m + 7 >> 3) << 3) - m)	
reserved_bits	u(v)
nalUnitHeaderBytes += ((m + 7) >> 3)	
}	
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {	
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte[NumBytesInRBSP++]	b(8)
rbsp_byte[NumBytesInRBSP++]	b(8)
i += 2	
emulation_prevention_three_byte /* equal to 0x03 */	f(8)
} else	
rbsp_byte[NumBytesInRBSP++]	b(8)
}	
}	

У інших прикладах відеокoder 20 і відеокoder 30 можуть кодувати пов'язані з масштабованими характеристиками синтаксичні елементи, використовуючи кодування з фіксованою довжиною, як показано нижче в прикладі з Таблиці 12.

Таблиця 12

layer_para_set() {	Дескриптор
vps_id	u(8)
priority_id	u(5)
temporal_id	u(3)
dependency_id	u(3)
quality_id	u(3)
view_idx	u(8)
layer_para_set_id	ue(v)
// cu hierarchy { {	

5

Нижче Таблиця 13 надає приклад синтаксису для набору параметрів зображення (PPS) відповідно до методик з даного розкриття винаходу. У цьому прикладі набору параметрів зображення не потрібно сигналізувати "seq_parameter_set_id" на відміну від PPS в традиційному HEVC.

10

Таблиця 13

pic_parameter_set_rbsp() {	Дескриптор
pic_parameter_set_id	ue(v)
...	
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data())	
pps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Приклади семантики для PPS з Таблиці 13 описуються нижче.

У цьому прикладі pps_extension_flag, що дорівнює 0, точно визначає, що ніякі синтаксичні елементи pps_extension_data_flag не присутні в структурі синтаксису Rbsp набору параметрів зображення. У цьому прикладі pps_extension_flag дорівнює 0 в бітових потоках, відповідних майбутньому стандарту HEVC. Значення 1 для pps_extension_flag може бути зарезервоване для майбутнього використання ITU-T|ISO/IEC. Декодери, наприклад відеокoder 30, можуть ігнорувати всі дані, які ідуть за значенням 1 для pps_extension_flag в одиниці NAL набору параметрів зображення.

15

У цьому прикладі pps_extension_data_flag може мати будь-яке значення. Це не обов'язково впливає на відповідність профілям, точно визначеним в майбутньому стандарті HEVC. Семантика для значень pps_extension_data_flag може призначатися в подальших розробках стандарту HEVC або розширеннях стандарту, не конфліктуючи з методиками з даного розкриття винаходу.

20

У методиках з даного розкриття винаходу ніякий id набору параметрів послідовності або id набору параметрів рівня не потрібно сигналізувати в PPS. Деякі інші синтаксичні елементи в PPS можна перемістити в LPS. Тобто відеокoder 20 і/або відеокoder 30 можуть

25

конфігуруватися для кодування одного або більше LPS, що включають в себе дані, аналогічні синтаксичним елементам, показаним в Таблиці 13.

Фіг. 6 - концептуальна схема, що ілюструє зразковий групувальний набір параметрів (GPS) і взаємозв'язку GPS з іншими наборами параметрів і заголовками слайсів. У цьому прикладі інші набори параметрів включають в себе LPS, SPS, PPS, набори параметрів адаптації (APS) типу 0 (наприклад, APS, що сигналізують параметри адаптивного контурного фільтра (ALF)), APS типу 1 (наприклад, APS, що сигналізують матрицю квантування) і інші набори параметрів. У цьому прикладі GPS включає в себе множину різних груп, причому кожна має унікальний ID GPS (також званий ID групи), де кожна група вказує конкретний набір з різних наборів параметрів за допомогою ID набору параметрів. Таким чином, заголовки слайсів повинні точно визначати тільки group_id для точного визначення кожного з наборів параметрів, відповідних групі, що має цей group_id.

Нижче Таблиці 14 і 15 надають альтернативні приклади синтаксису для RBSP групувального набору параметрів.

Таблиця 14

group_para_set_rbsp() {	Дескриптор
number_signaled_para_set_groups_minus1	ue(v)
for(i = 0; i <= number_signaled_para_set_groups_minus1; i++) {	
para_set_group_id[i]	ue(v)
lps_id[i]	ue(v)
pps_id[i]	ue(v)
for(j = 0; j < numParaSetTypes; j++)	
para_set_type_id[i][j]	
}	
gps_extension_flag	u(1)
if(gps_extension_flag)	
while(more_rbsp_data())	
gps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Таблиця 15

group_para_set_rbsp() {	Дескриптор
number_signaled_para_set_groups_minus1	ue(v)
for(i = 0; i <= number_signaled_para_set_groups_minus1; i++) {	
para_set_group_id[i]	ue(v)
lps_id[i]	ue(v)
pps_id[i]	ue(v)
aps_id[i]	ue(v)
}	
gps_extension_flag	u(1)
if(gps_extension_flag)	
while(more_rbsp_data())	
gps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

Кодери відео, наприклад відеокoder 20 і відеокoder 30, можуть конфігуруватися для кодування груповального набору параметрів згідно, наприклад, з Таблицею 14 або Таблицею 15. Приклади семантики для синтаксису наборів груповальних параметрів надаються нижче.

У цьому прикладі **number_signalled_para_set_groups_minus1** плюс 1 точно визначає число сигналізованих груп параметрів. Це значення може бути в діапазоні від 0 до 30, включно.

У цьому прикладі **para_set_group_id[i]** точно визначає ID i-ої сигналізованої групи наборів параметрів. Значення **para_set_group_id[i]** повинно бути в діапазоні від 0 до 31, включно.

У цьому прикладі **para_set_type_id[i][j]** точно визначає ID j-го типу набору параметрів для i-ої групи наборів параметрів.

У цьому прикладі **lps_id[i]** вказує id набору параметрів рівня, на який посилається група наборів параметрів з id групи, що дорівнює **para_set_group_id[i]**. Значення **dependency_id** і **view_idx** у LPS з **layer_para_set_id**, що дорівнює **lps_id[i]**, можуть бути ідентичні значенням **dependency_id** і **view_idx** відповідно у одиниці NAL групи наборів параметрів.

Значення **dependency_id** і **view_idx** у RBSP групування набору параметрів присутні в заголовку одиниці NAL цього RBSP в прикладах з Таблиць 14 і 15, і значення **dependency_id** і **view_idx** у LPS можуть бути присутніми або в заголовку одиниці NAL цього LPS, або в таблиці синтаксису LPS.

Як альтернатива значення **dependency_id** і **view_idx** у LPS з **layer_para_set_id**, що дорівнює **lps_id[i]**, можуть бути не ідентичні значенням **dependency_id** і **view_idx** відповідно у одиниці NAL групи наборів параметрів.

У цьому прикладі **pps_id[i]** вказує id набору параметрів зображення, на який посилається група наборів параметрів з id групи, що дорівнює **para_set_group_id[i]**.

У цьому прикладі **aps_id[i]** вказує id набору параметрів адаптації, на який посилається група наборів параметрів з id групи, що дорівнює **para_set_group_id[i]**.

У цьому прикладі **gps_extension_flag**, що дорівнює 0, задає, що ніякі синтаксичні елементи **gps_extension_data_flag** не присутні в структурі синтаксису RBSP групування набору параметрів. **gps_extension_flag** може дорівнювати 0 в бітових потоках, відповідних майбутньому стандарту HEVC. Значення 1 для **gps_extension_flag** може бути зарезервоване для майбутнього використання ITU-T/ISO/IEC. Декодери, наприклад відеокoder 30, можуть ігнорувати всі дані, які ідуть за значенням 1 для **gps_extension_flag** в одиниці NAL групування набору параметрів. Взагалі **gps_extension_data_flag** може мати будь-яке значення. Це не повинно впливати на відповідність профілям, точно визначеним в майбутньому стандарті HEVC.

У деяких прикладах замість **para_set_type_id[i][j]** може бути **aps_id[i][j]** з аналогічною семантикою, як **aps_id[i]**, яка описана вище.

Як показано на фіг. 6, замість посилання на ID набору параметрів зображення в заголовку слайсу, відповідно до методик з даного розкриття винаходу заголовки слайсу може посилатися

на ID групи наборів параметрів, відповідно непрямо посилаючись на LPS, PPS і APS кожного типу (наприклад, APS, що надають параметри ALF і матриці квантування).

Кодер відео може активізувати набір параметрів відео або набори параметрів рівнів, коли одиниця NAL VCL (що містить кодований слайс) посилається на набір параметрів, непрямо, наприклад, на основі принципу проектування H.264/AVC.

У деяких прикладах набори параметрів можна активізувати за допомогою специфічного типу одиниці NAL, а не кодованого слайсу. Наприклад, тип одиниці NAL цього специфічного типу (одиниця NAL активізації наборів параметрів) може активізувати один і рівно один VPS, при наявності в бітовому потоці. До того ж в різних альтернативах такий тип одиниці NAL може активізувати щонайменше один LPS. До того ж такий тип одиниці NAL може активізувати щонайменше один PPS. До того ж такий тип одиниці NAL може активізувати щонайменше один APS. Одиниця NAL активізації наборів параметрів може бути RBSP групувального набору параметрів. Одиниця NAL активізації наборів параметрів (PSA) може бути застосовна до однієї кодованої відеопослідовності. Одиниця NAL PSA може вважатися одиницею, що не є NAL VCL, тобто не має безпосереднього відношення до кодера відео. Синтаксис заголовка одиниці NAL у одиниці NAL PSA може бути таким же, як одиниця NAL VPS.

У деяких прикладах одиниця NAL PSA, при наявності в одиниці доступу, може передувати першій одиниці NAL VCL в одиниці доступу. Може бути присутня щонайменше одна одиниця NAL PSA в першій одиниці доступу кодованої відеопослідовності, наприклад IDR-зображенні. Декілька одиниць NAL PSA в одній і тій же кодованій відеопослідовності можуть містити однаковий id VPS; відповідно, ніякі різні набори параметрів відео не потрібно активізувати в одній і тій же кодованій відеопослідовності. Одиниця NAL PSA, при наявності в одиниці доступу, може передувати будь-якій одиниці NAL LPS, PPS, APS або SEI, при наявності. Одиниця NAL VPS, при наявності в одиниці доступу, може передувати будь-якій одиниці NAL LPS, PPS, APS або SEI, при наявності. До того ж в різних альтернативах одиниця NAL PSA, при наявності в одиниці доступу, може передувати одиниці NAL VPS, при наявності.

У деяких прикладах кодери відео, наприклад відеокoder 20 і відеодекодер 30, можуть конфігуруватися для використання синтаксису з Таблиці 16 для набору параметрів послідовності (SPS) на відміну від традиційного синтаксису SPS, наприклад в HEVC.

Таблица 16

seq_parameter_set_rbsp() {	Дескриптор
video_parameter_set_id	ue(v)
rep_format_idx	ue(v)
pcm_enabled_flag	u(1)
if(pcm_enabled_flag) {	
pcm_bit_depth_luma_minus1	u(4)
pcm_bit_depth_chroma_minus1	u(4)
}	
qpprime_y_zero_transquant_bypass_flag	u(1)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
for(i = 0; i <= max_temporal_layers_minus1; i++) {	
max_dec_pic_buffering[i]	ue(v)
num_reorder_pics[i]	ue(v)
max_latency_increase[i]	ue(v)
}	
restricted_ref_pic_lists_flag	u(1)
if(restricted_ref_pic_lists_flag)	
lists_modification_present_flag	u(1)
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
if(pcm_enabled_flag) {	
log2_min_pcm_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_coding_block_size	ue(v)
}	
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
scaling_list_enable_flag	
chroma_pred_from_luma_enabled_flag	u(1)
deblocking_filter_in_aps_enabled_flag	u(1)
seq_loop_filter_across_slices_enabled_flag	u(1)
asymmetric_motion_partitions_enabled_flag	u(1)
non_square_quadtrees_enabled_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
if(adaptive_loop_filter_enabled_flag)	
alf_coef_in_slice_flag	u(1)
if(pcm_enabled_flag)	
pcm_loop_filter_disable_flag	u(1)
temporal_id_nesting_flag	u(1)
if(log2_min_coding_block_size_minus3 == 0)	
inter_4x4_enabled_flag	u(1)
long_term_ref_pics_present_flag	u(1)
tiles_or_entropy_coding_sync_idc	u(2)

Таблиця 16

if(tiles_or_entropy_coding_sync_idc == 1) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_spacing_flag	u(1)
if(!uniform_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

Зразковий SPS з Таблиці 16 виключає з традиційного синтаксису SPS `profile_idc`, `reserved_zero_8bits`, `level_idc`, `chroma_format_idc`, `separate_colour_plane_flag` і відповідний умовний оператор "if", `max_temporal_layers_minus1`, `pic_width_in_luma_samples`, `pic_height_in_luma_samples`, `pic_cropping_flag`, `pic_crop_left_offset`, `pic_crop_right_offset`, `pic_crop_top_offset`, `pic_crop_bottom_offset` і відповідний умовний оператор "if", `bit_depth_luma_minus8`, `bit_depth_chroma_minus8`, `num_short_term_ref_pic_sets`, `short_term_ref_pic_set(i)` і відповідний умовний оператор "if". Крім того, зразковий SPS з Таблиці 16 додає `video_parameter_set_id` і `rep_format_idx`. Семантика для синтаксичних елементів, що залишилися, може бути такою ж, як задана в традиційному HEVC. Семантику для доданих елементів `video_parameter_set_id` і `rep_format_idx` можна задати наступним чином.

