



ДЕРЖАВНА СЛУЖБА
ІНТЕЛЕКТУАЛЬНОЇ
ВЛАСНОСТІ
УКРАЇНИ

УКРАЇНА

(19) **UA** (11) **114005** (13) **C2**
(51) МПК (2017.01)
H04N 7/00

(12) ОПИС ДО ПАТЕНТУ НА ВИНАХІД

(21) Номер заявки:	а 2015 04077	(72) Винахідник(и):	Ван Є-Куй (US)
(22) Дата подання заявки:	11.09.2013	(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:	10.04.2017	(74) Представник:	Мошинська Ніна Миколаївна, реєстр. №115
(31) Номер попередньої заявки відповідно до Паризької конвенції:	61/707,486, 61/708,404, 13/953,525	(56) Перелік документів, взятих до уваги експертизою:	Chen Y., Y-K. Wang. AHG10: On video parameter set for HEVC extensions . 10. JCT- VC meeting; 101. MPEG meeting; 11.07.2012 - 20.07.2012, Stockholm, (Joint Collaborative Team on Video Coding of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16). Nr: JCTVC-J0124. XP 030112486 Chen Y., Y-K. Wang. 3D-HLS: Video parameter set for 3D-HEVC JCT-3V meeting; 101. MPEG meeting; 16.07.2012 - 20.07.2012; Stockholm, (the Joint Collaborative Team on 3D Video Coding Extension Development of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16); Nr: JCT3V-A0099. XP 030130098 Chen Y., Y-K. Wang. AHG10: Video parameter set for HEVC base specification. JCT-VC meeting; 101. MPEG meeting, 11.07.2012 - 20.07.2012, Stockholm, (Joint Collaborative Team on Video Coding of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16); Nr: JCTVC-J0114, XP 030112476 Y-K. Wang, Chen Y. AHG9: On video parameter set. 11. JCT-VC meeting; 102. MPEG meeting, 10.10.2012 - 19.10.2012, Shanghai, (Joint Collaborative Team on Video Coding of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16); Nr: JCTVC-K0125, XP 030113007
(32) Дата подання попередньої заявки відповідно до Паризької конвенції:	28.09.2012, 01.10.2012, 29.07.2013		
(33) Код держави-учасниці Паризької конвенції, до якої подано попередню заявку:	US, US, US		
(41) Публікація відомостей про заявку:	10.08.2015, Бюл.№ 15		
(46) Публікація відомостей про видачу патенту:	10.04.2017, Бюл.№ 7		
(86) Номер та дата подання міжнародної заявки, поданої відповідно до Договору РСТ	PCT/US2013/059274, 11.09.2013		

(54) СИГНАЛІЗАЦІЯ ІДЕНТИФІКАТОРІВ РІВНІВ ДЛЯ РОБОЧИХ ТОЧОК ПРИ КОДУВАННІ ВІДЕО

(57) Реферат:

Описані методики стосуються кодування ідентифікаторів рівнів для робочих точок при кодуванні відео. В одному прикладі надається спосіб декодування відеоданих. Спосіб містить декодування синтаксичних елементів у наборі параметрів відео (VPS) в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту. Спосіб додатково

UA 114005 C2

містить декодування, за наявності, синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують інформацію про параметри гіпотетичного еталонного декодера (HRD), що має другий набір вмісту для першої робочої точки, причому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки, і причому декодування синтаксичних елементів містить декодування синтаксичних елементів, які вказують першу робочу точку і інформацію про параметри HRD тільки в рамках належних потоків бітів.



Фіг. 5

[0001] Дана заявка заявляє пріоритет за попередньою заявкою США № 61/707,486, поданою 28 вересня 2012 р., і попередньою заявкою США № 61/708,404, поданою 1 жовтня 2012 р., весь вміст яких включається в цей документ шляхом посилання.

ГАЛУЗЬ ТЕХНІКИ

5 [0002] Дане розкриття винаходу в цілому стосується обробки відеоданих, а конкретніше, - обробки робочих точок, що використовуються у відеоданих.

РІВЕНЬ ТЕХНІКИ

[0003] 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 записуючі пристрої, цифрові мультимедійні програвачі, відеоігрові пристрої, ігрові приставки, стільникові або супутникові радіотелефони, так звані «смартфони», пристрої для відеоконференцзв'язку, пристрої для потокового відео тощо. 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), який в цей час знаходиться в розробці, і розширеннях таких стандартів. Відеопристрої можуть ефективніше передавати, приймати, кодувати, декодувати і/або зберігати цифрову відеоінформацію за допомогою реалізації таких методик кодування відео.

20 [0004] Методики кодування відео включають в себе просторове (внутрішнє, intra-picture) прогнозування і/або часове (зовнішнє, inter-picture) прогнозування для зменшення або усунення надмірності, властиві відеопослідовностям. Для блокового кодування відео можна розбити слайс (slice) відео (наприклад, відеокадр або частину відеокадру) на відеоблоки, які також можуть називатися блоками дерева, одиницями кодування (CU) і/або вузлами кодування. 25 Відеоблоки в слайсі з внутрішнім кодуванням (intra-coded, I) зображення кодуються з використанням просторового прогнозування відносно еталонних вибірок у сусідніх блоках в тому самому зображенні. Відеоблоки в слайсі із зовнішнім кодуванням (inter-coded, P або B) зображення можуть використовувати просторове прогнозування відносно еталонних вибірок у сусідніх блоках в тому самому зображенні або часове прогнозування відносно еталонних вибірок в інших еталонних зображеннях. Зображення можуть називатися кадрами, а еталонні зображення можуть називатися еталонними кадрами.

[0005] Просторове або часове прогнозування дає в результаті блок з прогнозуванням для блока, який підлягає кодуванню. Залишкові дані являють собою різниці пікселів між вихідним блоком, який підлягає кодуванню, і блоком з прогнозуванням. Блок із зовнішнім кодуванням кодується відповідно до вектора руху, який вказує на блок еталонних вибірок, утворюючих блок з прогнозуванням, і залишкових даних, які 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 ділянки пікселів у ділянку перетворення, що дає в результаті залишкові коефіцієнти 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

45 [0006] В цілому, дане розкриття винаходу описує методики для сигналізації ідентифікаторів рівнів для робочих точок при кодуванні відео. Дані методики забезпечують підвищену ефективність сигналізації інформації, асоційованої з робочими точками, шляхом заборони кодування дублюючого вмісту щонайменше для одного з робочих точок або інформації про параметри гіпотетичного еталонного декодера (HRD), зв'язаної з робочими точками, в наборі параметрів відео (VPS) в рамках належного потоку бітів. Відповідно до методик належний потік бітів може не включати в себе синтаксичні елементи в VPS, які дублюють набір ідентифікаторів рівнів, асоційований з першою робочою точкою, як вміст для другої робочої точки. Крім того, відповідно до даних методик належний потік бітів може не включати в себе синтаксичні елементи в VPS, які дублюють набір параметрів HRD, асоційований з інформацією про параметри HRD для першої робочої точки, як вміст для другої робочої точки. Таким чином, 50 набір ідентифікаторів рівнів і набір параметрів HRD є унікальними для першої робочої точки в VPS.

[0007] В одному прикладі методики, описані в даному розкритті винаходу, належать до способу для декодування відеоданих. Спосіб включає в себе декодування синтаксичних елементів у наборі параметрів відео (VPS) в рамках належного потоку бітів, які вказують першу 60

робочу точку, яка має перший набір вмісту. Спосіб додатково включає в себе декодування, за наявності, синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують інформацію про параметри гіпотетичного еталонного декодера (HRD), яка має другий набір вмісту для першої робочої точки, причому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки, і причому декодування синтаксичних елементів містить декодування синтаксичних елементів, які вказують першу робочу точку та інформацію про параметри HRD тільки в рамках належних потоків бітів.

[0008] В іншому прикладі методики, описані в даному розкритті винаходу, належать до способу для кодування відеоданих. Спосіб включає в себе кодування синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту. Спосіб додатково включає в себе кодування синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, що яка другий набір вмісту для першої робочої точки, причому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки, і причому кодування синтаксичних елементів містить кодування синтаксичних елементів, які вказують першу робочу точку та інформацію про параметри HRD тільки з належних потоків бітів.

[0009] В іншому прикладі методики, описані в даному розкритті винаходу, належать до пристрою кодування відео. Пристрій кодування відео сконфігурований з можливістю кодування синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту. Пристрій кодування відео додатково сконфігурований з можливістю кодування, за наявності, синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки, причому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки, і причому відеокoder сконфігурований з можливістю кодування синтаксичних елементів, які вказують першу робочу точку та інформацію про параметри HRD тільки в рамках належних потоків бітів.

[0010] У ще одному прикладі методики, описані в цьому документі, належать до комп'ютерно-зчитуваного запам'ятовуючого носія. Комп'ютерно-зчитуваний запам'ятовуючий носій містить збережені на ньому команди, які при виконанні приписують процесору в пристрої для кодування відеоданих кодувати синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту. Команди при виконанні додатково приписують процесору кодувати, за наявності, синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки, причому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки, і причому процесор сконфігурований з можливістю кодування синтаксичних елементів, які вказують першу робочу точку та інформацію про параметри HRD тільки в рамках належних потоків бітів.

[0011] Методики, описані в цьому документі, також включають в себе приклад пристрою кодування відео, який включає в себе засіб для кодування синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту. Пристрій кодування відео додатково включає в себе засіб для кодування, за наявності, синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки, причому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки, і причому пристрій кодування відео сконфігурований з можливістю кодування синтаксичних елементів, який вказує першу робочу точку та інформацію про параметри HRD тільки в рамках належних потоків бітів.

[0012] Подробиці одного або більше прикладів викладаються на прикладених кресленнях і в описі нижче. Інші ознаки, цілі та переваги стануть зрозумілі з опису, креслень і з формули винаходу.

КОРОТКИЙ ОПИС КРЕСЛЕНЬ

[0013] Фіг. 1 - блок-схема, що ілюструє зразкову систему кодування і декодування відео, яка може використовувати методики, описані в даному розкритті винаходу.

[0014] Фіг. 2 - блок-схема, що ілюструє зразковий відеокoder, який може реалізувати методики, описані в даному розкритті винаходу.

[0015] Фіг. 3 - блок-схема, що ілюструє зразковий відеодекодер, який може реалізувати методики, описані в даному розкритті винаходу.

[0016] Фіг. 4 - блок-схема, що ілюструє зразковий набір пристроїв, які утворюють частину мережі.

5 [0017] Фіг. 5 - блок-схема алгоритму, що ілюструє зразковий спосіб для декодування робочих точок, що використовуються у відеоданих, відповідно до методик, описаних в даному розкритті винаходу.

[0018] Фіг. 6 - блок-схема алгоритму, що ілюструє зразковий спосіб для кодування робочих точок, що використовуються у відеоданих, відповідно до методик, описаних в даному розкритті винаходу.

ДОКЛАДНИЙ ОПИС

10 [0019] В цілому, дане розкриття винаходу описує методики для сигналізації ідентифікаторів рівнів для робочих точок при кодуванні відео. Робочі точки належать до субпотоків бітів, які можуть витягуватися з вихідного потоку бітів, який є масштабованим у часі і/або має декілька рівнів або видів. Субпотоки бітів можуть витягуватися з потоку бітів на основі значень ідентифікаторів рівнів та ідентифікаторів часових підрівнів, які ідентифікують робочу точку потоку бітів. Робочі точки сигналізуються в наборі параметрів відео (VPS) в рамках потоку бітів. Для кожної з робочих точок структура синтаксису робочих точок задає набір ідентифікаторів рівнів, що використовується для ідентифікації одиниць рівня абстракції мережі (NAL) в потоці бітів, які належать субпотіку бітів заданої робочої точки. Таким чином, одиниці NAL, які складають субпотік бітів заданої робочої точки, можна витягнути з вихідного потоку бітів на основі ідентифікаторів рівнів у одиниць NAL.

20 [0020] У деяких випадках можуть бути присутніми параметри гіпотетичного еталонного декодера (HRD), які належать до однієї або більше робочих точок. У цьому випадку інформація про параметри HRD сигналізується в VPS. Для кожної з однієї або більше робочих точок, які мають параметри HRD, синтаксична структура параметрів HRD задає набір параметрів HRD, який задає HRD, що використовується для перевірки, чи є належним субпотік бітів у заданої робочої точки.

30 [0021] Методики, описані в даному розкритті винаходу, включають в себе заборону дублюючого вмісту щонайменше для одного з робочих точок або інформації про параметри HRD, асоційованої з робочими точками, в VPS в рамках належного потоку бітів. Належний потік бітів може включати в себе синтаксичні елементи в VPS, які вказують першу робочу точку, яка має перший набір вмісту, який може включати в себе набір ідентифікаторів рівнів для першої робочої точки. Належний потік бітів також може включати в себе синтаксичні елементи в VPS, які вказують інформацію про параметри HRD, яка має другий набір вмісту, який може включати в себе набір параметрів HRD для першої робочої точки. Належний потік бітів може не включати в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки.

40 [0022] Відповідно до методик відеокодер сконфігурований з можливістю кодування тільки належних потоків бітів, і відеодекодер сконфігурований з можливістю декодування тільки належних потоків бітів. В одному прикладі в належному потоці бітів заборонені дублюючі набори ідентифікаторів рівнів для різних робочих точок, сигналізованих в одному VPS. В іншому прикладі в належному потоці бітів заборонені дублюючі набори параметрів HRD в інформації про параметри HRD для різних робочих точок, сигналізованих в одному VPS. Таким чином, заданий набір ідентифікаторів рівнів і заданий набір параметрів HRD є унікальними для першої робочої точки в VPS. Як інший приклад методики включають в себе кодування параметрів HRD для потоку бітів, причому кожна копія синтаксичної структури параметрів HRD включає в себе різний вміст.

50 [0023] Методики з даного розкриття винаходу підвищують ефективність сигналізації ідентифікаторів рівнів для робочих точок у належному потоці бітів. Наприклад, методики можуть підвищити ефективність за допомогою кодування тільки унікальних наборів вмісту для робочих точок і для інформації про параметри HRD, асоційованої з робочими точками, в одному VPS в рамках належного потоку бітів і заборони кодування дублюючого вмісту для різних робочих точок в одному VPS.

55 [0024] 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). Стандарти кодування відео додатково включають в себе розширення H.264 ITU-T з Масштабованого кодування відео (SVC) і Багатовидового кодування відео (MVC).

[0025] Крім того, є новий стандарт кодування відео, а саме Високоєфективне кодування відео (HEVC), що розробляється Об'єднаною командою з кодування відео (JCT-VC) з Експертної групи в галузі кодування відео (VCEG) ITU-T і Експертної групи з рухомих зображень (MPEG) ISO/IEC. Останнім робочим варіантом (WD) HEVC є робочий варіант 8, і надалі він називається WD8 HEVC під авторством Бросс (Bross) і інш., «High Efficiency Video Coding (HEVC) Text Specification Draft 8», липень 2012, Стокгольм, доступний станом на 14 травня 2013 р. за адресою http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip.

Хоча методики, описані в даному розкритті винаходу, описуються по відношенню до стандарту HEVC, аспекти даного розкриття винаходу не настільки обмежені, і їх можна поширити на інші стандарти кодування відео, а також на окремі методики кодування відео.

[0026] Фіг. 1 - блок-схема, що ілюструє зразкову систему 10 кодування і декодування відео, яка може використовувати методики, описані в даному розкритті винаходу. Як показано на фіг. 1, система 10 включає в себе пристрій-джерело 12, який формує кодовані відеодані, які пізніше повинні бути декодовані пристроєм-адресатом 14. Пристрій-джерело 12 і пристрій-адресат 14 можуть бути виконані у вигляді будь-якого з широкого діапазону пристроїв, що включає в себе настільні комп'ютери, ноутбуки (наприклад, 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 тощо. В деяких випадках пристрій-джерело 12 і пристрій-адресат 14 можуть бути обладнані для бездротового зв'язку.

[0027] У прикладі з фіг. 1 пристрій-джерело 12 включає в себе джерело 18 відео, відеокодер 20 та інтерфейс 22 виведення. Пристрій-адресат 14 включає в себе інтерфейс 28 введення, відеодекодер 30 і пристрій 32 відображення. В інших прикладах пристрій-джерело 12 і пристрій-адресат 14 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.

