



УКРАЇНА

(19) **UA** (11) **109506** (13) **C2**
(51) МПК (2015.01)
H03M 7/40 (2006.01)
H04N 7/00

ДЕРЖАВНА СЛУЖБА
ІНТЕЛЕКТУАЛЬНОЇ
ВЛАСНОСТІ
УКРАЇНИ

(12) ОПИС ДО ПАТЕНТУ НА ВІНАХІД

(21) Номер заявки:	а 2014 06229	(73) Власник(и): КВЕЛКОММ ІНКОРПОРЕЙТЕД , Attn: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121-1714, United States of America (US)
(22) Дата подання заявки:	05.10.2012	
(24) Дата, з якої є чинними права на винахід:	25.08.2015	(74) Представник: Мошинська Ніна Миколаївна, реєстр. №115
(31) Номер попередньої заявки відповідно до Паризької конвенції:	61/557,325, 61/561,911, 13/645,330	(56) Перелік документів, взятих до уваги експертизою: Sze (BoG coordinator) V: "BoG report on context reduction for CABAC", 6. JCT-VC meeting; 97. MPEG meeting; 14-7-2011 - 22- 7-2011; Torino; (Joint Collaborative Team on Video Coding of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16); 21.07.2011 Nguyen T; Marpe D; Schwarz H; Wiegand (Fraunhofer HHI) T: "Modified binarization and coding of MVD for PIPE/CABAC Modified binarization and coding of MVD for PIPE/CABAC", 6. JCT-VC meeting; 97. MPEG meeting; 14-7-2011 - 22-7-2011; Torino; (Joint Collaborative Team on Video Coding of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16); 16.07.2011 Hisao Sasai; Takahiro Nishi (Panasonic): "CE11: Context size reduction for the significance map", 96. MPEG meeting; 21-3- 2011 - 25-3-2011; Geneva; (Motion Picture Expert Group or ISO/IEC JTC1/SC29/WG11); 18.03.2011 "Test Model under Consideration for High- Efficiency Video Coding", 92. MPEG meeting; 19-4-2010 - 23-4-2010; Dresden; (Motion Picture Expert Group or ISO/IEC JTC1/SC29/WG11), 20100728 - ISSN 0000- 0029; 28.07.2010
(32) Дата подання попередньої заявки відповідно до Паризької конвенції:	08.11.2011, 20.11.2011, 04.10.2012	
(33) Код держави- учасниці Паризької конвенції, до якої подано попередню заявку:	US, US, US	
(41) Публікація відомостей про заявку:	10.09.2014, Бюл.№ 17	
(46) Публікація відомостей про видачу патенту:	25.08.2015, Бюл.№ 16	
(86) Номер та дата подання міжнародної заявки, поданої відповідно до Договору РСТ	PCT/US2012/059100, 05.10.2012	
(72) Винахідник(и): Чіень Вей-Цзюнь (US), Соле Рохальс Хоель (US), Карчевіч Марта (US)		

(54) СКОРОЧЕННЯ КІЛЬКОСТІ КОНТЕКСТІВ ДЛЯ КОНТЕКСТНО-АДАПТИВНОГО БІНАРНОГО АРИФМЕТИЧНОГО КОДУВАННЯ**(57) Реферат:**

Запропоноване скорочення кількості бінаризацій і/або контекстів, що використовуються в контекстно-адаптивному бінарному арифметичному кодуванні (CABAC) для кодування відео. Зокрема, дане розкриття пропонує методи, які дозволяють знизити кількість контекстів, що використовуються в CABAC, на величину аж до 56. Спосіб кодування відеоданих включає кодування прапора кодованого блока кольоровості Cb для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC), причому кодування

UA 109506 C2

прапора кодованого блока кольоровості S_b включає використання набору контекстів, що містить один або більше контекстів як частину САВАС; і кодування прапора кодованого блока кольоровості S_r з використанням САВАС, причому кодування прапора кодованого блока кольоровості S_r включає використання того самого набору контекстів, що і для прапора кодованого блока кольоровості S_b , як частину САВАС. Воно вибирає контекст з одного або більше контекстів на основі глибини перетворення одиниці перетворення, асоційованої з блоком відеоданих.

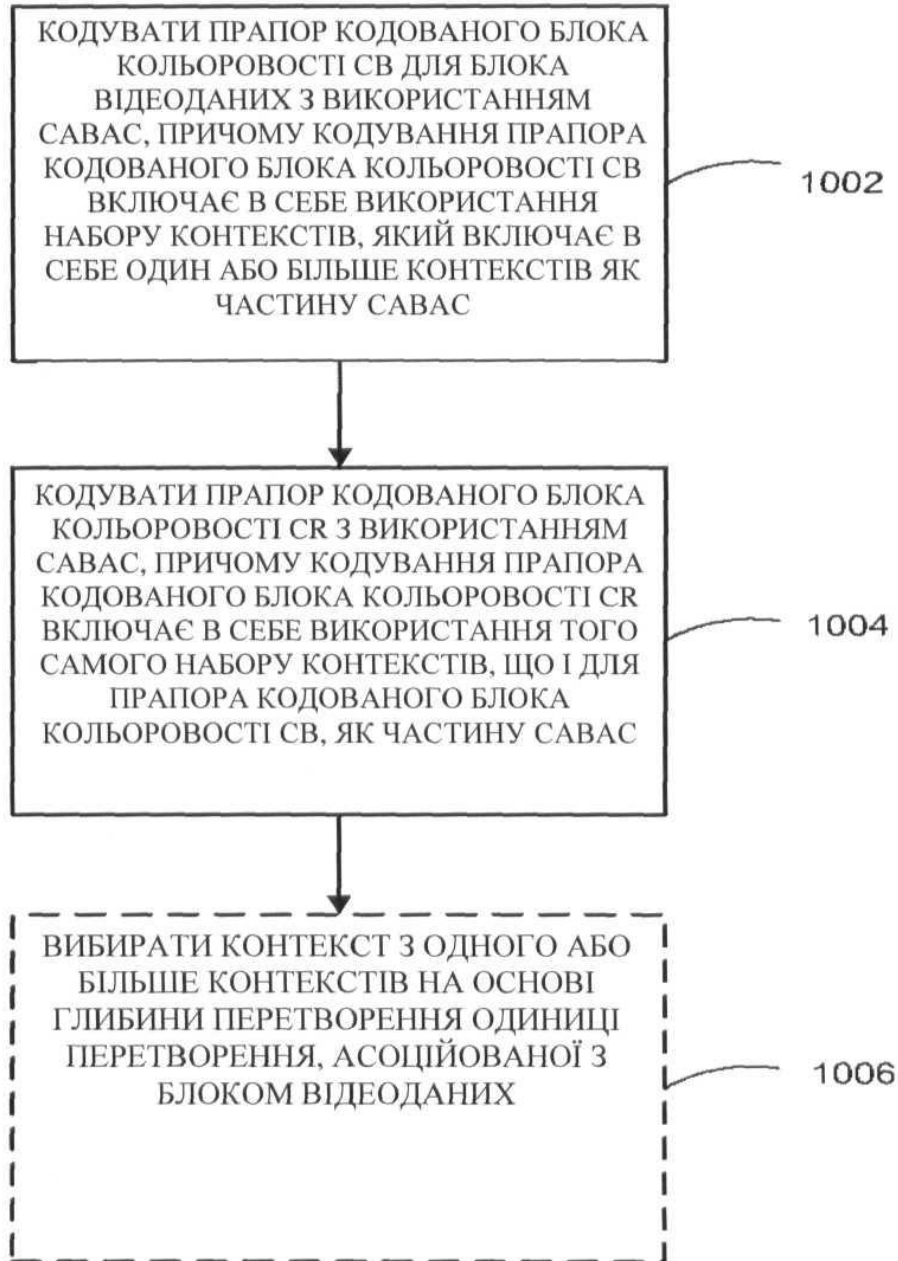


Fig. 10

Дана заявка заявляє пріоритет попередньої заявки США 61/557, 325, поданої 8 листопада 2011, і попередньої заявки США 61/561, 911, поданої 20 листопада 2011, обидві з яких включені в цей документ як посилання у всій їх повноті.

Галузь техніки

5 Даний винахід стосується кодування відео і, зокрема, контекстно-адаптивного бінарного арифметичного кодування (CABAC), що використовується в кодуванні відео.

Попередній рівень техніки

Можливості цифрового відео можуть бути включені в широкий діапазон пристроїв, включаючи цифрові телевізори, системи цифрового прямого мовлення, бездротові системи мовлення, персональні цифрові помічники (PDA), портативні або настільні комп'ютери, планшетні комп'ютери, електронні книги, цифрові камери, цифрові записуючі пристрої, цифрові медіаплеєри, ігрові відеопристрої, ігрові приставки, стільникові або супутникові радіотелефони, так звані "смартфони", пристрої відеотелеконференцій, пристрої потокового відео і тому подібне. Цифрові відеопристрої реалізують технології стиснення відео, такі як ті, що описані в стандартах, визначених MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, частина 10, Розширене кодування відео (AVC), стандарт Високоєфективного кодування відео (HEVC), що знаходиться в цей час в стадії розробки, і розширення таких стандартів. Відеопристрої можуть передавати, приймати, кодувати, декодувати і/або зберігати цифрову відеоінформацію більш ефективно шляхом реалізації таких методів стиснення відео.

20 Методи стиснення відео виконують просторове (всередині зображення, інтра-) прогнозування і/або часове (між зображеннями, інтер-) прогнозування для скорочення або усунення надмірності, властивої відеопослідовностями. Для блокового кодування, відео слайс (тобто відеозображення або частина відеозображення) може бути розділений на блоки відео, які також можуть згадуватися як блоки дерева, одиниці кодування (CU) і/або вузли кодування. 25 Блоки відео в інтра-кодованому (I) слайсі (сегменті) зображення кодуються з використанням просторового прогнозування відносно опорних вибірок у сусідніх блоках в тому самому зображенні. Блоки відео в інтер-кодованому (P або B) сегменті зображення можуть використовувати просторове прогнозування відносно опорних вибірок у сусідніх блоках в тому самому зображенні або часове прогнозування відносно опорних вибірок в інших опорних зображеннях. Зображення можуть згадуватися як кадри, і опорні зображення можуть згадуватися як опорні кадри.

Просторове або часове прогнозування призводить в результаті до блока прогнозування для блока, що підлягає кодуванню. Залишкові дані представляють піксельні різниці між вихідним блоком, що підлягає кодуванню, і блоком прогнозування. Інтер-кодований блок кодується 35 відповідно до вектора руху, який вказує на блок опорних вибірок, утворюючих блок прогнозування, і залишковими даними, які вказують різницю між кодованим блоком і блоком прогнозування. Інтра-кодований блок кодується відповідно до режиму інтра-кодування і залишкових даних. Для подальшого стиснення, залишкові дані можуть бути перетворені з піксельної ділянки в ділянку перетворення, призводячи в результаті до залишкових коефіцієнтів перетворення, які потім можуть бути квантовані. Квантовані коефіцієнти перетворення, спочатку впорядковані в двомірний масив, можуть скануватися, щоб сформувати одномірний вектор коефіцієнтів перетворення, і ентропійне кодування може бути застосоване для досягнення ще більшого стиснення.