У цьому прикладі `video_parameter_set_id` ідентифікує набір параметрів відео (VPS), на який посилається поточний SPS. Як альтернатива не потрібно сигналізувати `video_parameter_set_id`, і GPS може використовуватися для зв'язку SPS зі специфічним VPS.

У цьому прикладі `rep_format_idx` точно визначає індекс до формату представлення, сигналізованого в наборі параметрів відео, на який посилаються.

Як ще одна альтернатива Таблиця 17 надає інший приклад синтаксису для груповального набору параметрів. У цьому прикладі передбачається, що синтаксичний елемент ID набору параметрів відео відсутній в синтаксисі SPS, як описано вище.

Таблиця 17

group_para_set_rbsp() {	Дескриптор
gps_id	ue(v)
vps_id	ue(v)
sps_id	ue(v)
pps_id	ue(v)
num_ref_aps_ids	ue(v)
for(i = 0; i < num_ref_aps_ids; i++) {	
ref_aps_id[i]	ue(v)
ref_aps_param_type[i]	ue(v)
}	
gps_extension_flag	u(1)
if(gps_extension_flag)	
while(more_rbsp_data())	
gps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

Семантику для синтаксичних елементів з Таблиці 17 можна задати наступним чином.

У цьому прикладі **gps_id** точно визначає ідентифікатор групового набору параметрів (GPS).

5 У цьому прикладі **vps_id** точно визначає ідентифікатор набору параметрів відео, на який посилається GPS.

У цьому прикладі **sps_id** точно визначає ідентифікатор набору параметрів послідовності, на який посилається GPS.

10 У цьому прикладі **pps_id** точно визначає ідентифікатор набору параметрів зображення, на який посилається GPS.

У цьому прикладі **num_ref_aps_ids** точно визначає число наступних синтаксичних елементів **ref_aps_id[i]**. Значення **num_ref_aps_ids** повинно бути в діапазоні від 0 до 4, включно.

У цьому прикладі **ref_aps_id[i]** ідентифікує *i*-ий набір параметрів адаптації, на який посилається груповий набір параметрів.

15 Одне і те ж значення **ref_aps_id[i]** може бути присутнім в циклі більше одного разу, і відповідно за допомогою одного і того ж GPS можна посилатися більше ніж на один тип параметрів APS з одного і того ж APS, і він [тип] може застосовуватися до кодованих слайсів, що посилаються на GPS.

20 У цьому прикладі **ref_aps_param_type[i]** точно визначає тип параметрів APS, включених в *i*-ий набір параметрів адаптації, на який посилається груповий набір параметрів. Значення **ref_aps_param_type[i]** може бути в діапазоні від 0 до 3, включно. Значення від 0 до 3, включно, для **ref_aps_param_type[i]** відповідають типам параметрів APS у списку масштабування, фільтра зменшення блоковості, адаптивного зміщення вибірок (SAO) і ALF, відповідно. Значення **ref_aps_param_type[i]** для будь-яких двох різних значень *i* в деяких прикладах не повинні бути ідентичні.

25 У цьому прикладі **gps_extension_flag**, що дорівнює 0, точно визначає, що ніякі синтаксичні елементи **gps_extension_data_flag** не присутні в структурі синтаксису RBSP групування набору параметрів. **gps_extension_flag** може дорівнювати 0 в бітових потоках, відповідних майбутньому стандарту HEVC. Значення 1 для **gps_extension_flag** може бути зарезервоване для майбутнього використання ITU-T/ISO/IEC. Декодери, наприклад відеодекодер 30, можуть ігнорувати всі дані, які ідуть за значенням 1 для **gps_extension_flag** в одиниці NAL групування набору параметрів.

30 У цьому прикладі **gps_extension_data_flag** може мати будь-яке значення. Це не повинно впливати на відповідність профілям, точно визначеним в майбутньому стандарті HEVC.

Кодери відео, наприклад відеокодер 20 і відеодекодер 30, можуть застосовувати наступний процес для активізації наборів параметрів для однорівневих або одновидових бітових потоків, коли GPS точно визначається відповідно до Таблиці 17 або практично відповідає прикладу з Таблиці 17.

5 RBSP набору параметрів адаптації може включати в себе параметри, на які одиниці NAL кодованого слайсу в одному або більше кодованих зображеннях можуть посилатися непрямо за допомогою одного або більше групових наборів параметрів, на які посилаються одиниці NAL кодованого слайсу. Кожне RBSP набору параметрів адаптації спочатку може вважатися неактивним на початку роботи процесу декодування. Не більше одного RBSP набору
10 параметрів адаптації може вважатися активним для кожного типу параметрів APS в будь-який заданий момент протягом роботи процесу декодування, і активізація будь-якого конкретного RBSP набору параметрів адаптації для конкретного типу параметрів APS приводить до деактивації раніше активного RBSP набору параметрів адаптації (якщо є) для цього конкретного типу параметрів APS.

15 Коли RBSP набору параметрів адаптації (з конкретним значенням `aps_id`) не активне для конкретного типу параметрів APS, і на нього посилається одиниця NAL кодованого слайсу для цього конкретного типу параметрів APS (використовуючи це значення `aps_id`) непрямо за допомогою групового набору параметрів, на який посилається одиниця NAL кодованого слайсу, його можна активізувати для того конкретного типу параметрів APS. Це RBSP набору
20 параметрів адаптації називається активним RBSP набору параметрів адаптації для цього конкретного типу параметрів APS доти, поки воно не деактивується внаслідок активізації іншого RBSP набору параметрів адаптації для цього конкретного типу параметрів APS. RBSP набору параметрів адаптації з цим конкретним значенням `aps_id` може бути доступне процесу декодування перед його активізацією.

25 RBSP набору параметрів зображення може включати в себе параметри, на які одиниці NAL кодованого слайсу в одному або більше кодованих зображеннях можуть посилатися непрямо за допомогою одного або більше групових наборів параметрів, на які посилаються одиниці NAL кодованого слайсу. Кожне RBSP набору параметрів зображення спочатку може вважатися неактивним на початку роботи процесу декодування. Не більше одного RBSP набору
30 параметрів зображення може вважатися активним в будь-який заданий момент протягом роботи процесу декодування, і активізація будь-якого конкретного RBSP набору параметрів зображення приводить до деактивації раніше активного RBSP набору параметрів зображення (якщо є).

35 Коли RBSP набору параметрів зображення (з конкретним значенням `pic_parameter_set_id`) не активне, і на нього посилається одиниця NAL кодованого слайсу (використовуючи це значення `pic_parameter_set_id`) непрямо за допомогою групового набору параметрів, на який посилається одиниця NAL кодованого слайсу, його можна активізувати. Це RBSP набору параметрів зображення називається активним RBSP набору параметрів зображення доти, поки воно не деактивується внаслідок активізації іншого RBSP набору параметрів зображення. RBSP
40 набору параметрів зображення з цим конкретним значенням `pic_parameter_set_id` може бути доступне процесу декодування перед його активізацією.

Будь-яка одиниця NAL набору параметрів зображення, що містить значення `pic_parameter_set_id` для активного RBSP набору параметрів зображення для кодованого зображення, може мати такий же вміст, що і у активного RBSP набору параметрів зображення
45 для кодованого зображення, якщо вона не іде за крайньою одиницею NAL VCL в кодованому зображенні і не передує першій одиниці NAL VCL в іншому кодованому зображенні.

RBSP набору параметрів послідовності може включати в себе параметри, на які одиниці NAL кодованого слайсу в одному або більше кодованих зображеннях можуть посилатися непрямо за допомогою одного або більше групових наборів параметрів, на які посилаються
50 одиниці NAL кодованого слайсу, або на які можуть посилатися одна або більше одиниць NAL SEI, що містять повідомлення SEI про період буферизації. Кожне RBSP набору параметрів послідовності спочатку може вважатися неактивним на початку роботи процесу декодування. Не більше одного RBSP набору параметрів послідовності може вважатися активним в будь-який заданий момент протягом роботи процесу декодування, і активізація будь-якого конкретного
55 RBSP набору параметрів послідовності приводить до деактивації раніше активного RBSP набору параметрів послідовності (якщо є).

60 Коли RBSP набору параметрів послідовності (з конкретним значенням `seq_parameter_set_id`) вже не активне, і на нього посилається одиниця NAL кодованого слайсу непрямо за допомогою групового набору параметрів, на який посилається одиниця NAL кодованого слайсу (використовуючи це значення `seq_parameter_set_id`), або на нього посилається одиниця NAL

SEI, що містить повідомлення SEI про період буферизації (використовуючи це значення `seq_parameter_set_id`), його можна активізувати. Це RBSP набору параметрів послідовності називається активним RBSP набору параметрів послідовності доти, поки воно не деактивується внаслідок активізації іншого RBSP набору параметрів послідовності. RBSP набору параметрів послідовності з цим конкретним значенням `seq_parameter_set_id`, що міститься в одиниці доступу з `temporal_id`, що дорівнює 0, може бути доступне процесу декодування перед його активізацією. Активізоване RBSP набору параметрів послідовності повинно залишатися активним для всієї кодової відеопослідовності.

RBSP набору параметрів відео може включати в себе параметри, на які одиниці NAL кодового слайсу в одному або більше кодованих зображеннях можуть посилатися непрямо за допомогою одного або більше групових наборів параметрів, на які посилаються одиниці NAL кодового слайсу, або на які можуть посилатися одна або більше одиниць NAL SEI, що містять повідомлення SEI про період буферизації. Кожне RBSP набору параметрів відео спочатку може вважатися неактивним на початку роботи процесу декодування. Не більше одного RBSP набору параметрів відео може вважатися активним в будь-який заданий момент протягом роботи процесу декодування, і активізація будь-якого конкретного RBSP набору параметрів відео приводить до деактивації раніше активного RBSP набору параметрів відео (якщо є).

Коли RBSP набору параметрів відео (з конкретним значенням `video_parameter_set_id`) вже не активне, і на нього посилається одиниця NAL кодового слайсу непрямо за допомогою групового набору параметрів, на який посилається одиниця NAL кодового слайсу (використовуючи це значення `video_parameter_set_id`), або на нього посилається одиниця NAL SEI, що містить повідомлення SEI про період буферизації (використовуючи це значення `video_parameter_set_id`), його можна активізувати. Це RBSP набору параметрів відео називається активним RBSP набору параметрів відео доти, поки воно не деактивується внаслідок активізації іншого RBSP набору параметрів відео. RBSP набору параметрів відео з цим конкретним значенням `video_parameter_set_id`, що міститься в одиниці доступу з `temporal_id`, що дорівнює 0, повинно бути доступне процесу декодування перед його активізацією. Активізоване RBSP набору параметрів відео повинно залишатися активним для всієї кодової відеопослідовності.

Будь-яка одиниця NAL набору параметрів послідовності, що містить значення `seq_parameter_set_id` для активного RBSP набору параметрів послідовності для кодової відеопослідовності, може мати такий же вміст, що і у активного RBSP набору параметрів послідовності для кодової відеопослідовності, якщо вона не іде за крайньою одиницею доступу в кодованій відеопослідовності і не передує першій одиниці NAL VCL і першій одиниці NAL SEI, що містить повідомлення SEI про період буферизації (коли присутній), в іншій кодованій відеопослідовності.

Будь-яка одиниця NAL набору параметрів відео, що містить значення `video_parameter_set_id` для активного RBSP набору параметрів відео для кодової відеопослідовності, може мати такий же вміст, що і у активного RBSP набору параметрів відео для кодової відеопослідовності, якщо вона не іде за крайньою одиницею доступу в кодованій відеопослідовності і може передувати першій одиниці NAL VCL і першій одиниці NAL SEI, що містить повідомлення SEI про період буферизації (коли присутній), в іншій кодованій відеопослідовності.