[0028] Проілюстрована система 10 з фіг. 1 є усього лише одним прикладом. Методики з даного розкриття винаходу можуть виконуватися будь-яким пристроєм кодування і/або декодування цифрового відео. Хоча методики виконуються, як правило, пристроєм кодування відео або пристроєм декодування відео, методики також можуть виконуватися відеокодером/декодером, який звичайно називається «кодеком». Крім того, методики з даного розкриття винаходу також можуть виконуватися передпроцесором 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 кожний 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.

[0029] 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 відео в режимі реального часу, 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 в цілому і можуть застосовуватися до бездротових і/або дротових додатків.

[0030] У кожному випадку захоплене, заздалегідь захоплене або сформоване комп'ютером відео може кодуватися відеокодером 20. Кодовані відеодані можуть передаватися напяму пристрою-адресату 14 через інтерфейс 22 виведення пристрою-джерела 12. Кодовані відеодані також (або як альтернатива) можуть зберігатися в запам'ятовуючому пристрої для подальшого звернення пристрою-адресата 14 або інших пристроїв, для декодування і/або відтворення.

[0031] Лінія 16 зв'язку може включати в себе короткочасні носії, наприклад бездротову широкомовну або дротову мережну передачу, або запам'ятовуючі носії (тобто довготривалі запам'ятовуючі носії), наприклад жорсткий диск, флеш-накопичувач, компакт-диск, цифровий

відеодиск, диск Blu-ray або інші комп'ютерно-зчитувані носії. У деяких прикладах мережний сервер може приймати кодовані відеодані від пристрою-джерела 12 і надавати кодовані відеодані пристрою-адресату 14, наприклад, за допомогою мережної передачі. Аналогічним чином, обчислювальний пристрій з обладнання з виробництва носіїв, наприклад обладнання із штампування оптичних дисків, може прийняти кодовані відеодані від пристрою-джерела 12 і виробити диск, що містить кодовані відеодані. Тому в різних прикладах лінія 16 зв'язку може вважатися такою, що включає в себе один або більше комп'ютерно-зчитуваних носіїв різних видів. Лінія 16 зв'язку може бути виконана у вигляді будь-якого типу носія або пристрою, що допускає переміщення кодованих відеоданих від пристрою-джерела 12 до пристрою-адресата 14. В одному прикладі лінія 16 зв'язку може бути виконана 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 відповідно до стандарту зв'язку, наприклад протоколу 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 корисним для спрощення зв'язку від пристрою-джерела 12 до пристрою-адресата 14.

[0032] Інтерфейс 28 введення в пристрої-адресаті 14 приймає інформацію з лінії 16 зв'язку, яка може бути комп'ютерно-зчитуваним носієм. Інформація з лінії 16 зв'язку може включати в себе синтаксичну інформацію, задану відеокодером 20, яка також використовується відеодекодером 30, яка включає в себе синтаксичні елементи, які описують характеристики і/або обробку блоків та інших кодованих одиниць, наприклад GOP. Пристрій 32 відображення може об'єднуватися з пристроєм-адресатом 14 або бути зовнішнім по відношенню до нього. 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 світловипромінюючих діодах (OLED) або інший тип пристрою відображення.

[0033] Як альтернатива кодовані дані можуть виводитися з інтерфейсу 22 виведення в запам'ятовуючий пристрій 34. Аналогічним чином до кодованих даних можна звертатися із запам'ятовуючого пристрою 34 за допомогою інтерфейсу введення. Запам'ятовуючий пристрій 34 може включати в себе будь-який з ряду розподілених або локально доступних запам'ятовуючих носіїв, наприклад жорсткий диск, диски Blu-ray, DVD, CD-ROM, флеш-пам'ять, енергозалежний або енергонезалежний запам'ятовуючий пристрій або будь-які інші придатні цифрові запам'ятовуючі носії для зберігання кодованих відеоданих. У додатковому прикладі запам'ятовуючий пристрій 34 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 34 за допомогою потокової передачі або завантаження. Файловий сервер може бути будь-яким типом сервера, що допускає зберігання кодованих відеоданих і передачу тих кодованих відеоданих пристрою-адресату 14. Зразкові файлові сервери включають в себе веб-сервер (наприклад, для веб-сайта), FTP-сервер, мережні 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 або поєднання їх обох, яке підходить для звернення до кодованих відеоданих, збережених на файловому сервері. Передача кодованих відеоданих із запам'ятовуючого пристрою 34 може бути потоковою передачею, завантаженням або їх поєднанням.

[0034] Методики з даного розкриття винаходу не обов'язково обмежуються бездротовими додатками або настройками. Методики можуть застосовуватися до кодування відео в підтримку будь-якого з ряду мультимедійних додатків, таких як ефірні телепередачі, кабельні телепередачі, супутникові телепередачі, передачі потокового відео, наприклад по Інтернету, кодування цифрового відео для збереження на запам'ятовуючий носій, декодування цифрового відео, збереженого на запам'ятовуючому носії, або інших додатків. У деяких прикладах система 10 може бути сконфігурована з можливістю підтримки однонаправленої або двонаправленої передачі відео для підтримки таких додатків, як потокова передача відео, відтворення відео, телевізійне мовлення і/або відеотелефонія.

[0035] Відеокодер 20 і відеокодер 30 можуть працювати відповідно до будь-якого стандарту кодування відео, наприклад стандарту HEVC, що в цей час знаходиться в розробці, і можуть відповідати Тестовій моделі HEVC (НМ). Як альтернатива відеокодер 20 і відеокодер 30 можуть працювати відповідно до інших власних або промислових стандартів, наприклад стандарту H.264 ITU-T, який, як альтернатива, називається 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 та H.263 ITU-T. У деяких аспектах відеокодер 20 і відеокодер 30 можуть бути об'єднані зі звуковим кодером та декодером і можуть включати в себе придатні модулі мультимплексування-демультиплексування або інші апаратні засоби і програмне забезпечення, щоб справлятися з кодуванням звуку 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).

[0036] Стандарт H.264/MPEG-4 (AVC) ITU-T був сформульований Експертною групою в галузі кодування відео (VCEG) ITU-T разом з Експертною групою з рухомих зображень (MPEG) ISO/IEC як результат колективної співпраці, відомої як Об'єднана команда з відео (JVT). У деяких аспектах описані в даному розкритті винаходу методики можуть застосовуватися до пристроїв, які в цілому 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 з відео (JVT) продовжує працювати над розширеннями до H.264/MPEG-4 AVC.

[0037] Відеокодер 20 і відеокодер 30 можуть бути реалізовані у вигляді будь-якої з ряду придатних схем кодера або декодера, наприклад, одного або більше мікропроцесорів, цифрових процесорів сигналів (DSP), спеціалізованих інтегральних схем (ASIC), програмованих користувачем вентильних матриць (FPGA), дискретних логічних схем, програмного забезпечення, апаратних засобів, мікропрограмного забезпечення або будь-яких їх поєднань. Коли методики реалізуються частково в програмному забезпеченні, пристрій може зберігати команди для програмного забезпечення на придатному довготривалому комп'ютерно-зчитуваному носії і виконувати команди на апаратних засобах, що використовують один або більше процесорів, для виконання методик з даного розкриття винаходу. Кожний з відеокодера 20 і відеокодера 30 може включатися в один або більше кодерів або декодерів, будь-який з яких може вбудовуватися як частина об'єданого кодера/декодера (кодека) у відповідному пристрої. Пристрій, що включає в себе відеокодер 20 і/або відеокодер 30, може містити інтегральну схему, мікропроцесор і/або пристрій бездротового зв'язку, наприклад стільниковий телефон.

[0038] JCT-VC працює над розвитком стандарту HEVC. Робота із стандартизації HEVC ґрунтується на моделі пристрою кодування відео, що розвивається, яка називається Тестовою моделлю HEVC (НМ). НМ передбачає декілька додаткових можливостей у пристрої 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, НМ може надати цілих тридцять три режими кодування з внутрішнім прогнозуванням.

[0039] Взагалі, робоча модель НМ описує, що відеокадр або зображення можна розділити на послідовність блоків дерева або найбільших одиниць кодування (LCU), які включають в себе вибірки яскравості та кольоровості. Синтаксичні дані в потоці бітів можуть задавати розмір для LCU, яка є найбільшою одиницею кодування в показниках кількості пікселів. Слайс включає в себе деяку кількість 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) відповідно до квадродререва. Звичайно структура даних квадродререва включає в себе один вузол на CU, причому кореневий вузол відповідає блоку дерева. Якщо CU розділяється на чотири суб-CU, то вузол, що відповідає CU, включає в себе чотири листи, кожен з яких відповідає одній із суб-CU.

[0040] Кожний вузол в структурі даних квадродререва може надавати синтаксичні дані для відповідної CU. Наприклад, вузол в квадродреві може включати в себе прапор розділення, який 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 16x16 більше не розділяється, то чотири суб-CU 8x8 також будуть називатися листовими CU, хоча CU 16x16 ніколи не розділялася.

[0041] CU має схожу з макроблоком зі стандарту H.264 мету за винятком того, що CU не має відмінності в розмірі. Наприклад, блок дерева можна розділити на чотири дочірніх вузли (які також називаються суб-CU), і кожний дочірній вузол, в свою чергу, може бути батьківським вузлом і розділятися на чотири інших дочірніх вузли. Кінцевий, неподілений дочірній вузол, який називається листом квадродререва, містить вузол кодування, який також називається листовою CU. Синтаксичні дані, асоційовані з кодованим потоком бітів, можуть задавати максимальну кількість разів, яку можна ділити блок дерева, яка називається максимальною глибиною CU, а також можуть задавати мінімальний розмір вузлів кодування. Відповідно, потік бітів 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 та їх субблоки в H.264/AVC).

[0042] 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 8x8 пікселів до розміру блока дерева з максимальною кількістю 64x64 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 пропускання 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 відповідно до квадродререва. TU може мати квадратну або неквадратну (наприклад, прямокутну) форму.

[0043] Стандарт HEVC допускає перетворення відповідно до TU, які можуть відрізнятися для різних CU. TU звичайно мають розмір на основі розміру PU в даній CU, заданій для розділеної LCU, хоча це не завжди може бути так. TU звичайно мають такий самий розмір або менше, ніж PU. У деяких прикладах залишкові вибірки, відповідні CU, можна поділити на менші одиниці, використовуючи структуру квадродререва, відому як «залишкове квадродререво» (RQT). Листи RQT можуть називатися одиницями перетворення (TU). Значення різниці пікселів, асоційовані з TU, можна перетворити для створення коефіцієнтів перетворення, які можна квантувати.

[0044] Листова CU може включати в себе одну або більше одиниць прогнозування (PU). Звичайно PU представляє просторову ділянку, яка відповідає всій або частині відповідної CU, і може включати в себе дані для витягання еталонної вибірки для PU. Крім того, PU включає в себе дані, зв'язані з прогнозуванням. Наприклад, коли PU кодується у внутрішньому режимі, дані для PU можуть включатися в залишкове квадродререво (RQT), яке може включати в себе дані, що описують режим внутрішнього прогнозування для TU, що відповідає PU. Як інший приклад, коли PU кодується у зовнішньому режимі, PU може включати в себе дані, які задають один або більше векторів руху для PU. Дані, які 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), еталонне зображення, на яке вказує вектор руху, і/або список еталонних зображень (наприклад, Список 0, Список 1 або Список C) для вектора руху.

[0045] Листова CU, яка має одну або більше PU, також може включати в себе одну або більше одиниць перетворення (TU). Одиниці перетворення можна задавати з використанням RQT (що також називається структурою квадродререва 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 самий режим внутрішнього прогнозування. Тобто один і той самий 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, яка відповідає 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 самої CU. У деяких прикладах максимальний розмір листової TU може відповідати розміру відповідної листової CU.

[0046] Крім того, TU в листових CU також можуть асоціюватися з відповідними структурами даних квадродререва, які називаються залишковими квадродреревами (RQT). Тобто листова CU може включати в себе квадродререво, яке вказує, як листова CU розбивається на TU. Кореневий вузол квадродререва TU, як правило, відповідає листовій CU, тоді як кореневий вузол квадродререва CU, як правило, відповідає блоку дерева (або LCU). TU в RQT, які не розділяються, називаються листовими TU. Взагалі, дане розкриття винаходу використовує терміни «CU» та «TU», щоб посилатися на листову CU і листову TU відповідно, доки не вказане інше.

[0047] Відеопослідовність, як правило, включає в себе послідовність відеокадрів або зображень. Група зображень (GOP), як правило, містить послідовність з одного або більше відеозображень. GOP може включати синтаксичні дані в заголовок GOP, заголовок одного або більше зображень або куди-небудь в інше місце, які описують кількість зображень, включених в GOP. Кожний слайс зображення може включати в себе синтаксичні дані слайса, які описують режим кодування для відповідного слайса. Відеокодер 20 звичайно діє на відеоблоки в окремих слайсах відео, щоб кодувати відеодані. Відеоблок може відповідати вузлу кодування в CU. Відеоблоки можуть мати фіксовані розміри або розміри, що змінюються, і можуть відрізнятися за розміром відповідно до заданого стандарту кодування.

[0048] Як приклад НМ підтримує прогнозування в різних розмірах PU. Припускаючи, що розмір конкретної CU рівний $2N \times 2N$, НМ 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 $N \times N$. НМ також підтримує асиметричне розбиття для зовнішнього прогнозування в розмірах PU $2N \times nU$, $2N \times nD$, $nL \times 2N$ та $nR \times 2N$. При асиметричному розбитті один напрямок CU не розбивається, тоді як інший напрямок розбивається на 25% та 75%. Частина CU, що відповідає розбиттю 25%, вказується за допомогою «n» з подальшою вказівкою «Зверху», «Знизу», «Зліва» або «Праворуч». 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.

[0049] 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 у горизонтальному напрямку ($x=16$). Також блок $N \times N$, як правило, має N пікселів у вертикальному напрямку і N пікселів у горизонтальному напрямку, де N представляє ненегативне ціле значення. Пікселі в блоці можуть розташовуватися в рядках і стовпцях. Крім того, блокам не обов'язково мати таку саму кількість пікселів у горизонтальному напрямку, як і у вертикальному напрямку. Наприклад, блоки можуть містити $N \times M$ пікселів, де M не обов'язково рівне N.

[0050] Після кодування з внутрішнім прогнозуванням або зовнішнім прогнозуванням, що використовує PU в CU, відеокодер 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 ділянці (яка також називається ділянкою пікселів), а TU можуть містити коефіцієнти в ділянці перетворення після застосування перетворення, наприклад дискретного косинусного перетворення (DCT), цілочисельного перетворення, вейвлет-перетворення або концептуально схожого перетворення до залишкових відеоданих. Залишкові дані можуть відповідати різницям пікселів між пікселями некодованого зображення і значеннями прогнозування, що відповідають PU. Відеокодер 20 може утворити TU, що включають в себе залишкові дані для CU, а потім перетворити TU для створення коефіцієнтів перетворення для CU.

[0051] Після будь-яких перетворень для створення коефіцієнтів перетворення відеокодер 20 може виконати квантування коефіцієнтів перетворення. Квантування в цілому належить до процесу, в якому коефіцієнти перетворення квантуються, щоб зменшити по можливості обсяг даних, що використовується 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 менший бік до m-бітового значення, де n більше m.

[0052] Після квантування відеокодер може сканувати коефіцієнти перетворення, створюючи одновимірний вектор з двовимірної матриці, що включає в себе квантовані коефіцієнти перетворення. Сканування може бути призначене для вміщення коефіцієнтів з більшою енергією (і тому з меншою частотою) попереду масиву і вміщення коефіцієнтів з меншою енергією (і тому більшою частотою) позаду масиву. У деяких прикладах відеокодер 20 може використовувати зумовлений порядок сканування, щоб сканувати квантовані коефіцієнти перетворення для створення серіалізованого вектора, який можна ентропійно кодувати. В інших

прикладх 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.