Суть винаходу

45 Загалом, це розкриття описує методи для контекстно-адаптивного бінарного арифметичного кодування (CABAC) в процесі відеокодування. Зокрема, це розкриття пропонує скорочення кількості контекстів CABAC, що використовуються для одного або декількох синтаксичних елементів, необмежувальні приклади яких включають `pred_typ`, `merge_idx`, `inert_pred_flag`, `ref_idx_lx`, `cbf_cb`, `cbf_cr`, `coeff_abs_level_greater1_flag` та `coeff_abs_level_greater2_flag`. 50 Модифікації можуть скоротити до 56 контекстів з нехтувано малими змінами ефективності кодування. Скорочення контексту для синтаксичних елементів, що пропонуються, можуть бути використані окремо або в будь-якій комбінації.

В одному прикладі здійснення розкриття спосіб кодування відео може включати в себе визначення першого типу прогнозування для блока відеоданих у P слайсі (сегменті), 55 представлення першого типу прогнозування як синтаксичний елемент типу прогнозування P сегмента, визначення другого типу прогнозування для блока відеоданих у B сегменті, представлення другого типу прогнозування як синтаксичний елемент типу прогнозування B сегмента, визначення бінаризації P сегмента для синтаксичного елемента типу прогнозування P сегмента, визначення бінаризації B сегмента для синтаксичного елемента типу прогнозування B сегмента, при цьому синтаксичний елемент типу прогнозування P сегмента і синтаксичний

елемент типу прогнозування В сегмента визначаються з використанням тієї самої логіки бінаризації, і кодування відеоданих на основі бінаризації синтаксичного елемента типу прогнозування Р сегмента і синтаксичного елемента типу прогнозування В сегмента.

В іншому прикладі даного розкриття спосіб декодування відео може включати в себе відображення бінаризованого синтаксичного елемента типу прогнозування Р сегмента на тип прогнозування, використовуючи відображення бінаризації для блока відеоданих у Р сегменті, відображення бінаризованого синтаксичного елемента типу прогнозування В сегмента на тип прогнозування, використовуючи відображення бінаризації для блока відеоданих у В сегменті, і декодування відеоданих, основуючись на відображених типах прогнозування.

В іншому прикладі даного розкриття спосіб кодування відеоданих включає в себе визначення типу розділення для режиму прогнозування для блока відеоданих, кодування структурного елемента типу розділення синтаксичного елемента типу прогнозування для блока відеоданих з використанням CABAC з одним контекстом, причому цей один контекст є тим самим для будь-якого типу розділення, і кодування структурного елемента розміру розділення синтаксичного елемента типу прогнозування для блока відеоданих з використанням CABAC в режимі обходу.

В іншому прикладі даного розкриття спосіб декодування відеоданих містить прийом синтаксичного елемента типу прогнозування для блока відеоданих, які були кодовані з використанням CABAC, причому синтаксичний елемент типу прогнозування включає в себе структурний елемент типу розділення, що представляє тип розділення, і структурний елемент розміру розділення, що представляє розмір розділення, декодування структурного елемента типу розділення синтаксичного елемента типу прогнозування з використанням контекстно-адаптивного бінарного арифметичного кодування з одним контекстом, причому цей один контекст є тим самим для будь-якого типу розділення, і декодування структурного елемента типу розділення синтаксичного елемента типу прогнозування з використанням CABAC в режимі обходу.

В іншому прикладі даного розкриття спосіб кодування відеоданих включає в себе кодування прапора кодованого блока кольоровості С_b для блока відеоданих з використанням CABAC, причому кодування прапора кодованого блока кольоровості С_b включає в себе використання набору контекстів, що включає в себе один або більше контекстів як частину CABAC, і кодування прапора кодованого блока кольоровості С_r з використанням CABAC, причому кодування прапора кодованого блока кольоровості С_r містить використання того самого набору контекстів, що і для прапора кодованого блока кольоровості С_b, як частину CABAC.

Дане розкриття також описує вищезгадані способи в термінах пристроїв, сконфігурованих для виконання цих способів, а також в термінах зчитуваного комп'ютером носія для зберігання інструкцій, які, при їх виконанні, спонукають один або більше процесорів виконувати ці способи.

Деталі одного або декількох прикладів наведені на прикладених кресленнях і описі нижче. Інші ознаки, цілі та переваги будуть очевидні з опису і креслень і з формули винаходу.

Короткий опис креслень

На фіг. 1 представлена блок-схема, що ілюструє зразкову систему кодування і декодування відео, яка може використовувати методи, описані в даному розкритті.

На фіг. 2 представлена блок-схема, що ілюструє зразковий відеокoder, який може реалізувати способи, описані в даному розкритті.

На фіг. 3 представлена блок-схема, що ілюструє зразковий відеокoder, який може реалізувати способи, описані в даному розкритті.

На фіг. 4 показане концептуальне креслення, що демонструє як квадратні, так і неквадратні типи розділення.

На фіг. 5 показане концептуальне креслення, що демонструє асиметричні типи розділення.

На фіг. 6 представлена блок-схема послідовності операцій, що ілюструє приклад способу кодування відео згідно з даним розкриттям.

На фіг. 7 представлена блок-схема послідовності операцій, що ілюструє приклад способу декодування відео згідно з даним розкриттям.

На фіг. 8 представлена блок-схема послідовності операцій, що ілюструє приклад способу кодування відео згідно з даним розкриттям.

На фіг. 9 представлена блок-схема послідовності операцій, що ілюструє приклад способу декодування відео згідно з даним розкриттям.

На фіг. 10 представлена блок-схема послідовності операцій, що ілюструє приклад способу кодування відео згідно з даним розкриттям.

Докладний опис

Дане розкриття описує способи кодування даних, таких як відеодані. Зокрема, в даному розкритті описуються способи, які можуть сприяти ефективному кодуванню відеоданих з використанням процесів контекстно-адаптивного ентропійного кодування. Зокрема, в даному розкритті пропонується скорочення кількості контекстів CABAC, що використовуються для

5 кодування синтаксичних елементів, таких як `pred_typ`, `merge_idx`, `inert_pred_flag`, `ref_idx_lx`, `cbf_cb`, `cbf_cr`, `coeff_abs_level_greater1_flag` та `coeff_abs_level_greater2_flag`. Модифікації скорочують до 56 контекстів з нехтувано малими змінами ефективності кодування. Дане розкриття описує кодування відео для цілей ілюстрації. Однак способи, описані в даному описі, також можуть бути застосовні до кодування інших типів даних.

10 На фіг. 1 представлена блок-схема, що ілюструє приклад системи 10 кодування і декодування відео, яка може бути сконфігурована, щоб використовувати методи контекстно-адаптивного бінарного арифметичного кодування (CABAC) відповідно до прикладів даного розкриття. Як показано на фіг. 1, система 10 включає в себе пристрій-джерело 12, який передає

15 кодоване відео до пристрою-одержувача 14 через канал 16 зв'язку. Кодовані відеодані можуть також зберігатися на носії 34 зберігання даних або файловому сервері 36, і пристрій-одержувач 14 може звертатися до них за бажанням. При зберіганні на носії зберігання даних або файловому сервері відеокoder 20 може надати кодовані відеодані іншому пристрою, такому як

20 мережний інтерфейс, компакт-диск (CD), програматор або пристрій обладнання штампування Blu-Ray або цифрового відеодиска (DVD) або інші пристрої, для зберігання кодованих відеоданих на носії зберігання даних. Крім того, пристрій, окремий від відеокodера 30, такий як мережний інтерфейс, CD або DVD зчитувач або тому подібне, може витягувати кодовані відеодані з носія зберігання даних і надавати витягнуті дані на відеокoder 30.

Пристрій-джерело 12 і пристрій-одержувач 14 можуть включати в себе будь-який з широкого спектра пристроїв, включаючи настільні комп'ютери, ноутбуки (тобто портативні комп'ютери),

25 планшетні комп'ютери, телевізійні приставки, телефонні апарати, такі як так звані смартфони, телевізори, камери, пристрої відображення, цифрові медіаплеєри, ігрові консолі або тому подібне. У багатьох випадках такі пристрої можуть бути оснащені для бездротового зв'язку. Таким чином, канал 16 зв'язку може містити бездротовий канал, дротовий канал або комбінацію дротових і бездротових каналів, придатних для передачі кодованих відеоданих. Аналогічно

30 файловий сервер 36 може бути доступний для пристрою-одержувача 14 через будь-яке стандартне з'єднання передачі даних, в тому числі Інтернет-з'єднання. Це може включати в себе бездротовий канал (наприклад, з'єднання Wi-Fi), дротове з'єднання (наприклад, DSL, кабельний модем і або поєднання того і іншого, яке підходить для доступу до кодованих відеоданих, що зберігаються на файловому сервері.

35 Методи CABAC, відповідно до прикладів даного розкриття, можуть бути застосовані до кодування відео для підтримки будь-якого з множини мультимедійних додатків, таких як ефірне телевізійне мовлення, передачі кабельного телебачення, передачі супутникового телебачення, передачі потокового відео, наприклад, через Інтернет, кодування цифрового відео для зберігання на носії даних, декодування цифрового відео, збереженого на носії даних, або інших

40 додатків. У деяких прикладах система 10 може бути сконфігурована для підтримки односторонніх або двосторонніх передач відео для підтримки таких додатків, як потокове відео, відтворення відео, відеомовлення і/або відеотелефонія.

У прикладі за фіг. 1 пристрій-джерело 12 включає в себе джерело 18 відео, відеокoder 20, модулятор/демодулятор 22 і передавач 24. У пристрої-джерелі 12 джерело 18 відео може

45 включати в себе джерело, таке як пристрій захоплення відео, наприклад, відеокамеру, відеоархіви, що містять раніше захоплене відео, інтерфейс відеоканалу для прийому відео від постачальника відеоконтенту і/або систему комп'ютерної графіки для генерації даних комп'ютерної графіки як вихідного відео або комбінацію таких джерел. Як один приклад, якщо джерело 18 відео являє собою відеокамеру, пристрій-джерело 12 і пристрій-одержувач 14

50 можуть утворювати так звані камери-телефони або відеотелефони. Однак способи, описані в даному розкритті, можуть застосовуватися до кодування відео загалом і можуть бути застосовані до бездротових і/або дротових додатків або додатку, в якому кодовані відеодані збережені на локальному диску.

Захоплене, заздалегідь захоплене відео або відео, що генерується комп'ютером, може

55 кодуватися за допомогою відеокodера 20. Кодована інформація відео може модулюватися за допомогою модему 22 відповідно до стандарту зв'язку, наприклад, протоколу бездротового зв'язку, і передаватися до пристрою-одержувача 14 за допомогою передавача 24. Модем 22 може включати в себе різні змішувачі, фільтри, підсилювачі та інші компоненти, призначені для модуляції сигналу. Передавач 24 може включати в себе схеми, призначені для передачі даних,

60 в тому числі підсилювачі, фільтри і одну або більше антен.

Захоплене, заздалегідь захоплене відео або відео, що генерується комп'ютером, яке кодується за допомогою відеокодера 20, також може зберігатися на носії 34 зберігання даних або файловому сервері 36 для подальшого споживання. Носій 34 зберігання даних може включати Blu-Ray диски, DVD-диски, CD-ROM, флеш-пам'ять або будь-який інший підходящий цифровий носій зберігання даних для зберігання кодованого відео. Кодоване відео, збережене на носії 34 зберігання даних, може потім бути доступним для пристрою-одержувача 14 для декодування і відтворення. Хоча це не показано на фіг. 1, в деяких прикладах носій 34 зберігання даних і/або файловий сервер 36 може зберігати вихідний сигнал передавача 24.