Всі обмеження, які виражаються у взаємозв'язку між значеннями синтаксичних елементів (і значеннями змінних, введених з цих синтаксичних елементів) в наборах параметрів відео, наборах параметрів послідовності, наборах параметрів зображення і наборах параметрів адаптації і іншими синтаксичними елементами, є виразами обмежень, які можуть застосовуватися тільки до активних наборів параметрів відео, активного набору параметрів послідовності, активного набору параметрів зображення і активного набору параметрів адаптації для кожного конкретного типу параметрів APS. Якщо присутнє яке-небудь RBSP набору параметрів відео, яке не активізоване в бітовому потоці, то його синтаксичні елементи можуть мати значення, які відповідали б точно визначеним обмеженням, якби активізувалися за допомогою посилання в іншому відповідному бітовому потоці. Якщо присутнє яке-небудь RBSP набору параметрів послідовності, яке не активізоване в бітовому потоці, то його синтаксичні елементи можуть мати значення, які відповідали б точно визначеним обмеженням, якби активізувалися за допомогою посилання в іншому відповідному бітовому потоці. Якщо присутнє яке-небудь RBSP набору параметрів зображення, яке ніколи не активізоване в бітовому потоці, то його синтаксичні елементи можуть мати значення, які відповідали б точно визначеним обмеженням, якби активізувалися за допомогою посилання в іншому відповідному бітовому потоці. Якщо присутнє яке-небудь RBSP набору параметрів адаптації, яке ніколи не

активізоване в бітовому потоці, то його синтаксичні елементи можуть мати значення, які відповідали б точно визначеним обмеженням, якби активізувалися за допомогою посилання в іншому відповідному бітовому потоці.

Протягом роботи процесу декодування можуть вважатися діючими значення параметрів активного набору параметрів відео, активного набору параметрів послідовності, активного набору параметрів зображення і активного набору параметрів адаптації для кожного типу параметрів APS. Для інтерпретації повідомлень SEI значення параметрів з набору параметрів відео, набору параметрів послідовності, набору параметрів зображення і набору параметрів адаптації, які активні для роботи процесу декодування для одиниць NAL VCL кодованого зображення в одній і тій же одиниці доступу, можуть вважатися діючими, поки точно не визначене інше в семантиці повідомлення SEI.

Фіг. 7 - блок-схема послідовності операцій, що ілюструє зразковий спосіб для кодування відеоданих відповідно до методик з даного розкриття винаходу. Хоч описано відносно відеокодера 20, потрібно розуміти, що інші пристрої кодування відео можуть конфігуруватися для виконання способу з фіг. 7.

Спочатку в цьому прикладі відеокодер 20 приймає бітовий потік, що включає в себе один або більше рівнів необроблених відеоданих (100). Наприклад, джерело 18 відео (фіг. 1) може надати відеокодеру 20 багатовидові відеодані. Як альтернатива відеокодер 20 або його препроцесор може розділити бітовий потік необробленого відео на множину різних рівнів, наприклад рівнів просторового розрізнення, рівнів якості, часових рівнів або т. п. В ще одних прикладах бітовий потік може розбиватися на поєднання різних рівнів, наприклад будь-яке поєднання видів, рівнів просторового розрізнення, рівнів якості, часових рівнів або т. п.

Відеокодер 20 може визначити один або більше загальних параметрів для відповідних послідовностей серед набору рівнів (102). Відповідні послідовності можуть бути послідовностями, що мають відповідні положення у часі на різних рівнях. Тобто про першу послідовність, що має час T1 початку (в показниках часу відображення) і час T2 закінчення (знову в показниках часу відображення), і другу послідовність, що також має час T1 початку і час T2 закінчення, можна сказати, що вони відповідають одна одній. Зокрема, перша послідовність може утворювати частину першого рівня, а друга послідовність може утворювати частину другого, іншого рівня. "Послідовність" може включати в себе послідовність зображень, що ідуть одне за одним, в порядку декодування, наприклад, починаючи із зображення миттєвого оновлення декодування (IDR) і закінчуючи безпосередньо перед подальшим IDR-зображенням в порядку декодування. Взагалі, параметри можуть відповідати набору відповідних послідовностей одного або більше рівнів, наприклад N рівнів, де N - ціле число. Відеокодер 20 потім може кодувати VPS, що включає в себе дані для визначених параметрів (104). Наприклад, відеокодер 20 може кодувати VPS, відповідний одному з прикладів з Таблиці 2 або Таблиці 5.

Відеокодер 20 також може визначити загальні параметри для послідовностей в рамках одного рівня (106). Послідовність може містити одну з послідовностей, відповідних іншим послідовностям на інших рівнях, для яких кодувався VPS. Відеокодер 20 може кодувати набір параметрів послідовності (SPS), що включає в себе загальні параметри для послідовності (108). Таким чином, потрібно розуміти, що VPS і SPS є окремими структурами даних і відповідають різним типам відеоданих. Тоді як VPS може відповідати набору відповідних послідовностей серед множини рівнів, SPS відповідає одній послідовності на одному рівні. SPS може переважно відповідати SPS з H.264/AVC, SPS з H.264/AVC, яке розширене за допомогою MVC (проілюстровано в Таблиці 1 вище), майбутньому стандарту HEVC або прикладу з Таблиці 16, описаної вище. До того ж відеокодер 20 може кодувати набір параметрів зображення (PPS) для деякого зображення в послідовності (110). PPS може переважно відповідати SPS з H.264/AVC, майбутньому стандарту HEVC або прикладу з Таблиці 13, описаної вище. Хоч спосіб з фіг. 7 показує кодування тільки одного PPS, потрібно розуміти, можуть кодуватися декілька PPS. Одне або більше зображень можуть посилатися на один і той же PPS.

Відеокодер 20 потім може визначити, чи є останній рівень, для якого кодувалися SPS і PPS, крайнім рівнем (112). Якщо до крайнього рівня ще не зверталися (перехід "HI" етапу 112), то відеокодер 20 може вибрати наступний рівень і кодувати SPS і один або більше PPS для наступного рівня, наприклад, відповідно до етапів 106-110. Після того, як звернулися до крайнього рівня (перехід "TAK" етапу 112), відеокодер 20 може кодувати відеодані різних рівнів на основі даних VPS, SPS і PPS. Різні приклади кодування відеоданих, основуючись щонайменше частково на VPS, детальніше описуються нижче відносно фіг. 9-12.

Хоч і не показано в прикладі з фіг. 7, в деяких прикладах відеокодер 20 додатково може кодувати один або більше LPS і/або один або більше GPS, як описано вище. LPS можуть переважно відповідати прикладам з Таблиці 9, Таблиці 10 або Таблиць 12, тоді як GPS можуть

переважно відповідати прикладам з Таблиці 14, Таблиці 15 або Таблиць 17. У таких прикладах відеокoder 20 кодує відеодані також щонайменше частково на основі LPS і/або GPS.

Таким чином, спосіб з фіг. 7 представляє приклад способу, який включає в себе кодування набору параметрів відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з
5 одного або більше рівнів відеоданих посиляється на VPS, і кодування одного або більше рівнів відеоданих, основуючись щонайменше частково на VPS.

Фіг. 8 - блок-схема послідовності операцій, що ілюструє зразковий спосіб для декодування відеоданих відповідно до методик з даного розкриття винаходу. Хоч описано відносно відеодекодера 30, потрібно розуміти, що інші пристрої декодування відео можуть
10 конфігуруватися для виконання способу з фіг. 8.

Спочатку відеодекодер 30 приймає бітовий потік, що включає VPS, один або більше SPS і один або більше PPS для рівнів кодованих відеоданих (120). Відеодекодер 30 потім може декодувати VPS, який включає в себе загальні параметри для відповідних послідовностей серед одного або більше рівнів (122). Також відеодекодер 30 може декодувати набір параметрів
15 послідовності, що включає в себе загальні параметри для деякої послідовності одного рівня (124). Крім того, відеодекодер 30 може декодувати набір параметрів зображення, що включає в себе параметри для деякого зображення в послідовності (126). Як обговорювалося вище, одне або більше зображень можуть посилатися на один і той же PPS, і тому параметри в PPS можуть вважатися загальними для одного або більше зображень. Також відеодекодер 30 може
20 декодувати множину PPS для послідовності, хоч це не показано на фіг. 8.

Крім того, відеодекодер 30 може визначити, чи був останній рівень крайнім рівнем, до якого зверталися (128). Якщо найбільш останній рівень не був крайнім рівнем (перехід "HI" етапу 128), то відеодекодер 30 може перейти до декодування SPS і одного або більше PPS для подальшого рівня відповідно до етапів 124 і 126. З іншого боку, якщо найбільш останній рівень
25 був крайнім рівнем (перехід "TAK" етапу 128), то відеодекодер 30 може перейти до декодування відеоданих рівнів на основі VPS, SPS і PPS (130). Приклади кодування відеоданих, основуючись щонайменше частково на VPS, обговорюються детальніше відносно фіг. 9-12.

Хоч і не показано в прикладі з фіг. 8, в деяких прикладах відеодекодер 30 додатково може декодувати один або більше LPS і/або один або більше GPS, як описано вище. LPS можуть
30 переважно відповідати прикладам з Таблиці 9, Таблиці 10 або Таблиці 12, тоді як GPS можуть переважно відповідати прикладам з Таблиці 14, Таблиці 15 або Таблиці 17. У таких прикладах відеодекодер 30 декодує відеодані також щонайменше частково на основі LPS і/або GPS.

Таким чином, спосіб з фіг. 8 представляє приклад способу, який включає в себе кодування набору параметрів відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з
35 одного або більше рівнів відеоданих посиляється на VPS, і кодування одного або більше рівнів відеоданих, основуючись щонайменше частково на VPS.

Фіг. 9 - блок-схема послідовності операцій, що ілюструє зразковий спосіб кодування відеоданих, основуючись щонайменше частково на числі часових рівнів, які сигналізовані в VPS. Спосіб з фіг. 9 може виконуватися відеокoderом 20 і/або відеодекодером 30. Для прикладу
40 спосіб з фіг. 9 описується відносно відеодекодера 30.

У цьому прикладі відеодекодер 30 кодує (тобто декодує) VPS, що вказує число часових рівнів у відеоданих (150), наприклад один або більше рівнів, яким відповідає VPS. Наприклад, відеодекодер 30 може декодувати "cnt_t", яке описане вище відносно Таблиці 2. Як інший приклад відеодекодер 30 може декодувати num_temporal_layers_minus1, яке описане вище
45 відносно Таблиці 5.

На основі цього указання в цьому прикладі відеодекодер 30 декодує часові ідентифікатори для кожного з часових рівнів (152). Також відеодекодер 30 може визначити значення ідентифікаторів опорних зображень на основі числа часових рівнів (154). Наприклад, відеодекодер 30 може конфігуруватися для визначення, що для поточного зображення на рівні
50 N поточне зображення не буде використовувати для посилення зображення на рівні N+1 або вище. Тому відеодекодер 30 може визначити ідентифікатори для можливих опорних зображень на рівнях рівня N або нижче. Крім того, відеодекодер 30 може декодувати дані зображень на часовому рівні N, використовуючи опорні дані рівнів аж до (і включаючи) рівня N (156). Таким чином, фіг. 9 представляє приклад способу, який включає в себе кодування даних VPS, що
55 вказують максимальне число часових рівнів на одному або більше рівнях відеоданих, і кодування одного або більше рівнів, основуючись щонайменше частково на VPS.

Фіг. 10 - блок-схема послідовності операцій, що ілюструє зразковий спосіб кодування відеоданих, основуючись щонайменше частково на числі зображень, які повинні бути переупорядковані на одному або більше рівнях, і зображень, які повинні бути збережені в буфер

декодованих зображень. Спосіб з фіг. 10 може виконуватися відеокодером 20 і/або відеодекодером 30. Для прикладу спосіб з фіг. 10 описується відносно відеодекодера 30.

У цьому прикладі відеодекодер 30 декодує VPS, що вказує число зображень, які повинні бути переупорядковані на одному або більше рівнях відеоданих, і число зображень, які повинні
 5 бути збережені в буфер декодованих зображень (наприклад, запам'ятовуючий пристрій 82 опорних зображень) в заданий момент часу (160). Наприклад, відеодекодер 30 може декодувати синтаксичний елемент в VPS, практично відповідний `num_reorder_pics`, яке описане вище відносно Таблиці 16, і/або інформації про обмеження бітового потоку, яка точно визначає розмір DPB. У інших прикладах VPS міг би включати в себе тільки одне з двох, і не обов'язково
 10 обидва, з числа зображень, які повинні бути переупорядковані, і числа зображень, які повинні бути збережені в буфер декодованих зображень. Відеодекодер 30 потім може керувати буфером декодованих зображень (наприклад, запам'ятовуючим пристроєм 82 опорних зображень) на основі числа зображень, які повинні бути переупорядковані і/або збережені (162). Наприклад, відеодекодер 30 може видаляти зображення із запам'ятовуючого пристрою 82
 15 опорних зображень, коли в запам'ятовуючому пристрої 82 опорних зображень зберігається більше числа зображень, які повинні бути збережені.