[0053] Для виконання CABAC відеокодер 20 може призначити контекст в рамках контекстної моделі символу, який підлягає передачі. Контекст може належати, наприклад, до того, чи є сусідні значення символу ненульовими. Для виконання CAVLC відеокодер 20 може вибрати код змінної довжини для символу, який підлягає передачі. 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, використання VLC може досягти економії бітів у порівнянні, наприклад, з використанням кодових слів рівної довжини для кожного символу, який підлягає передачі. Визначення імовірності може ґрунтуватися на контексті, призначеному символу.

[0054] Відеокодер 20 додатково може відправляти відеодекодеру 30 синтаксичні дані, наприклад блокові синтаксичні дані, кадрові синтаксичні дані та основані на GOP синтаксичні дані, наприклад, в заголовку кадру, заголовку блока, заголовку слайса або заголовку GOP. Синтаксичні дані GOP можуть описувати кількість кадрів у відповідній GOP, а синтаксичні дані кадру можуть вказувати режим кодування/прогнозування, що використовується для кодування відповідного кадру.

[0055] WD8 HEVC також допускає встановлення обмеженої кількості підмножин синтаксису за допомогою «профілів» і «рівнів». WD8 HEVC проектується універсальним в тому значенні, що він обслуговує широкий діапазон додатків, швидкостей передачі бітів, розрізень, якості і послуг. Додатки повинні охоплювати, в тому числі, цифрові запам'ятовуючі носії, телевізійне мовлення і зв'язок в реальному часі. Під час створення WD8 HEVC розглянуті різні вимоги від типових додатків, розроблені необхідні алгоритмічні елементи, і все це об'єднане в єдиний синтаксис. Тому WD8 HEVC спростить обмін відеоданими між різними додатками. Однак, беручи до уваги практичність реалізації повного синтаксису WD8 HEVC, профілі і рівні надають засіб для обмеженої кількості підмножин синтаксису.

[0056] «Профіль» задається як підмножина всього синтаксису потоку бітів, який задається за допомогою WD8 HEVC. У межах, встановлених синтаксисом заданого профілю, все ж можна вимагати дуже великого розкиду в продуктивності кодерів і декодерів залежно від значень, що приймаються синтаксичними елементами в потоці бітів. Наприклад, заданий розмір декодованих зображень може вимагати дуже великого розкиду в продуктивності кодерів і декодерів. У багатьох додатках в цей час не практично і не економічно реалізовувати декодер, що допускає роботу з усіма гіпотетичними використаннями синтаксису в конкретному профілі.

[0057] Щоб подолати цю проблему, в рамках кожного профілю задаються «класи» і «рівні». Рівень класу є заданим набором обмежень, накладених на значення синтаксичних елементів у потоці бітів. Ці обмеження можуть бути простими межами значень. Як альтернатива вони можуть приймати вигляд обмежень арифметичних поєднань значень (наприклад, ширина зображення, помножена на висоту зображення, помножена на кількість зображень, декодованих в секунду). Рівень, заданий для більш низького класу, є більш обмеженим, ніж рівень, заданий для більш високого класу. Один і той самий набір рівнів задається для всіх профілів, причому більшість аспектів визначення кожного рівня є спільною для різних профілів. Окремі реалізації в рамках заданих обмежень можуть підтримувати різний рівень для кожного профілю, що підтримується. В іншому контексті рівень є значенням коефіцієнта перетворення перед масштабуванням. Профілі і рівні детальніше описуються в Додатку А до WD8 HEVC.

[0058] Кодований відеоконтент, що відповідає WD8 HEVC, використовує загальний синтаксис. Щоб одержати підмножину повного синтаксису, в потік бітів включаються прапори, параметри і інші синтаксичні елементи, які сигналізують наявність або відсутність синтаксичних елементів, які виникають пізніше в потоці бітів.

[0059] WD8 HEVC задає підрівень як часовий масштабований рівень у часовому масштабованому потоці бітів, рівень абстракції мережі (NAL), що складається з одиниць рівня відеокодування (VCL) з конкретним значенням змінної TemporalId і асоційовані одиниці NAL не на VCL. WD8 HEVC додатково задає представлення підрівня як підмножини потоку бітів, що складається з одиниць NAL конкретного підрівня і більш низьких підрівнів.

[0060] Підпункт 10.1 в WD8 HEVC описує підмножини потоку бітів і процес витягання для формування субпотоків бітів. Взагалі, WD8 HEVC описує витягання субпотоків бітів з потоку бітів на основі значень ідентифікаторів рівнів і ідентифікаторів часових підрівнів, які ідентифікують робочу точку потоку бітів.

[0061] Робоча точка є потоком бітів, створеним з іншого потоку бітів за допомогою операції процесу витягання субпотоків бітів з іншим потоком бітів, цільовим найбільшим TemporalId і списком цільових ідентифікаторів рівнів як вхідних даних. Робоча точка ідентифікується набором значень nuh_reserved_zero_6bits, позначеним як OpLayerIdSet, і значенням TemporalId, позначеним як OpTid, і асоційована підмножина потоку бітів, виведена як результат процесу витягання субпотоків бітів, який заданий в WD8 HEVC, з OpTid та OpLayerIdSet як вхідні дані, є незалежно декодованим. Якщо цільовий найбільший TemporalId у робочій точки рівний найбільшому значенню TemporalId в наборі рівнів, асоційованому зі списком цільових ідентифікаторів рівнів, то робоча точка ідентична набору рівнів. В іншому випадку робоча точка є підмножиною набору рівнів.

[0062] Процес витягання субпотоків бітів є заданим процесом, за допомогою якого з потоку бітів віддаляються одиниці NAL в потоці бітів, які не належать цільовому набору, визначеному цільовим найбільшим TemporalId і списком цільових ідентифікаторів рівнів, layerIdListTarget, при цьому вихідний субпотік бітів складається з одиниць NAL в потоці бітів, які належать цільовому набору. Вхідними даними в процес витягання субпотоків бітів є змінна tIdTarget і список targetDecLayerIdSet. Результатом процесу витягання субпотоків бітів є субпотік бітів. Субпотік бітів виводиться шляхом видалення з потоку бітів всіх одиниць NAL з TemporalId більше tIdTarget або з nuh_reserved_zero_6bits, що не входить в значення в targetDecLayerIdSet.

[0063] Будь-який субпотік бітів, який включається в результат процесу витягання субпотоків бітів, заданого в підпункті 10.1 в WD8 HEVC, з tIdTarget, рівним будь-якому значенню в діапазоні від 0 до 6 включно, і з targetDecLayerIdSet, що містить значення 0, відповідає WD8 HEVC. Потік бітів, що відповідає WD8 HEVC, може містити одну або більше одиниць NAL кодованого слайса з рівним 0 nuh_reserved_zero_6bits і рівним 0 TemporalId.

[0064] Поточний проект HEVC має наступні недоліки. По-перше, часовим підмножинам для кодованої відеопослідовності дозволено сигналізувати набір з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень, який відрізняється від такого у всієї кодованої відеопослідовності. Однак часовій підмножині немає сенсу використовувати відмінний простір профілю, і відсутні очевидні варіанти використання, в яких часовим підмножинам було б вигідно мати інші значення (наприклад, клас, сумісні профілі і зв'язані з профілем обмеження), відмінні від всієї кодованої відеопослідовності.

[0065] По-друге, в цей час набір значень nuh_reserved_zero_6bits (тобто ID рівнів) для кожної робочої точки, за винятком першої точки, сигналізується шляхом явної сигналізації кожного з ID рівнів, включених в кодовану відеопослідовність, яка буде декодована. Однак в більшості сценаріїв кодування з масштабованістю взаємозв'язок залежності рівнів є лінійним, і було б достатньо сигналізації тільки цільового ID рівня. Крім того, для нелінійного взаємозв'язку залежності рівнів, який типовий для сценаріїв багатовидового кодування, в яких сигналізується декілька ID рівнів, ID рівнів можуть кодуватися по-різному. Кодування ID рівнів, що розрізняються, є простим та ефективним, схожим, наприклад, на кодування значень лічильника послідовності зображень (ПОС) при сигналізації наборів короткострокових еталонних зображень. Нарешті, дозволено мати дублюючі набори сигналізованих ID рівнів.

[0066] По-третє, поточний проект HEVC також має недолік в тому, що дозволено мати дублюючі синтаксичні структури hrd_parameters() з однаковим вмістом.

[0067] Щоб усунути вищезазначені недоліки, в дане розкриття винаходу включаються наступні методики. По-перше, синтаксичні елементи для сигналізації простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для часових підрівнів видаляються із синтаксичної структури profile_tier_level(). Таким чином, економляться біти, що зайво використовуються в тих синтаксичних елементах для часових підрівнів.

[0068] По-друге, задається простий режим робочої точки, для якого треба сигналізувати тільки одне значення nuh_reserved_zero_6bits (тобто ID рівня). Сигналізація робочих точок змінюється, щоб зробити можливою сигналізацію тільки одного значення nuh_reserved_zero_6bits для робочих точок у простому режимі робочої точки. Коли сигналізується декілька значень nuh_reserved_zero_6bits (тобто ID рівнів), вони кодуються по-різному. Цей спосіб сигналізації робочих точок більш ефективний для будь-яких сценаріїв кодування з масштабованістю, багатовидового і/або 3DV-кодування (3-мірне відео) з лінійною залежністю рівнів. Крім того, відповідно до методик з даного розкриття винаходу в одному VPS

заборонені дублюючі набори ID рівнів, сигналізованих для різних робочих точок в VPS. Таким чином, заданий набір ідентифікаторів рівнів є унікальним для заданої робочої точки в VPS.

[0069] По-третє, відповідно до методик з даного розкриття винаходу в VPS заборонені дублюючі набори параметрів HRD, кодованих у синтаксичних структурах `hrd_parameters()`, асоційованих з робочими точками. Таким чином, заданий набір параметрів HRD в інформації про параметри HRD, асоційованій із заданою робочою точкою, є унікальним для заданої робочої точки в VPS.

[0070] Наприклад, відеокодер 20 кодує синтаксичні елементи в VPS в рамках належного потоку бітів. Синтаксичні елементи вказують першу робочу точку, яка має перший набір вмісту. Відеокодер 20 потім кодує синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки. Належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки. Таким чином, відеокодер 20 кодує кожен робочу точку з унікальним вмістом, наприклад різними наборами ID рівнів. Як додатковий приклад відеокодер 20 може кодувати параметри HRD для потоку бітів, причому кожна копія синтаксичної структури параметрів HRD включає в себе різний вміст.

[0071] Аналогічним чином відповідно до методик, описаних в даному розкритті винаходу, відеодекодер 30 декодує синтаксичні елементи в VPS в рамках належного потоку бітів. Синтаксичні елементи вказують першу робочу точку, яка має перший набір вмісту. У деяких прикладах відеодекодер 30 приймає від відеокодера 20 кодований належний потік бітів. Відеодекодер 30 потім декодує, за наявності, синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки. Належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки. Таким чином, відеодекодер 30 декодує кожен робочу точку з унікальним вмістом, наприклад різними наборами ID рівнів. Як додатковий приклад відеодекодер 30 може декодувати параметри HRD для потоку бітів, причому кожна копія синтаксичної структури параметрів HRD включає в себе різний вміст. У додаткових прикладах другий набір вмісту містить набір параметрів HRD для першої робочої точки, який задає HRD, що використовується для перевірки, чи є належним субпотік бітів першої робочої точки.

[0072] Синтаксичні елементи, які вказують першу робочу точку, можуть включати синтаксичну структуру `operation_point()` в VPS в рамках належного потоку бітів. Крім того, синтаксичні елементи, які вказують інформацію про параметри HRD, можуть містити синтаксичну структуру `hrd_parameters()` в VPS в рамках належного потоку бітів.

[0073] У вищенаведених прикладах належний потік бітів може не включати в себе синтаксичні елементи в VPS, які дублюють перший набір вмісту для другої робочої точки, причому перший набір вмісту унікальний для першої робочої точки в VPS. Крім того, належний потік бітів може не включати в себе синтаксичні елементи в VPS, які дублюють інформацію про параметри HRD, яка має другий набір вмісту для другої робочої точки, причому другий набір вмісту унікальний для першої робочої точки в VPS.

[0074] У деяких прикладах відеокодер 20 може додатково вказувати, чи використовується простий режим робочої точки для кожної з множини робочих точок в потоці бітів. Коли простий режим робочої точки використовується для конкретної точки з робочих точок, відеокодер 20 може кодувати тільки один цільовий ідентифікатор рівня для конкретної робочої точки. Як інший приклад відеокодер 20 може кодувати простір профілю, клас, сумісні профілі і/або зв'язані з профілем обмеження для потоку бітів, але не кодувати окремі простір профілю, клас, сумісні профілі або зв'язані з профілем обмеження для часових підрівнів потоку бітів.

[0075] Також відеодекодер 30 може приймати вказівку, чи використовується простий режим робочої точки для кожної з множини робочих точок в потоці бітів. Коли простий режим робочої точки використовується для конкретної точки з робочих точок, відеодекодер 30 може декодувати тільки один цільовий ідентифікатор рівня для конкретної робочої точки. Як інший приклад відеодекодер 30 може декодувати простір профілю, клас, сумісні профілі і/або зв'язані з профілем обмеження для потоку бітів, але не декодувати окремі простір профілю, клас, сумісні профілі або зв'язані з профілем обмеження для часових підрівнів потоку бітів.

[0076] Фіг. 2 - блок-схема, що ілюструє приклад відеокодера 20, який може реалізувати методики, описані в даному розкритті винаходу. Відеокодер 20 може виконувати внутрішнє і зовнішнє кодування відеоблоків у 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 (І-режим) може належати до будь-якого з декількох режимів просторового кодування. Зовнішні режими, наприклад однонаправлене прогнозування (Р-режим) або двонаправлене прогнозування (В-режим), можуть належати до будь-якого з декількох режимів часового кодування.

[0077] Як показано на фіг. 2, відеокодер 20 приймає поточний відеоблок у відеокадрі, який підлягає кодуванню. У прикладі з фіг. 2 відеокодер 20 включає в себе модуль 40 вибору режиму, запам'ятовуючий пристрій 64 еталонних кадрів, суматор 50, модуль 52 обробки з перетворенням, 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 обробки з внутрішнім прогнозуванням і модуль 48 розбиття. Для відновлення відеоблока відеокодер 20 також включає в себе модуль 58 зворотного квантування, модуль 60 обробки із зворотним перетворенням і суматор 62. Також може включатися фільтр зменшення блочності для фільтрації меж блока, щоб видалити артефакти блочності з відновленого відео. За бажанням фільтр зменшення блочності звичайно фільтрував би вивід суматора 62. Також можуть використовуватися додаткові фільтри (в циклі або після циклу) додатково до фільтра зменшення блочності. Такі фільтри скорочено не показані, але за бажанням можуть фільтрувати вивід суматора 50 (як фільтр в циклі).

[0078] Під час процесу кодування відеокодер 20 приймає відеокадр або слайс, які підлягають кодуванню. 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 еталонних кадрах, щоб забезпечити часове прогнозування. Модуль 46 обробки з внутрішнім прогнозуванням як альтернатива може виконувати кодування з внутрішнім прогнозуванням прийнятого відеоблока відносно одного або більше сусідніх блоків у тому самому кадрі або слайсі як блок, який підлягає кодуванню, щоб забезпечити просторове прогнозування. Відеокодер 20 може виконувати декілька проходів кодування, наприклад, для вибору придатного режиму кодування для кожного блока відеоданих.

[0079] 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 квадродрева, яка вказує розбиття LCU на суб-CU. CU листа в квадродереві можуть включати в себе одну або більше PU і одну або більше TU.

[0080] Модуль 40 вибору режиму може вибрати один з режимів кодування, внутрішній або зовнішній, наприклад, на основі помилкових результатів і надає результируючий блок з внутрішнім або зовнішнім кодуванням суматора 50, щоб сформувати дані залишкового блока, і суматора 62, щоб відновити кодований блок для використання як еталонного кадру. Модуль 40 вибору режиму також надає синтаксичні елементи, наприклад вектори руху, індикатори внутрішнього режиму, інформацію про розбиття та іншу таку синтаксичну інформацію, модулю 56 ентропійного кодування.

