



УКРАЇНА

(19) **UA** (11) **111492** (13) **C2**
(51) МПК**H04N 19/132** (2014.01)**H04N 19/129** (2014.01)**H04N 19/18** (2014.01)**H04N 19/70** (2014.01)ДЕРЖАВНА СЛУЖБА
ІНТЕЛЕКТУАЛЬНОЇ
ВЛАСНОСТІ
УКРАЇНИ**(12) ОПИС ДО ПАТЕНТУ НА ВІНАХІД**

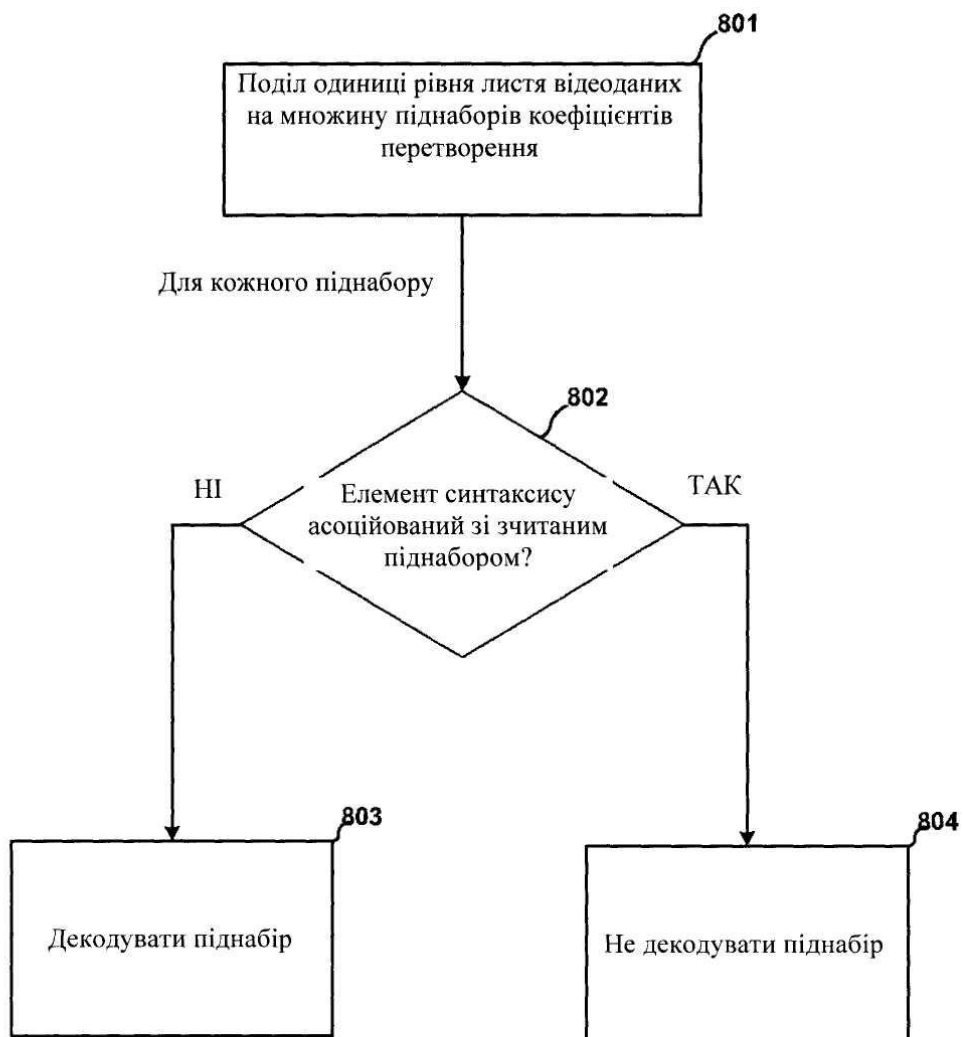
(21) Номер заявки: а 2014 00834	(72) Винахідник(и): Карчєвіч Марта (US), Ван Сянлінь (US), Го Лівей (US)
(22) Дата подання заявки: 29.06.2012	(73) Власник(и): КВЕЛКОММ ІНКОРПОРЕЙТЕД, Attn: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121-1714, United States of America (US)
(24) Дата, з якої є чинними права на винахід: 10.05.2016	(74) Представник: Мошинська Ніна Миколаївна, реєстр. №115
(31) Номер попередньої заявки відповідно до Паризької конвенції: 61/503,541, 61/552,341, 13/413,475	(56) Перелік документів, взятих до уваги експертизою: EP 1768415 A1, 28.03.2007 Louis Kerofsky et al: "A 4x4 CBP in H26L", 12. VCEG MEETING; 09.01.2001-12.01.2001; EIBSEE, DE; (VIDEO CODINGEXPERTS GROUP OF ITU-T SG.16), no. VCEG-L22, 04.01.2001, ISSN: 0000-0462 WIEGAND: "Editor's Proposed Revision of JVT-E146d37", 6. JVT MEETING; 63. MPEG MEETING; 09.12.2002- 12.12.2002; AWAJI, JP; (JOINT VIDEO TEAM OF ISO/IES JTC1/SC29/WG11 and ITU-T SG.16), no. JVT-F082r3, 13.12.2002, ISSN: 0000-0428, P. 60, 65 MATHIAS WIEN: "H.26L Core Experiment Description on Adaptive Block Transforms", 12. VCEG MEETING; 09.01.2001-12.01.2001; EIBSEE, DE; (VIDEO CODINGEXPERTS GROUP OF ITU-T SG.16), no. VCEG-L41, 14.02.2001, ISSN: 0000-0462, P. 4-5
(32) Дата подання попередньої заявки відповідно до Паризької конвенції: 30.06.2011, 27.10.2011, 06.03.2012	
(33) Код держави-учасниці Паризької конвенції, до якої подано попередню заявку: US, US, US	
(41) Публікація відомостей про заявку: 12.05.2014, Бюл.№ 9	
(46) Публікація відомостей про видачу патенту: 10.05.2016, Бюл.№ 9	
(86) Номер та дата подання міжнародної заявки, поданої відповідно до Договору РСТ PCT/US2012/044990, 29.06.2012	

(54) СИГНАЛІЗАЦІЯ ЕЛЕМЕНТІВ СИНТАКСИСУ ДЛЯ КОЕФІЦІЄНТІВ ПЕРЕТВОРЕННЯ ДЛЯ ПІДНАБОРІВ ОДИНИЦІ КОДУВАННЯ НА РІВНІ ЛИСТЯ**(57) Реферат:**

Дане розкриття описує способи для того, щоб кодувати коефіцієнти перетворення для блока відеоданих. Згідно з цими способами, відеокодер ділить одиницю рівня блока відеоданих на множину піднаборів коефіцієнтів перетворення. Відеокодер генерує, для піднабору з множини піднаборів коефіцієнтів перетворення, елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти. У деяких прикладах відеокодер може вибірково визначати, чи генерувати елемент синтаксису для кожного піднабору. Декодер може

UA 111492 C2

зчитувати ентропійно кодований бітовий потік, який включає в себе елемент синтаксису, і визначати, чи декодувати піднабір, на основі елемента синтаксису.



Фіг. 8

Дана заявка вимагає пріоритет попередньої заявки на патент США № 61/503,541, поданої 30 червня 2011 р., і попередньої заявки на патент США № 61/552,341, поданої 27 жовтня 2011 р., весь зміст кожної з яких включений тут по посиланню.

ГАЛУЗЬ ТЕХНІКИ

5 Дане розкриття стосується кодування і стиснення відео. Більш конкретно, дане розкриття направлене на способи для того, щоб сканувати квантовані коефіцієнти перетворення.

РІВЕНЬ ТЕХНІКИ

10 Цифрові можливості відео можуть бути включені в широкий діапазон пристроїв, включаючи цифрові телевізори, цифрові системи прямого мовлення, бездротові системи мовлення, персональні цифрові помічники (PDA), портативні або настільні комп'ютери, планшетні комп'ютери, зчитувачі електронних книг, цифрові камери, цифрові пристрої реєстрації, цифрові медіаплеєри, відеоігрові пристрої, пульти відеоігор, стільникові або супутникові радіотелефони, так звані "смартфони," пристрої організації відеотелеконференцій, пристрої потокової передачі відео і т. п. Цифрові відеопристрої реалізують способи стиснення відео, такі, як описані в 15 стандартах, визначених за допомогою MPEG-2, MPEG-4, ITU-T H.263 або ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), стандарту високоефективного кодування відео (HEVC), що розвивається на даний час, і розширеннях таких стандартів. Відеопристрої можуть передавати, приймати, кодувати, декодувати і/або зберігати цифрову відеоінформацію більш ефективно, реалізуючи такі способи стиснення відео.

20 Способи стиснення відео виконують просторове (всередині картинки) прогнозування і/або часове (між картинками) прогнозування, щоб зменшити або видалити надмірність, властиву відеопослідовностям. Для основаного на блоках кодування відео вирізка відео (тобто відеокадр або частина відеокадру) може бути розділена на відеоблоки, які можуть також згадуватися як 25 блоки дерева, одиниці кодування (CU) і/або вузли кодування. Відеоблоки у внутрішньокерованій (I) вирізці картини кодують, використовуючи просторове прогнозування відносно опорних вибірок в сусідніх блоках в тій же самій картинці. Відеоблоки у зовнішньокерованій (P або B) вирізці картини можуть використовувати просторове прогнозування відносно опорних вибірок в сусідніх блоках в тій же самій картинці або часове прогнозування відносно опорних вибірок в інших опорних картинках. Картинки можуть згадуватися як кадри, і опорні картини можуть згадуватися як опорні кадри.

Просторове або часове прогнозування приводить до прогнозуючого блока для блока, який повинен бути закодований. Залишкові дані представляють пікселні різниці між первинним блоком, який повинен бути закодований, і прогнозуючим блоком. Зовнішньокерований блок кодується згідно з вектором руху, який вказує на блок опорних вибірок, що формують 35 прогнозуючий блок, і залишковими даними, що вказують різницю між закодованим блоком і прогнозуючим блоком. Внутрішньокерований блок кодується згідно з режимом внутрішнього кодування і залишковими даними. Для подальшого стиснення залишкові дані можуть бути перетворені з піксельної області в область перетворення, приводячи до залишкових коефіцієнтів перетворення, які потім можуть квантуватися. Квантовані коефіцієнти перетворення, спочатку розміщені в двовимірному масиві, можуть бути скановані, щоб сформувати одновимірний вектор коефіцієнтів перетворення, і ентропійне кодування може бути застосоване, щоб досягнути ще більшої міри стиснення.

СУТЬ ВИНАХОДУ

45 При кодуванні відео, щоб стиснути об'єм даних, використовуваних для представлення відеоданих, відеокодер може виконувати ентропійне кодування відеоданих. Згідно зі способами, розкритими в даному описі, як частину ентропійного кодування, відеокодер ділить одиницю рівня листя (кінцевого рівня) відеоданих на множину піднаборів коефіцієнтів перетворення. Одиниця рівня листя, як описано тут, належить до нерозділеної одиниці структури даних відео, одним прикладом якої є кінцевий, нерозділений дочірній вузол структури даних квадродрева, 50 як описано більш детально нижче.

Для щонайменше одного з піднаборів кодер генерує елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти як частину ентропійно кодованого бітового потоку. Кодер визначає, чи сигналізувати елемент синтаксису для піднабору з множини піднаборів коефіцієнтів перетворення. Наприклад, кодер може визначити, 55 чи сигналізувати елемент синтаксису, на основі кількості потенційних ненульових коефіцієнтів в межах піднабору або на основі середньої кількості ненульових коефіцієнтів для піднабору, на основі статистики для однієї або більше раніше кодованих одиниць рівня листя відеоданих.

Декодер може зчитувати ентропійно кодований бітовий потік і визначати, чи декодувати коефіцієнти перетворення піднабору, на основі елемента синтаксису. У деяких прикладах 60 декодер може визначити, чи декодувати піднабір коефіцієнтів перетворення, на основі того, чи

включає ентропійно кодований бітовий потік в себе елемент синтаксису, асоційований з піднабором. Наприклад, якщо піднабір не включає в себе асоційований елемент синтаксису, декодер декодує піднабір. Однак, якщо піднабір дійсно включає в себе асоційований елемент синтаксису, декодер визначає, чи декодувати піднабір, на основі значення елемента синтаксису. Наприклад, якщо елемент синтаксису має перше значення, декодер декодує піднабір, але, якщо елемент синтаксису має друге, відмінне значення, декодер не декодує піднабір.

У деяких прикладах способи даного розкриття можуть підвищити ефективність кодування кодера або декодера. Наприклад, способи, описані тут, можуть зменшити кількість бітів, використовуваних кодером, щоб генерувати ентропійно кодований бітовий потік, який представляє відеодані.

У одному прикладі спосіб кодування блока відеоданих включає поділ одиниці рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення і генерування, для піднабору з множини піднаборів коефіцієнтів перетворення, елемента синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти.

У іншому прикладі пристрій може бути сконфігурований, щоб кодувати щонайменше одну одиницю рівня листя відеоданих. Пристрій може містити процесор, сконфігурований, щоб розділити одиницю рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення і генерувати, для піднабору з множини піднаборів коефіцієнтів перетворення, елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти.

У іншому прикладі пристрій може бути сконфігурований, щоб кодувати щонайменше одну одиницю рівня листя відеоданих, причому пристрій містить засіб для поділу одиниці рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення і засіб для генерування, для піднабору з множини піднаборів коефіцієнтів перетворення, елемента синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти.

У іншому прикладі спосіб декодування блока відеоданих включає поділ одиниці рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення, визначення, для піднабору з множини піднаборів коефіцієнтів перетворення, чи включає в себе згаданий піднабір асоційований елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти, визначення, на основі елемента синтаксису, чи декодувати піднабір.

У іншому прикладі пристрій може бути сконфігурований, щоб декодувати блок відеоданих, причому пристрій містить процесор, сконфігурований, щоб розділити одиницю рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення, визначати, для піднабору з множини піднаборів коефіцієнтів перетворення, чи включає в себе згаданий піднабір асоційований елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти, і визначати, на основі елемента синтаксису, чи декодувати піднабір.

У іншому прикладі пристрій може бути сконфігурований, щоб декодувати блок відеоданих, причому пристрій містить засіб для поділу одиниці рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення, засіб для визначення, для піднабору з множини піднаборів коефіцієнтів перетворення, чи включає в себе згаданий піднабір елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти, і засіб для визначення, на основі елемента синтаксису, чи декодувати піднабір.

Способи, описані в даному розкритті, можуть бути реалізовані в апаратному забезпеченні, програмному забезпеченні, програмно-апаратних засобах або будь-якій їх комбінації. Наприклад, різні способи можуть бути реалізовані або виконані одним або більше процесорами. Як використовується тут, процесор може належати до мікропроцесора, спеціалізованої інтегральної схеми (ASIC), програмованої користувачем вентиляційної матриці (FPGA), цифрового сигнального процесора (DSP) або іншої еквівалентної інтегральної або дискретної логічної схеми. Програмне забезпечення може бути виконане одним або більше процесорами. Програмне забезпечення, що містить інструкції для виконання способів, може бути спочатку збережене на зчитуваному комп'ютером носії і завантажено і виконане процесором.