Відеодекодер 30 також може визначити значення ідентифікаторів опорних зображень на основі числа зображень в DPB (тобто в запам'ятовуючому пристрої 82 опорних зображень) (164). Крім того, відеодекодер 30 може декодувати дані зображень на основі значень
 20 ідентифікаторів опорних зображень (166). Таким чином, спосіб з фіг. 10 представляє приклад способу, який включає в себе кодування даних VPS, що вказують число зображень, які повинні бути збережені в буфер декодованих зображень (DPB) під час декодування одного або більше рівнів, і способу, який включає в себе кодування даних VPS, що вказують число кадрів, які повинні бути переупорядковані на щонайменше одному з одного або більше рівнів.

Фіг. 11 - блок-схема послідовності операцій, що ілюструє зразковий спосіб кодування відеоданих, основуючись щонайменше частково на параметрах гіпотетичного еталонного декодера (HRD), сигналізованих в VPS. Спосіб з фіг. 11 може виконуватися відеокодером 20 і/або відеодекодером 30. Для прикладу спосіб з фіг. 11 описується відносно відеодекодера 30.

У цьому прикладі відеодекодер 30 декодує VPS, що вказує параметри HRD (170).
 30 Відеодекодер 30 додатково може визначити моменти видалення для зображень з буфера кодованих зображень (CPB) на основі параметрів HRD (172). Відеодекодер 30 потім може видалити дані з CPB на основі визначених моментів видалення (174) і декодувати дані, видалені з CPB. Відповідно, спосіб з фіг. 11 представляє приклад способу, який включає в себе кодування даних VPS, що вказують один або більше параметрів гіпотетичного еталонного
 35 декодера (HRD), і кодування даних одного або більше рівнів на основі параметрів HDR.

Фіг. 12 - блок-схема послідовності операцій, що ілюструє зразковий спосіб кодування відеоданих, основуючись щонайменше частково на даних розширення, сигналізованих в VPS. Спосіб з фіг. 12 може виконуватися відеокодером 20 і/або відеодекодером 30. Для прикладу спосіб з фіг. 12 описується відносно відеодекодера 30.

Відеодекодер 30 в цьому прикладі декодує дані VPS, що вказують, чи включає VPS дані розширення (180). Наприклад, відеодекодер 30 може декодувати `vps_extension_flag` в VPS. Відеодекодер 30 потім визначає, чи вказують дані, що VPS включає в себе дані розширення (182). Якщо дані вказують, що VPS включає в себе дані розширення (перехід "ТАК" етапу 182), то відеодекодер 30 кодує дані розширення VPS для одного або більше інструментів кодування розширення (184) і декодує відеодані, використовуючи інструменти кодування розширення і дані розширення (186). З іншого боку, якщо дані вказують, що VPS не включає в себе дані розширення (перехід "НІ" етапу 182), то відеодекодер 30 може декодувати відеодані, використовуючи традиційні інструменти кодування (188). Таким чином, спосіб з фіг. 12 представляє приклад способу, який включає в себе кодування даних VPS, що вказують, чи
 45 включає VPS розширення понад відповідний стандарт, і, коли VPS включає в себе це розширення - даних для цього розширення, а також кодування відеоданих на основі даних розширення VPS.

Треба визнати, що залежно від прикладу деякі дії або події в будь-якій з методик, описаних в цьому документі, можуть виконуватися в іншій послідовності, можуть додаватися, об'єднуватися
 55 або повністю пропускатися (наприклад, не всі описані дії або події необхідні для застосування методик на практиці). Крім того, в деяких прикладах дії або події можуть виконуватися одночасно, наприклад, за допомогою багатопотокової обробки, обробки переривань або декількох процесорів, а не послідовно.

У одному або більше прикладах описувані функції можуть бути реалізовані в апаратних засобах, програмному забезпеченні, мікропрограмному забезпеченні або будь-якому їх

поєднанні. Якщо вони реалізовані в програмному забезпеченні, то функції можуть зберігатися або передаватися у вигляді однієї або більше інструкцій або коду на зчитуваному комп'ютером носії і виконуватися апаратним модулем обробки. Зчитувані комп'ютером носії можуть включати в себе зчитувані комп'ютером носії інформації, які відповідають матеріальному носію, такому як носії інформації, або засоби зв'язку, що включають в себе будь-який носій, який полегшує перенесення комп'ютерної програми з одного місця в інше, наприклад, відповідно до деякого протоколу зв'язку. Таким чином, зчитувані комп'ютером носії загалом можуть відповідати (1) матеріальним зчитуваним комп'ютером носіям інформації, які не змінюються з часом, або (2) засобу зв'язку, такому як сигнал або несуча. Носії інформації можуть бути будь-якими доступними носіями, до яких можна звертатися за допомогою одного або більше комп'ютерів або одного або більше процесорів для витягання інструкції, коду і/або структури даних для реалізації методик, описані в даному розкритті винаходу. Комп'ютерний програмний продукт може включати в себе зчитуваний комп'ютером носій.

Як приклад, а не обмеження, такі зчитувані комп'ютером носії інформації можуть бути виконані у вигляді RAM, ROM, EEPROM, компакт-диска або іншого накопичувача на оптичних дисках, накопичувача на магнітних дисках або інших магнітних запам'ятовуючих пристроїв, флеш-пам'яті або будь-якого іншого носія, який може використовуватися для зберігання потрібного програмного коду у вигляді інструкцій або структур даних і до якого [носія] можна звертатися за допомогою комп'ютера. Також будь-яке з'єднання коректно називати зчитуванням комп'ютером носієм. Наприклад, якщо інструкції передаються з веб-сайта, сервера або іншого віддаленого джерела з використанням коаксіального кабелю, оптоволоконного кабелю, витой пари, цифрової абонентської лінії (DSL) або бездротових технологій, наприклад ІЧ-зв'язку, радіочастотного зв'язку і НВЧ-зв'язку, то коаксіальний кабель, оптоволоконний кабель, вита пара, DSL або бездротові технології, наприклад ІЧ-зв'язок, радіочастотний зв'язок і НВЧ-зв'язок, включаються у визначення носія. Однак потрібно розуміти, що зчитувані комп'ютером носії інформації і носії інформації не включають в себе з'єднання, несучі, сигнали або інші тимчасові носії, а замість цього орієнтовані на постійні, матеріальні носії інформації. Диски при використанні в даному документі включають в себе компакт-диск (CD), лазерний диск, оптичний диск, цифровий універсальний диск (DVD), гнучкий диск і диск Blu-ray, де диски (disks) звичайно відтворюють дані магнітним способом, тоді як диски (discs) відтворюють дані оптично за допомогою лазерів. Поєднання вищепереліченого також потрібно включити в поняття зчитуваних комп'ютером носіїв.

Інструкції можуть виконуватися одним або більше процесорами, наприклад одним або більше цифровими процесорами сигналів (DSP), універсальними мікропроцесорами, спеціалізованими інтегральними схемами (ASIC), програмованими користувачем логічними матрицями (FPGA) або іншими еквівалентними інтегральними або дискретними логічними схемами. Відповідно, термін "процесор" при використанні в даному документі може стосуватися будь-якої вищезазначеної структури або будь-якої іншої структури, придатної для реалізації описаних в цьому документі методик. До того ж в деяких особливостях функціональні можливості, описані в цьому документі, можуть бути надані в спеціалізованих апаратних і/або програмних модулях, сконфігурованих для кодування і декодування, або вбудовані в об'єднаний кодек. Також методики можна було б повністю реалізувати в одній або більше схемах або логічних елементах.

У ще одних прикладах дане розкриття винаходу передбачає зчитуваний комп'ютером носій, що містить збережену на ньому структуру даних, де структура даних включає в себе кодований бітовий потік відповідно до даного розкриття винаходу. Зокрема, кодований бітовий потік може включати в себе один або більше рівнів відеоданих і параметр відео (VPS) для одного або більше рівнів відеоданих, при цьому кожний з одного або більше рівнів відеоданих посиляється на VPS, і один або більше рівнів відеоданих кодуються, основууючись щонайменше частково на VPS.

Методики з даного розкриття винаходу можуть бути реалізовані в широкому спектрі пристроїв, включаючи бездротову телефонну трубку, інтегральну схему (IC) або набір IC (наприклад, набір мікросхем). Різні компоненти, модулі або блоки описуються в даному розкритті винаходу для підкреслення функціональних особливостей пристроїв, сконфігурованих для виконання розкритих методик, але не обов'язково вимагають реалізації за допомогою різних апаратних модулів. Точніше, як описано вище, різні модулі можуть об'єднуватися в апаратний модуль кодека або надаватися сукупністю взаємодіючих апаратних модулів, що включають в себе один або більше процесорів, як описано вище, в поєднанні з придатним програмним забезпеченням і/або мікропрограмним забезпеченням.

Описані різні приклади. Ці і інші приклади входять в обсяг нижченаведеної формули винаходу.

5

ФОРМУЛА ВИНАХОДУ

1. Спосіб кодування відеоданих, який включає етапи, на яких:
коднують набір параметрів відео (VPS) для бітового потоку, який містить множину рівнів відеоданих, при цьому кожний із згаданої множини рівнів відеоданих звертається до VPS, і при цьому кодування VPS містить:
кодування даних VPS, що вказують деяке число кадрів, які повинні бути переупорядковані в щонайменше одному зі згаданої множини рівнів відеоданих,
кодування даних VPS, що вказують деяке число зображень, які повинні бути збережені в буфері декодування зображень (DPB) під час декодування згаданої множини рівнів відеоданих;
кодування даних VPS, що вказують деяке максимальне число тимчасових рівнів в бітовому потоці, що включає в себе згадану множину рівнів відеоданих;
кодування даних VPS, що вказують деяке максимальне число видів в бітовому потоці, що включає в себе згадану множину рівнів відеоданих;
кодування інформації, яка задає відображення індексу вибірки на показник характеристик, при цьому кодування інформації, яка задає згадане відображення, містить кодування одного або більше з: відповідного просторового дозволу для кожного з множини індексів залежності, частоти кадрів для кожного з множини тимчасових індексів або ідентифікатора виду для кожного з множини індексів виду; і
кодування згаданої множини рівнів відеоданих, основуючись, щонайменше частково, на VPS.
2. Спосіб за п. 1, в якому кодування VPS додатково включає кодування даних VPS, що вказують один або більше наборів параметрів гіпотетичного еталонного декодера (HRD).
3. Спосіб за п. 1, в якому кодування VPS додатково включає кодування даних VPS, що вказують, чи включає в себе VPS розширення понад відповідного стандарту, і, коли VPS включає в себе це розширення, даних для цього розширення.
4. Спосіб за п. 1, в якому кодування згаданої множини рівнів відеоданих включає кодування згаданої множини рівнів відеоданих відповідно до Високоєфективного кодування відео (HEVC).
5. Спосіб за п. 1, в якому кодування згаданої множини рівнів відеоданих включає кодування згаданої множини рівнів відеоданих відповідно до щонайменше одного з Багатовидового кодування відео (MVC) або Масштабованого кодування відео (SVC).
6. Спосіб за п. 1, в якому кодування VPS включає кодування інформації, що точно визначає для однієї або більше розмірностей згаданої множини рівнів відеоданих одне або більше з: деякого числа рівнів пріоритету в згаданій множині рівнів відеоданих, деякого числа рівнів залежності в згаданій множині рівнів відеоданих, деякого числа часових рівнів в згаданій множині рівнів відеоданих або максимального числа рівнів якості для будь-якого з рівнів залежності в згаданій множині рівнів відеоданих.
7. Спосіб за п. 6, в якому, коли підмножина із згаданої множини рівнів відеоданих має однакове просторове розрізнення і однакову бітову глибину, кожний з рівнів згаданої підмножини відповідає різним рівням з рівнів залежності.
8. Спосіб за п. 7, в якому кодування інформації, що задає зазначене відображення, включає кодування інформації, яка точно визначає відповідний показник характеристик для кожного з множини індексів характеристик, коли показник характеристик, який задає характеристики деякої розмірності згаданої множини рівнів відеоданих, не знаходиться в діапазоні індексів від нуля до лічильника розмірності вибірки мінус 1, при цьому лічильник задається індексом.
9. Спосіб за п. 7, в якому кодування інформації, що задає зазначене відображення, включає кодування одного або більше з: пари специфічних значень глибини для яскравості і кольоровості для кожного з множини індексів бітової глибини, або специфічного показника формату дискретизації кольоровості для кожного з множини форматів дискретизації кольоровості.
10. Спосіб за п. 1, в якому кодування VPS включає кодування інформації, що задає параметри керування і один або більше прапорів дозволу/заборони інструментів.
11. Спосіб за п. 1, в якому параметри керування і згадані один або більше прапорів дозволу/заборони інструментів містять одне або більше з:
pcm_bit_depth_luma_minus1,
pcm_bit_depth_chroma_minus1,

loop_filter_across_slice_flag,
pcm_loop_filter_disable_flag,
temporal_id_nesting_flag,

одного або більше пов'язаних з елементом мозаїки синтаксичних елементів,
5 chroma_pred_from_luma_enabled_flag,
sample_adaptive_offset_enabled_flag,
adaptive_loop_filter_enabled_flag або
inter_4x4_enabled_flag.