[0081] Модуль 42 оцінки руху і модуль 44 компенсації руху можуть бути суттєво об'єднані, але ілюструються окремо з концептуальною метою. Оцінка руху, що виконується модулем 42 оцінки руху, є процесом формування векторів руху, які оцінюють рух для відеоблоків. Вектор руху може вказувати, наприклад, зміщення PU відеоблока в поточному відеокадрі або зображенні відносно блока з прогнозуванням в еталонному кадрі (або іншій кодованій одиниці) відносно поточного блока, що кодується в поточному кадрі (або іншій кодованій одиниці). Блок з прогнозуванням є блоком, який визнаний точно співпадаючим з блоком, який підлягає кодуванню, в показниках різниці пікселів, яка 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 еталонних зображеннях, збережених в запам'ятовуючому пристрої 64 еталонних кадрів. Наприклад, відеокодер 20 може інтерполювати значення положень однієї чверті пікселя, положень однієї восьмої пікселя або інших положень дробового пікселя в еталонному 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.

[0082] 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 еталонному зображенні. Еталонне зображення може вибиратися з першого списку еталонних зображень (Список 0) або другого списку еталонних зображень (Список 1), кожний з яких

ідентифікує одне або більше еталонних зображень, збережених в запам'ятовуючому пристрої 64 еталонних кадрів. Модуль 42 оцінки руху відправляє обчислений вектор руху модулю 56 ентропійного кодування і модулю 44 компенсації руху.

[0083] Компенсація руху, що виконується модулем 44 компенсації руху, може включати в себе вибірку або формування блока з прогнозуванням на основі вектора руху, визначеного модулем 42 оцінки руху. Знов у 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 еталонних 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 у слайсі відео.

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

[0085] Наприклад, модуль 46 обробки з внутрішнім прогнозуванням 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, некованим блоком, який кодувався для створення кодованого блока, а також швидкість передачі бітів (тобто кількість бітів), що використовується для створення кодованого блока. Модуль 46 обробки з внутрішнім прогнозуванням може обчислити відношення із спотворень і швидкостей для різних кодованих блоків, щоб визначити, який режим внутрішнього прогнозування демонструє найкраще значення спотворення залежно від швидкості передачі для блока.

[0086] Після вибору режиму внутрішнього прогнозування для блока модуль 46 обробки з внутрішнім прогнозуванням може надати модулю 56 ентропійного кодування інформацію, яка вказує вибраний режим внутрішнього прогнозування для блока. Модуль 56 ентропійного кодування може кодувати інформацію, яка вказує вибраний режим внутрішнього прогнозування. Відеокодер 20 може включити в переданий потік бітів конфігураційні дані, які можуть включати в себе множину таблиць індексів режимів внутрішнього прогнозування і множину таблиць індексів змінених режимів внутрішнього прогнозування (які також називаються таблицями відображення кодових слів), визначення контекстів кодування для різних блоків і вказівки найбільш імовірного режиму внутрішнього прогнозування, таблиці індексів режимів внутрішнього прогнозування і таблиці індексів зміненого режиму внутрішнього прогнозування для використання для кожного з контекстів.

[0087] 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 з перетворенням 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 з перетворенням може виконувати і інші перетворення, які концептуально схожі з DCT. Також могли б використовуватися вейвлет-перетворення, цілочисельні перетворення, перетворення поддіапазонів або інші типи перетворень. У будь-якому випадку модуль 52 обробки з перетворенням застосовує перетворення до залишкового блока, створюючи блок залишкових коефіцієнтів перетворення. Перетворення може перетворювати залишкову інформацію з ділянки значень пікселів у ділянку перетворення, наприклад частотну ділянку. Модуль 52

обробки з перетворенням може відправити результуючі коефіцієнти перетворення модулю 54 квантування.

[0088] Модуль 54 квантування квантує коефіцієнти перетворення для додаткового зменшення швидкості передачі бітів. Процес квантування може зменшити бітову глибину, асоційовану з деякими або всіма коефіцієнтами. Ступінь 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. Як альтернатива сканування може виконати модуль 56 ентропійного кодування.

[0089] 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 потік бітів може передаватися іншому пристрою (наприклад, відеодекодеру 30) або архівуватися для подальшої передачі або витягання.

[0090] Модуль 58 зворотного квантування і модуль 60 обробки із зворотним перетворенням застосовують зворотне квантування і зворотне перетворення відповідно, щоб відновити залишковий блок в ділянці пікселів, наприклад, для подальшого використання як еталонного блока. Модуль 44 компенсації руху може обчислити еталонний блок шляхом додавання залишкового блока до блока з прогнозуванням в одному з кадрів у запам'ятовуючому пристрої 64 еталонних кадрів. 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 еталонних кадрів. Відновлений відеоблок може використовуватися модулем 42 оцінки руху і модулем 44 компенсації руху як еталонний блок, щоб зовні кодувати блок в наступному відеокадрі.

[0091] Відеокодер 20 з фіг. 2 являє собою приклад відеокодера, сконфігурованого з можливістю реалізації однієї або більше методик, описаних в цьому документі. Відеокодер 20 сигналізує робочі точки в наборі параметрів відео (VPS), асоційованому з потоком бітів. Робочі точки належать до субпотоків бітів, які можуть витягуватися з вихідного потоку бітів, який є масштабованим у часі і/або має декілька рівнів або видів. У HEVC робоча точка може ідентифікуватися набором значень `nuh_reserved_zero_6bits`, позначеним як `OpLayerIdSet`, і значенням `TemporalId`, позначеним як `OpTid`. Як приклад вихідний потік бітів може включати в себе три рівні або види з різними просторовими розрізненнями і два часових масштабованих рівня з різними частотами кадрів. У цьому прикладі вихідний потік бітів включає в себе шість робочих точок з кожним з трьох просторових розрізнень, доступним на будь-якій частоті кадрів.

[0092] Для кожної з робочих точок, які відеокодер 20 сигналізує в VPS, асоційованому з потоком бітів, структура синтаксису робочих точок задає набір ідентифікаторів (ID) рівнів, що використовується для ідентифікації одиниць рівня абстракції мережі (NAL) в потоці бітів, які належать субпотіку бітів заданої робочої точки. Таким чином, одиниці NAL, які складають субпотік бітів заданої робочої точки, можна витягнути з вихідного потоку бітів на основі ідентифікаторів рівнів у одиниць NAL.

[0093] У деяких випадках відеокодер 20 може додатково кодувати параметри гіпотетичного еталонного декодера (HRD), які належать до однієї або більше робочих точок. У цьому випадку відеокодер 20 сигналізує інформацію про параметри HRD в VPS. Для кожної з однієї або більше робочих точок, які мають параметри HRD, синтаксична структура параметрів HRD задає набір параметрів HRD, який задає HRD, що використовується для перевірки, чи є належним субпотік бітів у заданої робочої точки.

[0094] Традиційно стандарти кодування відео допускають, щоб потоки бітів включали в себе дублюючі набори ідентифікаторів рівнів, які повинні бути сигналізовані для робочих точок, в VPS. Це означає, що дві або більше робочі точки можуть мати однаковий набір ідентифікаторів рівнів і тому ідентифікують однакові одиниці NAL для складання субпотоків бітів у робочих точок. Таким чином, декілька робочих точок можуть посилатися на однаковий вміст потоку бітів. Аналогічним чином, стандарти кодування відео допускають, щоб потоки бітів включали в себе дублюючі набори параметрів HRD, які повинні бути сигналізовані для робочих точок, в VPS. Це означає, що два або більше наборів параметрів HRD можуть мати однаковий вміст і тому

задають два ідентичних HRD для перевірки, чи є належними субпотіки бітів у зв'язаних робочих точках. В обох випадках сигналізація надмірна, і наявне неефективне використання бітів.

[0095] Методики з даного розкриття винаходу підвищують ефективність сигналізації інформації, асоційованої з робочими точками, шляхом заборони сигналізації дублюючого вмісту або для робочих точок, або параметрів HRD, зв'язаних з робочими точками, в VPS в рамках належного потоку бітів. Відповідно до методик з даного розкриття винаходу відеокoder 20 може кодувати синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту. Відеокoder 20 також може кодувати, за наявності, синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки. Належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки, і відеокoder 20 сконфігурований з можливістю кодування тільки належних потоків бітів. Як описано вище, перший набір вмісту може містити набір ідентифікаторів рівнів для першої робочої точки, а другий набір вмісту може містити набір параметрів HRD для першої робочої точки.

[0096] В одному прикладі відеокoder 20 буде кодувати тільки належний потік бітів так, що після кодування синтаксичних елементів у VPS в рамках потоку бітів, які вказують першу робочу точку, яка має перший набір ідентифікаторів рівнів, відеокoder 20 не кодує синтаксичні елементи в VPS, які дублюють перший набір ідентифікаторів рівнів для другої робочої точки. Методики з даного розкриття винаходу заборонили дублюючі набори ідентифікаторів рівнів для робочих точок в одному і тому самому VPS в рамках належного потоку бітів. Перший набір ідентифікаторів рівнів повинен бути унікальним для першої робочої точки в VPS, інакше потік бітів буде неналежним. Тобто, наприклад, неналежний потік бітів може містити перший набір ідентифікаторів рівнів для першої робочої точки, а також другий набір ідентифікаторів рівнів для другої робочої точки, причому другий набір включає в себе такі самі ідентифікатори рівнів, як і перший набір. Тому відеокoderу 20 не треба кодувати синтаксичні елементи в VPS належного потоку бітів, які дублюють набори ідентифікаторів рівнів для першої робочої точки і другої робочої точки.

[0097] В іншому прикладі можуть бути присутніми параметри HRD для першої робочої точки. У цьому випадку після кодування синтаксичних елементів у VPS в рамках потоку бітів, які вказують інформацію про параметри HRD, яка має перший набір параметрів HRD для першої робочої точки, відеокoder 20 не кодує синтаксичні елементи в VPS, які дублюють перший набір параметрів HRD для другої робочої точки. Методики з даного розкриття винаходу заборонили дублюючі набори параметрів HRD для робочих точок в одному і тому самому VPS в рамках належного потоку бітів. Перший набір параметрів HRD повинен бути унікальним для першої робочої точки в VPS, інакше потік бітів буде неналежним. Тому відеокoderу 20 не треба кодувати синтаксичні елементи в VPS належного потоку бітів, які дублюють набори параметрів HRD для першої робочої точки і другої робочої точки.

[0098] В іншому прикладі відеокoder 20 може вказувати, чи використовується простий режим робочої точки для кожної з множини робочих точок в потоці бітів, і коли простий режим робочої точки використовується для конкретної точки з робочих точок, може кодувати тільки один цільовий ідентифікатор рівня для конкретної робочої точки. Простий режим робочої точки є режимом, в якому для кожної робочої точки OpLayerIdSet включає в себе і тільки включає в себе конкретне значення nuh_reserved_zero_6bits і всі інші значення nuh_reserved_zero_6bits менше того конкретного значення nuh_reserved_zero_6bits. У деяких прикладах, коли відеокoder 20 кодує декілька значень nuh_reserved_zero_6bits (тобто ID рівнів), які сигналізуються, відеокoder 20 кодує ті значення по-різному. Таким чином, сигналізація робочих точок більш ефективна для будь-яких сценаріїв кодування з масштабованістю, багатовидового і/або 3DV-кодування з лінійною залежністю рівнів. Як описано вище, відеокoder 20 не кодує дублюючий набір ID рівнів в одному VPS.

[0099] Як інший приклад відеокoder 20 може кодувати простір профілю, клас, сумісні профілі і/або зв'язані з профілем обмеження для потоку бітів, але не кодувати окремі простір профілю, клас, сумісні профілі або зв'язані з профілем обмеження для часових підрівнів потоку бітів. Як описано вище, відеокoder 20 може кодувати параметри HRD для потоку бітів, причому кожна копія синтаксичної структури параметрів HRD включає в себе різний вміст.

[0100] Відеокoder 20 може використовувати наступні синтаксис і семантику. Зразкове корисне навантаження послідовності вихідних байтів (RBSP) набору параметрів відео задається в Таблиці 1 таким чином.

Синтаксис і семантика RBSP набору параметрів відео

ТАБЛИЦЯ 1

video_parameter_set_rbsp() {	Дескриптор
video_parameter_set_id	u(4)
vps_temporal_id_nesting_flag	u(1)
vps_reserved_zero_2bits	u(2)
vps_reserved_zero_6bits	u(6)
vps_max_sub_layers_minus1	u(3)
profile_tier_level(1, vps_max_sub_layers_minus1)	
vps_reserved_zero_12bits	u(12)
for(i = 0; i <= vps_max_sub_layers_minus1; i++) {	
vps_max_dec_pic_buffering[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
vps_num_hrd_parameters	ue(v)
for(i = 0; i < vps_num_hrd_parameters; i++) {	
vps_simple_op_mode_flag[i]	u(1)
if(i > 0)	
operation_point(i)	
hrd_parameters(i == 0, vps_max_sub_layers_minus1)	
}	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

[0101] Рівний 1 прапор **vps_simple_op_mode_flag[i]** задає, що використовується простий режим робочої точки для i^{oi} синтаксичної структури **operation_point_layer_ids()**. Коли **vps_simple_op_mode_flag[i]** рівний 0, цей прапор задає, що простий режим робочої точки не використовується для i^{oi} синтаксичної структури **operation_point()**.

[0102] Будь-які два примірники синтаксичної структури **hrd_parameters(i, vps_max_sub_layers_minus1)** та **hrd_parameters(j, vps_max_sub_layers_minus1)**, де i не рівне j , не повинні мати однакового вмісту. Таким чином, відеокодер 20 кодує синтаксичну структуру, яка включає в себе тільки унікальні параметри HRD.

[0103] Таблиця 2 надає синтаксис і семантику зразкового профілю, класу і рівня.

Синтаксис і семантика профілю, класу та рівня

ТАБЛИЦЯ 2

profile_tier_level(ProfilePresentFlag, MaxNumSubLayersMinus1) {	Дескриптор
if(ProfilePresentFlag) {	
general_profile_space	u(2)
general_tier_flag	u(1)
general_profile_idc	u(5)
for(i = 0; i < 32; i++)	
general_profile_compatibility_flag[i]	u(1)
general_reserved_zero_16bits [Ed. (GJS): Adjust semantics accordingly.]	u(16)
}	
general_level_idc	u(8)
for(i = 0; i < MaxNumSubLayersMinus1; i++) {	
if(ProfilePresentFlag)	
sub_layer_profile_present_flag[i]	u(1)
sub_layer_level_present_flag[i]	u(1)
if(sub_layer_profile_present_flag[i])	
sub_layer_profile_idc[i]	u(5)
if(sub_layer_level_present_flag[i])	
sub_layer_level_idc[i]	u(8)
}	
}	
}	

[0104] Рівний 1 прапор `sub_layer_profile_present_flag[i]`, коли `ProfilePresentFlag` рівний 1, задає, що інформація про профіль присутня в синтаксичній структурі `profile_tier_level()` для представлення підрівня з `TemporalId`, рівним `i`. Рівний 0 прапор `sub_layer_profile_present_flag[i]` задає, що інформація про профіль відсутня в синтаксичній структурі `profile_tier_level()` для представлення підрівня з `TemporalId`, рівним `i`. Коли відсутня, значення `sub_layer_profile_present_flag[i]` вважається рівним 0.

[0105] Рівний 1 прапор `sub_layer_level_present_flag[i]` задає, що інформація про рівень присутня в синтаксичній структурі `profile_tier_level()` для представлення підрівня з `TemporalId`, рівним `i`. Рівний 0 прапор `sub_layer_level_present_flag[i]` задає, що інформація про рівень відсутня в синтаксичній структурі `profile_tier_level()` для представлення підрівня з `TemporalId`, рівним `i`.

[0106] Синтаксичні елементи `sub_layer_profile_idc[i]` та `sub_layer_level_idc[i]` мають таку саму семантику, як і `general_profile_idc` і `general_level_idc` відповідно, але застосовуються до представлення підрівня з `TemporalId`, рівним `i`.