Файловий сервер 36 може бути будь-яким типом сервера, здатним зберігати кодоване відео і передавати це кодоване відео до пристрою-одержувача 14. Зразкові файлові сервери включають в себе веб-сервер (наприклад, для веб-сайта), FTP-сервер, пристрої зберігання, зв'язані з мережею (NAS), локальний диск або будь-який інший тип пристрою, здатного зберігати кодовані відеодані і передавати їх до пристрою-одержувача. Передача кодованих відеоданих з файлового сервера 36 може бути потоковою передачею, передачею завантаження або поєднанням того і іншого. Файловий сервер 36 може бути доступним для пристрою-одержувача 14 через будь-яке стандартне з'єднання передачі даних, в тому числі Інтернет-з'єднання. Це може включати в себе бездротовий канал (наприклад, з'єднання Wi-Fi), дротове з'єднання (наприклад, DSL, кабельний модем, Ethernet, USB і або поєднання того і іншого, яке підходить для доступу до кодованих відеоданих, що зберігаються на файловому сервері.

Пристрій-одержувач 14 в прикладі за фіг. 1 включає в себе приймач 26, модем 28, відеодекодер 30 і пристрій 32 відображення. Приймач 26 пристрою-одержувача 14 приймає інформацію по каналу 16, і модем 28 демодулює інформацію для одержання демодульованого потоку бітів для відеодекодера 30. Інформація, що передається через канал 16, може включати в себе різну синтаксичну інформацію, що генерується відеокодером 20, для використання відеодекодером 30 при декодуванні відеоданих. Такий синтаксис також може бути включений в кодовані відеодані, збережені на носії 34 зберігання даних або файловому сервері 36. Кожний з відеокодера 20 і відеодекодера 30 може бути частиною відповідного кодера-декодера (кодека), який здатний кодувати або декодувати відеодані.

Пристрій 32 відображення може бути вбудованим або зовнішнім по відношенню до пристрою-одержувача 14. У деяких прикладах пристрій-одержувач 14 може включати в себе вбудований пристрій відображення, а також може бути сконфігурований для взаємодії із зовнішнім пристроєм відображення. В інших прикладах пристрій-одержувач 14 може бути пристроєм відображення. Загалом, 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.

У прикладі на фіг. 1 канал 16 зв'язку може містити будь-яке дротове або бездротове середовище зв'язку, наприклад, радіочастотний (РЧ) спектр або одну або декількох фізичних ліній передачі, або будь-яку комбінацію бездротових і дротових середовищ передачі. Канал 16 зв'язку може бути частиною пакетної мережі, наприклад, локальної мережі, мережі широкого охоплення або глобальної мережі, такої як Інтернет. Канал 16 зв'язку звичайно являє собою будь-яке підходяще середовище зв'язку або набір різних середовищ зв'язку для передачі відеоданих від пристрою-джерела 12 до пристрою-одержувача 14, включаючи будь-яку підходящу комбінацію дротових або бездротових середовищ передачі. Канал 16 зв'язку може включати в себе маршрутизатори, комутатори, базові станції або будь-яке інше обладнання, яке може бути корисним для забезпечення зв'язку від пристрою-джерела 12 до пристрою-одержувача 14.

Відеокoder 20 і відеодекодер 30 можуть працювати відповідно до стандарту стиснення відео, такого як стандарт Високоєфективного кодування відео (HEVC), що знаходиться в цей час на стадії розробки Об'єднаної спільної групи з кодування відео (JCT-VC) ITU-T, Групи експертів з кодування відео (VCEG) і ISO/IEC Групи експертів з рухомого зображення (MPEG). Недавній проект стандарту HEVC, який називається "HEVC робочий проект 6" або "WD6", описується в документі JCTVC-H1003, Bross et al., "High efficiency video coding (HEVC) text specification draft 6", Об'єднаної спільної групи з кодування відео (JCT-VC) ITU-T SGI 6 WP3 та ISO/IEC JTC1/SC29/WG11, 8-е засідання: Сан-Хосе, Каліфорнія, США, лютий 2012 р., який за станом на 1 червня 2012 року можна скачати з http://phenix.int-evry.fr/jct/doc_end_user/documents/8_San%20Jose/wg11/JCTVC-1003-v22.zip.

Крім того, відеокoder 20 і відеодекодер 30 можуть діяти відповідно до інших пропріетарних або промислових стандартів, таких як стандарт ITU-T H.264, який альтернативно називається MPEG-4, частина 10, Розширене кодування відео (AVC), або розширення таких стандартів.

Методи даного розкриття, однак, не обмежується яким-небудь конкретним стандартом кодування. Інші приклади включають MPEG-2 та ITU-T H.263.

Хоча це не показано на фіг. 1, в деяких аспектах відеокoder 20 і відеокoder 30 можуть, кожний, бути інтегрований з аудіокoderом і декодером і можуть включати в себе відповідні блоки MUX-DEMUX або інші апаратні засоби і програмне забезпечення для обробки кодування аудіо і відео в загальному потоці даних або окремих потоках даних. Якщо застосовно, в деяких прикладах блоки MUX-DEMUX можуть відповідати протоколу мультимплектора ITU H.223 або іншим протоколам, таким як протокол користувацьких дейтаграм (UDP).

Відеокoder 20 і відеокoder 30, кожний, може бути реалізований як будь-яка з множини підходящих схем кодера, таких як один або більше мікропроцесорів, цифрові сигнальні процесори (DSP), спеціалізовані інтегральні схеми (ASIC), програмовані вентиляльні матриці (FPGA), дискретна логіка, програмне забезпечення, апаратні засоби, програмно-апаратні засоби або будь-які їх комбінації. Коли технології реалізовані частково в програмному забезпеченні, пристрій може зберігати інструкції для програмного забезпечення на підходящому постійному зчитуваному комп'ютером носії і виконувати інструкції в апаратних засобах з використанням одного або більше процесорів для виконання методів даного розкриття. Кожний з відеокoderа 20 і відеокoderа 30 може бути включений в один або більше кодерів або декодерів, кожний з яких може бути інтегрований як частина комбінованого кодера/декодера (кодека) у відповідному пристрої.

Відеокoder 20 може реалізовувати будь-який або всі методи даного розкриття для CABAC в процесі кодування відео. Крім того, відеокoder 30 може здійснювати деякі або всі з цих методів для CABAC в процесі кодування відео. Пристрій кодування відео, як описано в даному розкритті, може відноситися до відеокoderа або відеокoderа. Аналогічним чином блок кодування відео може відноситися до відеокoderа або відеокoderа. Крім того, кодування відео може відноситися до кодування відео або декодування відео.

В одному прикладі даного розкриття відеокoder 20 може бути сконфігурований, щоб визначати перший тип прогнозування для блока відеоданих у Р сегменті, представляти перший тип прогнозування як синтаксичний елемент типу прогнозування Р сегмента, визначати другий тип прогнозування для блока відеоданих у В сегменті, представляти другий тип прогнозування як синтаксичний елемент типу прогнозування В сегмента, визначати бінаризацію Р сегмента для синтаксичного елемента типу прогнозування Р сегмента, визначати бінаризацію В сегмента для синтаксичного елемента типу прогнозування В сегмента, при цьому синтаксичний елемент типу прогнозування Р сегмента і синтаксичний елемент типу прогнозування В сегмента визначаються з використанням тієї самої логіки бінаризації, і кодувати відеодані на основі бінаризації синтаксичного елемента типу прогнозування Р сегмента і синтаксичного елемента типу прогнозування В сегмента.

В іншому прикладі даного розкриття відеокoder 30 може бути сконфігурований, щоб відображати бінаризований синтаксичний елемент типу прогнозування Р сегмента на тип прогнозування, використовуючи відображення бінаризації для блока відеоданих у Р сегменті, відображати бінаризований синтаксичний елемент типу прогнозування В сегмента на тип прогнозування, використовуючи те саме відображення бінаризації для блока відеоданих у В сегменті, і декодувати відеодані, основуючись на відображених типах прогнозування.

В іншому прикладі даного розкриття відеокoder 20 може бути сконфігурований, щоб визначати тип розділення для режиму прогнозування для блока відеоданих, кодувати структурний елемент типу розділення синтаксичного елемента типу прогнозування для блока відеоданих з використанням CABAC з одним контекстом, причому цей один контекст є тим самим для будь-якого типу розділення, і кодувати структурний елемент розміру розділення синтаксичного елемента типу прогнозування для блока відеоданих з використанням CABAC в режимі обходу.

В іншому прикладі даного розкриття відеокoder 30 може бути сконфігурований, щоб приймати синтаксичний елемент типу прогнозування для блока відеоданих, які були кодовані з використанням CABAC, причому синтаксичний елемент типу прогнозування включає в себе структурний елемент типу розділення, що представляє тип розділення, і структурний елемент розміру розділення, що представляє розмір розділення, декодувати структурний елемент типу розділення синтаксичного елемента типу прогнозування з використанням CABAC з одним контекстом, причому цей один контекст є тим самим для будь-якого типу розділення, і декодувати структурний елемент розміру розділення синтаксичного елемента типу прогнозування з використанням CABAC в режимі обходу.

В іншому прикладі даного розкриття, як відеокoder 20, так і відеокoder 30 можуть бути сконфігуровані, щоб кодувати прапор кодованого блока кольоровості Сb для блока відеоданих з

використанням CABAC, причому кодування прапора кодованого блока кольоровості C_b включає в себе використання набору контекстів, що включає в себе один або більше контекстів як частину CABAC, і кодувати прапор кодованого блока кольоровості C_r з використанням CABAC, причому кодування прапора кодованого блока кольоровості C_r містить використання того самого набору контекстів, що і для прапора кодованого блока кольоровості C_b , як частину CABAC.

JCT-VC працює над розвитком стандарту HEVC. Зусилля HEVC стандартизації базуються на моделі пристрою кодування, що розвивається відео, яка називається тестовою моделлю HEVC (HM). HM передбачає декілька додаткових можливостей пристроїв кодування відео відносно існуючих пристроїв відповідно, наприклад, до ITU-T H.264/AVC. Наприклад, в той час як H.264 забезпечує дев'ять режимів кодування з інтра-прогнозуванням, HM може забезпечити тридцять три режими кодування з інтра-прогнозуванням. У наступному розділі будуть більш детально описані деякі аспекти HM.

Загалом, робоча модель HM описує, що відеокадр або зображення може бути розділено на послідовність блоків дерева або найбільших одиниць кодування (LCU), які включають в себе вибірки як яскравості, так і кольоровості. Блок дерева має мету, подібну до такої макроблока стандарту H.264. Сегмент (слайс) включає в себе ряд послідовних блоків дерева в порядку кодування. Відеокадр або зображення може бути розділено на один або більше сегментів. Кожний блок дерева може бути розділений на одиниці кодування (CU) відповідно до квадродревця. Наприклад, блок дерева, як кореневий вузол квадродревця, може бути розділений на чотири дочірні вузли, а кожний дочірній вузол, в свою чергу, може бути батьківським вузлом і може бути розділений на наступні чотири дочірні вузли. Остаточний, нероздільний дочірній вузол, як вузол листа квадродревця, включає в себе вузол кодування, тобто блок кодованого відео. Дані синтаксису, зв'язані з кодованим бітовим потоком, можуть визначати максимальну кількість разів розбиття блока дерева, а також можуть визначати мінімальний розмір вузлів кодування.

CU включає в себе вузол кодування і одиниці прогнозування (PU) і одиниці перетворення (TU), зв'язані з вузлом кодування. Розмір CU звичайно відповідає розміру вузла кодування і звичайно повинен бути квадратної форми. Розмір CU може знаходитися в діапазоні від 8×8 пікселів до розміру блока дерева максимально з 64×64 пікселів 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 на одну або більше PU. Режими розділення можуть відрізнятися залежно від того, чи пропускається CU, або кодована в прямому режимі, кодована в режимі інтра-прогнозування або кодована в режимі інтер-прогнозування. PU можуть бути розділені, щоб бути в неквадратній формі. Дані синтаксису, асоційовані з CU, також можуть описувати, наприклад, розділення CU на одну або більше TU відповідно до квадродревця. TU може бути квадратної або неквадратної форми.