Відповідно, дане розкриття також розглядає зчитувані комп'ютером носії даних, які містять інструкції, щоб примусити процесор (або інший обчислювальний пристрій) виконувати будь-які способи, описані в даному розкритті. У деяких випадках зчитуваний комп'ютером носій даних може бути частиною комп'ютерного продукту, що зберігає програму, який може бути проданий виготовлювачам і/або використовуватися в пристрої. Комп'ютерний програмний продукт може включати в себе зчитуваний комп'ютером носій, і в деяких випадках може також включати в себе пакувальні матеріали.

У одному прикладі дане розкриття описує зчитуваний комп'ютером носій даних, що зберігає інструкції, які, коли виконуються, примушують обчислювальний пристрій розділяти одиницю

рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення і генерувати, для піднабору з множини піднаборів коефіцієнтів перетворення, елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти.

У іншому прикладі дане розкриття описує зчитуваний комп'ютером носій даних, що зберігає інструкції, які, коли виконуються, примушують обчислювальний пристрій розділяти одиницю рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення, визначати, для піднабору з множини піднаборів коефіцієнтів перетворення, чи включає в себе згаданий піднабір асоційований елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти, і визначати, на основі елемента синтаксису, чи декодувати піднабір.

Деталі одного або більше прикладів сформульовані в прикладених кресленнях і описі нижче. Інші ознаки, об'єкти і переваги винаходу будуть очевидні з опису і креслень і з формули винаходу.

КОРОТКИЙ ОПИС КРЕСЛЕНЬ

ФІГ. 1 є блок-схемою, яка ілюструє один приклад системи кодування і декодування відео, сконфігурованої, щоб працювати згідно зі способами даного розкриття.

ФІГ. 2 є блок-схемою, яка ілюструє один приклад відеокодера, сконфігурованого, щоб працювати згідно зі способами даного розкриття.

ФІГ. 3 є блок-схемою, яка ілюструє один приклад відеодекодера, сконфігурованого, щоб працювати згідно зі способами даного розкриття.

ФІГ. 4 є концептуальною діаграмою, яка ілюструє один приклад одиниці рівня листя відеоданих, розділеної на множину піднаборів коефіцієнтів перетворення, сумісних з одним або більше аспектами даного розкриття.

ФІГ. 5 є блок-схемою, яка ілюструє один приклад способу кодування одиниці рівня листя відеоданих, сумісного з одним або більше аспектами даного розкриття.

ФІГ. 6 є блок-схемою, яка ілюструє інший приклад способу кодування одиниці рівня листя відеоданих, сумісного з одним або більше аспектами даного розкриття.

ФІГ. 7 є блок-схемою, яка ілюструє інший приклад способу кодування одиниці рівня листя відеоданих, сумісного з одним або більше аспектами даного розкриття.

ФІГ. 8 є блок-схемою, яка ілюструє один приклад способу декодування одиниці рівня листя відеоданих, сумісного з одним або більше аспектами даного розкриття.

ДОКЛАДНИЙ ОПИС

ФІГ. 1 є блок-схемою, що ілюструє зразкову систему 10 кодування і декодування відео, яка може використовувати способи, описані в даному розкритті. Як показано на фіг. 1, система 10 включає в себе пристрій-джерело 12, який генерує кодовані відеодані, які повинні бути декодовані в більш пізній час пристроєм 14 призначення. Пристрій-джерело 12 і пристрій 14 призначення можуть містити будь-який з широкого діапазону пристроїв, включаючи настільні комп'ютери, портативні комп'ютери (тобто ноутбуки), планшетні комп'ютери, телевізійні приставки, телефонні трубки, такі як так звані "смартфони", так звані "інтелектуальні" клавіатури, телевізори, камери, пристрої відображення, цифрові медіаплеєри, пульти відеоігор, пристрій потокової передачі відео або подібні. У деяких випадках пристрій-джерело 12 і пристрій 14 призначення можуть бути обладнані для бездротового зв'язку.

Пристрій 14 призначення може приймати кодовані відеодані, які повинні бути декодовані, по лінії 16 зв'язку. Лінія 16 зв'язку може містити будь-який тип носія або пристрою, здатного передавати кодовані відеодані від пристрою-джерела 12 на пристрій 14 призначення. У одному прикладі, лінія 16 зв'язку може містити комунікаційний носій, щоб дозволити пристрою-джерелу 12 передавати кодовані відеодані безпосередньо на пристрій 14 призначення в реальному часі. Закодовані відеодані можуть бути модульовані згідно зі стандартом зв'язку, таким як протокол бездротового зв'язку, і передані на пристрій 14 призначення. Комунікаційний носій може містити будь-який бездротовий або дротовий комунікаційний носій, такий як радіочастотного (РЧ) спектра або одну або більше фізичних ліній передачі. Комунікаційний носій може бути частиною основаної на пакетній передачі мережі, такої як локальна мережа, регіональна мережа або глобальна мережа, така як Інтернет. Комунікаційний носій може включати в себе маршрутизатори, перемикачі, базові станції або будь-яке інше обладнання, яке може бути корисним, щоб полегшити зв'язок від пристрою-джерела 12 на пристрій 14 призначення.

Альтернативно, кодовані дані можуть бути виведені з інтерфейсу 22 виведення на пристрій 32 зберігання. Аналогічно, до закодованих даних можна одержати доступ від пристрою 32 зберігання за допомогою інтерфейсу 28 введення. Пристрій 32 зберігання може включати в себе будь-який з множини розподілених або локально доступних носіїв зберігання даних, таких як накопичувач на жорстких дисках, диски Blu-ray, DVD, CD-ROM, флеш-пам'ять, енергозалежна

або енергоне залежна пам'ять або будь-які інші придатні цифрові носії даних, для того, щоб зберігати закодовані відеодані. У іншому прикладі пристрій 32 зберігання може відповідати файловому серверу або іншому проміжному пристрою зберігання, який може зберігати закодоване відео, згенероване вихідним пристроєм 12. Пристрій 14 призначення може одержати доступ до збережених відеоданих з пристрою 32 зберігання за допомогою передачі у вигляді потоку або завантаження. Файловий сервер може бути будь-яким типом сервера, здатного до того, щоб зберігати закодовані відеодані і передавати ці кодовані відеодані на пристрій 14 призначення. Зразкові файлові сервери включають в себе мережевий сервер (наприклад, для веб-сайта), FTP-сервер, пристрої мережевих сховищ даних (NAS) або локальний дисковий накопичувач. Пристрій 14 призначення може одержати доступ до кодованих відеоданих через будь-яке стандартне з'єднання даних, включаючи Інтернет-з'єднання. Воно може включати в себе бездротовий канал (наприклад, з'єднання Wi-Fi), дротове з'єднання (наприклад, DSL, кабельний модем і т. д.) або комбінацію обох, яка є придатною для того, щоб одержати доступ до закодованих відеоданих, що зберігаються на файловому сервері. Передача кодованих відеоданих від пристрою 32 зберігання може бути потоковою передачею, передачею завантаження або комбінацією обох.

Способи даного розкриття не обов'язково обмежені бездротовими додатками або параметрами настройки. Ці способи можуть бути застосовані до кодування відео на підтримання будь-якого з множини мультимедійних додатків, таких як радіомовлення телепередач, передач кабельного телебачення, передач супутникового телебачення, поточкових передач відео, наприклад через Інтернет, кодування цифрового відео для зберігання на запам'ятовуючому носії даних, декодування цифрового відео, збереженого на запам'ятовуючому носії даних, або інші додатки. У деяких прикладах система 10 може бути сконфігурована, щоб підтримувати односторонню або двосторонню відеопередачу, щоб підтримувати додатки, такі як передача у вигляді потоку відео, відтворення відео, мовлення відео і/або відеотелефонія.

У прикладі згідно з фіг. 1 пристрій-джерело 12 включає в себе відеоджерело 18, відеокодер 20 і інтерфейс 22 виведення. У деяких випадках інтерфейс 22 виведення може включати в себе модулятор/демодулятор (модем) і/або передавач. У пристрої-джерелі 12 відеоджерело 18 може включати в себе джерело, таке як відеопристрій захоплення, наприклад відеокамера, відеоархів, що містить раніше захоплене відео, інтерфейс подачі відео, щоб приймати відео від постачальника відеоконтенту, і/або систему комп'ютерної графіки для генерування даних комп'ютерної графіки як вихідного відео або комбінацію таких джерел. Як один приклад, якщо відеоджерело 18 є відеокамерою, пристрій-джерело 12 і пристрій 14 призначення можуть сформулювати так звані камерофони або відеотелефони. Однак, способи, описані в даному розкритті, можуть бути застосовні до кодування відео звичайно і можуть бути застосовані до бездротових і/або дротових додатків.

Захоплене, попередньо захоплене або комп'ютерногенероване відео може бути закодоване відеокодером 20. Закодовані відеодані можуть бути передані безпосередньо на пристрій 14 призначення через інтерфейс 22 виведення з пристрою-джерела 12. Закодовані відеодані можуть також (або альтернативно) зберігатися на пристрої 32 зберігання для більш пізнього доступу за допомогою пристрою 14 призначення або інших пристроїв, для декодування і/або відтворення.

Пристрій 14 призначення включає в себе інтерфейс 28 введення, відеодекодер 30 і пристрій 32 відображення. У деяких випадках інтерфейс 28 введення може включати в себе приймач і/або модем. Інтерфейс 28 введення з пристрою 14 призначення приймає закодовані відеодані по лінії 16 зв'язку. Закодовані відеодані, передані по лінії 16 зв'язку, або надані на пристрої 32 зберігання, можуть включати в себе множину елементів синтаксису, генерованих відеокодером 20, для використання відеодекодером, таким як відеодекодер 30, при декодуванні відеоданих. Такі елементи синтаксису можуть бути включені із закодованими відеоданими, переданими на комунікаційному носії, збережені на носії даних або збережені на файловому сервері.

Пристрій 32 відображення може бути інтегрованим з або зовнішнім до пристрою 14 призначення. У деяких прикладах пристрій 14 призначення може включати в себе інтегрований пристрій відображення і також бути конфігурованим, щоб взаємодіяти із зовнішнім пристроєм відображення. У інших прикладах пристрій 14 призначення може бути пристроєм відображення. Звичайно пристрій 32 відображення відображає декодовані відеодані користувачу і може містити будь-який з множини пристроїв відображення, таких як рідкокристалічний дисплей (LCD), плазмовий дисплей, дисплей на органічних світлодіодах (OLED) або інший тип пристрою відображення.

Відеокодер 20 і відеодекодер 30 можуть працювати згідно зі стандартом стиснення відео, таким як стандарт кодування відео високої продуктивності (HEVC), що знаходиться зараз в розвитку, і можуть відповідати Тестовій Моделі HEVC (HM). Альтернативно, відеокодер 20 і відеодекодер 30 можуть працювати згідно з іншими стандартами, що складають власність, або стандартами промисловості, такими як стандарт ITU-T H.264, альтернативно званий MPEG-4, Part 10, Advanced Video Coding (AVC), або розширеннями таких стандартів. Способи даного розкриття, однак, не обмежені ніяким конкретним стандартом кодування. Інші приклади включають в себе MPEG-2 і ITU-T H.263.

Хоч не показано на фіг. 1, в деяких аспектах відеокодер 20 і відеодекодер 30 можуть, кожний, інтегруватися з аудіокодером і декодером і можуть включати в себе відповідні блоки MUX-DEMUX (мультиплексорів-демультиплексорів) або інше апаратне забезпечення і програмне забезпечення, щоб виконувати кодування як аудіо, так і відео в загальному потоці даних або окремих потоках даних. Якщо застосовно, блоки MUX-DEMUX можуть відповідати протоколу мультиплексора ITU H.223 або іншим протоколам, таким як протокол дейтаграм користувача (UDP).

Відеокодер 20 і відеодекодер 30, кожний, можуть бути реалізовані як будь-яка з множини придатних схем кодера, таких як один або більше мікропроцесорів, цифрових сигнальних процесорів (DSPs), спеціалізованих інтегральних схем (ASIC), програмованих користувачем вентильних матриць (FPGA), дискретної логіки, програмного забезпечення, апаратного забезпечення, програмно-апаратних засобів або будь-яких їх комбінацій. Коли способи реалізуються частково в програмному забезпеченні, пристрій може зберігати інструкції для програмного забезпечення у придатному постійному зчитуваному комп'ютером носії і виконувати інструкції в апаратному забезпеченні, використовуючи один або більше процесорів, щоб виконати способи даного розкриття. Кожний відеокодер 20 і відеодекодер 30 може бути включений в один або більше кодерів або декодерів, будь-який з яких може бути інтегрованим як частина об'єднаного кодера/декодера (кодек) у відповідному пристрої.

JCT-VC працює над розвитком стандарту HEVC. Зусилля по стандартизації HEVC основані на моделі пристрою кодування відео, що розвивається, званий Тестова Модель HEVC (HM). HM передбачає декілька додаткових можливостей пристроїв кодування відео перед існуючими пристроями згідно, наприклад, з ITU-T H.264/AVC. Наприклад, тоді як H.264 забезпечує дев'ять режимів кодування з внутрішнім прогнозуванням, HM забезпечує цілих тридцять три режими кодування з внутрішнім прогнозуванням.

Звичайно робоча модель HM описує, що відеокадр або картинка можуть бути розділені на послідовність блоків дерева або найбільших одиниць кодування (LCU), які включають в себе вибірки як яскравості, так і кольоровості. Блок дерева має аналогічну мету, що і макроблок стандарту H.264. Вирізка включає в себе декілька послідовних блоків дерева в порядку кодування. Відеокадр або картинка можуть бути розділені на одну або більше вирізок. Кожний блок дерева може бути розділений на одиниці кодування (CUs) згідно з квадродревом. Наприклад, блок дерева, як вузол кореня квадродреву, може бути розділений на чотири дочірніх вузли, і кожний дочірній вузол може в свою чергу бути батьківським вузлом і бути розділений ще на чотири дочірніх вузли. Кінцевий нерозділений дочірній вузол, як вузол листа квадродреву, містить вузол кодування, тобто кодований відеоблок. Такий кінцевий нерозділений дочірній вузол структури даних відео згадується тут як одиниця рівня листа. Дані синтаксису, асоційовані з кодованим потоком бітів, можуть визначити максимальну кількість разів, скільки блок дерева може бути розділений, і можуть також визначати мінімальний розмір вузлів кодування.

CU включає в себе вузол кодування і одиниці прогнозування (PUs) і одиниці перетворення (TUs), асоційовані з вузлом кодування. Розмір CU відповідає розміру вузла кодування і повинен бути квадратним за формою. Розмір одиниці CU може змінюватися від 8×8 пікселів до розміру блока дерева з максимумом 64×64 пікселями або більше. Кожна CU може містити одну або більше PU і одну або більше TU. Дані синтаксису, асоційовані з CU, можуть описувати, наприклад, розділення CU в одну або більше одиниць PU. Режими розділення можуть відрізнятися між тим, чи є CU кодованою в режимі пропуску або прямому режимі, кодованою в режимі внутрішнього прогнозування або кодованою в режимі зовнішнього прогнозування. PU можуть бути розділені, щоб бути неквадратними за формою. Дані синтаксису, асоційовані з CU, можуть також описувати, наприклад, розділення CU в одну або більше TU згідно з квадродревом. TU може бути квадратною або неквадратною за формою.