[0107] Відеокoder 20, наприклад, може кодувати `sub_layer_profile_present_flag[i]` рівним 1 для вказівки, що інформація про профіль присутня в синтаксичній структурі `profile_tier_level()` для представлення підрівня з `TemporalId`, рівним `i`. Відеокoder 20 може не сигналізувати синтаксичні елементи для сигналізації простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для часових підрівнів у синтаксичній структурі `profile_tier_level()`.

[0108] Таблиця 3 надає синтаксис і семантику зразкової робочої точки.

Синтаксис і семантика робочої точки

ТАБЛИЦЯ 3

operation_point(opIdx) {	Дескриптор
op_first_present_layer_id [opIdx]	u(6)
if(!vps_simple_op_mode_flag[opIdx]) {	
op_num_layer_id_values_minus1 [opIdx]	ue(v)
for(i = 1; i <= op_num_layer_id_values_minus1[opIdx]; i++)	
op_layer_id_delta_minus1 [opIdx][i]	ue(v)
}	
}	

[0109] Синтаксична структура operation_point(opIdx) задає набір значень nuh_reserved_zero_6bits, включених в OpLayerIdSet у робочих точок, до яких застосовується opIdx-а синтаксична структура hrd_parameters() в наборі параметрів відео.

[0110] Коли vps_simple_op_mode_flag[opIdx] рівний 0, синтаксичний елемент op_first_present_layer_id[opIdx] задає перше (тобто 0-е) значення nuh_reserved_zero_6bits, включеного в OpLayerIdSet у робочих точок, до яких застосовується opIdx-а синтаксична структура hrd_parameters() в наборі параметрів відео. Коли vps_simple_op_mode_flag[opIdx] рівний 1, op_first_present_layer_id [opIdx] задає найбільше значення nuh_reserved_zero_6bits, включеного в OpLayerIdSet у робочих точок, до яких застосовується opIdx-а синтаксична структура hrd_parameters() в наборі параметрів відео.

[0111] Коли vps_simple_op_mode_flag[opIdx] рівний 0, синтаксичний елемент op_num_layer_id_values_minus1[opIdx] плюс 1 задає кількість значень nuh_reserved_zero_6bits, включених в OpLayerIdSet у робочих точок, до яких застосовується opIdx-а синтаксична структура hrd_parameters() в наборі параметрів відео. Значення op_num_layer_id_values_minus1[opIdx] повинні бути менше або рівні 63.

[0112] Коли vps_simple_op_mode_flag[opIdx] рівний 0, op_layer_id_delta_minus1[opIdx][i] плюс 1 задає різницю між $i^{\text{им}}$ значенням nuh_reserved_zero_6bits і $(i - 1)^{\text{им}}$ значенням nuh_reserved_zero_6bits, включеними в OpLayerIdSet у робочих точок, до яких застосовується opIdx-а синтаксична структура hrd_parameters() в наборі параметрів відео. Значення op_layer_id_delta_minus1[opIdx][i] повинне знаходитися в діапазоні від 0 до 63 включно.

[0113] Змінна NumOpLayerIdsMinus1[opIdx] виводиться таким чином.

```
if( vps_simple_op_mode_flag[ opIdx ] )
```

```
  NumOpLayerIdsMinus1[ opIdx ] = op_first_present_layer_id[ opIdx ]
```

```
else
```

```
  NumOpLayerIdsMinus1[ opIdx ] = op_num_layer_id_values_minus1[ opIdx ]
```

```
NumOpLayerIdsMinus1[ 0 ] вважається рівною 0.
```

[0114] Змінні OpLayerId[opIdx][i] для i в діапазоні від 0 до NumOpLayerIdsMinus1[opIdx] включно виводяться таким чином.

```
OpLayerId[ opIdx ][ 0 ] = vps_simple_op_mode_flag[ opIdx ] ? 0 :
```

```
op_first_present_layer_id[ opIdx ]
```

```
for( i = 1; i <= NumOpLayerIdsMinus1[ opIdx ]; i++ )
```

```
  if( vps_simple_op_mode_flag[ opIdx ] )
```

```
    OpLayerId[ opIdx ][ i ] = i
```

```
  else
```

```
    OpLayerId[ opIdx ][ i ] =
```

```
    OpLayerId[ opIdx ][ i - 1 ] + op_layer_id_delta_minus1[ opIdx ][ i ] + 1
```

Значення OpLayerId[0][0] вважається рівним 0.

[0115] Ніяке значення OpLayerId[opIdx][i] не повинне бути рівне OpLayerId[opIdx][j], коли i не рівне j, і i і j знаходяться в діапазоні від 0 до NumOpLayerIdsMinus1[opIdx] включно.

[0116] Ніякі два набори OpLayerId[opIdx1] та OpLayerId[opIdx2], де opIdx1 не рівне opIdx2, не повинні включати в себе однакові набори значень nuh_reserved_zero_6bits.

[0117] OpLayerIdSet у робочих точок, до яких застосовується opIdx-а синтаксична структура hrd_parameters() в наборі параметрів відео, встановлюється таким, що включає в себе і тільки включає в себе значення nuh_reserved_zero_6bits, рівні OpLayerId[opIdx][i], для i в діапазоні від 0 до NumOpLayerIdsMinus1[opIdx] включно.

[0118] Таким чином, відеокoder 20 може кодувати синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту, використовуючи описані в цьому документі методики. Належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки, і відеокoder сконфігурований з можливістю декодування тільки належних потоків бітів. Наприклад, відеокoder 20 кодує належний потік бітів так, що він не включає в себе синтаксичні елементи в VPS, які дублюють перший набір вмісту для другої робочої точки, причому перший набір вмісту унікальний для першої робочої точки в VPS.

[0119] Таблиця 4 надає альтернативний синтаксис і семантику зразкової робочої точки.

Альтернативний синтаксис і семантика робочої точки

ТАБЛИЦЯ 4

operation_point(opIdx) {	Дескриптор
if(!vps_simple_op_mode_flag[opIdx])	
op_num_layer_id_values_minus1 [opIdx]	ue(v)
for(i = 0; i <= op_num_layer_id_values_minus1 [opIdx]; i++)	
op_layer_id [opIdx][i]	u(6)
}	

[0120] Синтаксична структура operation_point(opIdx) задає набір значень nuh_reserved_zero_6bits, включених в OpLayerIdSet у робочих точок, до яких застосовується opIdx-а синтаксична структура hrd_parameters() в наборі параметрів відео.

[0121] Синтаксичний елемент op_num_layer_id_values_minus1[opIdx] плюс 1 задає кількість значень nuh_reserved_zero_6bits, включених в OpLayerIdSet у робочих точок, до яких застосовується opIdx-а синтаксична структура hrd_parameters() в наборі параметрів відео. Значення op_num_layer_id_values_minus1[opIdx] повинне бути менше або рівне 63. Коли відсутня, значення op_num_layer_id_values_minus1[opIdx] вважається рівним 0.

[0122] У потоках бітів, що відповідають описаним в цьому документі методикам, op_num_layer_id_values_minus1[opIdx] повинен бути рівний 0. Хоча в деяких прикладах необхідно, щоб значення op_num_layer_id_values_minus1[opIdx] було рівне 0, відеокodери, наприклад відеокoder 30, повинні дозволяти з'являтися і іншим значенням в синтаксисі op_num_layer_id_values_minus1[opIdx].

[0123] Синтаксичний елемент op_layer_id[opIdx][i] задає i^е значення nuh_reserved_zero_6bits, включеного в OpLayerIdSet у робочих точок, до яких застосовується opIdx^a синтаксична структура hrd_parameters() в наборі параметрів відео.

[0124] Змінна NumOpLayerIdsMinus1[opIdx] виводиться таким чином.

```
if( vps_simple_op_mode_flag[ opIdx ] )
```

```
    NumOpLayerIdsMinus1[ opIdx ] = op_layer_id[ opIdx ][ 0 ]
```

```
else
```

```
    NumOpLayerIdsMinus1[ opIdx ] = op_num_layer_id_values_minus1[ opIdx ]
```

NumOpLayerIdsMinus1[0] вважається рівною 0.

[0125] Змінні OpLayerId[opIdx][i] для i в діапазоні від 0 до NumOpLayerIdsMinus1[opIdx] включно виводяться таким чином.

```
for( i = 0; i <= NumOpLayerIdsMinus1[ opIdx ]; i++ )
```

```
    OpLayerId[ opIdx ][ i ] = vps_simple_op_mode_flag[ opIdx ] ? i :
```

```
    op_layer_id[ opIdx ][ i ]
```

Значення OpLayerId[0][0] вважається рівним 0.

[0126] Ніяке значення OpLayerId[opIdx][i] не повинне бути рівне OpLayerId[opIdx][j], коли i не рівне j, і i і j знаходяться в діапазоні від 0 до NumOpLayerIdsMinus1[opIdx] включно.

[0127] Ніякі два набори OpLayerId[opIdx1] і OpLayerId[opIdx2], де opIdx1 не дорівнює opIdx2, не повинні включати в себе однакові набори значень nuh_reserved_zero_6bits.

[0128] OpLayerIdSet у робочих точок, до яких застосовується opIdx^a синтаксична структура hrd_parameters() в наборі параметрів відео, встановлюється таким, що включає в себе і тільки включає в себе значення nuh_reserved_zero_6bits, рівні OpLayerId[opIdx][i], для i в діапазоні від 0 до NumOpLayerIdsMinus1[opIdx] включно.

[0129] Таким чином, відповідно до описаних в цьому документі методик відеокoder 20 може вказувати, чи використовується простий режим робочої точки для кожної з множини робочих

точок в потоці бітів. Коли простий режим робочої точки використовується для конкретної точки з робочих точок, відеокoder 20 може кодувати тільки один цільовий ідентифікатор рівня для конкретної робочої точки. Як інший приклад відеокoder 20 може кодувати простір профілю, клас, сумісні профілі і/або зв'язані з профілем обмеження для потоку бітів, але не кодувати окремі простір профілю, клас, сумісні профілі або зв'язані з профілем обмеження для часових підрівнів потоку бітів. Як додатковий приклад відеокoder 20 може кодувати параметри HRD для потоку бітів, причому кожна копія синтаксичної структури параметрів HRD включає в себе різний вміст.

[0130] Фіг. 3 - блок-схема, що ілюструє приклад відеодекодера 30, який може реалізувати методики, описані в даному розкритті винаходу. У прикладі з фіг. 3 відеодекодер 30 включає в себе модуль 70 ентропійного декодування, модуль 72 компенсації руху, модуль 74 обробки з внутрішнім прогнозуванням, модуль 76 зворотного квантування, модуль 78 обробки із зворотним перетворенням, запам'ятовуючий пристрій 82 еталонних кадрів і суматор 80. Відеодекодер 30 в деяких прикладах може виконувати прохід декодування, в цілому зворотний по відношенню до проходу кодування, описаного по відношенню до відеокoderу 20, який показаний на фіг. 2.

[0131] Під час процесу декодування відеодекодер 30 приймає від відеокoderа 20 потік бітів кодованого відео, який являє собою відеоблоки кодованого слайса відео і асоційовані синтаксичні елементи. Відеодекодер 30 може приймати потік бітів кодованого відео від мережного об'єкта 29. Мережний об'єкт 29 може бути, наприклад, сервером, інформованим про середовище мережним елементом (MANE), пультом відеомонтажу або іншим таким пристроєм, сконфігурованим з можливістю реалізації однієї або більше описаних вище методик. Мережний об'єкт 29 може включати в себе зовнішній засіб, сконфігурований з можливістю виконання методик з даного розкриття винаходу. Як описано вище, деякі з методик, описаних в даному розкритті винаходу, можна реалізувати за допомогою мережного об'єкта 29 перед тим, як мережний об'єкт 29 передає відеодекодеру 30 потік бітів кодованого відео. У деяких системах декодування відео мережний об'єкт 29 і відеодекодер 30 можуть бути частинами окремих пристроїв, тоді як в інших випадках функціональні можливості, описані по відношенню до мережного об'єкта 29, можуть виконуватися тим самим пристроєм, що містить відеодекодер 30.

[0132] Модуль 70 ентропійного декодування у відеодекодері 30 ентропійно декодує потік бітів, щоб сформувати квантовані коефіцієнти, вектори руху або індикатори режиму внутрішнього прогнозування, і інші синтаксичні елементи. Модуль 70 ентропійного декодування перенаправляє вектори руху та інші синтаксичні елементи в модуль 72 компенсації руху. Відеодекодер 30 може приймати синтаксичні елементи на рівні слайса відео і/або рівні відеоблока.

[0133] Коли слайс відео кодується у вигляді слайса з внутрішнім кодуванням (I), модуль 74 обробки з внутрішнім прогнозуванням може сформувати дані прогнозування для відеоблока поточного слайса відео на основі сигналізованого режиму внутрішнього прогнозування і даних з раніше декодованих блоків поточного кадру або зображення. Коли відеокадр кодується у вигляді слайса із зовнішнім кодуванням (тобто В, Р або GPB), модуль 72 компенсації руху створює блоки з прогнозуванням для відеоблока поточного слайса відео на основі векторів руху та інших синтаксичних елементів, прийнятих від модуля 70 ентропійного декодування. Блоки з прогнозуванням можуть створюватися з одного з еталонних зображень в одному зі списків еталонних зображень. Відеодекодер 30 може побудувати списки еталонних кадрів, Список 0 і Список 1, використовуючи методики побудови за умовчанням на основі еталонних зображень, збережених в запам'ятовуючому пристрої 82 еталонних кадрів.

[0134] Модуль 72 компенсації руху визначає інформацію прогнозування для відеоблока поточного слайса відео шляхом синтаксичного аналізу векторів руху та 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), що використовується для кодування відеоблоків у слайсі відео, типу слайса зовнішнього прогнозування (наприклад, В-слайс або Р-слайс), інформації побудови для одного або більше списків еталонних зображень для слайса, вектори руху для кожного відеоблока із зовнішнім кодуванням в слайсі, стан зовнішнього прогнозування для кожного відеоблока із зовнішнім кодуванням в слайсі та іншої інформації для декодування відеоблоків у поточному слайсі відео.

[0135] Модуль 72 компенсації руху також може виконати інтерполяцію на основі інтерполюючих фільтрів. Модуль 72 компенсації руху може використовувати інтерполюючі фільтри, які використовувалися відеокoderом 20 під час кодування відеоблоків, щоб обчислити

інтерпольовані значення для субцілих пікселів еталонних блоків. У цьому випадку модуль 72 компенсації руху може визначити інтерполюючі фільтри, що використовуються відеокодером 20, з прийнятих синтаксичних елементів і використовувати інтерполюючі фільтри для створення блоків з прогнозуванням.

5 [0136] Модуль 76 зворотного квантування зворотно квантує, тобто деквантує, квантовані коефіцієнти перетворення, надані в потоці бітів і декодовані модулем 70 ентропійного декодування. Процес зворотного квантування може включати в себе використання параметра QP_Y квантування, обчисленого відеодекодером 30 для кожного відеоблока в слайсі відео, щоб визначити ступінь квантування, а також ступінь зворотного квантування, яке потрібно застосувати. Модуль 78 обробки із зворотним перетворенням застосовує до коефіцієнтів перетворення зворотне перетворення, наприклад зворотне DCT, зворотне цілочисельне перетворення або концептуально схожий процес зворотного перетворення, щоб створити залишкові блоки в ділянці пікселів.

10 [0137] Після того, як модуль 72 компенсації руху сформує блок з прогнозуванням для поточного відеоблока на основі векторів руху та інших синтаксичних елементів, відеодекодер 30 утворює декодований відеоблок шляхом підсумовування залишкових блоків від модуля 78 обробки із зворотним перетворенням з відповідними блоками з прогнозуванням, сформованими модулем 72 компенсації руху. Суматор 90 являє собою компонент або компоненти, які виконують цю операцію підсумовування. За бажанням також можна застосувати фільтр зменшення блочності для фільтрації декодованих блоків, щоб видалити артефакти блочності. 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 еталонних кадрів, який зберігає еталонні зображення, що використовуються для подальшої компенсації руху. Запам'ятовуючий пристрій 82 еталонних кадрів також зберігає декодоване відео для подальшого представлення на пристрої відображення, наприклад пристрої 32 відображення з фіг. 1.