Новий стандарт HEVC допускає перетворення відповідно до TU, які можуть бути різними для різних CU. TU звичайно мають розмір залежно від розміру PU в межах даної CU, визначеної для розділеної LCU, хоча це може не завжди мати місце. TU, як правило, такого самого розміру або менше, ніж PU. У деяких прикладах залишкові вибірки, що відповідають CU, можуть бути розділені на більш дрібні одиниці, використовуючи структуру квадродревця, відому як "залишкове квадродререво" (RQT). Листові вузли RQT можуть згадуватися як одиниці перетворення (TU). Піксельні різниці значення, зв'язані з TU, можуть бути перетворені для одержання коефіцієнтів перетворення, які можуть бути квантовані.

Загалом, PU належить до даних, що відносяться до процесу прогнозування. Наприклад, коли PU є кодовою в інтра-режимі, PU може включати дані, що описують режим інтра-прогнозування для PU. Як інший приклад, коли PU кодована в інтер-режимі, PU може включати дані, що визначають вектор руху для PU. Дані, що визначають вектор руху для PU, можуть описувати, наприклад, горизонтальну складову вектора руху, вертикальну складову вектора руху, розрізнення для вектора руху (наприклад, точність в одну чверть пікселя або точність в одну восьму пікселя), опорне зображення, на яке вказує вектор руху, і/або список опорних зображень (наприклад, список 0, список 1 або список C) для вектора руху.

Загалом, TU використовується для процесів перетворення і квантування. Дана CU, що має одну або більше PU, може також включати в себе одну або більше одиниць перетворення (TU). Після прогнозування відеокодер 20 може обчислити залишкові значення з блока відео, ідентифікованого вузлом кодування, відповідно до PU. Вузол кодування потім оновлюється для посилення на залишкові значення замість вихідного блока відео. Залишкові значення містять піксельні різниці значення, які можуть перетворюватися в коефіцієнти перетворення, квантуватися і скануватися з використанням перетворень та іншої інформації перетворення,

визначеної в ТУ, щоб створити перетворені в послідовну форму коефіцієнти перетворення для ентропійного кодування. Вузол кодування може бути в черговий раз оновлений, щоб посилатися на ці перетворені в послідовну форму коефіцієнти перетворення. Дане розкриття звичайно використовує термін "блок відео", щоб посилатися на вузол кодування CU. У деяких конкретних випадках дане розкриття може також використовувати термін "блок відео", щоб посилатися на блок дерева, тобто LCU або CU, яка включає в себе вузол кодування і PU і TU.

Відеопослідовність звичайно включає в себе ряд відеокадрів або зображень. Група зображень (GOP) звичайно включає в себе послідовність з одного або більше відеозображень. GOP може включати синтаксичні дані в заголовку GOP, заголовку одного або більше зображень або в іншому місці, яке описує ряд зображень, включених в GOP. Кожний сегмент зображення може включати синтаксичні дані сегмента, які описують режим кодування для відповідного сегмента. Відеокодер 20 звичайно працює на блоках відео в окремих сегментах відео для кодування відеоданих. Блок відео може відповідати вузлу кодування всередині CU. Блоки відео можуть мати фіксовані або змінні розміри і можуть відрізнятися за розміром відповідно до заданого стандарту кодування.

Як приклад НМ підтримує прогнозування в різних розмірах PU. У припущенні, що розмір визначеної CU рівний $2N \times 2N$, НМ підтримує інтра-прогнозування в PU розмірів $2N \times 2N$ або $N \times N$ та інтер-прогнозування в симетричних PU розмірів $2N \times 2N$, $2N \times N$, $N \times 2N$ або $N \times N$. НМ також підтримує асиметричне розділення для інтер-прогнозування в PU розмірів $2N \times nU$, $2N \times nD$, $nL \times 2N$ та $nR \times 2N$. В асиметричному розділенні один напрямок CU не розділяється, а інший напрямок розділяється на 25 % та 75 %. Частина CU, що відповідає 25 % розділенню, означається за допомогою "n" з подальшою вказівкою "вгору", "вниз", "ліворуч" або "праворуч". Так, наприклад, " $2N \times nU$ " відноситься до $2N \times 2N$ CU, яка розділена горизонтально з $2N \times 0,5 N$ PU зверху і $2N \times 1,5 N$ PU знизу.

На фіг. 4 показана концептуальна діаграма, що демонструє як квадратні, так і неквадратні типи розділення для інтра-прогнозування та інтер-прогнозування. Розділення 102 є розділенням $2N \times 2N$ і може бути використане як для інтра-прогнозування, так для інтер-прогнозування. Розділення 104 є розділенням $N \times N$ і може бути використане як для інтра-прогнозування, так і для інтер-прогнозування. Розділення 106 є розділенням $2N \times N$ і в цей час використовується в HEVC для інтер-прогнозування. Розділення 108 є розділенням $N \times 2N$ і в цей час використовується в HEVC для інтер-прогнозування.

На фіг. 5 представлена концептуальна діаграма, що показує асиметричні типи розділення. Розділення 110 є розділенням $2N \times nU$ і в цей час використовується в HEVC для інтер-прогнозування. Розділення 112 є розділенням $2N \times nD$ і в цей час використовується в HEVC для інтер-прогнозування. Розділення 114 є розділенням $nL \times 2N$ і в цей час використовується в HEVC для інтер-прогнозування. Розділення 116 є розділенням $nR \times 2N$ і в цей час використовується в HEVC для інтер-прогнозування.

У цьому описі " $N \times N$ " та " N на N " можуть використовуватися взаємозамінним чином для позначення розмірів у пікселях блока відео по вертикалі і горизонталі, наприклад, 16×16 пікселів або 16 на 16 пікселів. Загалом, 16×16 блок буде мати 16 пікселів у вертикальному напрямку ($y=16$) та 16 пікселів у горизонтальному напрямку ($x=16$). Аналогічно блок $N \times N$, в загальному вигляді, має N пікселів у вертикальному напрямку і N пікселів в горизонтальному напрямку, де N являє собою ненегативне ціле число. Пікселі в блоці можуть бути впорядковані в рядки і стовпці. Крім того, блоки не обов'язково повинні мати однакове число пікселів у горизонтальному напрямку і у вертикальному напрямку. Наприклад, блоки можуть містити $N \times M$ пікселів, де M не обов'язково дорівнює N .

Після кодування з інтра-прогнозуванням або інтер-прогнозуванням з використанням PU для CU відеокодер 20 може обчислити залишкові дані, до яких застосовуються перетворення, вказані за допомогою ТУ в PU. Залишкові дані можуть відповідати різниці в пікселях між пікселями некодованого зображення і значеннями прогнозування, що відповідають CU. Відеокодер 20 може формувати залишкові дані для CU, а потім перетворювати залишкові дані для одержання коефіцієнтів перетворення.

Після будь-яких перетворень, щоб сформувані коефіцієнти перетворення, відеокодер 20 може виконувати квантування коефіцієнтів перетворення. Квантування, загалом, стосується процесу, в якому коефіцієнти перетворення квантуються, щоб по можливості зменшити обсяг даних, що використовуються для представлення коефіцієнтів, забезпечуючи подальше стиснення. Процес квантування може зменшити бітову глибину, асоційовану з деякими або усіма коефіцієнтами. Наприклад, n -бітове значення можна округлити вниз до m -бітового значення під час квантування, де n більше, ніж m .

У деяких прикладах відеокoder 20 може використовувати зумовлений порядок сканування для сканування квантованих коефіцієнтів перетворення для одержання перетвореного в послідовну форму вектора, який може ентропійно кодуватися. В інших прикладах відеокoder 20 може виконувати адаптивне сканування. Після сканування квантованих коефіцієнтів перетворення, щоб сформувати одномірний вектор, відеокoder 20 може ентропійно кодувати одномірний вектор, наприклад, відповідно до контекстно-адаптивного кодування змінної довжини (CAVLC), контекстно-адаптивного бінарного арифметичного кодування (CABAC), основаного на синтаксисі контекстно-адаптивного бінарного арифметичного кодування (SBAC), ентропійного кодування з розділенням інтервалу імовірності (PIPE) або іншим методом ентропійного кодування. Відеокoder 20 може також ентропійно кодувати синтаксичні елементи, асоційовані з кодованими відеоданими для використання відеодекодером 30 в декодування відеоданих.

Для виконання CABAC відеокoder 20 може призначити контекст в контекстній моделі для символу, що підлягає передачі. Контекст може відноситися, наприклад, до того, чи є сусідні значення символу ненульовими, чи ні. Для виконання CAVLC відеокoder 20 може вибрати код змінної довжини для символу, що підлягає передачі. Кодові слова в VLC можуть бути сконструйовані так, 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 в бітах в порівнянні, наприклад, з використанням кодових слів однакової довжини для кожного символу, що підлягає передачі. Визначення імовірності може бути основане на контексті, призначеному символу.

Дане розкриття стосується способів для ентропійних кодерів контекстно-адаптивного бінарного арифметичного кодування (CABAC) або інших ентропійних кодерів, таких як ентропійне кодування з розділенням інтервалу імовірності (PIPE) або зв'язаних кодерів. Арифметичне кодування є формою ентропійного кодування, що використовується в багатьох алгоритмах стиснення, які мають високу ефективність кодування, оскільки воно здатне відображати символи на кодові слова нецілочисельної довжини. Прикладом алгоритму арифметичного кодування є основане на контексті бінарне арифметичне кодування (CABAC), що використовується в H.264/AVC.

Загалом, кодування символів даних за допомогою CABAC включає в себе один або більше з наступних етапів:

(1) Бінаризація: Якщо символ, що підлягає кодуванню, не є бінарним значенням, він відображається на послідовності так званих "структурних елементів". Кожний структурний елемент може мати значення "0" або "1".

(2) Призначення контексту: Кожному структурному елементу (в звичайному режимі) призначається контекст. Контекстна модель визначає, як розраховується контекст для даного структурного елемента на основі інформації, доступної для структурного елемента, такої як значення раніше кодованих символів або кількість структурних елементів.

(3) Кодування структурних елементів: Структурні елементи кодуються арифметичним кодером. Для кодування структурного елемента арифметичний кодер вимагає як вхід імовірність значення структурного елемента, тобто імовірність того, що значення структурного елемента дорівнює "0", і імовірність того, що значення структурного елемента дорівнює "1". (Оцінена) імовірність кожного контексту представлена цілим числом, що називається "станом контексту". Кожний контекст має стан, і, таким чином, стан (тобто оцінена імовірність) є тим самим для структурних елементів, призначених одному контексту, і відрізняється між контекстами.

(4) Оновлення стану: Імовірність (стан) для вибраного контексту оновлюється на основі фактичного кодованого значення структурного елемента (наприклад, якщо значення структурного елемента було "1", імовірність "1" збільшується).

Потрібно зазначити, що ентропійне кодування з розділенням інтервалу імовірності (PIPE) використовує принципи, аналогічні принципам арифметичного кодування, і, таким чином, може також використовувати метод даного розкриття.