12. Спосіб за п. 1, в якому кодування VPS включає кодування інформації, що задає один або
10 більше дескрипторів робочих точок.

13. Спосіб за п. 12, в якому кодування інформації, що задає згадані один або більше
дескрипторів робочих точок, включає кодування інформації, що задає одне або більше з:
деякого числа максимальних робочих точок, залежності між різними рівнями або видами,
профілю і рівня для кожної з робочих точок або бітрейта для кожної з робочих точок.

14. Спосіб за п. 1, який додатково включає кодування відповідного набору параметрів
15 багаторівневої послідовності (LPS) для кожного із згаданої множини рівнів відеоданих, при
цьому кодування згаданої множини рівнів відеоданих, основуючись, щонайменше частково, на
VPS, включає кодування згаданої множини рівнів відеоданих, основуючись, щонайменше
частково, на VPS і відповідному LPS.

15. Спосіб за п. 14, в якому кодування відповідних LPS для кожного із згаданої множини рівнів
20 відеоданих включає кодування інформації, що задає указання розмірності вибірки, яке для
кожної розмірності вказує індекс до кожної розмірності.

16. Спосіб за п. 14, в якому кодування відповідних LPS для кожного із згаданої множини рівнів
відеоданих включає кодування інформації, що задає параметри керування і прапори
25 дозволу/заборони інструментів.

17. Спосіб за п. 16, в якому параметри керування і згадані один або більше прапорів
дозволу/заборони інструментів містять одне або більше з:

pcm_bit_depth_luma_minus1,
pcm_bit_depth_chroma_minus1,

30 loop_filter_across_slice_flag,
pcm_loop_filter_disable_flag,
одного або більше пов'язаних з елементом мозаїки синтаксичних елементів,
chroma_pred_from_luma_enabled_flag,
sample_adaptive_offset_enabled_flag,
35 adaptive_loop_filter_enabled_flag або
ієрархії одиниць кодування (CU).

18. Спосіб за п. 14, в якому кодування відповідних LPS для кожного із згаданої множини рівнів
відеоданих включає кодування інформації, що задає інформацію одного або більше інших
наборів параметрів, застосовуваних до щонайменше одного зі слайсу, групи слайсів,
40 зображення і декількох зображень, що звертаються до загального набору параметрів
зображення (PPS).

19. Спосіб за п. 1, який додатково включає кодування одного або більше наборів параметрів
зображення (PPS), так що PPS не звертаються до VPS, не звертаються до наборів параметрів
багаторівневої послідовності (LPS) згаданої множини рівнів відеоданих.

20. Спосіб за п. 19, в якому кодування згаданої множини рівнів відеоданих, основуючись,
45 щонайменше частково, на VPS, включає кодування згаданої множини рівнів відеоданих,
основується, щонайменше частково, на VPS, PPS і LPS, так, що, коли синтаксичний елемент
одного з PPS конфліктує з VPS або відповідним одним з LPS, кодують відповідний рівень із
згаданої множини рівнів відеоданих на основі синтаксичного елемента згаданого одного з PPS.

21. Спосіб за п. 1, який додатково включає кодування групувального набору параметрів (GPS),
50 який групує разом всі набори параметрів, в тому числі VPS, для згаданої множини рівнів
відеоданих.

22. Спосіб за п. 21, в якому кодування GPS включає кодування інформації, що задає
ідентифікатор GPS, причому спосіб додатково включає кодування інформації заголовка слайсу,
55 що відповідає ідентифікатору GPS.

23. Спосіб за п. 1, в якому кодування згаданої множини рівнів відеоданих включає декодування
згаданої множини рівнів відеоданих, і в якому кодування VPS містить аналіз VPS.

24. Спосіб за п. 1, в якому кодування згаданої множини рівнів відеоданих містить кодування
згаданої множини рівнів відеоданих, і в якому кодування VPS містить побудову VPS.

60 25. Пристрій для кодування відеоданих, що містить відеокодер, сконфігурований для:

- кодування набору параметрів відео (VPS) для бітового потоку, який містить множину рівнів відеоданих, при цьому кожен із згаданої множини рівнів відеоданих звертається до VPS, і при цьому відеокoder, сконфігурований для кодування VPS, сконфігурований для: кодування даних VPS, що вказують деяке число кадрів, які повинні бути переупорядковані в щонайменше одному
- 5 зі згаданої множини рівнів відеоданих, кодування даних VPS, які вказують деяке число зображень, які повинні бути збережені в буфері декодованих зображень (DPB) під час декодування згаданої множини рівнів відеоданих, кодування даних VPS, що вказують деяке максимальне число тимчасових рівнів в бітовому потоці, що включає в себе згадану множину рівнів відеоданих;
- 10 кодування даних VPS, що вказують деяке максимальне число видів в бітовому потоці, що включає в себе згадану множину рівнів відеоданих;
- кодування інформації, яка задає відображення індексу вибірки на показник характеристик, при цьому для кодування інформації, яка задає згадане відображення, відеокoder сконфігурований для кодування одного або більше з: відповідного просторового дозволу для кожного з множини
- 15 індексів залежності, частоти кадрів для кожного з множини тимчасових індексів або ідентифікатора виду для кожного з множини індексів виду; і
- кодування згаданої множини рівнів відеоданих, основуючись, щонайменше частково, на VPS.
26. Пристрій за п. 25, в якому відеокoder, який сконфігурований для кодування VPS, додатково сконфігурований для кодування даних VPS, що вказують один або більше наборів параметрів гіпотетичного еталонного декодера (HRD).
- 20 27. Пристрій за п. 25, в якому відеокoder, який сконфігурований для кодування VPS, додатково сконфігурований для кодування даних VPS, що вказують, чи включає VPS в себе розширення понад відповідного стандарту, і, коли VPS включає в себе це розширення, даних для цього розширення.
- 25 28. Пристрій за п. 25, в якому відеокoder сконфігурований для кодування згаданої множини рівнів відеоданих відповідно до одного з Високоєфективного кодування відео (HEVC), Багатовидового кодування відео (MVC) і Масштабованого кодування відео (SVC).
29. Пристрій за п. 25, в якому відеокoder містить відеодекодер, і при цьому пристрій додатково містить пристрій відображення, який сконфігурований для відображення відеоданих.
- 30 30. Пристрій за п. 25, в якому відеокoder містить кодувальник відео, і при цьому пристрій додатково містить камеру, яка сконфігурована для генерування відеоданих.
31. Пристрій за п. 25, при цьому пристрій містить щонайменше одне з: інтегральної схеми; мікропроцесора; і пристрою бездротового мобільного зв'язку, який включає в себе відеокoder.
32. Пристрій за п. 25, в якому відеокoder, який сконфігурований для кодування VPS, сконфігурований для кодування інформації, яка задає один або більше дескрипторів робочих точок.
- 35 33. Пристрій за п. 32, в якому для кодування інформації, яка задає згадані один або більше дескрипторів робочих точок, відеокoder сконфігурований для кодування інформації, яка задає одне або більше з: деякого числа максимальних робочих точок, залежності між різними рівнями або видами, профілю і рівня для кожної з робочих точок або бітрейта для кожної з робочих точок.
- 40 34. Пристрій для кодування відеоданих, що містить: засіб для кодування набору параметрів відео (VPS) для бітового потоку, який містить множину рівнів відеоданих, при цьому кожен із згаданої множини рівнів відеоданих звертається до VPS, і при цьому засіб для кодування VPS
- 45 містить: засіб для кодування даних VPS, що вказують деяке число кадрів, які повинні бути переупорядковані в щонайменше одному із згаданої множини рівнів відеоданих, засіб для кодування даних VPS, що вказують деяке число зображень, які повинні бути збережені в буфері декодованих зображень (DPB) під час декодування згаданої множини рівнів відеоданих;
- засіб для кодування даних VPS, що вказують деяке максимальне число тимчасових рівнів в бітовому потоці, що включає в себе згадану множину рівнів відеоданих;
- 50 засіб для кодування даних VPS, що вказують деяке максимальне число видів в бітовому потоці, що включає в себе згадану множину рівнів відеоданих;
- засіб для кодування інформації, яка задає відображення індексу вибірки на показник характеристик, при цьому засіб для кодування інформації, яка задає згадане відображення,
- 55 містить засіб для кодування одного або більше з: відповідного просторового дозволу для кожного з множини індексів залежності, частоти кадрів для кожного з множини тимчасових індексів або ідентифікатора виду для кожного з множини індексів виду; і
- засіб для кодування згаданої множини рівнів відеоданих, основуючись, щонайменше частково, на VPS.

35. Пристрій за п. 34, в якому засіб для кодування VPS додатково містить засіб для кодування даних VPS, що вказують один або більше наборів параметрів гіпотетичного еталонного декодера (HRD).

5 36. Пристрій за п. 34, в якому засіб для кодування VPS додатково містить засіб для кодування даних VPS, що вказують, чи включає VPS в себе розширення понад відповідного стандарту, і, коли VPS включає в себе це розширення, даних для цього розширення.

37. Пристрій за п. 34, в якому засіб для кодування VPS містить засіб для кодування згаданої множини рівнів відеоданих відповідно до одного з Високоєфективного кодування відео (HEVC), Багатовидового кодування відео (MVC) і Масштабованого кодування відео (SVC).

10 38. Пристрій за п. 34, в якому засіб для кодування VPS містить засіб для кодування інформації, яка задає один або більше дескрипторів робочих точок.

39. Пристрій за п. 38, в якому засіб для кодування інформації, яка задає згадані один або більше дескрипторів робочих точок, містить засіб для кодування інформації, яка задає одне або більше з: деякого числа максимальних робочих точок, залежності між різними рівнями або видами, профілю і рівня для кожної з робочих точок або бітрейта для кожної з робочих точок.

15 40. Довготривалий зчитуваний комп'ютером носій інформації, який зберігає інструкції, які, при виконанні, спонукають процесор: кодувати набір параметрів відео (VPS) для бітового потоку, який містить множину рівнів відеоданих, при цьому кожен із згаданої множини рівнів відеоданих звертається до VPS, і при цьому інструкції, які спонукають процесор кодувати VPS, містять інструкції, які спонукають процесор: кодувати дані VPS, що вказують деяке число кадрів, які повинні бути переупорядковані в щонайменше одному із згаданої множини рівнів відеоданих, кодувати дані VPS, що вказують деяке число зображень, які повинні бути збережені в буфері декодованих зображень (DPB) під час декодування згаданої множини рівнів відеоданих; кодувати дані VPS, що вказують деяке максимальне число тимчасових рівнів в бітовому потоці, що включає в себе згадану множину рівнів відеоданих; кодувати дані VPS, що вказують деяке максимальне число видів в бітовому потоці, що включає в себе згадану множину рівнів відеоданих;

20 кодувати інформацію, яка задає відображення індексу вибірки на показник характеристик, при цьому інструкції, які спонукають процесор кодувати інформацію, яка задає згадане відображення, додатково містять інструкції, які, при виконанні, спонукають процесор кодувати одне або більше з: відповідного просторового дозволу для кожного з множини індексів залежності, частоти кадрів для кожного з множини тимчасових індексів або ідентифікатора виду для кожного з множини індексів виду; і кодувати згадану множину рівнів відеоданих, основуючись, щонайменше частково, на VPS.