Стандарт HEVC враховує перетворення згідно з одиницями TU, які можуть бути різними для різних CU. TU типово мають розміри на основі розмірів PU в межах заданої CU, визначеної для розділеної LCU, хоч це може не завжди мати місце. TU типово мають той же розмір або

менший, ніж PU. У деяких прикладах залишкові вибірки, відповідні CU, можуть бути підрозділені на менші одиниці, використовуючи структуру квадродерева, відому як "залишкове квадродерево" (RQT). Листові вузли RQT можуть згадуватися як одиниці перетворення (TUs). Такий листовий вузол TU є одним прикладом одиниці рівня листя, як описано тут. Значення піксельних різниць, асоційовані з TU, можуть бути перетворені, щоб сформувати коефіцієнти перетворення, які можуть бути квантовані.

Звичайно PU включає в себе дані, що стосуються процесу прогнозування. Наприклад, коли PU є кодованою у внутрішньому режимі, PU може включати в себе дані, що описують режим внутрішнього прогнозування для PU. Як інший приклад, коли PU є кодованою у зовнішньому режимі, PU може включати в себе дані, що визначають вектор руху для PU. Дані, що визначають вектор руху для PU, можуть описувати, наприклад, горизонтальний компонент вектора руху, вертикальний компонент вектора руху, розрізнення для вектора руху (наприклад, піксельну точність в одну чверть або піксельну точність в одну восьму), опорну картинку, на яку вектор руху вказує, і/або список опорних картинок (наприклад, Список 0, Список 1 або Список C) для вектора руху.

Звичайно TU використовується для процесів квантування і перетворення. Задана CU, що має одну або більше PU, може також включати в себе одну або більше одиниць перетворення (TU). Після прогнозування відеокoder 20 може обчислити залишкові значення, відповідні цій PU. Залишкові значення містять значення піксельної різниці, які можуть бути перетворені в коефіцієнти перетворення, квантовані і скановані з використанням одиниць TU, щоб сформувати перетворені в послідовну форму коефіцієнти перетворення для ентропійного кодування. Дане розкриття типово використовує термін "відеоблок", щоб посилатися на вузол кодування в CU. У деяких конкретних випадках дане розкриття може також використовувати термін "відеоблок", щоб посилатися на блок дерева, тобто LCU, або CU, що включає в себе вузол кодування і одиниці PU і одиниці TU.

Відеопослідовність типово включає в себе ряд відеокадрів або картинок. Група картинок (GOP) звичайно містить послідовність з однієї або більше відеокартинок. GOP може включати в себе дані синтаксису в заголовку GOP, заголовку однієї або більше картинок GOP або в іншому місці, що описує декілька картинок, включених в GOP. Кожна вирізка картини може включати в себе дані синтаксису вирізки, які описують режим кодування для відповідної вирізки. Відеокoder 20 типово оперує над відеоблоками в межах індивідуальних вирізок відео, щоб кодувати відеодані. Відеоблок може відповідати вузлу кодування в межах CU. Відеоблоки можуть мати фіксований або змінний розміри і можуть відрізнятися за розміром згідно з вказаним стандартом кодування.

Як приклад HM підтримує прогнозування в PU різних розмірів. Передбачаючи, що розмір конкретної CU дорівнює $2N \times 2N$, HM підтримує внутрішнє прогнозування в розмірах PU, що дорівнюють $2N \times 2N$ або $N \times N$, і зовнішнє прогнозування в симетричних розмірах PU, що дорівнюють $2N \times 2N$, $2N \times N$, $N \times 2N$ або $N \times N$. HM також підтримує асиметричне розділення для зовнішнього прогнозування в розмірах PU, що дорівнюють $2N \times nU$, $2N \times nD$, $nL \times 2N$ і $nR \times 2N$. При асиметричному розділенні один напрям CU не розділяється, в той час як інший напрям розділяється на 25 % і 75 %. Частина CU, відповідна 25 %-ому розділенню, вказується за допомогою "n", що супроводжується індикацією "Верхня" (Up), "Ліва" (Left) або "Права" (Right). Таким чином, наприклад, " $2N \times nU$ " стосується CU розміром $2N \times 2N$, яка розділена горизонтально з PU $2N \times 0,5N$ зверху і PU $2N \times 1,5N$ внизу.

У даному описі " $N \times N$ " і " N на N " можуть використовуватися взаємозамінно, щоб посилатися на піксельні розмірності блока в термінах вертикального і горизонтального вимірювань, наприклад 16×16 пікселів або 16 на 16 пікселів. Звичайно блок 16×16 буде мати 16 пікселів у вертикальному напрямі ($y=16$) і 16 пікселів в горизонтальному напрямі ($x=16$). Аналогічно, блок $N \times N$ звичайно має N пікселів у вертикальному напрямі і N пікселів в горизонтальному напрямі, де N представляє ненегативне цілочислове значення. Піксели в блоці можуть бути розміщені в рядках і колонках. Крім того, блоки не повинні обов'язково мати однакову кількість пікселів в горизонтальному напрямі як у вертикальному напрямі. Наприклад, блоки можуть містити $N \times M$ пікселів, де M не обов'язково дорівнює N .

Після кодування з внутрішнім прогнозуванням або із зовнішнім прогнозуванням, використовуючи одиниці PU в CU, відеокoder 20 може обчислити залишкові дані для одиниць TU в CU. Одиниці PU можуть містити піксельні дані в просторовій області (також званій піксельною областю), і одиниці TU можуть містити коефіцієнти в області перетворення після застосування перетворення, наприклад дискретного косинусного перетворення (DCT), цілочислового перетворення, вейвлет-перетворення або концептуально подібного перетворення, до залишкових відеоданих. Залишкові дані можуть відповідати піксельним різницям між пікселями

незакодованої картини і значеннями прогнозування, відповідними одиницям PU. Відеокодер 20 може сформувати одиниці TU, включаючи залишкові дані для CU, і потім перетворити одиниці TU, щоб сформувати коефіцієнти перетворення для цієї CU.

Після будь-якого перетворення, щоб сформувати коефіцієнти перетворення, відеокодер 20 може виконати квантування коефіцієнтів перетворення. Квантування звичайно належить до процесу, в якому коефіцієнти перетворення квантуються, щоб можливо зменшити об'єм даних, використовуваних, щоб представити коефіцієнти, що забезпечують подальше стиснення. Процес квантування може зменшити глибину в бітах, асоційовану з деякими або всіма коефіцієнтами. Наприклад, значення n -біт може бути округлене в меншу сторону до значення m -біт під час квантування, де n більше, ніж m .

У деяких прикладах відеокодер 20 може використовувати попередньо заданий порядок сканування, щоб сканувати квантовані коефіцієнти перетворення, щоб сформувати перетворений в послідовну форму вектор, який може бути ентропійно кодований. У інших прикладах відеокодер 20 може виконати адаптивне сканування. Після сканування квантованих коефіцієнтів перетворення, щоб сформувати одновимірний вектор, відеокодер 20 може ентропійно кодувати одновимірний вектор, наприклад, згідно з контекстно-адаптивним кодуванням із змінною довжиною коду (CAVLC), контекстно-адаптивним двійковим арифметичним кодуванням (CABAC), основаним на синтаксисі контекстно-адаптивним двійковим арифметичним кодуванням (SBAC), ентропійним кодуванням з розділенням інтервалу імовірності (PIPE) або іншими методиками ентропійного кодування. Відеокодер 20 може також ентропійно кодувати елементи синтаксису, асоційовані із закодованими відеоданими, для використання відеодекодером 30 при декодуванні відеоданих.

Щоб виконати CABAC, відеокодер 20 може надати контекст в межах контекстної моделі символу, який повинен бути переданий. Контекст може стосуватися того, наприклад, чи є сусідні значення символу ненульовими чи ні. Щоб виконати CAVLC, відеокодер 20 може вибрати код із змінною довжиною слова для символу, який повинен бути переданий. Кодові слова в VLC можуть бути побудовані таким чином, що відносно більш короткі коди відповідають більш імовірним символам, в той час як більш довгі коди відповідають менш імовірним символам. Таким чином, використання VLC може досягати економії бітів, наприклад, при використанні кодових слів рівної довжини для кожного символу, який повинен бути переданий. Визначення імовірності може бути основане на контексті, призначеному на символ.

Відповідно до цього розкриття, відеокодер 20 з пристрою-джерела 12 може сканувати коефіцієнти перетворення одиниці рівня листя відеоданих (наприклад, листовий вузол квадродререва або іншої структури даних), яка включає в себе двовимірну матрицю коефіцієнтів перетворення (наприклад, де кожний відповідає пікселам відображеного зображення) в одновимірний вектор, який представляє коефіцієнти перетворення. Згідно зі способами, розкритими в даному описі, виконуючи таке сканування, відеокодер 20 може розділити одиницю рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення. Для кожного з піднаборів одиниці рівня листя кодер 20 може визначити, чи сигналізувати до декодера як частину ентропійно кодованого бітового потоку елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти. Кодер 20 може визначити, чи сигналізувати елемент синтаксису для конкретного піднабору, на основі визначення, чи поліпшить сигналізація елемента синтаксису ефективність кодування. Щоб визначити, чи поліпшить сигналізація елемента синтаксису ефективність кодування, кодер 20 може застосувати одне або більше правил, як описано більш детально нижче. Кодер 20 може вивести ентропійно кодований бітовий потік, який включає в себе блок відеоданих. Ентропійно кодований бітовий потік може потім бути зчитаний і декодований декодером, щоб відновити двовимірну матрицю, яка представляє одиницю рівня листя відеоданих.

Зворотне декодування коефіцієнтів перетворення може також бути виконане відеодекодером 30 з пристрою 14 призначення. Таким чином, відеодекодер 30 може відобразити коефіцієнти одновимірного вектора коефіцієнтів перетворення, які представляють блок відеоданих, в позиції в двовимірній матриці коефіцієнтів перетворення, щоб відновити двовимірну матрицю коефіцієнтів перетворення. Згідно зі способами, розкритими в даному описі, декодер 30 може зчитати одновимірну матрицю, яка представляє одиницю рівня листя відеоданих, і розділити цю одиницю рівня листя на множину піднаборів коефіцієнтів перетворення. Для кожного з піднаборів декодер 30 може визначити, чи декодувати коефіцієнти перетворення піднабору. Наприклад, якщо декодер 30 не зчитує в ентропійно кодованому бітовому потоці елемент синтаксису, який вказує, чи має конкретний піднабір ненульові коефіцієнти, то декодер 30 декодує коефіцієнти піднабору. Однак, якщо декодер 30 зчитує такий елемент синтаксису, асоційований з конкретним піднабором, декодер 30 може визначити, чи

декодувати коефіцієнти перетворення цього піднабору, на основі значення елемента синтаксису. Наприклад, якщо елемент синтаксису вказує, що піднабір включає в себе коефіцієнти, відмінні від нуля, декодер 30 декодує коефіцієнти перетворення цього піднабору. Однак, якщо елемент синтаксису вказує, що піднабір не включає в себе ненульових коефіцієнтів, декодер 30 не декодує коефіцієнти перетворення цього піднабору.

Способи, описані тут, можуть підвищити ефективність кодування відео. Наприклад, розділення блока відеоданих на множину піднаборів коефіцієнтів перетворення і сигналізація елементів синтаксису, які вказують, чи включають піднабори в себе ненульові коефіцієнти, як описано тут, можуть зменшити кількість бітів, необхідних для представлення відеоданих, що може підвищити рівень ефективності кодера/декодера.

Фіг. 2 є блок-схемою, що ілюструє зразковий відеокодер 20, який може реалізувати способи зовнішнього прогнозування, описані в даному розкритті. Відеокодер 20 може виконати внутрішнє і зовнішнє кодування відеоблоків в межах відеовірізок. Внутрішнє кодування покладається на просторове прогнозування, щоб зменшити або видалити просторову надмірність у відео в межах заданого відеокадру або картинки. Зовнішнє кодування покладається на часове прогнозування, щоб зменшити або видалити часову надмірність у відео в межах суміжних кадрів або картинок відеопослідовності. Внутрішній режим (І режим) може посылатися на будь-який з декількох основаних на просторовому стисненні режимів. Зовнішні режими, такі як однонаправлене прогнозування (Р режим) або двонаправлене прогнозування (В режим), можуть посылатися на будь-який з декількох основаних на часовому стисненні режимів.

У прикладі згідно з фіг. 2 відеокодер 20 включає в себе блок 35 розділення, блок 41 прогнозування, пам'ять 64 опорних картинок, суматор 50, модуль 52 перетворення, блок 54 квантування і блок 56 ентропійного кодування. Блок 41 прогнозування включає в себе блок 42 оцінки руху, блок 44 компенсації руху і модуль 46 внутрішнього прогнозування. Для реконструкції відеоблока відеокодер 20 також включає в себе блок 58 зворотного квантування, модуль 60 інверсного перетворення і суматор 62. Фільтр видалення блоковості (не показаний на фіг. 2) може також бути включений, щоб фільтрувати границі блока, щоб видалити артефакти блоковості з відновленого відео. Якщо бажано, фільтр видалення блоковості типово фільтрує вихідні дані суматора 62. Додаткові фільтри контуру (в контурі або після контуру) можуть також використовуватися на доповнення до фільтра видалення блоковості.

Як показано на фіг. 2, відеокодер 20 приймає відеодані, і блок 35 розділення розділяє дані на відеоблоки. Це розділення може також включати в себе розділення на вирізки, мозаїчні елементи або інші великі блоки, а також розділення відеоблока, наприклад, згідно зі структурою квадродерева одиниць LCU і CU. Відеокодер 20 загалом ілюструє компоненти, які кодують відеоблоки в межах вирізки відео, яка повинна бути закодована. Вирізка може бути розділена на множинні відеоблоки (і можливо на набори відеоблоків, названих мозаїчними елементами). Блок 41 прогнозування може вибрати один з множини можливих режимів кодування, таких як один з множини режимів внутрішнього кодування або один з множини режимів зовнішнього кодування, для поточного відеоблока на основі результатів помилки (наприклад, швидкості кодування і рівня спотворення). Блок 41 прогнозування може видати результуючий внутрішньо-або зовнішньокодований блок до суматора 50, щоб генерувати залишкові дані блока, і до суматора 62, щоб відновити закодований блок для використання як опорної картинки.

Модуль 46 внутрішнього прогнозування в межах модуля 41 прогнозування може виконати внутрішнє прогнозує кодування поточного відеоблока відносно одного або більше сусідніх блоків в тому ж самому кадрі або вирізці, що і поточний блок, який повинен бути закодований, щоб забезпечити просторове стиснення. Блок 42 оцінки руху і блок 44 компенсації руху в межах модуля 41 прогнозування виконують зовнішнє прогнозує кодування поточного відеоблока відносно одного або більше прогнозуючих блоків в одній або більше опорних картинках, щоб забезпечити часове стиснення.