CABAC в H.264/AVC і HEVC використовує стани, і кожний стан неявно зв'язаний з імовірністю. Існують варіанти CABAC, в яких імовірність символу ("0" або "1") використовується безпосередньо, тобто імовірність (або її цілочисельна версія) є станом. Наприклад, такі варіанти CABAC описані в "Description of video coding technology proposal by France Telecom, NTT, NTT DOCOMO, Panasonic and Technicolor", JCTVC-A114, 1st JCT-VC Meeting, Dresden, DE, квітень 2010 р., далі згадується як "JCTVC-A114", і A. Alshin and E. Alshina, "Multi-parameter probability update for CABAC", JCTVC-F254, 6th JCT-VC Meeting, Torino, IT, липень 2011, далі згадується як "JCTVC-F254".

У даному розкритті запропоноване скорочення кількості бінаризацій і/або контекстів, що використовуються в CABAC. Зокрема, дане розкриття пропонує методи, які можуть знизити кількість контекстів, що використовуються в CABAC, на величину до 56. З числом на 56 менше контекстів експериментальні результати показують, 0,00 %, 0,01 % та -0,13 % зміни частоти спотворення бітів (BD) у високоефективному інтра-режимі тільки, довільному доступі і тестових умовах малої затримки відповідно. Таким чином, скорочення в кількості необхідних контекстів знижує потреби в зберіганні як в кодері, так і декодері без істотного впливу на ефективність кодування.

У даному розкритті запропоноване скорочення кількості контекстів CABAC, що використовуються для синтаксичних елементів `pred_type`, `merge_idx`, `inert_pred_flag`, `ref_idx_lx`, `cbf_cb`, `cbf_cr`, `coeff_abs_level_greater1_flag` та `coeff_abs_level_greater2_flag`. Модифікації скорочують до 56 контекстів з нехтувано малими змінами в ефективності кодування. Запропоновані скорочення контексту для синтаксичних елементів, наведених вище, можуть бути використані окремо або в будь-якій комбінації.

Синтаксичний елемент `pred_type` включає в себе режим прогнозування (`pred_mode_flag`) і тип розділення (`part_mode`) для кожної одиниці кодування. Синтаксичний елемент `pred_mode_flag`, рівний 0, визначає, що поточна одиниця кодування кодована в режимі інтер-прогнозування. Синтаксичний елемент `pred_mode_flag`, рівний 1, вказує, що поточна одиниця кодування кодована в режимі інтра-прогнозування. Синтаксичний елемент `part_mode` визначає режим розділення поточної одиниці кодування.

Синтаксичний елемент `merge_idx[x0][y0]` визначає індекси кандидатів злиття списку кандидатів злиття, де `x0`, `y0` визначають місцеположення (`x0`, `y0`) верхньої лівої вибірки яскравості блока прогнозування, що розглядається, відносно верхньої лівої вибірки яскравості зображення. Коли `merge_idx[x0][y0]` відсутній, то робиться висновок, що він дорівнює 0. Список кандидатів злиття є списком сусідніх одиниць кодування з поточними одиницями, з яких інформація про рух може бути скопійована.

Синтаксичний елемент `inter_pred_flag[x0][y0]` визначає, чи використовується моно-прогнозування або бі-прогнозування для поточної одиниці прогнозування. Індекси масиву `x0`, `y0` визначають місцеположення (`x0`, `y0`) верхньої-лівої вибірки яскравості блока прогнозування, що розглядається, відносно верхньої-лівої вибірки яскравості зображення.

Синтаксичний елемент `ref_idx_lx` відноситься до конкретного опорного зображення всередині списку опорних зображень.

Синтаксичні елементи `cbf_cb`, `cbf_cr` вказують, чи містять блоки перетворення кольоровості (`Cb` та `Cr`, відповідно) ненульові коефіцієнти перетворення. Синтаксичний елемент `cbf_cb[x0][y0][trafoDepth]`, рівний 1, визначає, що блок перетворення `Cb` містить один або більше рівнів коефіцієнтів перетворення, не рівних 0. Індекси масиву `x0`, `y0` вказують місцеположення (`x0`, `y0`) верхньої-лівої вибірки яскравості блока перетворення, що розглядається, відносно верхньої-лівої вибірки яскравості зображення. Індекс масиву `trafoDepth` вказує поточний рівень підрозділу одиниці кодування на блоки з метою кодування перетворення. Індекс масиву `trafoDepth` рівний 0 для блоків, які відповідають одиницям кодування. Коли `cbf_cb[x0][y0][trafoDepth]` не присутній, і режимом прогнозування є не інтра-прогнозування, значення `cbf_cb[x0][y0][trafoDepth]` виводиться як рівне 0.

Синтаксичний елемент `cbf_cr[x0][y0][trafoDepth]`, рівний 1, вказує, що блок перетворення `Cr` містить один або більше рівнів коефіцієнтів перетворення, не рівних 0. Індекси масиву `x0`, `y0` вказують місцеположення (`x0`, `y0`) верхньої-лівої вибірки яскравості блока перетворення, що розглядається, відносно верхньої-лівої вибірки яскравості зображення. Індекс масиву `trafoDepth` вказує поточний рівень підрозділу одиниці кодування на блоки з метою кодування перетворення. Індекс масиву `trafoDepth` рівний 0 для блоків, які відповідають одиницям кодування. Коли `cbf_cr[x0][y0][trafoDepth]` не присутній, і режимом прогнозування не є інтра-прогнозування, значення `cbf_cr[x0][y0][trafoDepth]` виводиться рівним 0.

Синтаксичний елемент `coeff_abs_level_greater1_flag[n]` вказує для позиції `n` сканування, чи є рівні коефіцієнтів перетворення більше, ніж 1. Коли `coeff_abs_level_greater1_flag[n]` не присутній, то робиться висновок, що він дорівнює 0.

Синтаксичний елемент `coeff_abs_level_greater2_flag[n]` вказує для позиції `n` сканування, чи є рівні коефіцієнтів перетворення більше, ніж 2. Коли `coeff_abs_level_greater2_flag[n]` не присутній, робиться висновок, що він дорівнює 0.

В одній пропозиції для HEVC різні бінаризації за синтаксичним елементом `pred_type` використовуються в Р та В сегментах, як показано в таблиці 1. Дане розкриття пропонує використовувати ті самі бінаризації для Р та В сегментів. Приклади наведені в таблицях 2-4. Таблиця 5 показує вплив ефективності кодування на Р сегмент в стандартних умовах

тестування (див., наприклад, F Bossen, "Common test conditions and software reference configurations", JCTVC-F900).

Таблиця 1

Бінаризація для pred_type в одній пропозиції для HEVC

Тип сег-мента	Значення pred_type	Режим прогнозування	Режим розділення	Рядок структурного елемента		
				cLog2CUsSize > Log2Min CUsSize (1)	cLog2CUsSize = 3 && !inter_4x4_enabled_flag (2)	cLog2CUsSize > 3 inter_4x4_enabled_flag (3)
I	0	MODE_INTRA	PART_2N×2N	-	1	1
	1	MODE_INTRA	PART_N×N	-	0	0
P	0	MODE_INTRA	PART_2N×2N	01	0 1	0 1
	1	MODE_INTRA	PART_2N×N	0 011	0 01	0 01
	2	MODE_INTRA	PART_N×2N	0 0011	0 00	0 001
	4	MODE_INTRA	PART_2N×nU	0 0100	-	-
	5	MODE_INTRA	PART_2N×nD	0 0101	-	-
	6	MODE_INTRA	PART_nL×2N	0 00100	-	-
	7	MODE_INTRA	PART_nR×2N	0 00101	-	-
	3	MODE_INTRA	PART_N×N	-	-	0 000
	4	MODE_INTRA	PART_2N×2N	1	11	11
	5	MODE_INTRA	PART_N×N	-	10	10
B	0	MODE_INTRA	PART_2N×2N	1	1	1
	1	MODE_INTRA	PART_2N×N	011	01	01
	2	MODE_INTRA	PART_N×2N	0011	001	001
	4	MODE_INTRA	PART_2N×nU	0100	-	-
	5	MODE_INTRA	PART_2N×nD	0101	-	-
	6	MODE_INTRA	PART_nL×2N	00100	-	-
	7	MODE_INTRA	PART_nR×2N	00101	-	-
	3	MODE_INTRA	PART_N×N	-	-	0001
	4	MODE_INTRA	PART_2N×2N	000	000 0	0000 0
	5	MODE_INTRA	PART_N×N	-	000 1	0000 1

- 5 Як видно з таблиці 1, I сегменти (наприклад, сегменти, які включають тільки блоки інтра-прогнозування) включають в себе два різних типи прогнозування (pred_type). Один рядок структурного елемента (бінаризація) використовується для блока інтра-прогнозування з типом розділення 2N×2N, а інший рядок структурного елемента використовується для блока інтра-прогнозування з типом розділення N×N. Як показано в таблиці 1, рядок структурного елемента, що використовується для I сегментів, не залежить від розміру CU.

- 10 Для P та B сегментів, в Таблиці 1, різні рядки структурного елемента використовуються для кожного значення pred_type. Знову значення pred_type залежить як від режиму прогнозування (інтер-прогнозування або інтра-прогнозування), так і використовуваного типу розділення. Для P та B сегментів фактичний використовуваний рядок структурного елемента додатково залежить від розміру кодованої CU і від того, чи підтримується інтер-прогнозування для розміру 4 × 4 блока.

- 15 Перший стовпчик під рядком структурного елемента застосовується для ситуації, коли логарифмічна функція розміру CU кодованої CU перевищує логарифмічну функцію мінімально допустимого розміру CU. Згідно з одним прикладом в HEVC, перший стовпчик рядків структурного елемента використовується, якщо cLog2CUsSize > Log2MinCUsSize. Логарифмічна функція використовується для створення меншого числа, так що може бути використаний менший послідовний індекс.

- 20 Якщо логарифмічна функція від розміру CU кодованої CU еквівалентна логарифмічній функції мінімально допустимого розміру CU (тобто cLog2CUsSize == Log2MinCUsSize), то один зі стовпчиків 2 та 3 під рядком структурного елемента в таблиці 1 використовується для вибору

бінарizaції. Столпчик 2 використовується, коли логарифмічна функція від розміру CU кодованої CU еквівалентна 3, і інтер-прогнозування для 4×4 CU не підтримується (тобто $cLog2CUsi\text{ze} == 3 \&\& !inter_4 \times 4_enabled_flag$). Столпчик 3 використовується, коли логарифмічна функція від розміру CU для кодованої CU більше, ніж 3, або коли інтер-прогнозування для 4×4 CU підтримується (тобто $cLog2CUsi\text{ze} > 3 || inter_4 \times 4_enabled_flag$).

У таблиці 2 нижче показані приклади бінарizaції, де P та B сегменти використовують ті самі рядки структурного елемента, відповідно до одного або більше прикладів, описаних в даному розкритті. Як показано в таблиці 2, P сегменти використовують ті самі бінарizaції, що використовуються для B сегментів у таблиці 1. Таким чином, немає необхідності зберігати і використовувати окремий набір контекстів для P та B сегментів. Таким чином, загальна кількість контекстів, необхідних для кодування синтаксичного елемента `pred_type`, знижується. Крім того, тільки одне відображення (замість двох) між логікою рядка структурного елемента (показано в столпчиках (1)-(3)) і фактичним рядком структурного елемента повинне бути збережене.