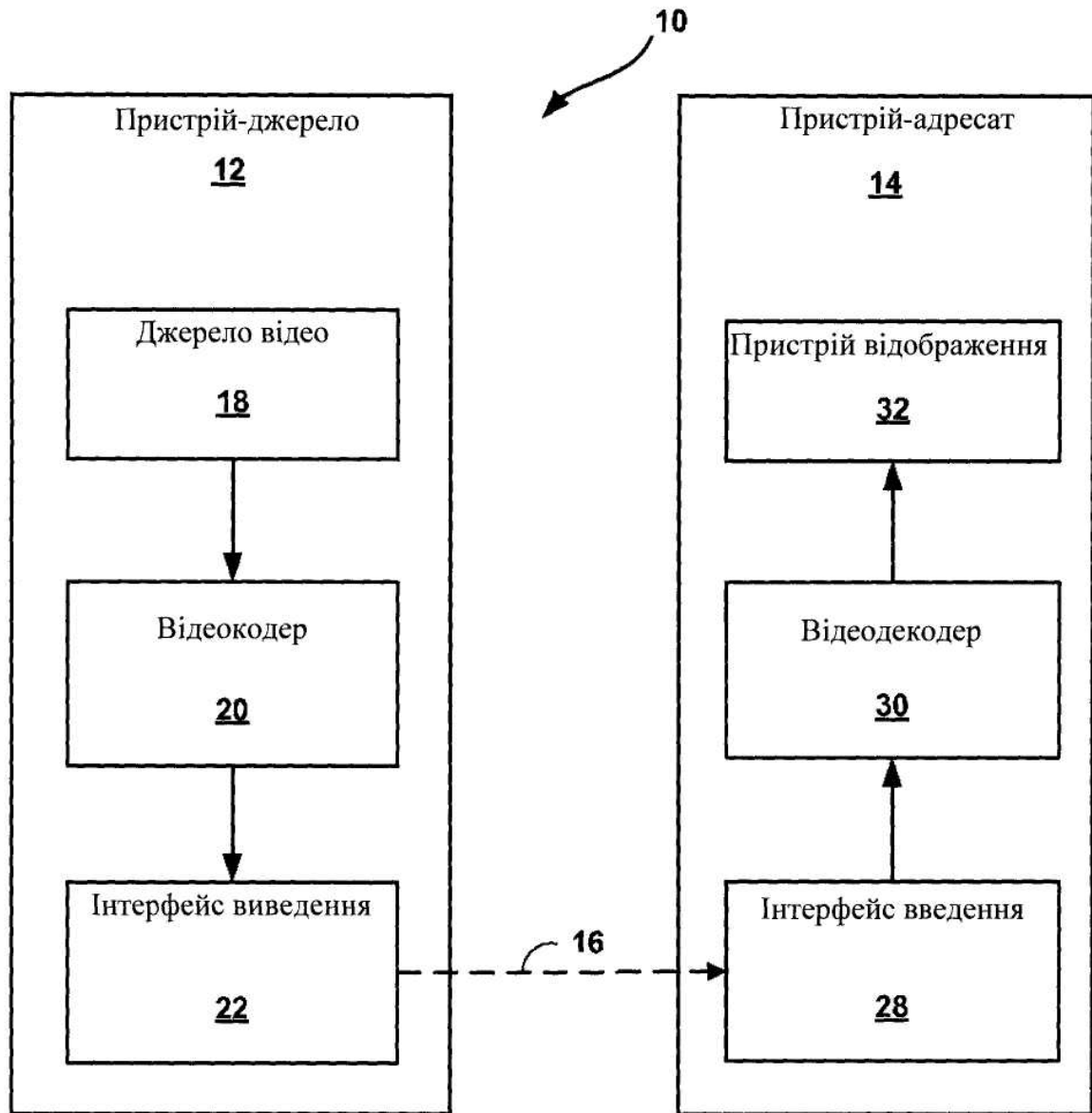
35 41. Довготривалий зчитуваний комп'ютером носій інформації за п. 40, в якому інструкції, які спонукають процесор кодувати VPS, додатково містять інструкції, які, при виконанні, спонукають процесор кодувати дані VPS, що вказують один або більше наборів параметрів гіпотетичного еталонного декодера (HRD).

40 42. Довготривалий зчитуваний комп'ютером носій інформації за п. 40, в якому інструкції, які при виконанні, спонукають процесор кодувати VPS, додатково містять інструкції, які спонукають процесор кодувати дані VPS, що вказують, чи включає VPS в себе розширення понад відповідного стандарту, і, коли VPS включає в себе це розширення, дані для цього розширення.

45 43. Довготривалий зчитуваний комп'ютером носій інформації за п. 40, в якому інструкції, які спонукають процесор кодувати згадану множину рівнів відеоданих, містять інструкції, які, при виконанні, спонукають процесор кодувати згадану множину рівнів відеоданих відповідно до одного з Високоєфективного кодування відео (HEVC), Багатовидового кодування відео (MVC) і Масштабованого кодування відео (SVC).

50 44. Довготривалий зчитування комп'ютером носій інформації за п. 40, в якому інструкції, які спонукають процесор кодувати VPS, додатково містять інструкції, які, при виконанні, спонукають процесор кодувати інформацію, яка задає один або більше дескрипторів робочих точок.

55 45. Довготривалий зчитування комп'ютером носій інформації за п. 44, в якому інструкції, які спонукають процесор кодувати інформацію, яка задає один або більше дескрипторів робочих точок, додатково містять інструкції, які, при виконанні, спонукають процесор кодувати інформацію, яка задає одне або більше з: деякого числа максимальних робочих точок, залежності між різними рівнями або видами, профілю і рівня для кожної з робочих точок або бітрейта для кожної з робочих точок.



Фіг. 1

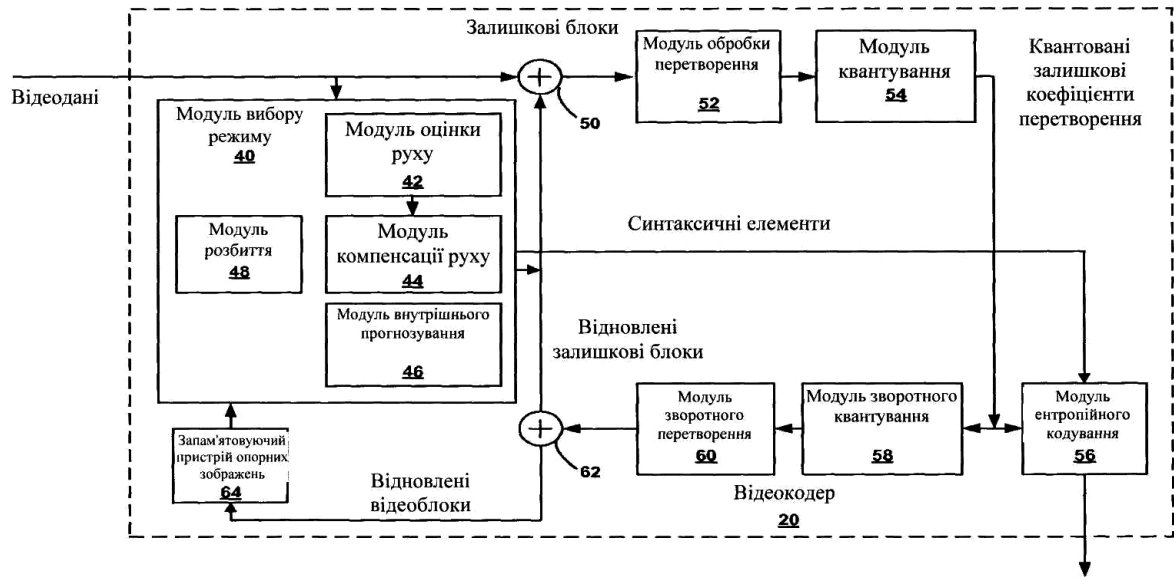


Fig. 2

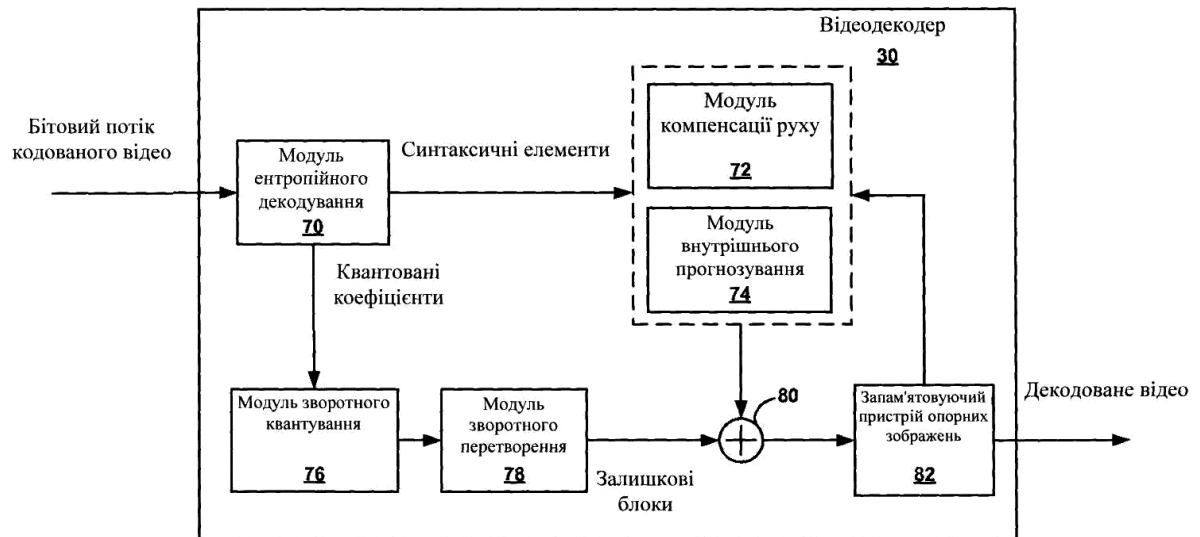
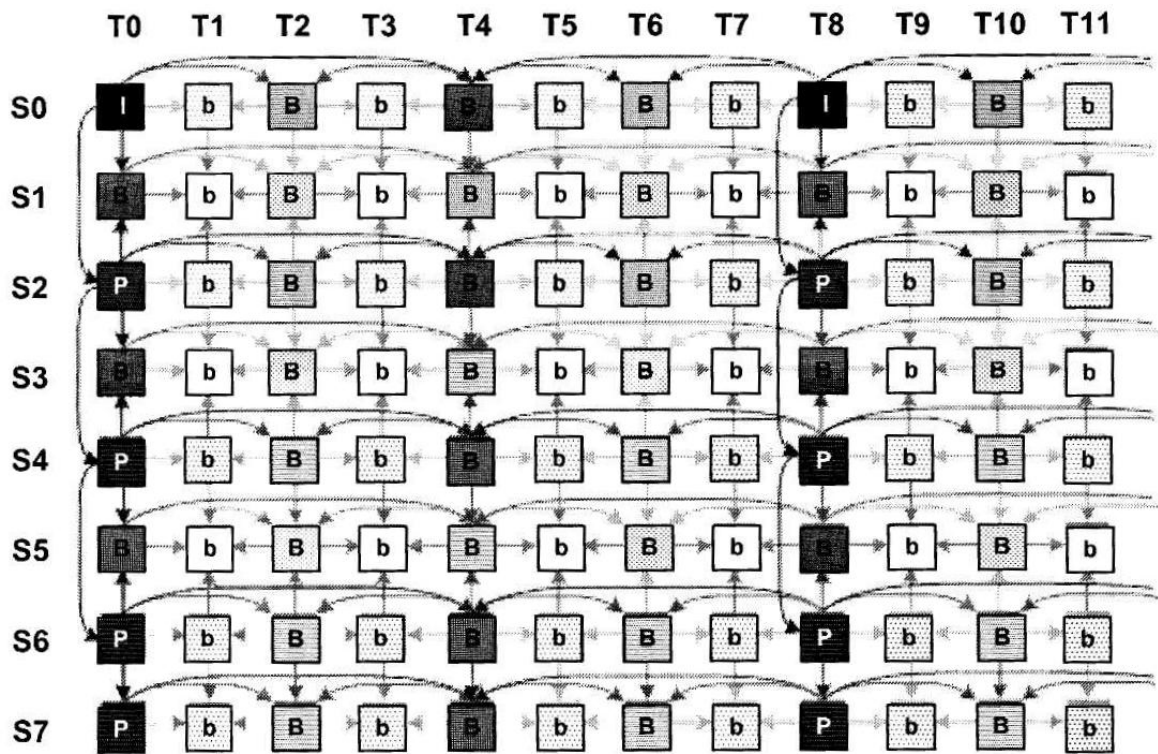
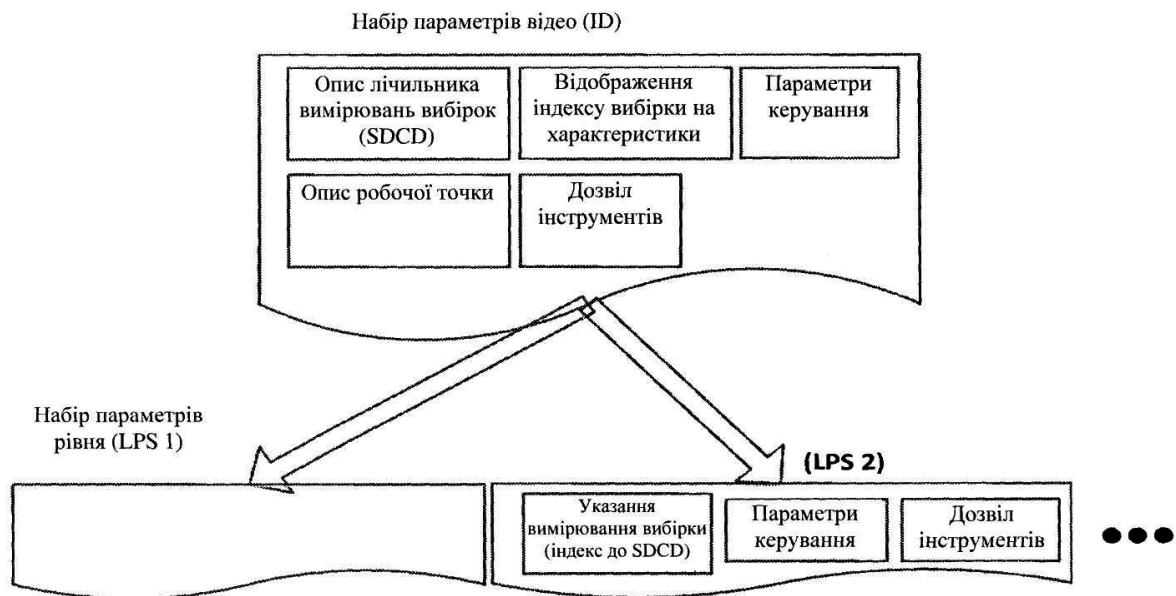