Блок 42 оцінки руху може бути сконфігурований, щоб визначити режим зовнішнього прогнозування для вирізки відео згідно із попередньо визначеним шаблоном для відеопослідовності. Попередньо визначений шаблон може визначати вирізки відео в послідовності як Р-вирізки, В-вирізки або вирізки GPB. Блок 42 оцінки руху і блок 44 компенсації руху можуть бути високоінтегровані, але ілюстровані по окремості в концептуальних цілях. Оцінка руху, виконана блоком 42 оцінки руху, є процесом генерування векторів руху, які оцінюють руху для відеоблоків. Вектор руху, наприклад, може вказувати зміщення RU відеоблока в межах поточного відеокадру або картинки відносно прогнозуючого блока в межах опорної картинки.

Прогнозуючий блок є блоком, який виявлений як такий, що близько відповідає RU відеоблока, який повинен бути кодований, в термінах піксельної різниці, яка може бути визначена

сумою абсолютних різниць (SAD), сумою різниць квадратів (SSD) або іншими метриками відмінності. У деяких прикладах відеокoder 20 може обчислювати значення для субцілочислових піксельних позицій опорних картинок, збережених в пам'яті 64 опорних картинок. Наприклад, відеокoder 20 може інтерполювати значення піксельних позицій в одну чверть, піксельних позицій в одну восьму або інших фракційних піксельних позицій опорної картини. Тому блок 42 оцінки руху може виконувати пошук руху відносно повних піксельних позицій і фракційних піксельних позицій і вивести вектор руху з фракційною піксельною точністю.

Блок 42 оцінки руху обчислює вектор руху для PU відеоблока у зовнішньокодованій вирізці, порівнюючи позицію PU з позицією прогнозуючого блока опорної картини. Опорна картинка може бути вибрана з першого списку опорних картинок (Список 0) або другого списку опорних картинок (Список 1), кожний з яких ідентифікує одну або більше опорних картинок, збережених в пам'яті 64 опорних картинок. Блок 42 оцінки руху посилає обчислений вектор руху в блок 56 ентропійного кодування і блок 44 компенсації руху.

Компенсація руху, виконана блоком 44 компенсації руху, може включати вибірку або генерування прогнозуючого блока, на основі вектора руху, визначеного за допомогою оцінки руху, можливо виконуючи інтерполяцію до субпіксельної точності. Після прийому вектора руху для PU поточного відеоблока блок 44 компенсації руху може визначити місцезнаходження прогнозуючого блока, на який вектор руху вказує в одному зі списків опорних картинок. Відеокoder 20 формує залишковий відеоблок, віднімаючи піксельні значення прогнозуючого блока з піксельних значень поточного кодованого блока відео, формуючи значення піксельних різниць. Значення піксельних різниць формують залишкові дані для блока і можуть включати в себе компоненти різниці і яскравості, і кольоровості. Суматор 50 представляє компонент або компоненти, які виконують цю операцію віднімання. Блок 44 компенсації руху може також генерувати елементи синтаксису, асоційовані з відеоблоками і відеовирізкою, для використання відеокoderом 30 при декодуванні відеоблоків вирізки відео.

Після того, як блок 44 компенсації руху генерує прогнозуючий блок для поточного відеоблока, відеокoder 20 формує залишковий відеоблок, віднімаючи прогнозуючий блок з поточного відеоблока. Залишкові відеодані в залишковому блоці можуть бути включені в одну або більше TU і подані до модуля 52 перетворення. Модуль 52 перетворення перетворює залишкові відеодані в залишкові коефіцієнти перетворення, використовуючи перетворення, такі як дискретне косинусне перетворення (DCT) або концептуально подібне перетворення. Модуль 52 перетворення може перетворити залишкові відеодані з піксельної області до області перетворення, такої як частотна область.

Модуль 52 перетворення може послати одержувані коефіцієнти перетворення в блок 54 квантування. Блок 54 квантування квантує коефіцієнти перетворення, щоб додатково зменшити частоту проходження бітів. Процес квантування може зменшити глибину в бітах, асоційовану з деякими або всіма коефіцієнтами. Міра квантування може бути змінена, регулюючи параметр квантування.

Після квантування блок 56 ентропійного кодування ентропійно кодує квантовані коефіцієнти перетворення. Наприклад, блок 56 ентропійного кодування може виконати контекстно-адаптивне кодування із змінною довжиною коду (CAVLC), контекстно-адаптивне двійкове арифметичне кодування (CABAC), основане на синтаксисі контекстно-адаптивне двійкове арифметичне кодування (SBAC), ентропійне кодування з розділенням інтервалу імовірності (PIPE) або інший спосіб ентропійного кодування або методу. Після ентропійного кодування за допомогою блока 56 ентропійного кодування кодований потік бітів може бути переданий до відеокoderа 30 або заархівований для більш пізньої передачі або пошуку відеокoderом 30. Блок 56 ентропійного кодування може також ентропійно кодувати вектора руху і інші елементи синтаксису для поточної закодованої вирізки відео. У деяких прикладах блок 56 ентропійного кодування може потім виконати сканування матриці, що включає в себе квантовані коефіцієнти перетворення, щоб генерувати одновимірний вектор коефіцієнтів перетворення ентропійно кодованого бітового потоку.

Згідно зі способами даного розкриття при скануванні матриці коефіцієнтів перетворення, щоб генерувати одновимірний вектор, блок 56 ентропійного кодування може розділити одиницю рівня листя відеоданих, таку як нерозділений дочірній вузол структури квадродерева, як описано вище, на множину піднаборів коефіцієнтів перетворення, які менше, ніж одиниця рівня листя. Наприклад, блок 56 ентропійного кодування може розділити одиницю рівня листя відеоданих на множину піднаборів, де кожний має геометричну форму в межах одиниці рівня листя. У деяких прикладах блок 56 ентропійного кодування може розділити коефіцієнти перетворення одиниці рівня листя на менші компонування прямокутної форми відеоданих. У інших прикладах блок 56 ентропійного кодування може розділити коефіцієнти перетворення

одиниці рівня листя на піднабори трикутної форми коефіцієнтів перетворення. У ще інших прикладах блок 56 ентропійного кодування може розділити коефіцієнти перетворення одиниці рівня листя на піднабори, що мають інші форми, або навіть піднабори, які не відповідають ніякій конкретній геометричній формі. Замість цього блок 56 ентропійного кодування може розділити

5 множину коефіцієнтів перетворення одиниці рівня листя згідно з порядком сканування коефіцієнтів перетворення, який може бути фіксованим або адаптивним. Наприклад, блок 56 ентропійного кодування може розділити множину коефіцієнтів перетворення одиниць рівня листя на множину піднаборів коефіцієнтів перетворення, на основі порядку сканування (наприклад, адаптивного або фіксованого порядку сканування) коефіцієнтів перетворення.

10 Піднабори можуть мати подібні або різні кількості коефіцієнтів в різних прикладах.

Згідно зі способами, розкритими в даному описі, як тільки одиниця рівня листя коефіцієнтів перетворення була розділена на множину піднаборів, блок 56 ентропійного кодування може генерувати, для щонайменше деяких з множини піднаборів, елемент синтаксису, який вказує, чи включає в себе відповідний піднабір які-небудь ненульові коефіцієнти. У деяких прикладах блок

15 56 ентропійного кодування може генерувати такий елемент синтаксису для кожного піднабору з множини піднаборів коефіцієнтів перетворення. У інших прикладах блок 56 ентропійного кодування може вибірково визначити, чи генерувати такий елемент синтаксису для кожного піднабору з множини піднаборів. Рішення, чи генерувати такий елемент синтаксису, може бути

20 основане на тому, чи передбачає така сигналізація додаткове стиснення або поліпшену ефективність кодування. Якщо так, елемент синтаксису може генеруватися, але в іншому випадку і кодер 20, і декодер 30 можуть бути запрограмовані, щоб знати, що індикація не сигналізується в цьому випадку.

Згідно зі способами даного розкриття, щоб визначити, чи поліпшить генерування такого елемента синтаксису ефективність стиснення або кодування, блок 56 ентропійного кодування

25 може застосувати одне або більше правил, як описано більш детально нижче. Як один приклад блок 56 ентропійного кодування може визначити, чи генерувати елемент синтаксису для конкретного піднабору, на основі кількості потенційних ненульових коефіцієнтів цього піднабору. Наприклад, блок 56 ентропійного кодування може визначити, чи генерувати елемент синтаксису, на основі порівняння багатьох потенційних відмінних від нуля коефіцієнтів цього

30 піднабору з порогом.

Як один приклад такого способу, щоб визначити кількість потенційних ненульових коефіцієнтів піднабору, блок 56 ентропійного кодування визначає кількість коефіцієнтів піднабору, які мають більш ранню позицію, ніж останній ненульовий коефіцієнт цієї одиниці рівня листя, і порівнюють визначену кількість коефіцієнтів з порогом $thNoCoeff$, як показано в

35 прикладі згідно з фіг. 6 і описано більш детально нижче. Згідно з цим прикладом, якщо визначена кількість потенційних ненульових коефіцієнтів більше, ніж поріг $thNoCoeff$, блок 56 ентропійного кодування генерує елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти. Однак, якщо визначена кількість потенційних ненульових коефіцієнтів менше ніж або дорівнює порогу $thNoCoeff$, блок 56 ентропійного

40 кодування може не генерувати елемент синтаксису. Як один приклад, блок 56 ентропійного кодування визначає, чи сигналізувати елемент синтаксису, який вказує, чи включає в себе конкретний піднабір які-небудь ненульові коефіцієнти, на основі псевдокоду Прикладу 1 нижче:

Приклад 1

$noCoeff[xS][yS] > thNoCoeff$,

45 де $noCoeff[xS][yS]$ вказує кількість потенційних ненульових коефіцієнтів піднабору, і $thNoCoeff$ є пороговим значенням.

У інших прикладах блок 56 ентропійного кодування може визначити, чи сигналізувати елемент синтаксису, який вказує, чи включає в себе піднабір коефіцієнтів перетворення які-небудь ненульові коефіцієнти, на основі інших правил. Наприклад, блок 56 ентропійного

50 кодування може визначити, чи сигналізувати елемент синтаксису для піднабору, на основі середньої кількості ненульових коефіцієнтів піднабору, як описано більш детально нижче відносно прикладу згідно з фіг. 7. Згідно з цим прикладом, коли одиниці рівня листя відеоданих кодуються, блок 56 ентропійного кодування може зібрати і зберегти статистику, що вказує, як часто коефіцієнти в позиціях в межах кожного піднабору є ненульовими. На основі такої

55 збереженої статистики блок 56 ентропійного кодування може визначити, чи генерувати елемент синтаксису. Наприклад, блок 56 ентропійного кодування може порівняти середню кількість ненульових коефіцієнтів для піднабору з порогом $tfiAvgCoeff$. Якщо середня кількість ненульових коефіцієнтів піднабору менше, ніж поріг, блок 56 ентропійного кодування генерує елемент синтаксису. Однак, якщо середня кількість ненульових коефіцієнтів піднабору більше

60 ніж або дорівнює порогу $tfiAvgCoeff$, блок 56 ентропійного кодування не генерує елемент

синтаксису на основі припущення, що в цьому випадку дуже ймовірно, що піднабір містить щонайменше один ненульовий коефіцієнт.

Як інший приклад для піднабору, який містить останній ненульовий коефіцієнт (відносно зигзагоподібного сканування, перший ненульовий коефіцієнт при зворотному зигзагоподібному скануванні) одиниці рівня листя, той факт, що цей піднабір містить останній ненульовий коефіцієнт, означає, що це піднабір повинен включати в себе щонайменше один ненульовий коефіцієнт, і тому можна передбачити, що будь-який піднабір, який включає в себе останній ненульовий коефіцієнт, повинен включати в себе щонайменше один ненульовий коефіцієнт. Відповідно, немає необхідності генерувати елемент синтаксису для піднабору, який включає в себе останній ненульовий коефіцієнт, оскільки цей піднабір, як можна передбачити, включає в себе щонайменше один ненульовий коефіцієнт.

Як один приклад, блок 56 ентропійного кодування може визначити, чи сигналізувати елемент синтаксису, який вказує, чи включає конкретний піднабір в себе які-небудь ненульові коефіцієнти, на основі псевдокоду Прикладу 2 нижче:

Приклад 2

```
(noCodedCoeff[xS][yS]+noSubBlks[xS][yS]/2)/noSubBlks<tbAvrgCoeff,
```

тут noCodedCoeff[xS][yS] вказує кількість ненульових коефіцієнтів в піднаборі. Згідно з псевдокодом з Прикладу 1, блок 56 ентропійного кодування ініціалізує значення в масиві змінних noCoeff[xS][yS] в 0 раніше, ніж кодувати одиницю рівня листя відеоблока даних і надати значення цьому масиву, на основі значення елемента синтаксису lastPos, який вказує позицію останнього ненульового коефіцієнта в межах одиниці рівня листя відеоданих:

```
for (pos=0; pos<=lastPos; pos++){
  xC=ScanOrder [0][pos];
  yC=ScanOrder [!][pos];
  noCoeff[xC/M][yC/M]++;
}
```

Згідно з псевдокодом Прикладу 2, після того, як одиниця рівня листя закодована, блок 56 ентропійного кодування може оновити масив змінних noSubBlks і noCodedCoeff таким чином:

```
for (pos=0; pos<=lastPos; pos++){
  xC=ScanOrder [0][pos];
  yC=ScanOrder [!][pos];
  if (transCoeffLevel[xC][yC]!=0){
    noCodedCoeff[xC/M][yC/M]++;
  }
}
for (xS=0; xS<N/M; xS++){
  for (yS=0; yS<N/M; yS++){
    if (noCoeff[xS][yS]>0){
      noSubBlks[xS][yS]++;
    }
  }
}
```

При використанні псевдокоду з Прикладу 2 вище, блок 56 ентропійного кодування може визначити, чи сигналізувати елемент синтаксису, який вказує, чи включає в себе піднабір коефіцієнтів перетворення одиниці рівня листя які-небудь ненульові коефіцієнти, на основі середньої кількості ненульових коефіцієнтів піднабору для раніше закодованих одиниць рівня листя відеоданих.

Блок 58 зворотного квантування і модуль 60 зворотного перетворення застосовують зворотне квантування і зворотне перетворення, відповідно, щоб відновити залишковий блок в піксельній області для більш пізнього використання як опорного блока опорної картинки. Блок 44 компенсації руху може обчислити опорний блок за допомогою підсумовування залишкового блока і прогнозуючого блока однієї з опорних картинок в межах одного зі списків опорних картинок. Блок 44 компенсації руху може також застосувати один або більше фільтрів інтерполяції до відновленого залишкового блока, щоб обчислити субцілочислові піксельні значення для використання в оцінці руху. Суматор 62 додає відновлений залишковий блок до блока прогнозування з компенсацією руху, сформованого блоком 44 компенсації руху, щоб сформувану опорний блок для зберігання в пам'яті 64 опорних картинок. Опорний блок може використовуватися блоком 42 оцінки руху і блоком 44 компенсації руху як опорний блок, щоб зовнішньо прогнозувати блок в подальшому відеокадрі або картинці.