Таблиця 2

Бінарizaція для `pred_type` в одному прикладі даного розкриття

Тип сегмента	Значення <code>pred_type</code>	Режим прогнозування	Режим розділення	Рядок структурного елемента		
				cLog2CU Size> Log2Min CUsi\text{ze} (1)	cLog2CUsi\text{ze} = = Log2Min CUsi\text{ze}	
					cLog2CUsi\text{ze} = = 3 && !inter_4× 4_enabled_flag (2)	cLog2CUsi\text{ze}> 3 inter_4× 4_enabled_flag (3)
I	0	MODE_INTRA	PART_2N×2N	-	1	1
	1	MODE_INTRA	PART_N×N	-	0	0
P або B	0	MODE_INTRA	PART_2N×2N	1	1	1
	1	MODE_INTRA	PART_2N×N	011	01	01
	2	MODE_INTRA	PART_N×2N	0011	001	001
	4	MODE_INTRA	PART_2N×nU	0100	-	-
	5	MODE_INTRA	PART_2N×nD	0101	-	-
	6	MODE_INTRA	PART_nL×2N	00100	-	-
	7	MODE_INTRA	PART_nR×2N	00101	-	-
	3	MODE_INTRA	PART_N×N	-	-	0001
	4	MODE_INTRA	PART_2N×2N	000	000 0	0000 0
	5	MODE_INTRA	PART_N×N	-	000 1	0000 1

У таблиці 3 нижче показаний інший приклад бінарizaції для `pred_type`. У цьому прикладі B сегменти використовують ті самі бінарizaції, що і P сегменти з Таблиці 1. Таблиця 4 нижче показує додатковий приклад, де P сегменти і B сегменти використовують ті самі бінарizaції. Таблиці 2-4 призначені тільки для того, щоб показати приклади загальних бінарizaцій між P та B сегментами. Будь-які бінарizaції або правила бінарizaції можуть бути використані так, що синтаксичні елементи `pred_type` для P та B сегментів спільно використовують ті самі бінарizaції.

Відеокодер 20 і відеодекодер 30 можуть зберігати ті самі правила відображення і таблиці відображення (наприклад, як показано в таблицях 2-4) для використання з P та B сегментами. Кодування і декодування CABAC можуть бути застосовані до синтаксичного елемента `pred_type` з використанням цих відображень.

Таким чином, відеокодер 20 може бути сконфігурований, щоб визначати перший тип прогнозування для блока відеоданих у P сегменті, представляти перший тип прогнозування як синтаксичний елемент типу прогнозування P сегмента, визначати другий тип прогнозування для блока відеоданих у B сегменті, представляти другий тип прогнозування як синтаксичний елемент типу прогнозування B сегмента, визначати бінарizaцію P сегмента для синтаксичного елемента типу прогнозування P сегмента, визначати бінарizaцію B сегмента для синтаксичного елемента типу прогнозування B сегмента, причому синтаксичний елемент типу прогнозування P сегмента і синтаксичний елемент типу прогнозування B сегмента визначаються з використанням тієї самої логіки бінарizaції, і кодувати відеодані на основі бінарizaцій синтаксичного елемента типу прогнозування P сегмента і синтаксичного елемента типу прогнозування B сегмента.

Відеокодер 20 може бути додатково сконфігурований для бінаризації синтаксичного елемента типу прогнозування Р сегмента з визначеною бінаризацією Р сегмента, бінаризації синтаксичного елемента типу прогнозування В сегмента з визначеною бінаризацією В сегмента, застосування контекстно-адаптивного бінарного арифметичного кодування (CABAC) до бінаризованого синтаксичного елемента типу прогнозування Р сегмента і застосування контекстно-адаптивного бінарного арифметичного кодування (CABAC) до бінаризованого синтаксичного елемента типу прогнозування В сегмента.

Аналогічним чином відеокодер 30 може бути сконфігурований для відображення бінаризованого синтаксичного елемента типу прогнозування Р сегмента на тип прогнозування з використанням відображення бінаризації для блока відеоданих у Р сегменті, відображення бінаризованого синтаксичного елемента типу прогнозування В сегмента на тип прогнозування з використанням того самого відображення бінаризації для блока відеоданих у В сегменті і декодування відеоданих на основі відображених типів прогнозування.

Відеокодер 30 може бути додатково сконфігурований, щоб приймати підданий контекстно-адаптивному бінарному арифметичному кодуванню синтаксичний елемент типу прогнозування Р сегмента, який вказує тип прогнозування для блока відеоданих у Р сегменті, приймати підданий контекстно-адаптивному бінарному арифметичному кодуванню синтаксичний елемент типу прогнозування В сегмента, який вказує тип прогнозування для блока відеоданих у В сегменті, декодувати синтаксичний елемент типу прогнозування Р сегмента, щоб сформувати бінаризований синтаксичний елемент типу прогнозування Р сегмента, і декодувати синтаксичний елемент типу прогнозування В сегмента, щоб сформувати бінаризований синтаксичний елемент типу прогнозування В сегмента.

Таблиця 3

Бінаризація для pred_type в іншому прикладі даного розкриття

Тип сегмента	Значення pred_type	Режим прогнозування	Режим розділення	Рядок структурного елемента		
				cLog2CUsSize > Log2Min CUsSize (1)	cLog2CUsSize = 3 && !inter_4x4_enabled_flag (2)	cLog2CUsSize > 3 inter_4x4_enabled_flag (3)
I	0	MODE_INTRA	PART_2N×2N	-	1	1
	1	MODE_INTRA	PART_N×N	-	0	0
Р або В	0	MODE_INTRA	PART_2N×2N	0 1	0 1	0 1
	1	MODE_INTRA	PART_2N×N	0 011	0 01	0 01
	2	MODE_INTRA	PART_N×2N	0 0011	0 00	0 001
	4	MODE_INTRA	PART_2N×nU	0 0100	-	-
	5	MODE_INTRA	PART_2N×nD	0 0101	-	-
	6	MODE_INTRA	PART_nL×2N	0 00100	-	-
	7	MODE_INTRA	PART_nR×2N	0 00101	-	-
	3	MODE_INTRA	PART_N×N	-	-	0 000
	4	MODE_INTRA	PART_2N×2N	1	11	11
	5	MODE_INTRA	PART_N×N	-	10	10

Таблиця 4

Бінаризація для pred_type в іншому прикладі даного розкриття

Тип сегмента	Значення pred_type	Режим прогнозування	Режим розділення	Рядок структурного елемента		
				cLog2CUsSize > Log2Min CUsSize (1)	cLog2CUsSize = = Log2Min CUsSize cLog2CUsSize = = 3 && !inter_4x4_enabled_flag (2)	cLog2CUsSize > 3 inter_4x4_enabled_flag (3)
I	0	MODE_INTRA	PART_2N×2N	-	1	1
	1	MODE_INTRA	PART_N×N	-	0	0
P або B	0	MODE_INTRA	PART_2N×2N	1	1	1
	1	MODE_INTRA	PART_2N×N	011	01	01
	2	MODE_INTRA	PART_N×2N	001	00	001
	4	MODE_INTRA	PART_2N×nU	0100	-	-
	5	MODE_INTRA	PART_2N×nD	0101	-	-
	6	MODE_INTRA	PART_nL×2N	0000	-	-
	7	MODE_INTRA	PART_nR×2N	0001	-	-
	3	MODE_INTRA	PART_N×N	-	-	000
	4	MODE_INTRA	PART_2N×2N	000	000 0	0000 0
	5	MODE_INTRA	PART_N×N	-	000 1	0000 1

Таблиця 5 нижче показує продуктивність кодування з використанням загальної бінаризації для P та B сегментів, показаних в Таблиці 2. Як можна бачити з Таблиці 5, незначна ефективність кодування втрачається (або не втрачається взагалі) при використанні загальних бінаризацій. НЕ (висока ефективність) низької затримки P є загальною тестовою умовою для бінаризацій однонаправлено прогнозованих (P) сегментів. Класи A-E представляють різні розрізнення кадрів. Клас A відповідає розрізненню 2k × 4k. Клас B відповідає розрізненню 1920 × 1080. Клас C відповідає розрізненню WVGA. Клас D відповідає розрізненню WQVGA. Клас E відповідає розрізненню 720P. Зміна від 0,1 до 0,2 відсотка при тестовій умові НЕ низької затримки P, як правило, вважається незначною.

Таблиця 5

Продуктивність кодування для єдиної бінаризації за pred_type

частота BD	НЕ низької затримки P		
	Y	U	V
Клас A			
Клас B	0,02 %	0,16 %	0,26 %
Клас C	0,01 %	0,05 %	-0,12 %
Клас D	-0,02 %	-0,10 %	-0,12 %
Клас E	0,02 %	0,03 %	0,05 %
Всього	0,01 %	0,04 %	0,03 %
код. T [%]			
декод. T [%]			

Опційно ті самі бінаризації (не обмежуючись таблицями 2-4) для типу прогнозування (включає розмір прогнозування і/або режим прогнозування) можуть бути спільно використані в двох і більше різних типах сегментів інтер- прогнозування. Сегменти інтер-прогнозування можуть включати в себе, без обмеження вказаним:

- P сегмент: сегмент підтримує тільки однонаправлене прогнозування руху;
- B сегмент: сегмент підтримує однонаправлене і двонаправлене прогнозування руху;
- У кодуванні відео, що масштабується: поліпшений шар може спільно використовувати ті самі бінаризації з базовим шаром;

d. У багатовидовому кодуванні: різні види (представлення) можуть спільно використовувати ті самі бінаризації.

Коли підтримується асиметричне розділення, чотири контексти, порівну розділені на два набори контекстів, використовуються для CABAC на останніх двох структурних елементах для сигналізації синтаксичного елемента `pred_type` для асиметричних розділень (тобто `PART_2N×nU`, `PART_2N×nD`, `PART_nL×2N`, `PART_nR×2N`). Залежно від того, чи виконане розділення в горизонтальному напрямку або вертикальному напрямку, застосовується один набір контекстів. Передостанній структурний елемент (тобто структурний елемент типу розділення; `part_mode`) визначає, чи має поточна CU симетричні розділення або асиметричні розділення. Останній структурний елемент (тобто структурний елемент розміру розділення; `part_mode`) визначає, чи є розміром першого розділення одна чверть або три чверті від розміру CU. Таблиця 6 показує приклад контекстів передостаннього (тип розділення) і останнього (розмір розділення) для синтаксичного елемента `pred_type`.

Таблиця 6

Контексти для останніх двох структурних елементів синтаксичного елемента `pred_type`

Структурний елемент	Контекст
Тип розділення (симетричний або асиметричний)	Набір 1 контекстів (2 контексти, один для вертикального розділення, 1 для горизонтального розділення)
Розмір розділення (перше розділення дорівнює $\frac{1}{4}$ CU або $\frac{3}{4}$ CU)	Набір 2 контекстів (2 контексти, один для $\frac{1}{4}$ CU і один для $\frac{3}{4}$ CU)

Дане розкриття пропонує використовувати один контекст для передостаннього структурного елемента (тобто структурного елемента типу розділення) і використовувати режим обходу на останньому структурному елементі (тобто структурному елементі розміру розділення). У результаті, число контекстів зменшується з 4 до 1. Таблиця 7 показує приклад контексту, що використовується згідно з даним прикладом даного розкриття. Таблиця 8 показує продуктивність кодування, зв'язану із змінами, що пропонуються. Висока ефективність (HE) довільного доступу є тестовою умовою з кадрами довільного доступу. HE низької затримки B є тестовою умовою, яка допускає двонаправлене прогнозування.