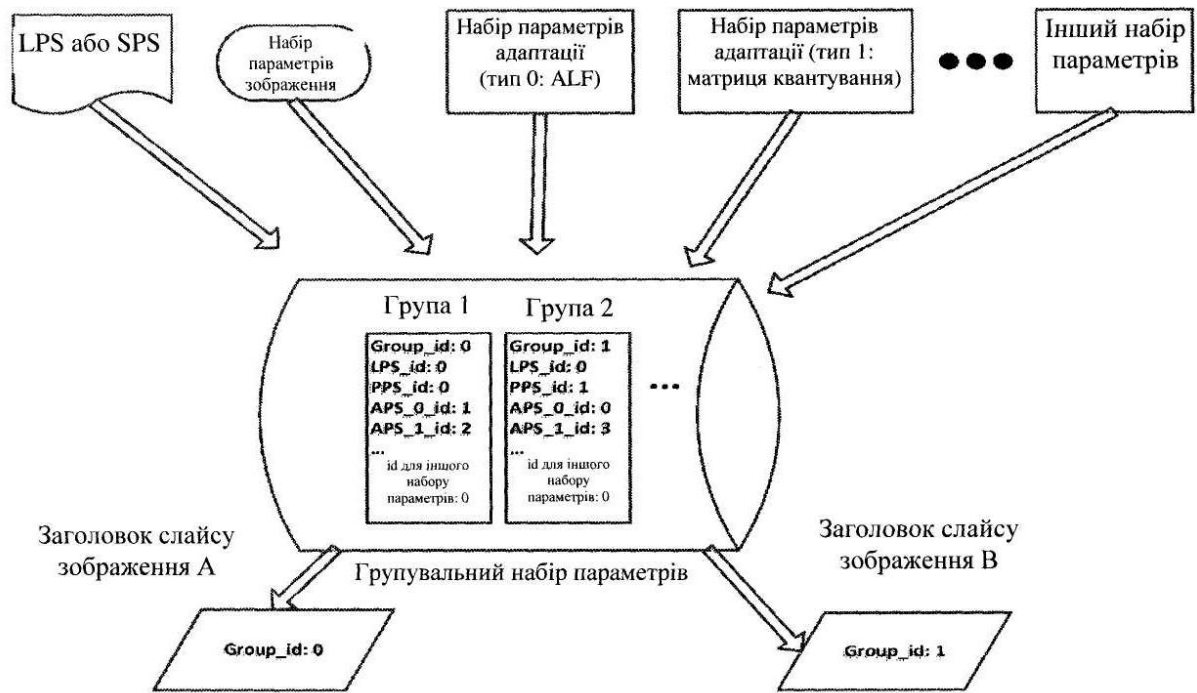
Fig. 3



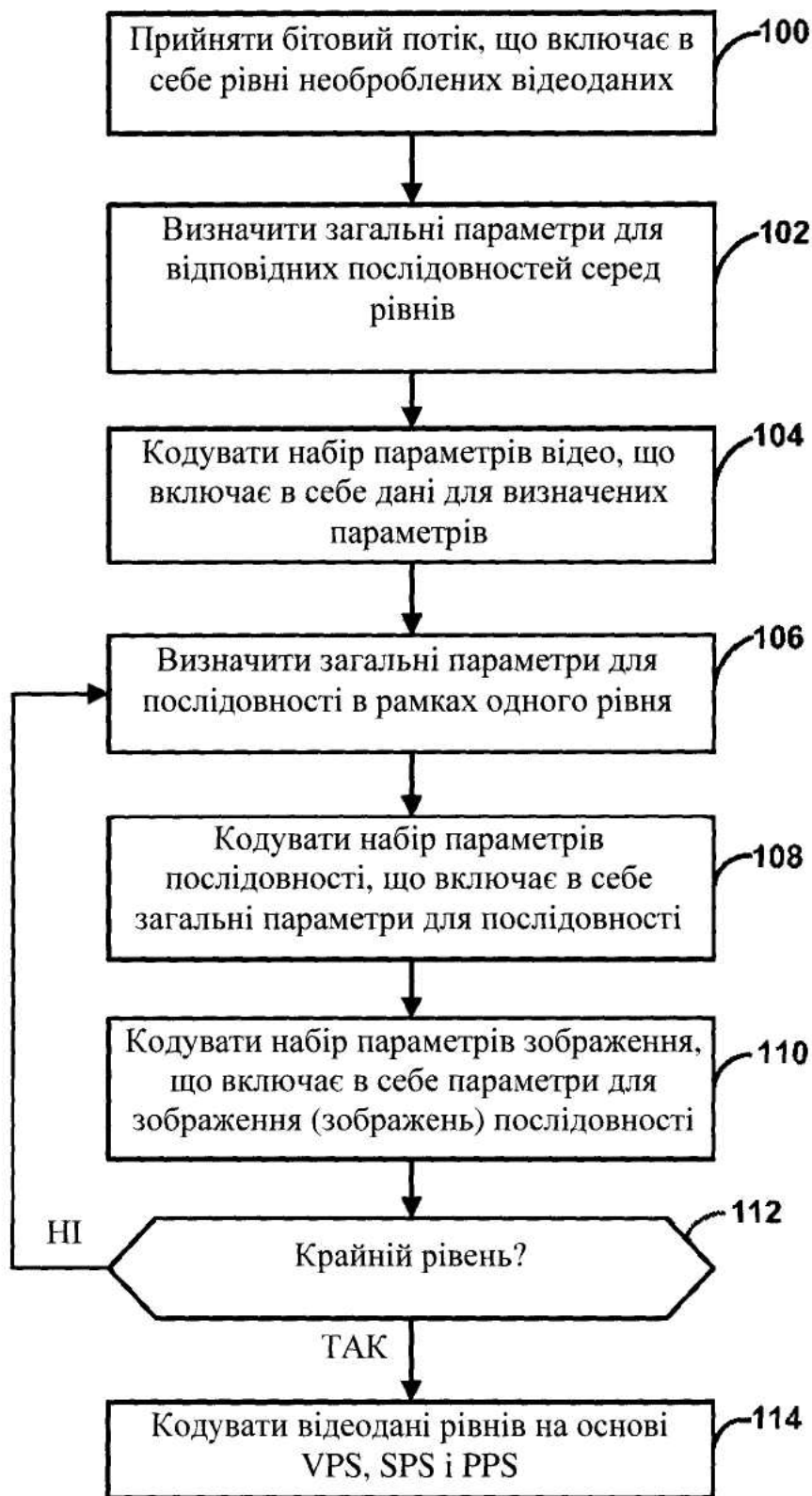
Фіг. 4



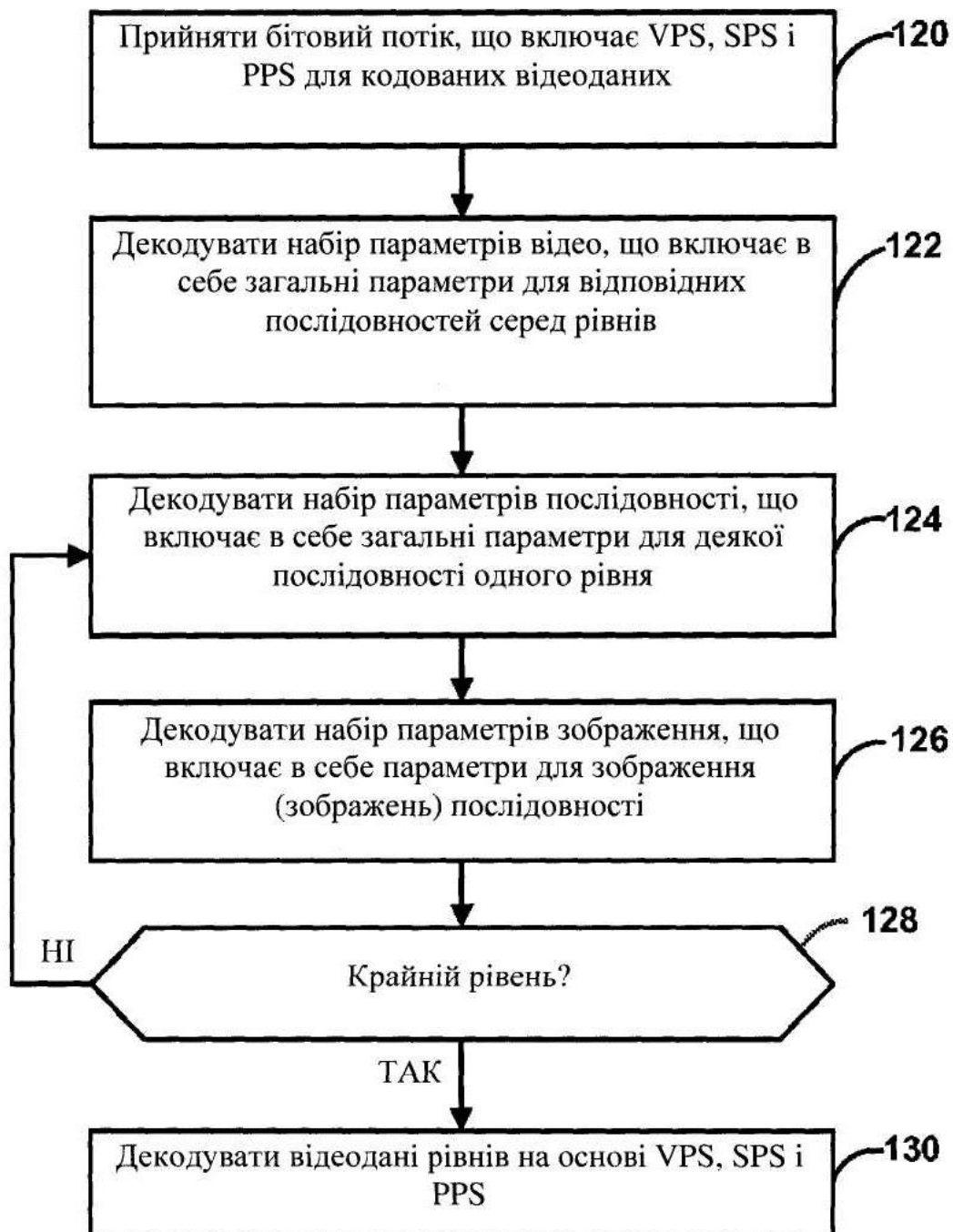
Фіг. 5



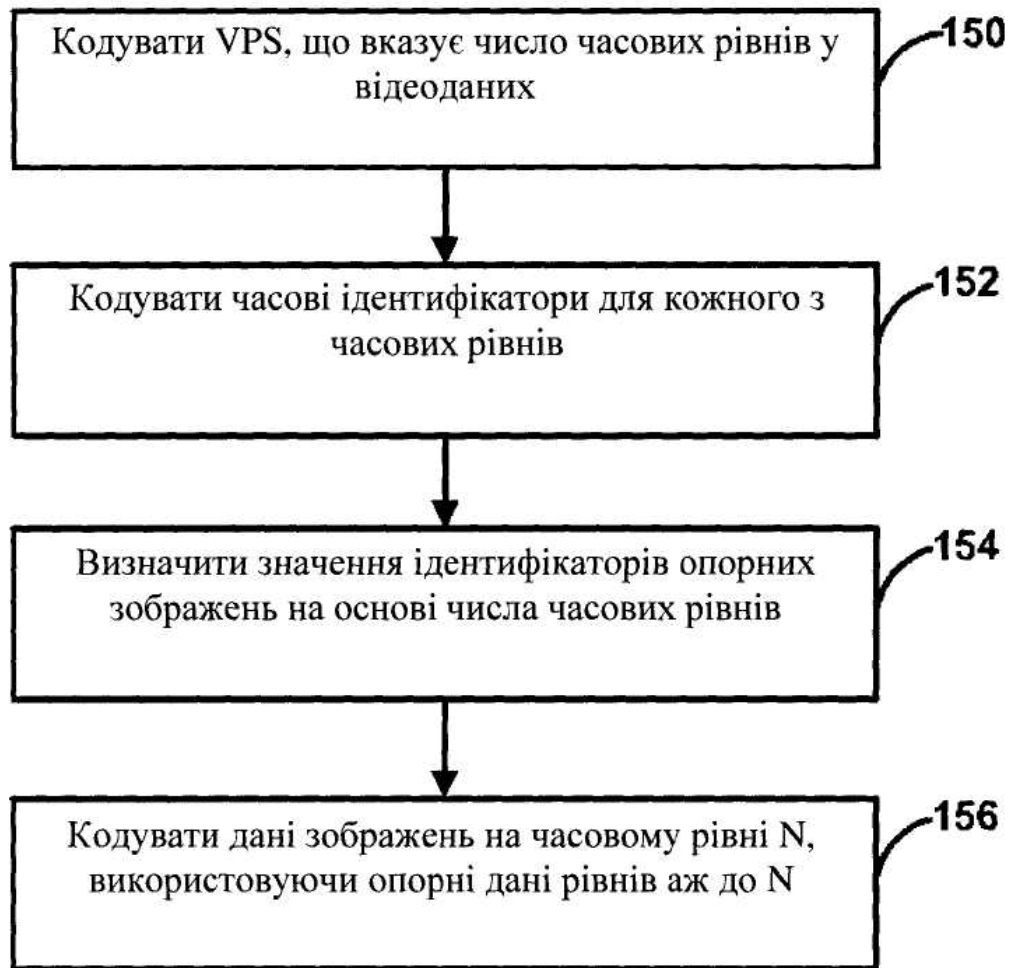