Фіг. 3 є блок-схемою, що ілюструє зразковий відеодекодер 30, який може реалізувати способи зовнішнього прогнозування, описані в даному розкритті. У прикладі згідно з фіг. 3 відеодекодер 30 включає в себе блок 80 ентропійного декодування, модуль 81 прогнозування, блок 86 зворотного квантування, блок 88 зворотного перетворення, суматор 90 і пам'ять 92 опорних картинок. Модуль 81 прогнозування включає в себе блок 82 компенсації руху і модуль 84 внутрішнього прогнозування. Відеодекодер 30 в деяких прикладах може виконати прохід декодування, загалом зворотний проходу кодування, описаному відносно відеокодера 20 на фіг. 2.

Під час процесу декодування відеодекодер 30 приймає кодований відеопотік бітів, який представляє відеоблоки кодованої вирізки відео і асоційовані елементи синтаксису, від відеокодера 20. Блок 80 ентропійного декодування відеокодера 30 ентропійно декодує потік бітів, щоб генерувати квантовані коефіцієнти вектора руху і інші елементи синтаксису. Блок 80 ентропійного декодування направляє вектори руху і інші елементи синтаксису до модуля 81 прогнозування. Відеодекодер 30 може прийняти елементи синтаксису на рівні вирізки відео і/або рівні блока відео.

Блок 80 ентропійного декодування може зчитувати одновимірний вектор коефіцієнтів перетворення, декодований блоком ентропійного декодування, і відновити двовимірну матрицю коефіцієнтів перетворення з одновимірного вектора. Згідно зі способами, розкритими в даному описі, блок 80 ентропійного декодування може зчитувати одновимірну матрицю, яка представляє одиницю рівня листя відеоданих, і розділити цю одиницю рівня листя на множину піднаборів коефіцієнтів перетворення. Піднабори коефіцієнтів перетворення можуть мати прямокутну, трикутну або будь-яку іншу форму або компонування. Для кожного з піднаборів блок 80 зворотного ентропійного декодування визначає, чи декодувати коефіцієнти перетворення піднабору. Наприклад, якщо блок 80 ентропійного декодування не зчитує елемент синтаксису, який вказує, чи має конкретний піднабір ненульові коефіцієнти, то блок 86 зворотного квантування декодує коефіцієнти піднабору. Однак, якщо блок 80 ентропійного декодування зчитує такий елемент синтаксису, асоційований з конкретним піднабором, блок 80 ентропійного декодування може визначити, чи декодувати коефіцієнти перетворення піднабору, на основі значення елемента синтаксису. Наприклад, якщо елемент синтаксису вказує, що піднабір включає в себе ненульові коефіцієнти, блок 80 ентропійного декодування декодує коефіцієнти перетворення піднабору. Однак, якщо елемент синтаксису вказує, що піднабір не включає в себе ненульових коефіцієнтів, блок 80 ентропійного декодування не декодує коефіцієнти перетворення цього піднабору.

Коли відеовирізка кодована як внутрішньокодована (I) вирізка, модуль 84 внутрішнього прогнозування з модуля 81 прогнозування може генерувати дані прогнозування для відеоблока поточної вирізки відео на основі сигналізованого режиму внутрішнього прогнозування і даних від раніше декодованих блоків поточного кадру або картинки. Коли відеокадр кодований як зовнішньокодована (тобто B, P або GPB) вирізка, блок 82 компенсації руху модуля 81 прогнозування формує прогнозуючі блоки для відеоблока поточної вирізки відео на основі векторів руху і інших елементів синтаксису, прийнятих від блока 80 ентропійного декодування. Прогнозуючі блоки можуть бути сформовані з однієї з опорних картинок в межах одного зі списків опорних картинок. Відеодекодер 30 може побудувати списки опорних кадрів, Список 0 і Список 1, використовуючи способи конструювання за умовчанням, на основі опорних картинок, збережених в пам'яті 92 опорних картинок.

Блок 82 компенсації руху визначає інформацію прогнозування для відеоблока поточної вирізки відео, за допомогою синтаксичного розбору вектора руху і інших елементів синтаксису, і використовує цю інформацію прогнозування, щоб сформувати прогнозуючі блоки для поточного відео декодованого блока. Наприклад, блок 82 компенсації руху використовує деякі з прийнятих елементів синтаксису, щоб визначити режим прогнозування (наприклад, внутрішнє або зовнішнє прогнозування), використовуваний для кодування відеоблоків вирізки відео, тип вирізки із зовнішнім прогнозуванням (наприклад, B-вирізка, P-вирізка або вирізка GPB), інформацію конструювання для одного або більше зі списків опорних картинок для вирізки, векторів руху для кожного зовнішньокодованого відеоблока цієї вирізки, статусу зовнішнього прогнозування для кожного зовнішньокодованого відеоблока цієї вирізки і іншої інформації, щоб декодувати відеоблоки в поточній вирізці відео.

Блок 82 компенсації руху може також виконати інтерполяцію, основу на фільтрах інтерполяції. Блок 82 компенсації руху може використовувати фільтри інтерполяції, які використовуються відеокодером 20 під час кодування відеоблоків, щоб обчислити інтерпольовані значення для субцілочислових пікселів опорних блоків. У цьому випадку блок 82 компенсації руху може визначити фільтри інтерполяції, використовувані відеокодером 20, з

прийнятих елементів синтаксису і використовувати ці фільтри інтерполяції, щоб сформувати прогнозуючі блоки. Блок 86 зворотного квантування зворотно квантує, тобто деквантує, квантовані коефіцієнти перетворення, надані в потоці бітів і декодовані блоком 80 ентропійного декодування.

У деяких прикладах процес зворотного квантування може включати в себе використання параметра квантування, обчисленого відеокодером 20 для кожного відеоблока у відеовирізці, щоб визначити міру квантування і, аналогічно, міру зворотного квантування, яке повинно бути застосоване. Модуль 88 зворотного перетворення застосовує зворотне перетворення, наприклад зворотне DCT, зворотне цілочислове перетворення, або концептуально подібний процес зворотного перетворення до коефіцієнтів перетворення, щоб сформувати залишкові блоки в піксельній області.

Після того, як блок 82 компенсації руху генерує прогнозуючий блок для поточного відеоблока, на основі векторів руху і інших елементів синтаксису, відеодекодер 30 формує декодований відеоблок за допомогою підсумовування залишкових блоків від модуля 88 зворотного перетворення з відповідними прогнозуючими блоками, генерованими блоком 82 компенсації руху. Суматор 90 представляє компонент або компоненти, які виконують цю операцію підсумовування. Якщо бажано, фільтр видалення блоковості може також бути застосований, щоб фільтрувати декодовані блоки, щоб видалити артефакти блоковості. Інші фільтри контуру (або в контурі кодування, або після контуру кодування) можуть також використовуватися, щоб згладити піксельні переходи або інакше поліпшити якість відео. Декодовані відеоблоки в заданому кадрі або картинці потім зберігаються в пам'яті 92 опорних картинок, яка зберігає опорні картини, використовувані для подальшої компенсації руху. Пам'ять 92 опорних картинок також зберігає декодоване відео для більш пізнього представлення на пристрої відображення, такому як пристрій 32 відображення з фіг. 1.

Фіг. 4 є концептуальною діаграмою, яка зображає один приклад одиниці 410 рівня листя, розділеної на множину піднаборів коефіцієнтів перетворення, сумісних зі способами, розкритими в даному описі. Одиниця 410 рівня листя, зображена на фіг. 4, може містити нерозділений дочірній вузол відеоданих, наприклад листовий вузол структури квадродерева (RQT). Як показано на фіг. 4, кожна множина піднаборів 420 включає в себе множину коефіцієнтів перетворення 412 (шістнадцять коефіцієнтів перетворення в прикладі на фіг. 1), які можуть бути або можуть не бути ненульовими коефіцієнтами з амплітудою, більше ніж нуль. Як також показано на фіг. 4, останній ненульовий коефіцієнт (відносно зигзагоподібного сканування, перший ненульовий коефіцієнт зворотного зигзагоподібного сканування) одиниці 410 рівня листя має позицію (7, 7) в межах піднабору (1, 1) одиниці 410 рівня листя.

Як описано вище, як тільки відеокодер 20 розділив одиницю 410 рівня листя на множину піднаборів 420 коефіцієнтів перетворення, відеокодер 20 може визначити, чи генерувати елемент синтаксису, який вказує, чи включає в себе згаданий піднабір які-небудь ненульові коефіцієнти. Наприклад, відеокодер 20 може визначити, чи генерувати елемент синтаксису, асоційований з конкретним блоком, на основі кількості потенційних ненульових коефіцієнтів в блоці.

Згідно з прикладом на фіг. 4 коефіцієнт в позиції (7, 7) є останнім ненульовим коефіцієнтом одиниці 420 рівня листя, означаючи, що всі коефіцієнти перетворення після позиції (7, 7) згідно із зигзагоподібним порядком сканування (незатемнені коефіцієнти в прикладі на фіг. 4) мають амплітуду нуль. Згідно з одним прикладом відеокодер 20 може визначити кількість потенційних ненульових коефіцієнтів в піднаборі на основі останньої позиції ненульового коефіцієнта в межах одиниці 410 рівня листя.

Відеокодер 20 може, для кожного відповідного піднабору одиниці 410 рівня листя, визначити потенційну кількість ненульових коефіцієнтів (наприклад, коефіцієнти зворотного зигзагоподібного сканування, які йдуть за останнім ненульовим коефіцієнтом сканування в місцеположенні (7, 7) піднабору). Наприклад, відеокодер 20 може визначити, що всі шістнадцять коефіцієнтів кожного з піднаборів (0, 0), (0, 1), (1, 0), (1, 1) і (2, 0), зображених на фіг. 4, є потенційно ненульовими коефіцієнтами, оскільки всі коефіцієнти піднаборів знаходяться перед останнім ненульовим коефіцієнтом в цьому скануванні. Відеокодер може також визначити, що п'ятнадцять коефіцієнтів піднабору (0, 2) можуть потенційно бути ненульовими коефіцієнтами, шість коефіцієнтів кожного з піднаборів (2, 1) і (3, 0) можуть потенційно бути ненульовими коефіцієнтами, і три коефіцієнти піднаборів (1, 1) і (0, 3) можуть потенційно бути ненульовими коефіцієнтами.

Кодер 20 може визначити, чи генерувати елемент синтаксису для піднаборів 420, зображених на фіг. 4, на основі визначеної кількості потенційних ненульових коефіцієнтів для кожного піднабору. Наприклад, кодер 20 може порівняти визначену кількість потенційних

ненульових коефіцієнтів для кожного піднабору з порогом $thNoCoeff$. Якщо визначена кількість потенційних ненульових коефіцієнтів більше, ніж поріг, кодер 20 може генерувати елемент синтаксису, який вказує, чи включає в себе згаданий піднабір ненульові коефіцієнти. Однак, якщо визначена кількість потенційних ненульових коефіцієнтів менше ніж або дорівнює порогові, кодер 20 може не генерувати елемент синтаксису.

Згідно з одним конкретним прикладом, поріг $thNoCoeff$ може мати значення п'ять (5). З посиланням на приклад на фіг. 4 кодер 20 може сигналізувати елемент синтаксису, який вказує, чи включають в себе піднабори (0, 0), (0, 1), (1, 0), (1, 1), (0, 2), (2, 0), (2, 1) і (3, 0) ненульові коефіцієнти, оскільки кожний з цих піднаборів 420 включає в себе більше потенційних ненульових коефіцієнтів, ніж значення поріг $thNoCoeff$ п'ять. Однак, для піднаборів (1, 2), (0, 3), (3, 1), (2, 2), (1, 3), (3, 2), (2, 3) і (3, 3), відеокодер 20 може не генерувати елемент синтаксису, оскільки ці піднабори включають в себе менше, ніж п'ять потенційних ненульових коефіцієнтів. Наприклад, відеокодер 20 може кодувати коефіцієнти перетворення піднаборів (1, 1) і (0, 2) і визначити, чи включають в себе ці піднабори які-небудь ненульові коефіцієнти. Якщо відеокодер 20 визначає, що будь-який з піднаборів (1, 2) і (0, 3) включає в себе які-небудь ненульові коефіцієнти, відеокодер може генерувати елемент синтаксису зі значенням один (1), асоційованим з цим піднабором. Інакше, якщо відеокодер 20 визначає, що будь-який з піднаборів (1, 1) і (0, 2) не включає в себе відмінний від нуля коефіцієнт, відеокодер 20 може також генерувати елемент синтаксису зі значенням нуль (0), асоційованим з цим піднабором. Відеокодер 20 може також генерувати елемент синтаксису зі значенням (0) або (1), асоційованим з кожним з піднаборів (0, 0), (0, 1), (1, 0), (2, 0), (2, 1) і (3, 0).

Елементи синтаксису, які вказують, чи мають піднабори (0, 0), (0, 1), (1, 0), (1, 1), (0, 2), (2, 0), (2, 1) і (3, 0) ненульові коефіцієнти (наприклад, окремі елементи синтаксису, визначені для кожного відповідного піднабору), можуть бути виведені відеокодером 20 як частина ентропійно кодованого бітового потоку і зчитані декодером 30. Як частина відновлення одиниці 410 рівня листя, декодер 30 може розділити дані, що представляють блок 410, на множину субблоків. Для кожного субблока декодер 30 може визначити, чи включає в себе ентропійно кодований бітовий потік елемент синтаксису, який вказує, чи включає в себе згаданий піднабір ненульові коефіцієнти. Якщо декодер 30 зчитує субблок, який не включає в себе такий елемент синтаксису, декодер 30 декодує коефіцієнти піднабору. Однак, якщо декодер 30 зчитує такий елемент синтаксису, асоційований з піднабором, декодер 30 може використовувати значення елемента синтаксису, щоб визначити, чи декодувати коефіцієнти піднабору. Наприклад, згідно з прикладом, сформульованим вище, якщо декодер 30 зчитує значення елемента синтаксису одиниця (1) для піднабору (1, 1), декодер може декодувати цей піднабір. Однак, якщо декодер 30 зчитує значення елемента синтаксису нуль (0) для піднабору (1, 1), декодер 30 не декодує коефіцієнти перетворення цього субблока.

Фіг. 5 є блок-схемою, яка ілюструє один приклад способу кодування відеоданих, сумісних з одним або більше аспектами даного розкриття. Спосіб згідно з фіг. 5 описаний як виконуваний кодером 20, зображеним на фіг. 1 і 2, однак будь-який інший пристрій може використовуватися, щоб виконати спосіб згідно з фіг. 5.

Як показано на фіг. 5, кодер 20 ділить листовий блок відеоданих на множину піднаборів, де кожний включає в себе множинні коефіцієнти перетворення (501). Як один приклад, множина піднаборів може містити піднабори прямокутної форми, як зображено в прикладі на фіг. 4, або будь-якої іншої геометричної форми або шаблону. Одиниця рівня листя відеоданих може містити одиницю кодування найбільш низького рівня зі стандарту кодування відео, таку як нерозділений дочірній вузол структури квадродререва, як описано вище.