Таблиця 7

Контексти для останніх двох структурних елементів синтаксичного елемента `pred_type` відповідно до прикладу даного розкриття

Структурний елемент	Контекст
Тип розділення (симетричний або асиметричний)	Набір 1 контекстів (1 контекст)
Розмір розділення (перше розділення дорівнює $\frac{1}{4}$ CU або $\frac{3}{4}$ CU)	Режим обходу (жодного контексту)

Таблиця 8

Продуктивність кодування запропонованого способу для `pred_type`

Частота BD	HE всіх інтра			HE довільного доступу			HE низької затримки B		
	Y	U	V	Y	U	V	Y	U	V
Клас A				0,03 %	-0,17 %	-0,29 %			
Клас B				0,02 %	-0,03 %	0,04 %	0,01 %	0,00 %	0,24 %
Клас C				-0,01 %	-0,03 %	-0,02 %	-0,01 %	-0,03 %	0,02 %
Клас D				0,01 %	0,07 %	-0,05 %	0,01 %	0,06 %	0,03 %
Клас E							0,00 %	0,30 %	0,39 %
Всього				0,01 %	-0,04 %	-0,07 %	0,00 %	0,06 %	0,01 %
код. T [%]									
декод. T [%]									

Таким чином, згідно з цим прикладом, відекодер 20 може бути сконфігурований, щоб

визначати тип розділення для режиму прогнозування для блока відеоданих, кодувати структурний елемент типу розділення синтаксичного елемента типу прогнозування для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування з одним контекстом, причому один контекст є однаковим для будь-якого типу розділення, і

5 кодувати структурний елемент розміру розділення синтаксису типу прогнозування для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування в режимі обходу.

Крім того, згідно з цим прикладом, відеодекодер 30 може бути сконфігурований, щоб приймати синтаксичний елемент типу прогнозування для блока відеоданих, який був кодований з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC), причому синтаксичний елемент типу прогнозування включає в себе структурний елемент типу розділення, що представляє тип розділення, і структурний елемент розміру розділення, що представляє розмір розділення, декодувати структурний елемент типу розділення синтаксичного елемента типу прогнозування з використанням контекстно-адаптивного бінарного арифметичного кодування з одним контекстом, причому один контекст є однаковим для будь-якого типу розділення, і декодувати структурний елемент розміру розділення синтаксису типу прогнозування для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування в режимі обходу.

В іншому прикладі при кодуванні прямокутного типу розділення режим обходу або один контекст може бути використаний для структурного елемента, який вказує, чи є режимом розділення $PART_nL \times 2N$ або $PART_nR \times 2N$, або ж режимом є $PART_2N \times nU$, $PART_2N \times nD$. Використання режиму обходу або одного контексту застосовне, тому що імовірність будь-якого режиму розділення, що використовується, близька до 50 %. Також опційно режим обходу або один контекст може бути використаний для структурного елемента, який вказує, чи є режим симетричним розділенням або асиметричним розділенням.

Наступний приклад даного розкриття стосується сигналізації в режимі "злиття" інтер-прогнозування. У режимі злиття кодер інструктує декодер, через сигналізацію бітового потоку синтаксису прогнозування, копіювати вектор руху, опорний індекс (що ідентифікує опорне зображення, в даному списку опорних зображень, на яке вказує вектор руху) і напрямок прогнозування руху (яке ідентифікує список опорних зображень (Список 0 або Список 1), тобто з точки зору того, чи передує у часі опорний кадр або йде за поточним кадром) від вибраного вектора руху-кандидата для поточної частини зображення, яке повинне кодуватися. Це виконується за допомогою сигналізації в бітовому потоці індексу в список векторів руху-кандидатів, що ідентифікує вибраний вектор руху-кандидат (тобто конкретний кандидат просторового прогнозувальника вектора руху (MVP) або кандидат часового MVP).

Таким чином, в режимі злиття синтаксис прогнозування може включати в себе прапор, що ідентифікує режим (в цьому випадку режим злиття), та індекс (`merge_idx`), що ідентифікує вибраний вектор руху-кандидат. У деяких випадках вектор руху-кандидат буде в причинній частині по відношенню до поточної частини. Тобто вектор руху-кандидат буду вже декодованим декодером. Таким чином, декодер вже прийняв і/або визначив вектор руху, опорний індекс і напрямок руху прогнозування для причинної частини. Таким чином, декодер може просто витягнути вектор руху, опорний індекс і напрямок прогнозування руху, асоційовані з причинною частиною, з пам'яті і скопіювати ці значення як інформацію руху для поточної частини. Для відновлення блока в режимі злиття декодер одержує блок прогнозування, використовуючи одержану інформацію руху для поточної частини, і додає залишкові дані до блока прогнозування для відновлення кодованого блока.

У HM4.0 один з п'яти кандидатів злиття сигналізується, коли поточна PU знаходиться в режимі злиття. Зрізаний унарний код використовується для представлення синтаксичного елемента `merge_idx`. В одній пропозиції для HEVC, для CABAC, кожний структурний елемент використовує один контекст. Дане розкриття пропонує використовувати один контекст повторно в усіх чотирьох структурних елементах, як показано в таблиці 9.

Таблиця 9

Контексти для останніх двох структурних елементів синтаксичного елемента `pred_type`

Структурний елемент	Контекст
Структурний елемент 0-3 для <code>merge_idx</code>	Набір 1 контексту (той самий контекст для всіх структурних елементів)

Таблиця 10 показує продуктивність кодування, зв'язану з цим прикладом.

Таблиця 10

Продуктивність кодування запропонованого способу для merge_idx

Частота BD	HE всіх інтра			HE довільного доступу			HE низької затримки B		
	Y	U	V	Y	U	V	Y	U	V
Клас A				0,00 %	-0,20 %	-0,07 %			
Клас B				0,01 %	0,03 %	0,03 %	0,01 %	-0,08 %	-0,22 %
Клас C				0,00 %	0,04 %	0,00 %	0,00 %	-0,08 %	-0,09 %
Клас D				0,03 %	0,07 %	0,05 %	0,05 %	-0,24 %	0,44 %
Клас E							-0,04 %	0,08 %	0,78 %
Всього				0,01 %	-0,05 %	-0,02 %	-0,01 %	-0,09 %	0,17 %
код. T [%]									
декод. T [%]									

Опційно більш ніж один контекст може бути використаний в кодуванні індексу злиття, при цьому деякі структурні елементи спільно використовують той самий контекст, і деякі структурні елементи використовують різні контексти. Як один приклад тільки послідовні структурні елементи (bin) спільно використовують той самий контекст. Наприклад, bin2 та bin3 можуть спільно використовувати один контекст; bin2 та bin4 не можуть спільно використовувати один і той самий контекст, якщо тільки bin3 також не використовує цей контекст.

Як інший приклад припустимо, що загальна кількість структурних елементів індексу злиття дорівнює N (перший структурний елемент є bin0, останній структурний елемент є bin $N-1$). У порогів, thres_i , $i=1, \dots, y$, використовуються для визначення контексту, що спільно використовується в кодуванні індексу злиття. У цьому прикладі наступні правила вказують, як контексти спільно використовуються між структурними елементами:

1. $0 < Y < N$ (є менше порогів, ніж структурних елементів)
2. $\text{thres}_i < \text{thres}_{i+1}$
3. $0 < \text{thres}_1$
4. $\text{thres}_Y = N$
5. bin _{i} буде спільно використовувати один контекст, де $i = \{\text{thres}_Y, \dots, \text{thres}_{i+1}-1\}$

На основі цих правил попередній спосіб, в якому один контекст багаторазово використовується в усіх чотирьох структурних елементах, можна розглядати як один випадок, де $N=4$, $Y=1$, $\text{thres}_1=4$. Тому структурні елементи від 0 до 3 спільно використовують той самий контекст.

Інший приклад включає в себе встановлення $N=4$, $Y=2$, $\text{thres}_1=2$, $\text{thres}_2=4$. У цьому прикладі bin0 та bin1 спільно використовують одні і ті самі контексти, і bin2 та bin3 спільно використовують одні і ті самі контексти.

Прапор інтер-прогнозування (inter_pred_flag) визначає, чи використовується моно-прогнозування або бі-прогнозування для поточної PU. У деяких прикладах індекс контексту для прапора інтер-прогнозування дорівнює поточній глибині CU. Оскільки є чотири можливих глибини CU (0-3), то існує чотири можливих контексти для кодування inter_pred_flag.

Дане розкриття припускає, що індекс контексту, який використовується для вибору контексту для кодування inter_pred_flag, дорівнює поточній глибині CU (наприклад, рівень декомпозиції квадродрева для CU), але обмежений вибраним порогом (тобто меншим з поточної глибини CU або порога). Поріг може бути вибраний рівнем 2 в одному прикладі. Альтернативно індекс контексту може бути рівним максимальній глибині CU мінус поточну глибину CU і обмежений зверху вибраним порогом. Альтернативно зумовлена таблиця відображення може бути спроектована для вибору індексу контексту за заданою глибиною CU. Таблиця відображення може бути реалізована у вигляді набору логіки. У результаті три контексти використовуються для кодування синтаксичного елемента між inter_pred_flag.

Таблиця 11 показує продуктивність кодування, коли таблиця ініціалізації змінюється, але кількість контекстів не змінюється. Таблиця 12 показує продуктивність кодування способу, що пропонується, який зменшує кількість контекстів з 4 до 3.

Таблиця 11

Продуктивність кодування НМ4.0 з модифікованою ініціалізацією АВАС за `inter_pred_flag`

Частота BD	НЕ всіх інтра			НЕ довільного доступу			НЕ низької затримки В		
	Y	U	V	Y	U	V	Y	U	V
Клас А				0,03 %	-0,15 %	-0,11 %			
Клас В				-0,03 %	0,01 %	-0,03 %	0,03 %	-0,02 %	0,11 %
Клас С				0,00 %	-0,12 %	0,06 %	-0,03 %	-0,16 %	0,01 %
Клас D				-0,01 %	-0,04 %	0,01 %	-0,09 %	0,51 %	0,20 %
Клас Е							0,10 %	-0,03 %	0,65 %
Всього				-0,01 %	-0,07 %	-0,02 %	0,00 %	0,07 %	0,14 %
код. Т [%]									
декод. Т [%]									

Таблиця 12

Продуктивність кодування запропонованого способу скорочення контексту за `inter_pred_flag`

Частота BD	НЕ всіх інтра			НЕ довільного доступу			НЕ низької затримки В		
	Y	U	V	Y	U	V	Y	U	V
Клас А				0,05 %	-0,11 %	-0,14 %			
Клас В				-0,01 %	-0,03 %	0,02 %	0,00 %	-0,01 %	0,15 %
Клас С				-0,02 %	-0,14 %	-0,02 %	0,01 %	0,01 %	0,03 %
Клас D				0,03 %	-0,01 %	-0,01 %	-0,09 %	-0,12 %	0,01 %
Клас Е							-0,12 %	0,08 %	0,45 %
Всього				0,01 %	-0,07 %	-0,03 %	-0,04 %	-0,01 %	0,03 %
код. Т [%]									
декод. Т [%]									

Індекс опорного кадру (`ref_idx_lx`) сигналізується за допомогою зрізаного унарного коду по відношенню до активного опорного кадру в асоційованому списку (наприклад, Список 0 або Список 1). Три контексти використовуються для кодування індексу опорного кадру. Один контекст - для структурного елемента 0, один контекст - для структурного елемента 1 і один контекст використовується для решти частини структурних елементів. Таблиця 13 показує приклад призначень контексту для структурних елементів унарного коду для `ref_idx_lx`.

Таблиця 13

Призначення контексту для структурних елементів `ref_idx_lx`

Структурні елементи унарного коду <code>ref_idx_lx</code>	Контекст
Структурний елемент 0	Контекст 1
Структурний елемент 1	Контекст 2
Структурні елементи 2-N (N - загальна кількість структурних елементів)	Контекст 3

Дане розкриття пропонує використання двох контекстів для кодування унарного коду для `ref_idx_lx`: один контекст для структурного елемента 0 і інший контекст для решти структурних елементів. Таблиця 14 показує приклад призначення контексту для структурних елементів унарного коду для `ref_idx_lx` згідно з цим прикладом даного розкриття. Таблиця 15 показує продуктивність кодування, асоційовану із запропонованими модифікаціями.