Фіг. 6



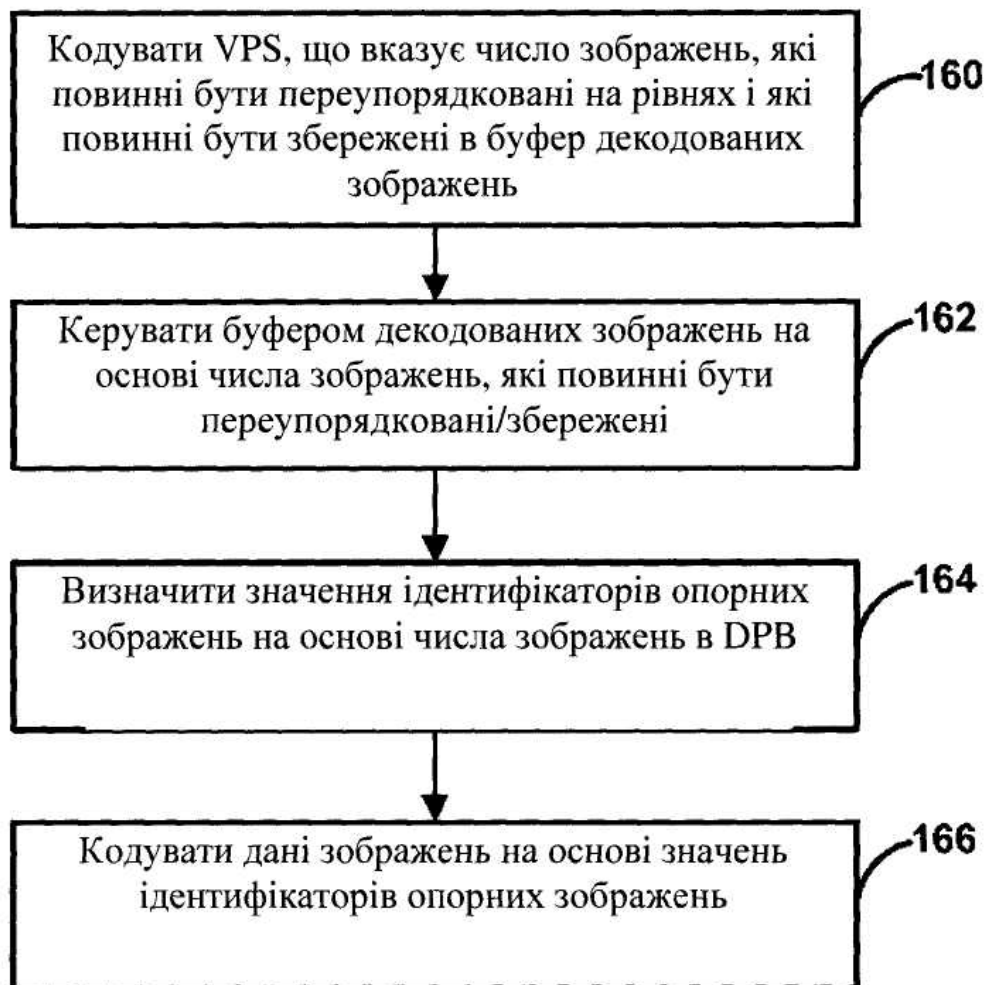
Фіг. 7



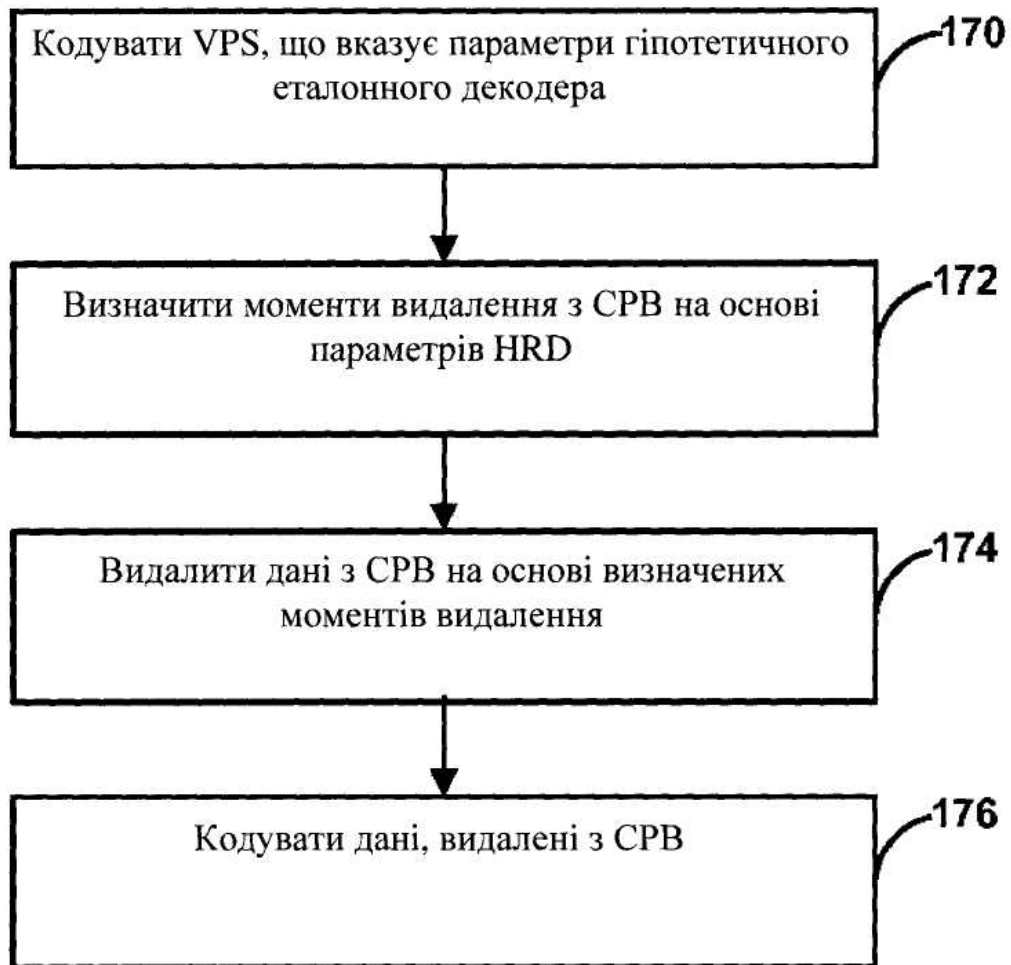
Фіг. 8



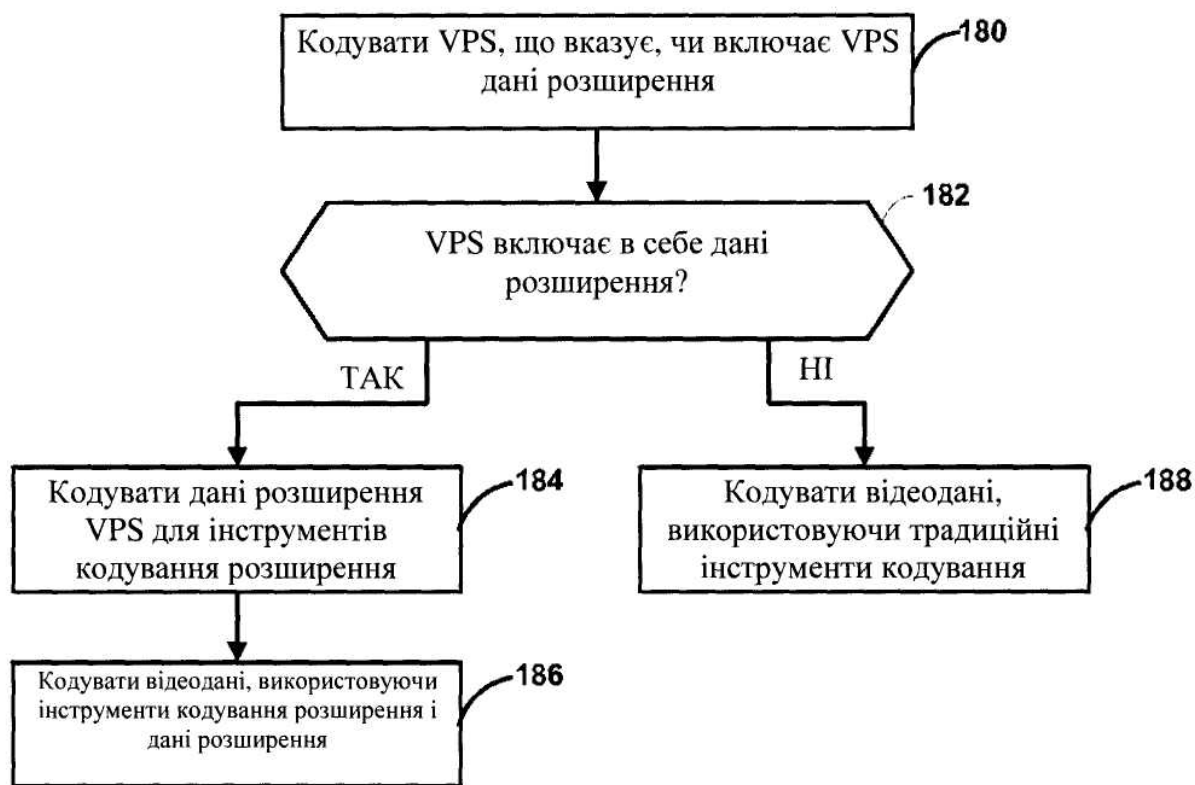
Фіг. 9



Фіг. 10



Фіг. 11



Фіг.12