Як також показано на фіг. 5, кодер 20 генерує для щонайменше одного з множини піднаборів елемент синтаксису, який вказує, чи включає в себе згаданий піднабір ненульові коефіцієнти (502). У деяких прикладах кодер 20 може генерувати такий елемент синтаксису, асоційований з кожним з множини піднаборів. У інших прикладах кодер 20 може вибірково визначити, чи генерувати такий елемент синтаксису для кожного з піднаборів. Наприклад, кодер 20 може вибірково визначити, чи генерувати елемент синтаксису, на основі того, чи поліпшить генерування елемента синтаксису ефективність кодування і/або стиснення. Службові витрати, асоційовані з указанням, що конкретний піднабір фактично включає в себе ненульові коефіцієнти, може погіршити стиснення в деяких випадках, і у цих випадках такої зайвої сигналізації можна уникнути, і кодер і декодер можуть кодувати/декодувати піднабір без якого-небудь визначення того, чи включає в себе згаданий піднабір ненульові коефіцієнти.

У деяких прикладах кодер 20 може визначити, чи генерувати елемент синтаксису, що поліпшить ефективність кодування, на основі кількості потенційних ненульових коефіцієнтів в піднаборі, як описано більш детально нижче відносно фіг. 6, або на основі середньої кількості

коефіцієнтів кожного піднабору, як описано більш детально нижче відносно фіг. 7. Якщо кодер 20 генерує елемент синтаксису, асоційований з піднабором, кодер 20 може вивести елемент синтаксису як частину ентропійно кодованого бітового потоку. Ентропійно кодований бітовий потік може бути зчитаний декодером 30, як зображено на фіг. 3, і елемент синтаксису може використовуватися декодером 30, щоб декодувати одиницю рівня листя, як описано більш детально відносно фіг. 8 нижче.

Фіг. 6 є блок-схемою, яка ілюструє один приклад способу кодування відеоданих, сумісного з одним або більше аспектами даного розкриття. Спосіб згідно з фіг. 6 описаний як виконуваний кодером 20, зображеним на фіг. 2, однак будь-який пристрій може використовуватися, щоб виконати способи згідно з фіг. 6. Згідно з прикладом на фіг. 6 кодер 20 визначає, чи поліпшить генерування елемента синтаксису, асоційованого з піднабором коефіцієнтів перетворення, ефективність кодування, на основі кількості потенційних ненульових коефіцієнтів для цього піднабору.

Як показано в прикладі на фіг. 6, кодер 20 ділить одиницю рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення (601). Як один приклад, множина піднаборів може містити піднабори прямокутної форми, як зображено в прикладі на фіг. 4, або будь-якої іншої геометричної форми або шаблону. Одиниця рівня листя відеоданих може містити одиницю кодування найбільш низького рівня для стандарту кодування відео, таку як нерозділений дочірній вузол структури квадродререва, як описано вище.

Згідно з прикладом на фіг. 6 відеокдер 20 визначає, чи генерувати елемент синтаксису, асоційований з кожним з множини піднаборів, на основі кількості потенційних ненульових коефіцієнтів кожного піднабору. Наприклад, як показано на фіг. 6, кодер 20 визначає для кожного піднабору кількість потенційних ненульових коефіцієнтів піднабору (602). Щоб визначити кількість потенційних ненульових коефіцієнтів, кодер 20 може визначити, скільки коефіцієнтів кожного піднабору мають позицію перед останнім ненульовим коефіцієнтом одиниці рівня листя, як описано вище відносно прикладу на фіг. 4.

Як також показано на фіг. 6, кодер 20 може порівняти визначену кількість потенційних ненульових коефіцієнтів з пороговим значенням $thNoCoeff$ (603). Як також показано на фіг. 6, якщо визначена кількість потенційних ненульових коефіцієнтів більше, ніж порогове значення $thNoCoeff$, кодер 20 генерує елемент синтаксису, асоційований з піднабором (604). Однак, якщо визначена кількість потенційних ненульових коефіцієнтів менше ніж або дорівнює пороговому значенню $thNoCoeff$, кодер 20 не генерує елемент синтаксису, асоційований з піднабором (605). Згідно з цими прикладами, якщо кількість потенційних ненульових коефіцієнтів менше ніж або дорівнює пороговому значенню $thNoCoeff$, можна передбачити, що кодування/декодування піднабору повинно бути виконане. У цьому способі елемент синтаксису не генерується для тих піднаборів, які включають в себе дуже мало потенційних ненульових коефіцієнтів, які можуть зменшити кількість бітів інформації, повідомленої кодером 20, щоб представити одиницю рівня листя відеоданих. Це має місце тому, що, коли піднабір містить дуже мало потенційних ненульових коефіцієнтів, кількість бітів, необхідних для того, щоб сигналізувати ці ненульові коефіцієнти, є відносно малою, що в свою чергу приводить до відносно великих службових витрат в сигналізації елемента синтаксису, яка вказує, чи має піднабір ненульові коефіцієнти. У результаті краще не сигналізувати елемент синтаксису і замість цього кодувати коефіцієнти піднабору безпосередньо, коли піднабір містить дуже мало потенційних ненульових коефіцієнтів (тобто менше ніж $thNoCoeff$).

Фіг. 7 є блок-схемою, яка ілюструє інший приклад способу кодування відеоданих, сумісних з одним або більше аспектами даного розкриття. Спосіб згідно з фіг. 7 описаний як виконуваний кодером 20, зображеним на фіг. 2, однак будь-який пристрій може використовуватися, щоб виконати способи згідно з фіг. 7. Згідно з прикладом на фіг. 7, кодер 20 визначає, чи поліпшить генерування елемента синтаксису ефективність кодування, на основі середньої кількості ненульових коефіцієнтів для кожного піднабору.

Як показано на фіг. 7, кодер 20 ділить одиницю рівня листя відеоданих на множину піднаборів коефіцієнтів перетворення (701). Як також показано на фіг. 7, для кожного піднабору кодер 20 визначає середню кількість ненульових коефіцієнтів для піднабору. Для кодера 20 може бути сконфігуровано збирати статистики, які вказують, як часто коефіцієнти в позиціях в межах кожного піднабору раніше закодованих одиниць рівня листя відеоданих включають в себе ненульові коефіцієнти. Наприклад, кодер 20 може підтримувати один або більше лічильників, які підраховують, коли блоки рівня листя кодуються, як часто коефіцієнти в межах кожного піднабору є ненульовими і/або скільки коефіцієнтів кожного піднабору є ненульовими. Згідно з цим прикладом, коли кодер 20 кодує нову одиницю рівня листя відеоданих, кодер 20 може одержати доступ до такого лічильника, щоб визначити середню кількість ненульових

коефіцієнтів для кожного піднабору. У деяких прикладах лічильник може бути скинений для кожного піднабору (наприклад, ініціалізований) періодично, наприклад з кожним відеокадром або вирізкою.

Як також показано на фіг. 7, кодер 20 порівнює визначену середню кількість ненульових коефіцієнтів з порогом $th_{AvgCoeff}$ (703). Як також показано на фіг. 7, якщо визначена середня кількість ненульових коефіцієнтів менше, ніж поріг $th_{AvgCoeff}$, кодер 20 генерує елемент синтаксису, асоційований з піднабором (704). Однак, якщо визначена середня кількість ненульових коефіцієнтів більше ніж або дорівнює порогу, кодер 20 не генерує елемент синтаксису, асоційований з піднабором (705).

Як описано вище, кодер 20 може генерувати ентропійно кодований бітовий потік, який включає в себе щонайменше один елемент синтаксису, що вказує, чи включає в себе піднабір коефіцієнтів перетворення які-небудь ненульові коефіцієнти. Кодер 20 може визначити, чи генерувати елемент синтаксису, на основі кількості потенційних ненульових коефіцієнтів піднабору, як показано на фіг. 6, або на основі середньої кількості ненульових коефіцієнтів, як показано на фіг. 7. Декодер 30 може зчитувати ентропійно кодований бітовий потік, включаючи щонайменше один елемент синтаксису, і використовувати цей щонайменше один елемент синтаксису, щоб декодувати ентропійно кодований бітовий потік.

Фіг. 8 є блок-схемою, яка ілюструє один приклад способу, який може бути виконаний декодером, сумісним з одним або більше аспектами даного розкриття. Спосіб згідно з фіг. 8 описаний як виконуваний декодером 30, зображеним на фіг. 3, однак будь-який пристрій може використовуватися, щоб виконати спосіб згідно з фіг. 8.

Як показано на фіг. 8, декодер 30 може розділити одиницю рівня листя відеоданих на множину піднаборів, де кожний включає в себе множинні коефіцієнти перетворення (801). Як один приклад, множина піднаборів може містити піднабори прямокутної форми, як зображено в прикладі на фіг. 4, або будь-якої іншої геометричної форми або шаблону. Одиниця рівня листя відеоданих може містити одиницю кодування найбільш низького рівня для стандарту кодування відео, таку як нерозділений дочірній вузол структури квадродреву, як описано вище. Одиниця рівня листя даних представлена ентропійно кодованим бітовим потоком, який включає в себе щонайменше один елемент синтаксису, що вказує, чи включає в себе піднабір з множини піднаборів які-небудь ненульові коефіцієнти.

Як також показано на фіг. 8, для кожного з піднаборів декодер 30 визначає, чи включає в себе ентропійно кодований бітовий потік елемент синтаксису, асоційований з піднабором (802). Як також показано на фіг. 8, якщо піднабір не включає в себе такий асоційований елемент синтаксису, декодер 30 декодує піднабір (наприклад, коефіцієнти перетворення піднабору) (803). Як також показано на фіг. 8, якщо піднабір включає в себе такий асоційований елемент синтаксису, декодер 30 може використовувати асоційований елемент синтаксису, щоб визначити, чи декодувати піднабір (804). Наприклад, якщо елемент синтаксису має перше значення (наприклад, один (1)), декодер 30 декодує піднабір. Однак, якщо елемент синтаксису має друге значення (наприклад, нуль (0)), декодер 30 не декодує піднабір.

У одному або більше прикладах функції, описані тут, можуть бути реалізовані щонайменше частково в апаратному забезпеченні, такому як спеціалізовані компоненти апаратного забезпечення або процесор. Більш широко, способи можуть бути реалізовані в апаратному забезпеченні, процесорах, програмному забезпеченні, програмно-апаратних засобах або будь-якій їх комбінації. Якщо реалізовані в програмному забезпеченні, функції можуть бути збережені на або передані як одна або більше інструкцій або код по зчитуваному комп'ютером носію і виконані оснований на апаратному забезпеченні блоком обробки. Зчитуваний комп'ютером носій може включати в себе зчитуваний комп'ютером носій даних, який відповідає матеріальному носію, такому як запам'ятовуючі носії даних, або комунікаційні носії, що включають в себе будь-який носій, який полегшує передачу комп'ютерної програми від одного місця до іншого, наприклад, згідно з протоколом зв'язку. Таким чином зчитуваний комп'ютером носій загалом може відповідати (1) матеріальному зчитуваному комп'ютером носію даних, який є постійним, або (2) комунікаційному носію, такому як сигнал або несуча. Запам'ятовуючі носії даних можуть бути будь-яким доступним носієм, до якого можуть одержати доступ один або більше комп'ютерів або один або більше процесорів, щоб витягнути інструкції, код і/або структури даних для реалізації способів, описаних в даному розкритті. Комп'ютерний програмний продукт може включати в себе зчитуваний комп'ютером носій.

За допомогою прикладу, а не обмеження, такі зчитувані комп'ютером носії даних можуть містити RAM, ROM, EEPROM, CD-ROM або інший оптичний дисковий запам'ятовуючий пристрій, магнітний дисковий запам'ятовуючий пристрій або інші магнітні пристрої зберігання, флеш-пам'ять або будь-який інший носій, який може використовуватися, щоб зберегти бажаний

програмний код в формі інструкцій або структур даних і до якого може одержати доступ комп'ютер. Також, будь-яке з'єднання належно називають зчитуванням комп'ютером носієм. Наприклад, якщо інструкції передані від веб-сайта, сервера або іншого віддаленого джерела, використовуючи коаксіальний кабель, волоконно-оптичний кабель, виту пару, цифрову абонентську лінію (DSL) або бездротові технології, такі як інфрачервона, радіо- і мікрохвильова, то ці коаксіальний кабель, волоконно-оптичний кабель, вита пара, DSL або бездротові технології, такі як інфрачервона, радіо- і мікрохвильова, включені у визначення носія. Треба мати на увазі, однак, що зчитувані комп'ютером носії даних і запам'ятовуючі носії даних не включають в себе з'єднання, несучі, сигнали або інші тимчасові носії, але замість цього направлені на постійні матеріальні носії даних. Диск і диск, як використовується тут, включають в себе компакт-диск (CD), лазерний диск, оптичний диск, цифровий універсальний диск (DVD), дискету і диск blu-ray, де диски (disks) звичайно відтворюють дані магнітним чином, в той час як диски (discs) відтворюють дані оптично за допомогою лазерів. Комбінації вищезазначеного повинні також бути включені в рамки зчитуваного комп'ютером носія.

Інструкції можуть бути виконані одним або більше процесорами, такими як один або більше цифрових сигнальних процесорів (DSPs), мікропроцесори загального призначення, спеціалізовані інтегральні схеми (ASIC), програмовані користувачем логічні матриці (FPGA) або інші еквівалентні інтегральні або дискретні логічні схеми. Відповідно, термін "процесор", як використовується тут, може стосуватися будь-якої відомої структури або будь-якої іншої структури, придатної для реалізації способів, описаних тут. Також, в деяких аспектах функціональні можливості, описані тут, можуть бути надані в межах спеціалізованого апаратного забезпечення і/або програмних модулів, сконфігурованих для кодування і декодування або вбудовані в об'єднаний кодек. Також, способи могли бути повністю реалізовані в одній або більше схемах або логічних елементах.

Способи даного розкриття можуть бути реалізовані в широкій різноманітності пристроїв або апаратів, включаючи бездротову телефонну трубку, інтегральну схему (IC) або набір IC (наприклад, мікропроцесорний набір). Різні компоненти, модулі або блоки описані в даному розкритті, щоб підкреслити функціональні аспекти пристроїв, сконфігурованих, щоб виконувати розкриті способи, але не обов'язково вимагати реалізації різними блоками апаратного забезпечення. Замість цього, як описано вище, різні блоки можуть бути об'єднані в блоці апаратного забезпечення кодека або надані колекцією взаємодіючих блоків апаратного забезпечення, включаючи один або більше процесорів, як описано вище, в сполученні з придатним програмним забезпеченням і/або програмно-апаратними засобами.

Були описані різні приклади. Ці і інші приклади знаходяться в рамках нижченаведеної формули винаходу.

ФОРМУЛА ВИНАХОДУ

1. Спосіб кодування блока відеоданих, що містить:

розділення блока перетворення відеоданих на множину піднаборів коефіцієнтів перетворення; визначення для кожного відповідного піднабору коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, чи генерувати відповідний елемент синтаксису, який вказує, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення будь-які ненульові коефіцієнти; і

генерування відповідного елемента синтаксису для кожного з піднаборів коефіцієнтів перетворення зі згаданої множини піднаборів коефіцієнтів перетворення, для яких визначено генерувати цей елемент синтаксису.

2. Спосіб за п. 1, в якому визначення для відповідного піднабору коефіцієнтів перетворення, чи генерувати відповідний елемент синтаксису, включає визначення, чи поліпшить генерування відповідного елемента синтаксису ефективність кодування.

3. Спосіб за п. 1, в якому визначення, чи генерувати відповідний елемент синтаксису, включає порівняння кількості потенційних ненульових коефіцієнтів згаданого відповідного піднабору коефіцієнтів перетворення зі щонайменше одним порогом; і, якщо кількість потенційних ненульових коефіцієнтів згаданого відповідного піднабору коефіцієнтів перетворення більше, ніж поріг, генерування відповідного елемента синтаксису.