15 [0138] Відеодекодер 30 з фіг. 3 являє собою приклад відеодекодера, сконфігурованого з можливістю реалізації однієї або більше методик, описаних в цьому документі. Методики з даного розкриття винаходу підвищують ефективність сигналізації інформації, асоційованої з робочими точками, шляхом заборони сигналізації дублюючого вмісту або для робочих точок, або параметрів HRD, зв'язаних з робочими точками, в VPS в рамках належного потоку бітів. Відповідно до методик з даного розкриття винаходу відеодекодер 30 може декодувати синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту. Крім того, відеодекодер 30 може декодувати, за наявності, синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки. Належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки, і відеодекодер 30 сконфігурований з можливістю декодування тільки належних потоків бітів. Як описано вище, перший набір вмісту може містити набір ідентифікаторів рівнів для першої робочої точки, а другий набір вмісту може містити набір параметрів HRD для першої робочої точки.

20 [0139] В одному прикладі після декодування синтаксичних елементів у VPS в рамках потоку бітів, які вказують першу робочу точку, яка має перший набір ідентифікаторів рівнів, відеодекодер 30 продовжить декодування, тільки якщо потік бітів є належним потоком бітів, який не включає в себе синтаксичні елементи в VPS, які дублюють перший набір ідентифікаторів рівнів для другої робочої точки. Методики з даного розкриття винаходу заборонили дублюючі набори ідентифікаторів рівнів для робочих точок в одному і тому самому VPS в рамках належного потоку бітів. Перший набір ідентифікаторів рівнів повинен бути унікальним для першої робочої точки в VPS, інакше потік бітів буде неналежним. Відеодекодер 30 декодує тільки синтаксичні елементи в VPS належного потоку бітів, які не включають в себе дублюючі набори ідентифікаторів рівнів для першої робочої точки і другої робочої точки. У випадку, коли відеодекодер 30 приймає неналежний потік бітів, відеодекодер 30 може перейти в стан відмови і відкинути весь потік бітів.

25 [0140] В іншому прикладі можуть бути присутніми параметри HRD для першої робочої точки. У цьому випадку після кодування синтаксичних елементів у VPS в рамках потоку бітів, які вказують інформацію про параметри HRD, яка має перший набір параметрів HRD для першої робочої точки, відеодекодер 30 продовжить декодування, тільки якщо потік бітів є належним потоком бітів, який не включає в себе синтаксичні елементи в VPS, які дублюють перший набір параметрів HRD для другої робочої точки. Методики з даного розкриття винаходу заборонили

дублюючі набори параметрів HRD для робочих точок в одному і тому самому VPS в рамках належного потоку бітів. Перший набір параметрів HRD повинен бути унікальним для першої робочої точки в VPS, інакше потік бітів буде неналежним. Відеодекодер 30 декодує тільки синтаксичні елементи в VPS належного потоку бітів, які не включають в себе дублюючі набори параметрів HRD для першої робочої точки і другої робочої точки. У випадку, коли відеодекодер 30 приймає неналежний потік бітів, відеодекодер 30 може перейти в стан відмови і відкинути весь потік бітів.

[0141] Як інший приклад відеодекодер 30 може приймати вказівку, чи використовується простий режим робочої точки для кожної з множини робочих точок в потоці бітів. Коли простий режим робочої точки використовується для конкретної точки з робочих точок, відеодекодер 30 може декодувати тільки один цільовий ідентифікатор рівня для конкретної робочої точки. Як інший приклад відеодекодер 30 може декодувати простір профілю, клас, сумісні профілі і/або зв'язані з профілем обмеження для потоку бітів, але не декодувати окремі простір профілю, клас, сумісні профілі або зв'язані з профілем обмеження для часових підрівнів потоку бітів. Як описано вище, відеодекодер 30 може декодувати параметри HRD для потоку бітів, причому кожна копія синтаксичної структури параметрів HRD включає в себе різний вміст.

[0142] Фіг. 4 - блок-схема, що ілюструє зразковий набір пристроїв, які утворюють частину мережі 100. У цьому прикладі мережа 100 включає в себе маршрутизатори 104A, 104B (маршрутизатори 104) і транскодуючий пристрій 106. Маршрутизатори 104 і транскодуючий пристрій 106 призначені для представлення невеликої кількості пристроїв, які можуть утворювати частину мережі 100. У мережу 100 також можуть включатися інші мережні пристрої, наприклад комутатори, концентратори, шлюзи, брандмауери, мости та інші такі пристрої. Крім того, на мережному тракті між серверним пристроєм 102 і клієнтським пристроєм 108 можуть надаватися додаткові мережні пристрої. У деяких прикладах серверний пристрій 102 може відповідати пристрою-джерелу 12 (фіг. 1), тоді як клієнтський пристрій 108 може відповідати пристрою-адресату 14 (фіг. 1).

[0143] Звичайно маршрутизатори 104 реалізують один або більше протоколів маршрутизації для обміну мережними даними по мережі 100. У деяких прикладах маршрутизатори 104 можуть бути сконфігуровані з можливістю виконання посередницьких операцій або операцій кешування. Тому в деяких прикладах маршрутизатори 104 можуть називатися пристроями-посередниками. Звичайно маршрутизатори 104 виконують протоколи маршрутизації для виявлення маршрутів в мережі 100. Шляхом виконання таких протоколів маршрутизації маршрутизатор 104B може виявити мережний маршрут від себе до серверного пристрою 102 через маршрутизатор 104A.

[0144] Методики з даного розкриття винаходу можуть бути реалізовані мережними пристроями, наприклад маршрутизаторами 104 і транскодуючим пристроєм 106, але можуть бути реалізовані також клієнтським пристроєм 108. Таким чином, маршрутизатори 104, транскодуючий пристрій 106 і клієнтський пристрій 108 являють собою приклади пристроїв, сконфігурованих з можливістю виконання методик з даного розкриття винаходу, включаючи методики, перераховані в розділі формули винаходу в даному розкритті винаходу. Крім того, пристрої з фіг. 1, показаний на фіг. 2 кодер і показаний на фіг. 3 декодер також є типовими пристроями, які можуть бути сконфігуровані з можливістю виконання методик з даного розкриття винаходу, включаючи методики, перераховані в розділі формули винаходу в даному розкритті винаходу.

[0145] Як описано в даному розкритті винаходу, «відеокодер» може використовуватися для посилання в цілому на пристрій кодування відео і пристрій декодування відео. Крім того, «кодування відео» може належати до кодування відео або декодування відео.

[0146] Фіг. 5 - блок-схема алгоритму, що ілюструє зразковий спосіб 200 декодування відеоданих відповідно до методик, описаних в даному розкритті винаходу. Спосіб 200 може виконуватися будь-яким пристроєм або системою, сконфігурованими з можливістю обробки відеоданих, наприклад відеодекодером 30 з фіг. 1 або 3.

[0147] Спосіб 200 включає в себе декодування синтаксичних елементів у наборі параметрів відео (VPS) в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту (202). Наприклад, відеодекодер 30 може декодувати синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту. У деяких прикладах синтаксичні елементи, які вказують першу робочу точку, мають в VPS синтаксичну структуру `operation_point()` в рамках належного потоку бітів.

[0148] Спосіб 200 додатково включає в себе декодування синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують інформацію про параметри гіпотетичного еталонного декодера (HRD), яка має другий набір вмісту для першої робочої точки (204).

Відповідно до методик з даного розкриття винаходу належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки. Наприклад, відеодекодер 30 декодує, за наявності, синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки. Відеодекодер 30 сконфігурований з можливістю декодування тільки належних потоків бітів, і належні потоки бітів не включають в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки. У деяких прикладах синтаксичні елементи, які вказують інформацію про параметри HRD, мають в VPS синтаксичну структуру `hrd_parameters()` в рамках належного потоку бітів. У деяких прикладах декодування синтаксичних елементів включає в себе декодування синтаксичних елементів, які вказують першу робочу точку та інформацію про параметри HRD тільки в рамках належних потоків бітів.

[0149] Спосіб 200 додатково включає в себе декодування відеоданих в належному потоці бітів на основі декодованих синтаксичних елементів (206). Наприклад, відеодекодер 30 декодує відеодані в належному потоці бітів щонайменше частково на основі синтаксичних елементів, які вказують першу робочу точку, яка має перший набір вмісту. Відеодекодер 30 додатково декодує відеодані в належному потоці бітів щонайменше частково на основі синтаксичних елементів, за наявності, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки.

[0150] У деяких прикладах перший набір вмісту містить набір ідентифікаторів рівнів для першої робочої точки, який ідентифікує одну або більше одиниць рівня абстракції мережі (NAL) належного потоку бітів, які належать субпотіку бітів першої робочої точки. У деяких прикладах другий набір вмісту містить набір параметрів HRD для першої робочої точки, який задає HRD, що використовується для перевірки, чи є належним субпотік бітів першої робочої точки. В інших прикладах належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють перший набір вмісту для другої робочої точки, причому перший набір вмісту унікальний для першої робочої точки в VPS. У додаткових прикладах належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють інформацію про параметри HRD, яка має другий набір вмісту для другої робочої точки, причому другий набір вмісту унікальний для першої робочої точки в VPS.

[0151] У деяких прикладах, коли набір ідентифікаторів рівнів для першої робочої точки включає в себе два або більше ідентифікаторів рівнів, спосіб 200 додатково містить декодування двох або більше ідентифікаторів рівнів по-різному. Тобто відеодекодер 30 по-різному декодує два або більше ідентифікаторів рівнів, коли набір ідентифікаторів рівнів для першої робочої точки включає в себе два або більше ідентифікаторів рівнів.

[0152] В іншому прикладі спосіб 200 може включати в себе декодування одного або більше синтаксичних елементів у рамках належного потоку бітів, які вказують, чи використовується простий режим робочої точки для першої і другої робочих точок. Коли простий режим робочої точки використовується для конкретної точки з першої і другої робочих точок, спосіб 200 може включати в себе декодування тільки цільового ідентифікатора рівня для конкретної робочої точки. В одному прикладі відеодекодер 30 декодує один або більше синтаксичних елементів у рамках належного потоку бітів, які вказують, чи використовується простий режим робочої точки для першої і другої робочих точок. Коли простий режим робочої точки використовується для конкретної точки з першої і другої робочих точок, відеодекодер 30 декодує тільки цільовий ідентифікатор рівня для конкретної робочої точки. У деяких прикладах конкретна робоча точка ідентифікується цільовим ідентифікатором рівня і часовим ідентифікатором, який вказує часовий підрівень належного потоку бітів.

[0153] Спосіб 200 може додатково включати в себе явне декодування кожного ідентифікатора рівня в наборі ідентифікаторів рівнів для першої робочої точки на основі вказівки, що простий режим робочої точки не використовується для першої робочої точки. Наприклад, відеодекодер 30 декодує ідентифікатор рівня в наборі ідентифікаторів рівнів для першої робочої точки на основі вказівки, що простий режим робочої точки не використовується для першої робочої точки. Спосіб 200 може додатково включати в себе декодування множини цільових ідентифікаторів рівнів, що розрізняються, для конкретної робочої точки, коли простий режим робочої точки не використовується для конкретної точки з робочих точок. Наприклад, відеодекодер 30 по-різному декодує множину цільових ідентифікаторів рівнів для конкретної робочої точки, коли простий режим робочої точки не використовується для конкретної точки з робочих точок (наприклад, коли `vsp_simple_or_mode_flag[i]` відсутній в VPS або рівний нулю).

[0154] У деяких прикладах спосіб 200 додатково включає в себе декодування синтаксичних елементів, які вказують одне або більше з простору профілю, класу, сумісних профілів і

зв'язаних з профілем обмежень для всієї декодованої відеопослідовності належного потоку бітів, причому належний потік бітів не включає в себе синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для одного або більше часових підрівнів декодованої відеопослідовності. Наприклад, відеодекодер

5

30 декодує синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для всієї декодованої відеопослідовності належного потоку бітів. Належний потік бітів не включає в себе синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для одного або більше часових підрівнів декодованої відеопослідовності.

[0155] Синтаксичні елементи можуть вказувати, що одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для часових підрівнів видаляються зі синтаксичної структури `profile_tier_level()` в VPS в рамках належного потоку бітів. У деяких прикладах спосіб 200 додатково включає в себе декодування одного або більше синтаксичних елементів, які вказують, чи присутня інформація про профіль для кожного з часових підрівнів належного потоку бітів. Коли інформація про профіль присутня для конкретного підрівня з часових підрівнів, спосіб 200 може додатково включати в себе декодування інформації про профіль для конкретного часового підрівня. Наприклад, відеодекодер 30 декодує один або більше синтаксичних елементів, які вказують, чи присутня інформація про профіль для кожного з часових підрівнів належного потоку бітів, і коли інформація про профіль присутня для конкретного підрівня з часових підрівнів, відеодекодер 30 декодує інформацію про профіль для конкретного часового підрівня.

10

15

20

[0156] Аналогічним чином спосіб 200 додатково включає в себе декодування одного або більше синтаксичних елементів, які вказують, чи присутня інформація про рівень для кожного з часових підрівнів потоку бітів. Коли інформація про рівень присутня для конкретного підрівня з часових підрівнів, спосіб 200 додатково включає в себе декодування інформації про рівень для конкретного часового підрівня. Наприклад, відеодекодер 30 декодує один або більше синтаксичних елементів, які вказують, чи присутня інформація про рівень для кожного з часових підрівнів потоку бітів, і коли інформація про рівень присутня для конкретного підрівня з часових підрівнів, відеодекодер 30 декодує інформацію про рівень для конкретного часового підрівня. У деяких прикладах часові підрівні належного потоку бітів ідентифікуються часовими ідентифікаторами.

25

30

[0157] Фіг. 6 - блок-схема алгоритму, що ілюструє зразковий спосіб 300 кодування відеоданих відповідно до методик, описаних в даному розкритті винаходу. Спосіб 300 може виконуватися будь-яким пристроєм або системою, сконфігурованими з можливістю обробки відеоданих, наприклад відеокодером 20 з фіг. 1 та 2.

35

[0158] Спосіб 300 включає в себе кодування синтаксичних елементів у наборі параметрів відео (VPS) в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту (302). Наприклад, відеокодер 20 може кодувати синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують першу робочу точку, яка має перший набір вмісту. У деяких прикладах синтаксичні елементи, які вказують першу робочу точку, мають в VPS синтаксичну структуру `operation_point()` в рамках належного потоку бітів.

40

[0159] Спосіб 300 додатково включає в себе кодування синтаксичних елементів у VPS в рамках належного потоку бітів, які вказують інформацію про параметри гіпотетичного еталонного декодера (HRD), яка має другий набір вмісту для першої робочої точки (304). Відповідно до методик з даного розкриття винаходу належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки. Наприклад, відеокодер 20 кодує, за наявності, синтаксичні елементи в VPS в рамках належного потоку бітів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки. Відеокодер 20 сконфігурований з можливістю кодування тільки належних потоків бітів, і належні потоки бітів не включають в себе синтаксичні елементи в VPS, які дублюють щонайменше один з першого або другого набору вмісту для другої робочої точки. У деяких прикладах синтаксичні елементи, які вказують інформацію про параметри HRD, мають в VPS синтаксичну структуру `hrd_parameters()` в рамках належного потоку бітів.

45

50

[0160] Спосіб 300 додатково включає в себе кодування відеоданих в належному потоці бітів на основі кодованих синтаксичних елементів (306). Наприклад, відеокодер 20 кодує відеодані в належному потоці бітів щонайменше частково на основі синтаксичних елементів, які вказують першу робочу точку, яка має перший набір вмісту. Відеокодер 20 додатково кодує відеодані в належному потоці бітів щонайменше частково на основі синтаксичних елементів, які вказують інформацію про параметри HRD, яка має другий набір вмісту для першої робочої точки.

55

60

[0161] У деяких прикладах перший набір вмісту містить набір ідентифікаторів рівнів для першої робочої точки, який ідентифікує одну або більше одиниць рівня абстракції мережі (NAL) належного потоку бітів, які належать субпотіку бітів першої робочої точки. У деяких прикладах другий набір вмісту містить набір параметрів HRD для першої робочої точки, який задає HRD, що використовується для перевірки, чи є належним субпотік бітів першої робочої точки. В інших прикладах належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють перший набір вмісту для другої робочої точки, причому перший набір вмісту унікальний для першої робочої точки в VPS. У додаткових прикладах належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють інформацію про параметри HRD, яка має другий набір вмісту для другої робочої точки, причому другий набір вмісту унікальний для першої робочої точки в VPS.

[0162] У деяких прикладах, коли набір ідентифікаторів рівнів для першої робочої точки включає в себе два або більше ідентифікаторів рівнів, спосіб 300 додатково містить кодування двох або більше ідентифікаторів рівнів по-різному. Тобто відеокoder 20 по-різному кодує два або більше ідентифікаторів рівнів, коли набір ідентифікаторів рівнів для першої робочої точки включає в себе два або більше ідентифікаторів рівнів.

[0163] В іншому прикладі спосіб 300 може включати в себе кодування одного або більше синтаксичних елементів у рамках належного потоку бітів, які вказують, чи використовується простий режим робочої точки для першої і другої робочих точок. Коли простий режим робочої точки використовується для конкретної точки з першої і другої робочих точок, спосіб 300 може включати в себе кодування тільки цільового ідентифікатора рівня для конкретної робочої точки. В одному прикладі відеокoder 20 кодує один або більше синтаксичних елементів у рамках належного потоку бітів, які вказують, чи використовується простий режим робочої точки для першої і другої робочих точок. Коли простий режим робочої точки використовується для конкретної точки з першої і другої робочих точок, відеокoder 20 кодує тільки цільовий ідентифікатор рівня для конкретної робочої точки. У деяких прикладах конкретна робоча точка ідентифікується цільовим ідентифікатором рівня і часовим ідентифікатором, який вказує часовий підрівень належного потоку бітів.

[0164] Спосіб 300 може додатково включати в себе явне кодування кожного ідентифікатора рівня в наборі ідентифікаторів рівнів для першої робочої точки на основі вказівки, що простий режим робочої точки не використовується для першої робочої точки. Наприклад, відеокoder 20 кодує ідентифікатор рівня в наборі ідентифікаторів рівнів для першої робочої точки на основі вказівки, що простий режим робочої точки не використовується для першої робочої точки. Спосіб 300 може додатково включати в себе кодування множини цільових ідентифікаторів рівнів, що розрізняються, для конкретної робочої точки, коли простий режим робочої точки не використовується для конкретної точки з робочих точок. Наприклад, відеокoder 20 по-різному кодує множину цільових ідентифікаторів рівнів для конкретної робочої точки, коли простий режим робочої точки не використовується для конкретної точки з робочих точок (наприклад, коли `vsp_simple_or_mode_flag[i]` відсутній в VPS або рівний нулю).

[0165] У деяких прикладах спосіб 300 додатково включає в себе кодування синтаксичних елементів, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для всієї кодованої відеопослідовності належного потоку бітів, причому належний потік бітів не включає в себе синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для одного або більше часових підрівнів кодованої відеопослідовності. Наприклад, відеокoder 20 кодує синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для всієї кодованої відеопослідовності належного потоку бітів. Належний потік бітів не включає в себе синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для одного або більше часових підрівнів декодованої відеопослідовності.

[0166] Синтаксичні елементи можуть вказувати, що одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для часових підрівнів видаляються з синтаксичної структури `profile_tier_level()` в VPS в рамках належного потоку бітів. У деяких прикладах спосіб 300 додатково включає в себе кодування одного або більше синтаксичних елементів, які вказують, чи присутня інформація про профіль для кожного з часових підрівнів належного потоку бітів. Коли інформація про профіль присутня для конкретного підрівня з часових підрівнів, спосіб 300 може додатково включати в себе кодування інформації про профіль для конкретного часового підрівня. Наприклад, відеокoder 20 кодує один або більше синтаксичних елементів, які вказують, чи присутня інформація про профіль для кожного з часових підрівнів належного потоку бітів, і коли інформація про профіль присутня для

конкретного підрівня з часових підрівнів, відеокoder 20 кодує інформацію про профіль для конкретного часового підрівня.

[0167] Аналогічним чином спосіб 300 додатково включає в себе кодування одного або більше синтаксичних елементів, які вказують, чи присутня інформація про рівень для кожного з часових підрівнів потоку бітів. Коли інформація про рівень присутня для конкретного підрівня з часових підрівнів, спосіб 300 додатково включає в себе кодування інформації про рівень для конкретного часового підрівня. Наприклад, відеокoder 20 кодує один або більше синтаксичних елементів, які вказують, чи присутня інформація про рівень для кожного з часових підрівнів потоку бітів, і коли інформація про рівень присутня для конкретного підрівня з часових підрівнів, відеокoder 20 кодує інформацію про рівень для конкретного часового підрівня. У деяких прикладах часові підрівні належного потоку бітів ідентифікуються часовими ідентифікаторами.

[0168] Потрібно зазначити, 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 або повністю пропускати (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 більше процесорів, а не послідовно.

[0169] В одному або більше прикладах функції, що описуються, 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 команд або коду на комп'ютерно-зчитуваному носії і виконуватися апаратним модулем обробки. Комп'ютерно-зчитувані носії можуть включати в себе запам'ятовуючі комп'ютерно-зчитувані носії, які відповідають матеріальному носію, такому як запам'ятовуючі носії, 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 якого-небудь протоколу зв'язку. Таким чином, комп'ютерно-зчитувані носії в цілому можуть відповідати (1) матеріальним комп'ютерно-зчитуваним запам'ятовувачим носіям, які є довготривалими, або (2) засобу зв'язку, такому як сигнал або несуча. Запам'ятовуючі носії можуть бути будь-якими доступними носіями, до яких можна звертатися за допомогою одного або більше комп'ютерів або одного або більше процесорів для витягання команд, коду і/або структур даних для реалізації методик, описаних в даному розкритті винаходу. Комп'ютерний програмний продукт може включати в себе комп'ютерно-зчитуваний носій.

[0170] Як приклад, а не обмеження, такі комп'ютерно-зчитувані запам'ятовуючі носії можуть бути виконані у вигляді RAM, ROM, EEPROM, компакт-диска або іншого накопичувача на оптичних дисках, накопичувача на магнітних дисках або інших магнітних запам'ятовувачих пристроїв, флеш-пам'яті або будь-якого іншого носія, який може використовуватися для зберігання потрібного програмного коду у вигляді команд або структур даних і до якого можна звертатися за допомогою комп'ютера. Також будь-яке з'єднання коректно називати комп'ютерно-зчитуваним носієм. Наприклад, якщо команди передаються з веб-сайта, сервера або іншого віддаленого джерела з використанням коаксіального кабелю, оптоволоконного кабелю, витої пари, цифрової абонентської лінії (DSL) або бездротових технологій, наприклад ІК-зв'язку, радіочастотного зв'язку і СНЧ-зв'язку, то коаксіальний кабель, оптоволоконний кабель, вита пара, DSL або бездротові технології, наприклад ІК-зв'язок, радіочастотний зв'язок і СНЧ-зв'язок, включаються у визначення носія. Однак потрібно розуміти, що комп'ютерно-зчитуваний запам'ятовуючі носії і запам'ятовуючі носії не включають в себе з'єднання, несучі, сигнали або інші короткочасні носії, а замість цього орієнтовані на довготривалі, матеріальні запам'ятовуючі носії. Оптичні і магнітні 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, причому магнітні диски звичайно відтворюють дані магнітним способом, тоді як оптичні диски відтворюють дані оптично за допомогою лазерів. Поєднання вищепереліченого також потрібно включити в сферу комп'ютерно-зчитуваних носіїв.

[0171] Команди 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 належати до будь-якої вищезазначеної структури або до будь-якої іншої структури, придатної для реалізації описаних в цьому документі методик. Крім того, в деяких аспектах функціональні можливості, описані в цьому документі, можуть бути надані в спеціалізованих апаратних і/або програмних модулях, сконфігурованих з можливістю кодування і декодування, або вбудовані в

об'єднаний кодек. Також методики можна було б повністю реалізувати в одній або більше схемах або логічних елементах.

[0172] 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 з можливістю 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 за допомогою сукупності взаємодіючих апаратних модулів, що включають в себе один або більше процесорів, як описано вище, в поєднанні з придатним програмним забезпеченням і/або мікропрограмним забезпеченням.

[0173] Описані різні приклади. Ці та інші приклади входять в обсяг нижченаведеної формули винаходу.

- 15 Посилальні позиції
- 10 система кодування і декодування відео
- 12 пристрій-джерело
- 14 пристрій-адресат
- 16 лінія зв'язку
- 20 18 джерело відео
- 20 відеокодер
- 22 інтерфейс виведення
- 28 інтерфейс введення
- 29 мережний об'єкт
- 25 30 відеодекодер
- 32 пристрій відображення
- 34 запам'ятовуючий пристрій
- 40 модуль вибору режиму
- 42 модуль оцінки руху
- 30 44, 72 модуль компенсації руху
- 46, 74 модуль обробки з внутрішнім прогнозуванням
- 48 модуль розбиття
- 50, 62, 80, 90 суматор
- 52 модуль обробки з перетворенням
- 35 54 модуль квантування
- 56 модуль ентропійного кодування
- 58, 76 модуль зворотного квантування
- 60, 78 модуль обробки із зворотним перетворенням
- 64, 82 запам'ятовуючий пристрій еталонних кадрів
- 40 70 модуль ентропійного декодування
- 100 мережа
- 102 серверний пристрій
- 104 маршрутизатори
- 106 транскодуючий пристрій
- 45 108 клієнтський пристрій

ФОРМУЛА ВИНАХОДУ

1. Спосіб декодування відеоданих, що містить етапи, на яких:

- 50 для кожного прийнятого потоку бітів з множини прийнятих потоків бітів відеоданих визначають, що прийнятий потік бітів є неналежним потоком бітів, якщо прийнятий потік бітів включає в себе синтаксичні елементи в наборі параметрів відео (VPS), які дублюють вміст для інформації про параметри гіпотетичного еталонного декодера (HRD) для різних робочих точок, сигналізованих в VPS;
- 55 для кожного прийнятого потоку бітів з множини прийнятих потоків бітів відеоданих визначають, що прийнятий потік бітів є належним потоком бітів, якщо прийнятий потік бітів не включає в себе ніякі синтаксичні елементи в VPS, які дублюють вміст для інформації про параметри HRD для згаданих різних робочих точок, сигналізованих в VPS;
- 60 щонайменше для одного потоку бітів з множини прийнятих потоків бітів, визначеного як належний:

декодувати перші синтаксичні елементи в VPS в рамках потоку бітів, які вказують першу інформацію про параметри HRD; і

декодувати, за наявності, другі синтаксичні елементи в VPS в рамках потоку бітів, які вказують другу інформацію про параметри HRD, яка відрізняється від першої інформації про параметри HRD.

2. Спосіб за п. 1, в якому потік бітів, визначений як належний, не включає в себе синтаксичні елементи в VPS, які дублюють другу інформацію про параметри HRD.

3. Спосіб за п. 1, який додатково містить етап, на якому декодують один або більше синтаксичних елементів у рамках потоку бітів, визначеного як належного, причому один або більше декодованих синтаксичних елементів вказують, чи використовується простий режим робочої точки для однієї або більше з зазначених різних робочих точок.

4. Спосіб за п. 3, який додатково містить етап, на якому декодують тільки цільовий ідентифікатор рівня з набору ідентифікаторів рівнів для першої робочої точки із зазначених різних робочих точок на основі вказівки, що простий режим робочої точки використовується для першої робочої точки із зазначених різних робочих точок, причому перша робоча точка ідентифікується цільовим ідентифікатором рівня і часовим ідентифікатором, який вказує часовий підрівень потоку бітів, визначеного як належний.

5. Спосіб за п. 3, який додатково містить етап, на якому явно декодують кожний ідентифікатор рівня з набору ідентифікаторів рівнів для першої робочої точки із зазначених різних робочих точок на основі вказівки, що простий режим робочої точки не використовується для першої робочої точки із зазначених різних робочих точок.

6. Спосіб за п. 1, в якому перша інформація про параметри HRD задає інформацію про параметри HRD, що використовується для перевірки, чи є належним субпотік бітів першої робочої точки із зазначених різних робочих точок.

7. Спосіб за п. 1, в якому синтаксичні елементи, які вказують першу інформацію про параметри HRD, містять в VPS синтаксичну структуру `hrd_parameters()` (параметри_hrd()) в рамках потоку бітів, визначеного як належний.

8. Спосіб за п. 1, який додатково містить етап, на якому декодують синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для всієї кодованої відеопослідовності потоку бітів, визначеного як належний, причому потік бітів, визначений як належний, не включає в себе синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для одного або більше часових підрівнів кодованої відеопослідовності.

9. Спосіб за п. 8, в якому синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для часових підрівнів, не включаються в синтаксичну структуру `profile_tier_level()` (профіль_клас_рівень()) в VPS в рамках потоку бітів, визначеного як належний.

10. Спосіб кодування відеоданих, що містить етапи, на яких:

кодують потік бітів відеоданих так, що потік бітів є належним потоком бітів,

причому належні потоки бітів включають в себе синтаксичні елементи в наборі параметрів відео (VPS), які дублюють вміст для інформації про параметри гіпотетичного еталонного декодера (HRD) для різних робочих точок,

причому належний потік бітів не включає в себе ніякі синтаксичні елементи в VPS, які дублюють вміст для інформації про параметри HRD для згаданих різних робочих точок, і

причому кодування потоку бітів містить етапи, на яких:

кодують перші синтаксичні елементи в рамках належного потоку бітів, причому перші синтаксичні елементи вказують першу інформацію про параметри HRD; і

кодують другі синтаксичні елементи в VPS в рамках належного потоку бітів, причому другі синтаксичні елементи вказують другу інформацію про параметри HRD, яка відрізняється від першої інформації про параметри HRD.

11. Спосіб за п. 10, в якому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють другу інформацію про параметри HRD.

12. Спосіб за п. 10, який додатково містить етап, на якому:

кодують, у рамках належного потоку бітів, один або більше синтаксичних елементів, які вказують, чи використовується простий режим робочої точки для однієї або більше із зазначених різних робочих точок.

13. Спосіб за п. 12, який додатково містить етап, на якому кодують тільки цільовий ідентифікатор рівня набору ідентифікаторів рівнів для першої робочої точки із зазначених різних робочих точок на основі вказівки, що простий режим робочої точки використовується для

першої робочої точки, причому перша робоча точка ідентифікується цільовим ідентифікатором рівня і часовим ідентифікатором, який вказує часовий підрівень належного потоку бітів.

14. Спосіб за п. 13, який додатково містить етап, на якому явно кодують кожний ідентифікатор рівня набору ідентифікаторів рівнів для першої робочої точки із зазначених робочих точок на основі вказівки, що простий режим робочої точки не використовується для першої робочої точки.

15. Спосіб за п. 10, в якому перша інформація про параметри HRD задає інформацію про параметри HRD, використовувану для перевірки, чи є належним субпотік бітів першої робочої точки зі згаданих різних робочих точок.

16. Спосіб за п. 10, в якому синтаксичні елементи, які вказують першу інформацію про параметри HRD, містять в VPS синтаксичну структуру `hrd_parameters()` в рамках належного потоку бітів.

17. Спосіб за п. 10, який додатково містить етап, на якому кодують треті синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для всієї кодованої відеопослідовності належного потоку бітів, причому належний потік бітів не включає в себе синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для одного або більше часових підрівнів кодованої відеопослідовності.

18. Спосіб за п. 17, в якому треті синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для часових підрівнів, не включаються в синтаксичну структуру `profile_tier_level()` в VPS в рамках належного потоку бітів.

19. Пристрій кодування відео, який містить:

запам'ятовуючий пристрій, сконфігурований з можливістю зберігання відеоданих одного або більше потоків бітів; і

відеокодер, сконфігурований з можливістю:

кодування належного потоку бітів, причому визначення, чи є потік бітів належним, задається так, що:

потоки бітів, які включають в себе синтаксичні елементи в наборі параметрів відео (VPS), які дублюють інформацію про параметри гіпотетичного еталонного декодера (HRD) для різних робочих точок, визначаються як неналежні потоки бітів, і

потоки бітів, які не включають в себе ніякі синтаксичні елементи в VPS, які дублюють інформацію про параметри HRD для згаданих різних робочих точок, визначаються як належні потоки бітів, і

причому для кодування належного потоку бітів, відеокодер сконфігурований з можливістю:

кодування перших синтаксичних елементів в VPS в рамках належного потоку бітів, причому перші синтаксичні елементи вказують першу інформацію про параметри HRD; і

кодування, за наявності, других синтаксичних елементів у VPS в рамках належного потоку бітів, причому другі синтаксичні елементи вказують другу інформацію про параметри HRD, яка відрізняється від першої інформації про параметри HRD.