4. Спосіб за п. 3, який додатково включає:

визначення кількості потенційних ненульових коефіцієнтів на основі того, скільки коефіцієнтів згаданого відповідного піднабору коефіцієнтів перетворення мають позицію після останнього ненульового коефіцієнта блока перетворення.

5. Спосіб за п. 1, в якому визначення, чи генерувати відповідний елемент синтаксису, включає:

визначення для множини попередньо закодованих блоків перетворення відеоданих середньої кількості ненульових коефіцієнтів, включених до згаданого відповідного піднабору коефіцієнтів перетворення;

5 порівняння середньої кількості ненульових коефіцієнтів, включених до згаданого відповідного піднабору коефіцієнтів перетворення, з порогом; і,
якщо середня кількість ненульових коефіцієнтів менше, ніж поріг, генерування відповідного елемента синтаксису.

6. Спосіб за п. 1, в якому визначення, чи генерувати відповідний елемент синтаксису, включає визначення, чи включає в себе відповідний піднабір коефіцієнтів перетворення останній
10 ненульовий коефіцієнт для блока перетворення відеоданих, при цьому відповідний елемент синтаксису, який вказує, чи включає в себе відповідний піднабір коефіцієнтів перетворення будь-які ненульові коефіцієнти, не генерується, якщо відповідний піднабір коефіцієнтів перетворення включає в себе останній ненульовий коефіцієнт.

7. Пристрій, сконфігурований, щоб кодувати щонайменше один блок перетворення відеоданих,
15 що містить:

процесор, сконфігурований, щоб:

розділяти блок перетворення відеоданих на множину піднаборів коефіцієнтів перетворення;

20 визначати для кожного відповідного піднабору коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, чи генерувати відповідний елемент синтаксису, який вказує, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення будь-які ненульові коефіцієнти; і

генерувати відповідний елемент синтаксису для кожного з піднаборів коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, для яких визначено генерувати цей елемент синтаксису.

25 8. Пристрій за п. 7, в якому процесор також сконфігурований, щоб визначати для відповідного піднабору коефіцієнтів перетворення, чи генерувати відповідний елемент синтаксису, на основі того, чи поліпшить генерування відповідного елемента синтаксису ефективність кодування.

9. Пристрій за п. 7, в якому процесор також сконфігурований, щоб:

30 визначати, чи генерувати відповідний елемент синтаксису, на основі порівняння кількості потенційних ненульових коефіцієнтів відповідного піднабору коефіцієнтів перетворення щонайменше з одним порогом; і,

якщо кількість потенційних ненульових коефіцієнтів відповідного піднабору коефіцієнтів перетворення більша, ніж поріг, генерувати відповідний елемент синтаксису.

35 10. Пристрій за п. 9, в якому процесор також сконфігурований, щоб визначати кількість потенційних ненульових коефіцієнтів на основі того, скільки коефіцієнтів відповідного піднабору коефіцієнтів перетворення мають позицію після останнього ненульового коефіцієнта блока перетворення.

11. Пристрій за п. 7, в якому процесор також сконфігурований, щоб:

40 визначати, для множини попередньо закодованих блоків перетворення відеоданих, середню кількість ненульових коефіцієнтів, включених до відповідного піднабору коефіцієнтів перетворення;

порівнювати середню кількість ненульових коефіцієнтів, включених до відповідного піднабору коефіцієнтів перетворення, з порогом; і,

45 якщо середня кількість ненульових коефіцієнтів менша, ніж поріг, генерувати відповідний елемент синтаксису.

12. Пристрій за п. 7, в якому процесор також сконфігурований, щоб визначати, чи включає відповідний піднабір коефіцієнтів перетворення в себе останній ненульовий коефіцієнт для блока перетворення відеоданих, в якому відповідний елемент синтаксису, який вказує, чи
50 включає відповідний піднабір коефіцієнтів перетворення в себе будь-які ненульові коефіцієнти, не генерується, якщо відповідний піднабір коефіцієнтів перетворення включає в себе останній ненульовий коефіцієнт.

13. Зчитуваний комп'ютером носій даних, який зберігає інструкції, які, коли виконуються, змушують обчислювальний пристрій:

розділяти блок перетворення відеоданих на множину піднаборів коефіцієнтів перетворення;

55 визначати для кожного відповідного піднабору коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, чи генерувати відповідний елемент синтаксису, який вказує, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення будь-які ненульові коефіцієнти; і

генерувати відповідний елемент синтаксису для кожного з піднаборів коефіцієнтів перетворення зі згаданої множини піднаборів коефіцієнтів перетворення, для яких визначено генерувати цей елемент синтаксису.

14. Зчитуваний комп'ютером носій даних за п. 13, в якому інструкції також змушують обчислювальний пристрій:

визначати, для відповідного піднабору коефіцієнтів перетворення, чи генерувати відповідний елемент синтаксису, на основі того, чи поліпшить генерування відповідного елемента синтаксису ефективність кодування.

15. Зчитуваний комп'ютером носій даних за п. 13, в якому інструкції також змушують обчислювальний пристрій:

визначати, чи генерувати відповідний елемент синтаксису, на основі порівняння кількості потенційних ненульових коефіцієнтів відповідного піднабору коефіцієнтів перетворення зі щонайменше одним порогом; і,

якщо кількість потенційних ненульових коефіцієнтів відповідного піднабору коефіцієнтів перетворення більша, ніж поріг, генерувати відповідний елемент синтаксису.

16. Зчитуваний комп'ютером носій даних за п. 15, в якому інструкції також змушують обчислювальний пристрій визначати кількість потенційних ненульових коефіцієнтів, на основі того, скільки коефіцієнтів відповідного піднабору коефіцієнтів перетворення мають позицію після останнього ненульового коефіцієнта блока перетворення.

17. Зчитуваний комп'ютером носій даних за п. 13, в якому інструкції також змушують обчислювальний пристрій:

визначати, для множини попередньо закодованих блоків перетворення відеоданих, середню кількість ненульових коефіцієнтів, включених до відповідного піднабору коефіцієнтів перетворення;

порівнювати середню кількість ненульових коефіцієнтів, включених до відповідного піднабору коефіцієнтів перетворення, з порогом; і,

якщо середня кількість ненульових коефіцієнтів менше, ніж поріг, генерувати відповідний елемент синтаксису.

18. Зчитуваний комп'ютером носій даних за п. 13, в якому інструкції також змушують обчислювальний пристрій визначати, чи включає відповідний піднабір коефіцієнтів перетворення в себе останній ненульовий коефіцієнт для блока перетворення відеоданих, в якому відповідний елемент синтаксису, який вказує, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення будь-які ненульові коефіцієнти, не генерується, якщо відповідний піднабір коефіцієнтів перетворення включає в себе останній ненульовий коефіцієнт.

19. Пристрій, сконфігурований, щоб кодувати щонайменше один блок перетворення відеоданих, що містить:

засіб для розділення блока перетворення відеоданих на множину піднаборів коефіцієнтів перетворення;

засіб для визначення для кожного відповідного піднабору коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, чи генерувати відповідний елемент синтаксису, який вказує, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення будь-які ненульові коефіцієнти; і

засіб для генерування відповідного елемента синтаксису для кожного з піднаборів коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, для яких визначено генерувати згаданий елемент синтаксису.

20. Пристрій за п. 19, який додатково містить засіб для визначення, для згаданого відповідного піднабору коефіцієнтів перетворення, чи генерувати відповідний елемент синтаксису, на основі того, чи поліпшить генерування відповідного елемента синтаксису ефективність кодування.

21. Пристрій за п. 19, який додатково містить:

засіб для визначення, чи генерувати відповідний елемент синтаксису, на основі порівняння кількості потенційних ненульових коефіцієнтів відповідного піднабору коефіцієнтів перетворення зі щонайменше одним порогом; і

засіб для, якщо кількість потенційних ненульових коефіцієнтів відповідного піднабору коефіцієнтів перетворення більша, ніж поріг, генерування відповідного елемента синтаксису.

22. Пристрій за п. 21, який додатково містить засіб для визначення кількості потенційних ненульових коефіцієнтів, на основі того, скільки коефіцієнтів відповідного піднабору коефіцієнтів перетворення мають позицію після останнього ненульового коефіцієнта блока перетворення.

23. Пристрій за п. 19, в якому засіб для визначення, чи генерувати відповідний елемент синтаксису, містить:

засіб для визначення, для множини попередньо закодованих блоків перетворення відеоданих, середньої кількості ненульових коефіцієнтів, включених до відповідного піднабору коефіцієнтів перетворення;

5 засіб для того, щоб порівняти середню кількість ненульових коефіцієнтів, включених до відповідного піднабору коефіцієнтів перетворення, з порогом; і

засіб для, якщо середня кількість ненульових коефіцієнтів менша, ніж поріг, генерування відповідного елемента синтаксису.

10 24. Пристрій за п. 19, який додатково містить засіб для визначення, чи включає відповідний піднабір коефіцієнтів перетворення в себе останній ненульовий коефіцієнт для блока перетворення відеоданих, в якому відповідний елемент синтаксису, який вказує, чи включає відповідний піднабір коефіцієнтів перетворення в себе будь-які ненульові коефіцієнти, не генерується, якщо відповідний піднабір коефіцієнтів перетворення включає в себе останній ненульовий коефіцієнт.

25. Спосіб декодування блока відеоданих, що включає:

15 розділення блока перетворення відеоданих на множину піднаборів коефіцієнтів перетворення; і визначення, для кожного відповідного піднабору коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, чи включає в себе згаданий відповідний піднабір відповідний елемент синтаксису, який вказує, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення будь-які ненульові коефіцієнти; і

20 визначення, для кожного відповідного піднабору коефіцієнтів перетворення, чи декодувати відповідний піднабір коефіцієнтів перетворення, на основі відповідного елемента синтаксису.

26. Спосіб за п. 25, який додатково включає, якщо відповідний піднабір коефіцієнтів перетворення не включає в себе відповідний елемент синтаксису, декодування відповідного піднабору коефіцієнтів перетворення.

25 27. Спосіб за п. 26, який додатково включає, якщо відповідний піднабір коефіцієнтів перетворення даних включає в себе відповідний елемент синтаксису, визначення, чи декодувати відповідний піднабір коефіцієнтів перетворення, на основі вказівки.

28. Спосіб за п. 27, який додатково включає:

30 якщо відповідний елемент синтаксису, асоційований з відповідним піднабором коефіцієнтів перетворення, має перше значення, декодування відповідного піднабору коефіцієнтів перетворення; і,

якщо відповідний елемент синтаксису має друге значення, відмінне від першого значення, недекодування відповідного піднабору коефіцієнтів перетворення.

29. Спосіб за п. 25, який додатково включає:

35 якщо відповідний піднабір коефіцієнтів перетворення не включає в себе відповідний елемент синтаксису, визначення, чи включає відповідний піднабір коефіцієнтів перетворення в себе окремий елемент синтаксису, який вказує, що останній ненульовий коефіцієнт включений у відповідний піднабір коефіцієнтів перетворення; і

40 декодування відповідного піднабору коефіцієнтів перетворення, якщо відповідний піднабір коефіцієнтів перетворення включає в себе окремий елемент синтаксису, який вказує, що останній ненульовий коефіцієнт включений у відповідний піднабір коефіцієнтів перетворення.

30. Пристрій, сконфігурований, щоб декодувати блок відеоданих, що містить процесор, сконфігурований, щоб:

45 розділяти блок перетворення відеоданих на множину піднаборів коефіцієнтів перетворення;

визначати, для кожного відповідного піднабору коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, чи включає в себе згаданий відповідний піднабір відповідний елемент синтаксису, який вказує, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення будь-які ненульові коефіцієнти; і

50 визначати, для кожного відповідного піднабору коефіцієнтів перетворення, чи декодувати відповідний піднабір коефіцієнтів перетворення, на основі відповідного елемента синтаксису.

31. Пристрій за п. 30, в якому процесор також сконфігурований, щоб, якщо відповідний піднабір коефіцієнтів перетворення не включає в себе відповідний елемент синтаксису, декодувати відповідний піднабір коефіцієнтів перетворення.

55 32. Пристрій за п. 31, в якому процесор також сконфігурований, щоб, якщо відповідний піднабір коефіцієнтів перетворення включає в себе відповідний елемент синтаксису, визначати, чи декодувати відповідний піднабір коефіцієнтів перетворення, на основі вказівки.

33. Пристрій за п. 32, в якому процесор також сконфігурований, щоб:

60 якщо відповідний елемент синтаксису, асоційований з відповідним піднабором коефіцієнтів перетворення, має перше значення, декодувати відповідний піднабір коефіцієнтів перетворення; і,

якщо відповідний елемент синтаксису має друге значення, відмінне від першого значення, не декодувати відповідний піднабір коефіцієнтів перетворення.

34. Пристрій за п. 30, в якому процесор також сконфігурований, щоб:

5 якщо відповідний піднабір коефіцієнтів перетворення не включає в себе відповідний елемент синтаксису, визначати, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення окремий елемент синтаксису, який вказує, що останній ненульовий коефіцієнт включений у відповідний піднабір коефіцієнтів перетворення; і

10 декодувати відповідний піднабір коефіцієнтів перетворення, якщо відповідний піднабір коефіцієнтів перетворення включає в себе окремий елемент синтаксису, який вказує, що останній ненульовий коефіцієнт включений у відповідний піднабір коефіцієнтів перетворення.

35. Зчитуваний комп'ютером носій даних, який зберігає інструкції, які, коли виконуються, змушують обчислювальний пристрій:

розділяти блок перетворення відеоданих на множину піднаборів коефіцієнтів перетворення;

15 визначати, для кожного відповідного піднабору коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення відповідний елемент синтаксису, який вказує, чи включає відповідний піднабір коефіцієнтів перетворення в себе будь-які ненульові коефіцієнти; і

20 визначати, для кожного відповідного піднабору коефіцієнтів перетворення, чи декодувати відповідний піднабір коефіцієнтів перетворення, на основі відповідного елемента синтаксису.

36. Зчитуваний комп'ютером носій даних за п. 35, в якому інструкції також змушують обчислювальний пристрій: якщо відповідний піднабір коефіцієнтів перетворення не включає в себе відповідний елемент синтаксису, декодувати відповідний піднабір коефіцієнтів перетворення.

25 37. Зчитуваний комп'ютером носій даних за п. 36, в якому інструкції також змушують обчислювальний пристрій, якщо відповідний піднабір коефіцієнтів перетворення включає в себе відповідний елемент синтаксису, визначати, чи декодувати відповідний піднабір коефіцієнтів перетворення, на основі вказівки.

38. Зчитуваний комп'ютером носій даних за п. 37, в якому інструкції також змушують обчислювальний пристрій:

30 якщо відповідний елемент синтаксису, асоційований з відповідним піднабором коефіцієнтів перетворення, має перше значення, декодувати відповідний піднабір коефіцієнтів перетворення; і,

якщо відповідний елемент синтаксису має друге значення, відмінне від першого значення, не декодувати відповідний піднабір коефіцієнтів перетворення.