20. Пристрій кодування відео за п. 19, в якому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють другу інформацію про параметри HRD.

21. Пристрій кодування відео за п. 19, в якому відеокодер додатково сконфігурований з можливістю кодування одного або більше синтаксичних елементів у рамках належного потоку бітів, причому зазначений один або більше синтаксичних елементів вказує, чи використовується простий режим робочої точки для одної або більше із зазначених різних робочих точок.

22. Пристрій кодування відео за п. 21, в якому відеокодер додатково сконфігурований з можливістю кодування тільки цільового ідентифікатора рівня з набору ідентифікаторів рівнів для першої робочої точки із зазначених різних робочих точок на основі вказівки, що простий режим робочої точки використовується для першої робочої точки із зазначених різних робочих точок, причому перша робоча точка ідентифікується цільовим ідентифікатором рівня і часовим ідентифікатором, який вказує часовий підрівень належного потоку бітів.

23. Пристрій кодування відео за п. 21, в якому відеокодер додатково сконфігурований з можливістю явного кодування кожного ідентифікатора рівня з набору ідентифікаторів рівнів для першої робочої точки із зазначених різних робочих точок на основі вказівки, що простий режим робочої точки не використовується для першої робочої точки із зазначених різних робочих точок.

24. Пристрій кодування відео за п. 19, в якому перша інформація про параметри HRD задає інформацію про параметри HRD, використовувану для перевірки, чи є належним субпотік бітів першої робочої точки зі згаданих різних робочих точок.

25. Пристрій кодування відео за п. 19, в якому синтаксичні елементи, які вказують першу інформацію про параметри HRD, містять в VPS синтаксичну структуру `hrd_parameters()` в рамках належного потоку бітів.

5 26. Пристрій кодування відео за п. 19, в якому відеокoder додатково сконфігурований з можливістю кодування третіх синтаксичних елементів, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для всієї кодованої відеопослідовності належного потоку бітів, причому належний потік бітів не включає в себе синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для одного або більше часових підрівнів кодованої відеопослідовності.

10 27. Пристрій кодування відео за п. 26, в якому треті синтаксичні елементи, які вказують одне або більше з простору профілю, класу, сумісних профілів і зв'язаних з профілем обмежень для часових підрівнів, не включаються в синтаксичну структуру `profile_tier_level()` в VPS в рамках належного потоку бітів.

15 28. Пристрій кодування відео за п. 19, в якому відеокoder містить відеокoder, сконфігурований з можливістю:

декодування перших синтаксичних елементів в VPS, які вказують першу інформацію про параметри HRD, і

20 декодування, за наявності, других синтаксичних елементів в VPS, які вказують другу інформацію про параметри HRD, яка відрізняється від першої інформації про параметри HRD.

29. Пристрій кодування відео за п. 19, в якому відеокoder містить відеокoder, сконфігурований з можливістю:

кодування перших синтаксичних елементів у VPS, які вказують першу інформацію про параметри HRD, і

25 кодування, за наявності, других синтаксичних елементів у VPS, які вказують другу інформацію про параметри HRD, яка відрізняється від першої інформації про параметри HRD.

30. Постійний комп'ютерно-зчитуваний запам'ятовуючий носій, що містить збережені на ньому команди, які при виконанні приписують процесору пристрої для кодування відеоданих:

кодувати належний потік бітів, причому визначення, чи є потік бітів належним, задається так, що

30 потоки бітів, які включають в себе синтаксичні елементи в наборі параметрів відео (VPS), які дублюють інформацію про параметри гіпотетичного еталонного декодера (HRD) для різних робочих точок, визначаються як неналежні потоки бітів, і

потоки бітів, які не включають в себе ніякі синтаксичні елементи в VPS, які дублюють інформацію про параметри HRD для згаданих різних робочих точок, визначаються як належні

35 потоки бітів, і

причому команди, які наказують процесору кодувати належний потік бітів, включають в себе команди, які при виконанні наказують процесору:

кодувати перші синтаксичні елементи в VPS в рамках належного потоку бітів, причому перші синтаксичні елементи вказують першу інформацію про параметри HRD; і

40 кодувати, при наявності, другі синтаксичні елементи в VPS в рамках належного потоку бітів, причому другі синтаксичні елементи вказують другу інформацію про параметри HRD, яка відрізняється від першої інформації про параметри HRD.

31. Постійний комп'ютерно-зчитуваний запам'ятовуючий носій за п. 30, в якому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють другу інформацію про параметри HRD.

45 32. Пристрій кодування відео, який містить:

засіб для кодування належного потоку бітів, причому визначення, чи є потік бітів належним, задається так, що

потоки бітів, які включають в себе синтаксичні елементи в наборі параметрів відео (VPS), які

50 дублюють інформацію про параметри гіпотетичного еталонного декодера (HRD) для різних робочих точок, визначаються як неналежні потоки бітів, і

потоки бітів, які не включають в себе ніякі синтаксичні елементи в VPS, які дублюють інформацію про параметри HRD для згаданих різних робочих точок, визначаються як належні потоки бітів, і

55 причому засіб для кодування належного потоку бітів включає в себе:

засіб для кодування перших синтаксичних елементів в VPS в рамках належного потоку бітів, причому перші синтаксичні елементи вказують першу інформацію про параметри HRD; і

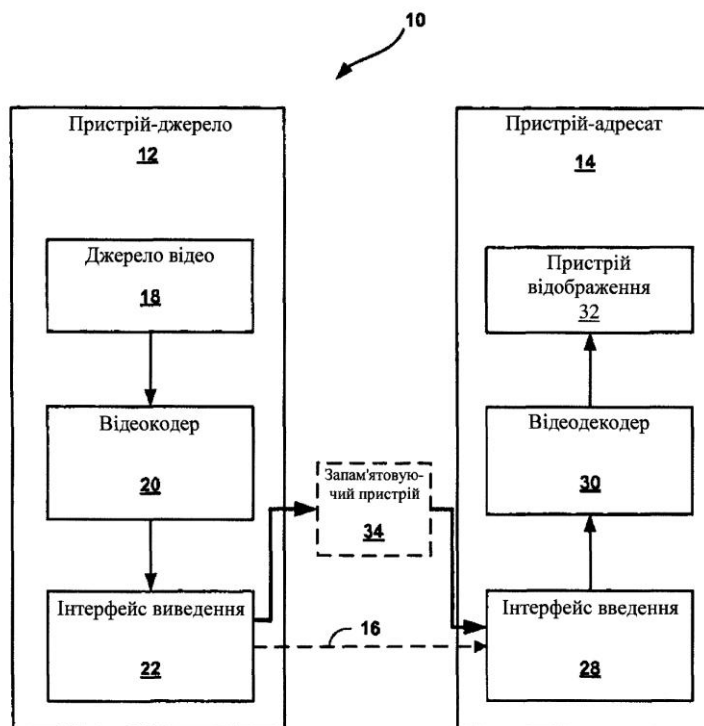
засіб для кодування, при наявності, других синтаксичних елементів в VPS в рамках належного потоку бітів, причому другі синтаксичні елементи вказують другу інформацію про параметри HRD, яка відрізняється від першої інформації про параметри HRD.

60

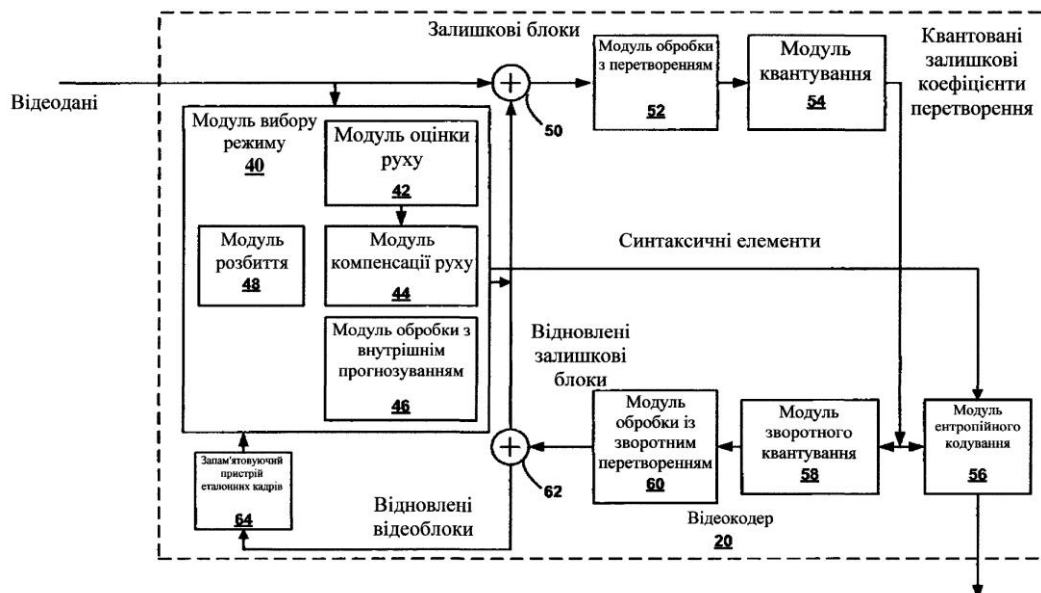
33. Пристрій кодування відео за п. 32, в якому належний потік бітів не включає в себе синтаксичні елементи в VPS, які дублюють другу інформацію про параметри HRD.

34. Пристрій кодування відео за п. 32, в якому перша інформація про параметри HRD задає HRD, що використовується для перевірки, чи є належним субпотік бітів першої робочої точки із зазначених різних робочих точок.

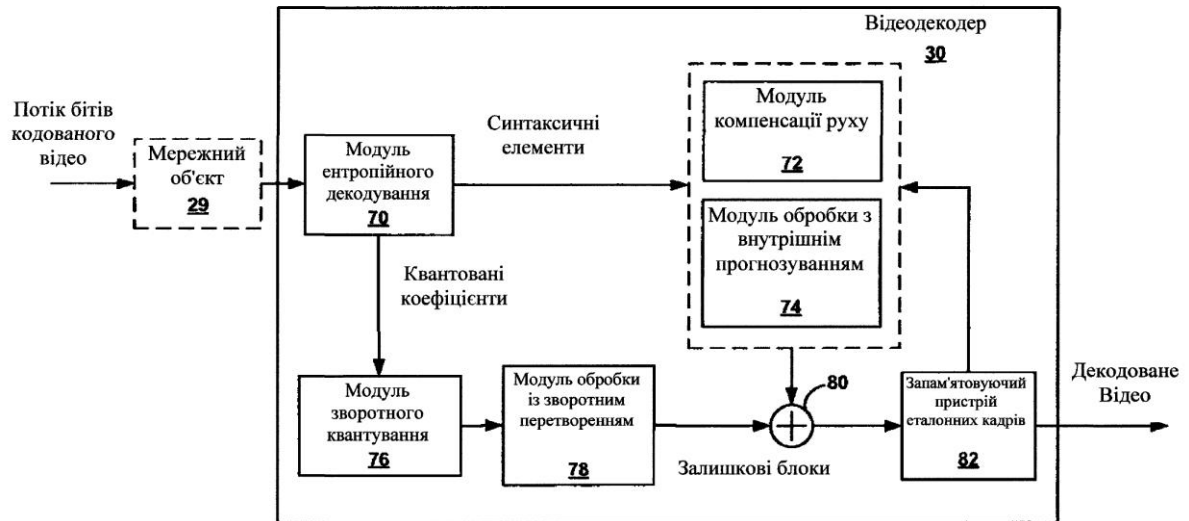
5



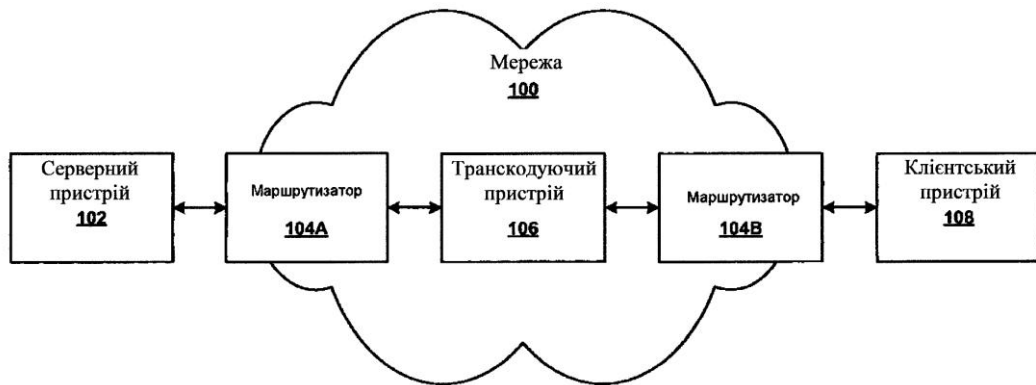
Фіг. 1



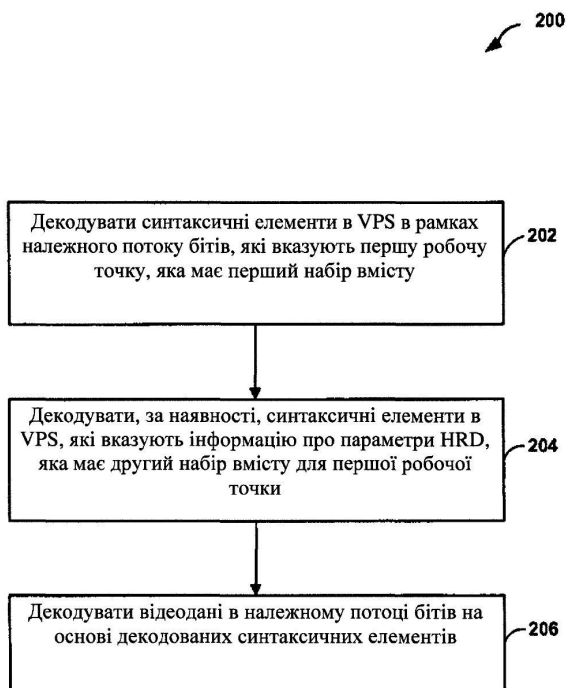
Фіг. 2



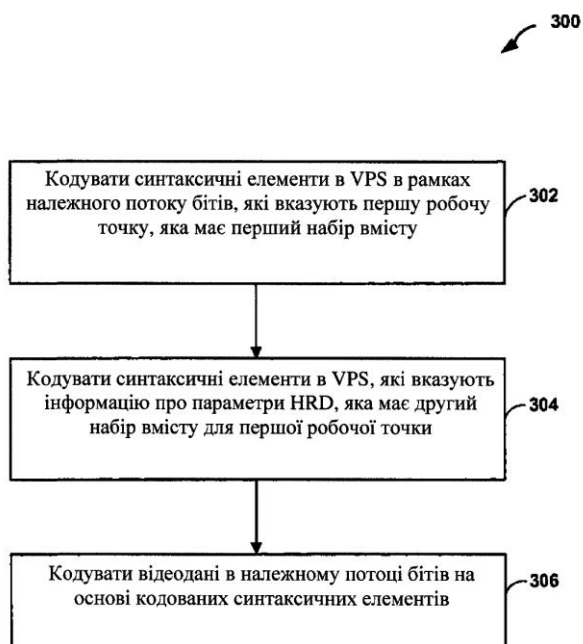
Фіг. 3



Фіг. 4



Фіг. 5



Фіг. 6

Комп'ютерна верстка А. Крулевський

Державна служба інтелектуальної власності України, вул. Василя Липківського, 45, м. Київ, МСП, 03680, Україна

ДП "Український інститут інтелектуальної власності", вул. Глазунова, 1, м. Київ – 42, 01601