35 39. Зчитуваний комп'ютером носій даних за п. 35, в якому інструкції також змушують обчислювальний пристрій:

40 якщо відповідний піднабір коефіцієнтів перетворення не включає в себе відповідний елемент синтаксису, визначати, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення окремий елемент синтаксису, який вказує, що останній ненульовий коефіцієнт включений у відповідний піднабір коефіцієнтів перетворення; і

декодувати відповідний піднабір коефіцієнтів перетворення, якщо відповідний піднабір коефіцієнтів перетворення включає в себе окремий елемент синтаксису, який вказує, що останній ненульовий коефіцієнт включений у відповідний піднабір коефіцієнтів перетворення.

40. Пристрій, сконфігурований, щоб декодувати блок відеоданих, що містить:

45 засіб для розділення блока перетворення відеоданих на множину піднаборів коефіцієнтів перетворення;

засіб для визначення, для кожного відповідного піднабору коефіцієнтів перетворення з множини піднаборів коефіцієнтів перетворення, чи включає в себе згаданий відповідний піднабір відповідний елемент синтаксису, який вказує, чи включає відповідний піднабір коефіцієнтів перетворення в себе будь-які ненульові коефіцієнти; і

50 засіб для визначення, для кожного відповідного піднабору коефіцієнтів перетворення, чи декодувати відповідний піднабір коефіцієнтів перетворення, на основі елемента синтаксису.

41. Пристрій за п. 40, який додатково містить засіб для, якщо відповідний піднабір коефіцієнтів перетворення не включає в себе відповідний елемент синтаксису, декодування відповідного піднабору коефіцієнтів перетворення.

55 42. Пристрій за п. 41, який додатково містить засіб для, якщо відповідний піднабір коефіцієнтів перетворення включає в себе відповідний елемент синтаксису, визначення декодувати відповідний піднабір коефіцієнтів перетворення, на основі вказівки.

43. Пристрій за п. 42, який додатково містить:

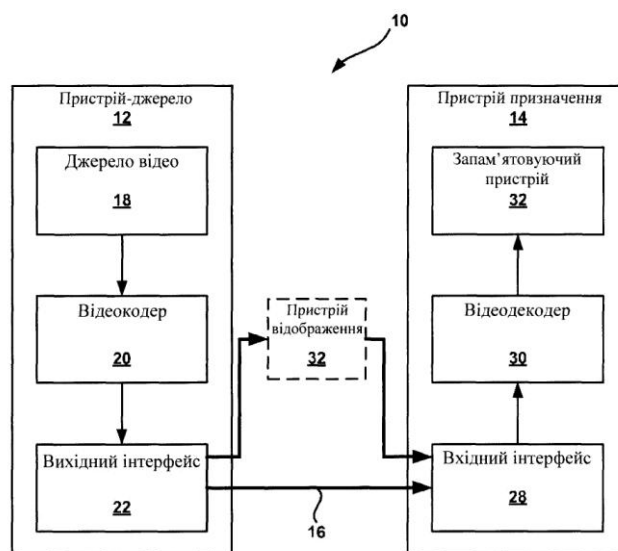
засіб для, якщо відповідний елемент синтаксису, асоційований з відповідним піднабором коефіцієнтів перетворення, має перше значення, декодування відповідного піднабору коефіцієнтів перетворення; і

засіб для, якщо відповідний елемент синтаксису має друге значення, відмінне від першого значення, недекодування відповідного піднабору коефіцієнтів перетворення.

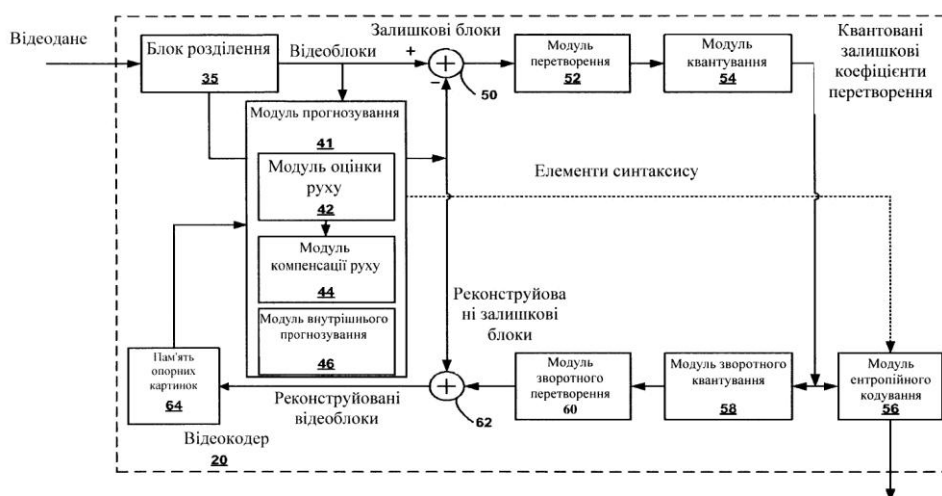
44. Пристрій за п. 40, який додатково містить:

якщо відповідний піднабір коефіцієнтів перетворення не включає в себе відповідний елемент синтаксису, засіб для визначення, чи включає в себе згаданий відповідний піднабір коефіцієнтів перетворення окремий елемент синтаксису, який вказує, що останній ненульовий коефіцієнт включений у відповідний піднабір коефіцієнтів перетворення; і

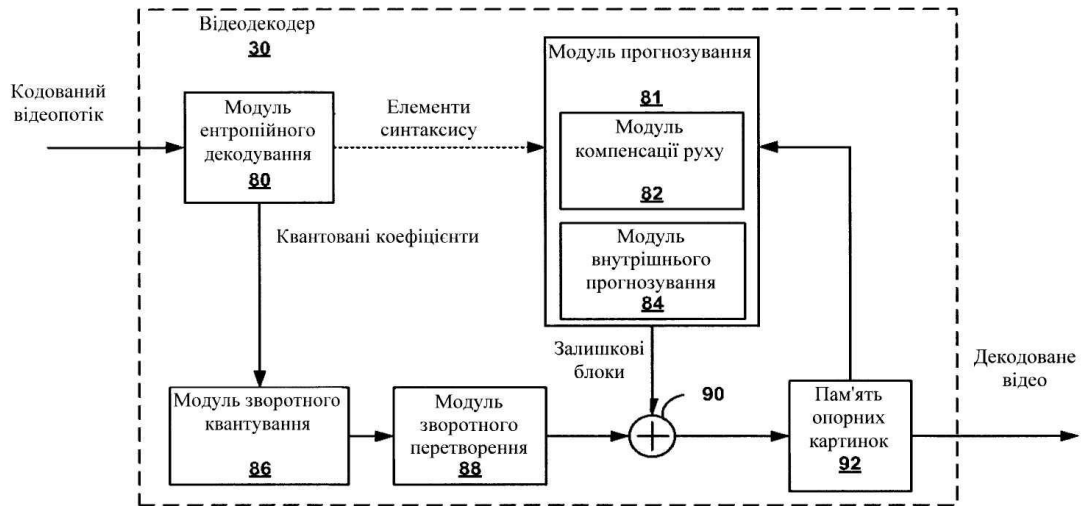
засіб для того, щоб декодувати відповідний піднабір коефіцієнтів перетворення, якщо відповідний піднабір коефіцієнтів перетворення включає в себе окремий елемент синтаксису, який вказує, що останній ненульовий коефіцієнт включений у відповідний піднабір коефіцієнтів перетворення.



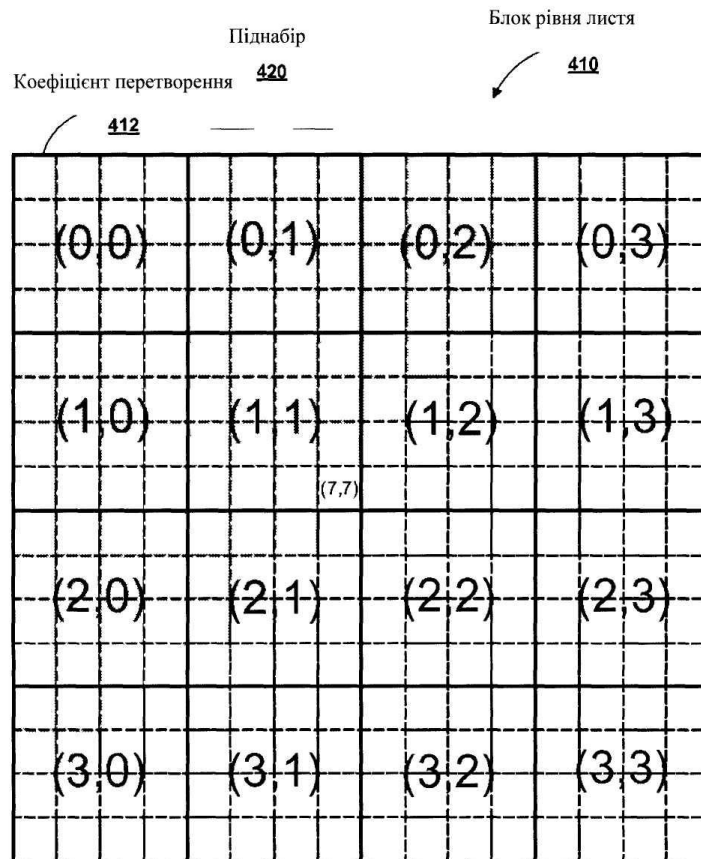
Фіг. 1



Фіг. 2



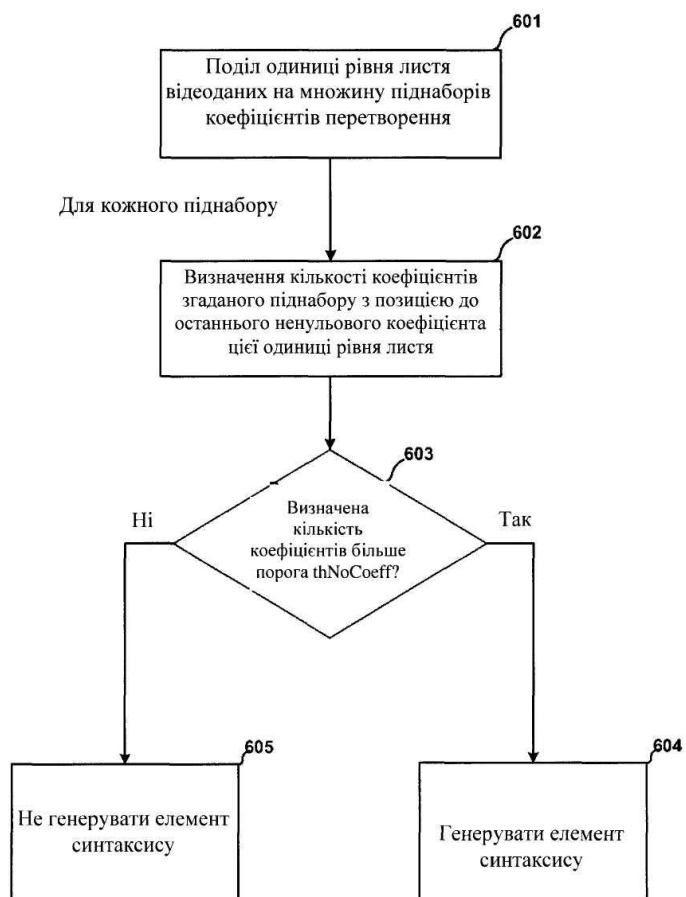
Фіг. 3



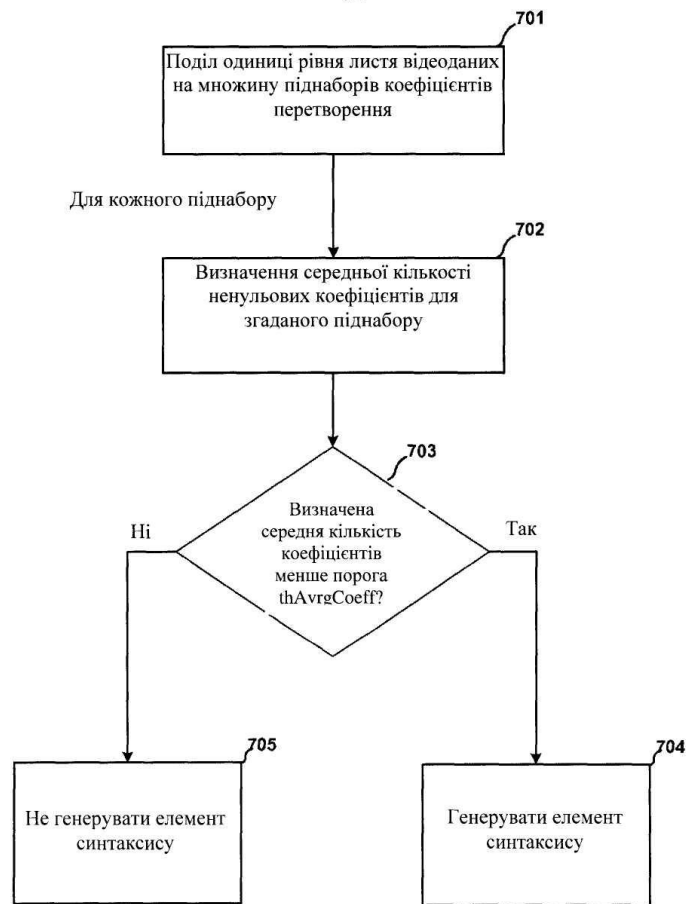
Фіг. 4



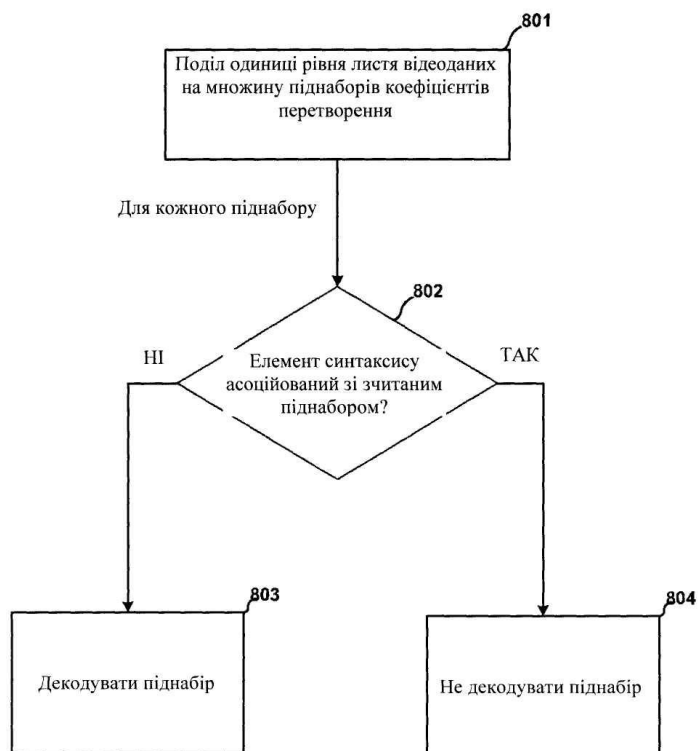
Фіг. 5



Фіг. 6



Фіг. 7



Фіг. 8

Комп'ютерна верстка В. Мацело

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