Таблиця 14

Призначення контексту для структурних елементів ref_idx_lx

Структурні елементи унарного коду ref_idx_lx	Контекст
Структурний елемент 0	Контекст 1
Структурні елементи 1-N (N - загальна кількість структурних елементів)	Контекст 2

Таблиця 15

Продуктивність кодування запропонованого способу на ref_idx_lx

Частота BD	НЕ всіх інтра			НЕ довільного доступу			НЕ низької затримки В		
	Y	U	V	Y	U	V	Y	U	V
Клас А				-0,01 %	-0,11 %	-0,16 %			
Клас В				-0,01 %	0,00 %	-0,01 %	0,01 %	-0,12 %	0,01 %
Клас С				-0,01 %	0,02 %	0,03 %	-0,04 %	-0,04 %	-0,07 %
Клас D				0,03 %	0,06 %	0,11 %	-0,06 %	0,19 %	-0,09 %
Клас Е							-0,06 %	-0,34 %	0,48 %
Всього				0,00 %	-0,01 %	-0,01 %	-0,03 %	-0,09 %	0,05 %
код. Т [%]									
декод. Т [%]									

Для синтаксичних елементів прапора кодованого блока кольоровості (cbf_cb та cbf_cr) два різних набори контекстів (5 контекстів у кожному наборі контекстів) використовуються для САВАС. Індекс фактичного контексту, що використовується в кожному наборі, дорівнює поточній глибині перетворення, асоційованій з прапором кодованого блока кольоровості, який кодується. Таблиця 16 показує набори контекстів для прапорів кодованого блока кольоровості cbf_cb та cbf_cr.

Таблиця 16

Набори контекстів для cbf_cb та cbf_cr

Прапор кодованого блока кольоровості	Набір контекстів
cbf_cb	Набір контекстів 1 (5 контекстів)
cbf_cr	Набір контекстів 2 (5 контекстів)

Дане розкриття пропонує, що cbf_cb та cbf_cr спільно використовують один набір контекстів. Індекс дійсного контексту, що використовується в кожному наборі, може все ще бути рівним поточній глибині перетворення, асоційованій з прапором кодованого блока кольоровості, який кодується. Таблиця 17 показує набори контекстів для прапорів кодованого блока кольоровості cbf_cb та cbf_cr згідно з прикладами даного розкриття. Таблиця 18 показує продуктивність кодування, асоційовану із запропонованими модифікаціями.

Таблиця 17

Набори контекстів для cbf_cb та cbf_cr згідно з прикладами даного розкриття

Прапор кодованого блока кольоровості	Набір контекстів
cbf_cb	Набір контекстів 1 (5 контекстів)
cbf_cr	Набір контекстів 1 (5 контекстів)

Таблиця 18

Продуктивність кодування запропонованого способу на cbf_cb та cbf_cr

Частота BD	HE всіх інтра			HE довільного доступу			HE низької затримки B		
	Y	U	V	Y	U	V	Y	U	V
Клас A	-0,01 %	0,59 %	-1,06 %	0,02 %	0,56 %	-1,71 %			
Клас B	0,00 %	0,59 %	-1,07 %	-0,01 %	0,68 %	-1,32 %	0,01 %	1,06 %	1,89 %
Клас C	-0,01 %	0,17 %	-0,75 %	0,00 %	0,09 %	-0,63 %	-0,01 %	0,21 %	0,97 %
Клас D	0,00 %	0,17 %	-0,51 %	0,04 %	-0,23 %	-0,80 %	0,04 %	-0,36 %	0,45 %
Клас E	0,00 %	0,36 %	-0,24 %				0,04 %	0,36 %	0,40 %
Всього	0,00 %	0,21 %	-0,70 %	0,01 %	0,30 %	-1,13 %	0,02 %	0,36 %	0,87 %
код. T [%]									
декод. T [%]									

Таким чином, згідно з цим прикладом, як відеокодер 20, так і відеодекодер 30 можуть бути сконфігуровані, щоб кодувати прапор кодованого блока кольоровості Cb для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC), причому CABAC використовує набір контекстів, що включає в себе один або більше контекстів, і кодувати прапор кодованого блока кольоровості Cr з використанням CABAC, причому CABAC використовує той самий набір контекстів, як і для прапора кодованого блока кольоровості Cb. Відеокодер 20 і відеодекодер 30 можуть бути додатково сконфігуровані для вибору контексту з одного або більше контекстів на основі глибини перетворення блока перетворення блока, асоційованого з блоком відеоданих.

В одній пропозиції для HEVC є дванадцять наборів контекстів як для coeff_abs_level_greater1_flag, так і coeff_abs_level_greater2_flag. coeff_abs_level_greater1_flag вказує, чи має коефіцієнт перетворення абсолютне значення більше, ніж 1. coeff_abs_level_greater2_flag вказує, чи має коефіцієнт перетворення абсолютне значення більше, ніж 2. Набори контекстів однаково призначаються для компонента яскравості і кольоровості, тобто 6 наборів контекстів для яскравості і 6 контекстів для кольоровості. Кожний набір контекстів складається з 5 контекстів. Індекс набору контекстів, ctxSet, вибраний на основі попереднього рівня coeff_abs_level_greater1_flag. Для coeff_abs_level_greater1_flag індекс контексту в наборі контекстів, greater1Ctx, визначається на основі кінцевих одиниць до максимум 4. Індекс контексту може бути представлений у вигляді:

$$\text{ctxIdx_level_greater1} = (\text{ctxSet} * 5) + \text{Min}(4, \text{greater1Ctx}) \quad (1)$$

Для coeff_abs_level_greater2_flag індекс контексту в наборі контекстів, greater2Ctx, ґрунтується на числі coeff_abs_level_greater1_flag, що складає від 1 до максимум 4. Індекс контексту може бути представлений у вигляді:

$$\text{ctxIdx_level_greater2} = (\text{ctxSet} * 5) + \text{Min}(4, \text{greater2Ctx}) \quad (2)$$

greater1Ctx оснований на числі для значущих коефіцієнтів і числі коефіцієнтів, які більше 1. З іншого боку, greater2Ctx оснований на числі коефіцієнтів, які більше, ніж 1.

У деяких прикладах різне число контекстів може бути використане в різних наборах контекстів, в тому числі, наприклад:

1. Набори контекстів для рівня більшого, ніж 1, або рівня більшого, ніж 2, можуть мати різне число контекстів. Наприклад, набори контекстів 0 та 3 можуть мати 5 контекстів, а інші набори контекстів могли б мати 2 контексти.

2. Набори контекстів для коефіцієнта яскравості можуть мати різне число контекстів в порівнянні з наборами контекстів для компонента кольоровості. Наприклад, набір контекстів 0 для яскравості може мати 5 контекстів, а набір контекстів 0 для кольоровості може мати 4 контексти.

3. Набір контекстів для рівня більшого, ніж 1, може мати відмінне число контекстів, в порівнянні з набором контекстів для рівня більшого, ніж 2. Наприклад, набір контекстів 0 для рівня більшого, ніж 1, може мати 5 контекстів, і набір контекстів 0 для рівня більшого, ніж 2, міг би мати тільки 3 контексти.

В інших прикладах може бути використане інше число для наборів контекстів для кодування більше 1 або більше 2, в тому числі, наприклад:

1. Набори контекстів для коефіцієнта яскравості можуть мати відмінне число наборів контекстів по відношенню до наборів контекстів, що використовуються для компонента кольоровості. Наприклад, яскравість могла б використовувати 6 контекстів, а кольоровість могла б використовувати 4 контексти.

5 2. Набори контекстів для більше 1 можуть мати інше число контекстів по відношенню до наборів контекстів, що використовуються більше 2. Наприклад, у випадку більше 1 могло б використовуватися 6 контекстів, а у випадку більше 2 могло б використовуватися 4 контексти.

Опційно використовується метрика для визначення того, який контекст використовується в наборі контекстів, і діапазон значення метрики більше, ніж число контекстів у наборі контекстів. В одному такому аспекті один контекст міг би бути асоційований з одним або більше значень метрики. Спільне використання контексту, переважно, обмежене безперервними значеннями. Наприклад, нехай значення метрики дорівнює y . $y=2$ асоційоване з контекстом 3, і $y=1$ і $y=4$ також можуть бути асоційовані з контекстом 3. Однак якщо $y=3$ асоційоване з контекстом 4, $y=4$ не може бути асоційоване з контекстом 3.

15 Наприклад, для `coeff_abs_level_greater1_flag` набори контекстів 0 та 3 мають 5 контекстів, а набори контекстів 1, 2, 4 та 5 мають 2 контексти. Для `coeff_abs_level_greater2_flag` набори контекстів 0, 1 та 2 мають 5 контекстів, а набори контекстів 3, 4 та 5 мають 2 контексти. Це може бути представлено у вигляді:

$ctxIdx_level_greater1 = (ctxSet * 5) + \text{Min}(\text{Thres_greater1}, \text{greater1Ctx})$ (3)

20 якщо $ctxSet=0$ або $ctxSet=3$, $\text{Thres_greater1}=4$;

в іншому випадку, $\text{Thres_greater1}=1$

$ctxIdx_level_greater2 = (ctxSet * 5) + \text{Min}(\text{Thres_greater2}, \text{greater2Ctx})$ (4)

якщо $ctxSet < 3$, $\text{Thres_greater2}=4$;

в іншому випадку, $\text{Thres_greater2}=1$

25 Thres_greater1 та Thres_greater2 можна вибрати різним чином залежно від наступних ситуацій:

1. Компонент яскравості або кольоровості,

2. Набори контекстів.

30 Як інший приклад для `coeff_abs_level_greater1_flag` набори контекстів 0 та 3 мають 5 контекстів, і набори контекстів 1, 2, 4 та 5 мають 3 контексти. Для `coeff_abs_level_greater2_flag` набори контекстів 0, 1 та 2 мають 5 контекстів, і набори контекстів 3, 4 та 5 мають 2 контексти. Це може бути представлено у вигляді:

$ctxIdx_level_greater1 = (ctxSet * 5) + \text{greater1Ctx_mapped}(3)$ $ctxIdx_level_greater2 = (ctxSet * 5) + \text{greater2Ctx_mapped}(4)$

35 У таких прикладах відображення може бути таким, як показано в Таблицях 19 та 20:

Таблиця 19

greater1Ctx	0	1	2	3	(3
ctxSet 0	0	1	2	3	4
ctxSet 1	0	1	1	2	2
ctxSet 2	0	1	1	1	2
ctxSet 3	0	1	2	3	4
ctxSet 4	0	1	2	2	2
ctxSet 5	0	1	1	2	2

Таблиця 20

greater2Ctx	0	1	2	3	(3
ctxSet 0	0	1	2	3	4
ctxSet 1	0	1	1	1	1
ctxSet 2	0	1	1	1	1
ctxSet 3	0	1	2	3	4
ctxSet 4	0	1	1	1	1
ctxSet 5	0	1	1	1	1

40 Таблиці ініціалізації CABAC для `coeff_abs_level_greater1_flag` та `coeff_abs_level_greater2_flag` також модифіковані для наборів контекстів для Thres_greater1 або Thres_greater2 , рівного 1. Модифікації переміщують ініціалізацію п'ятого контексту вперед, щоб

стати ініціалізацією другого контексту. Цей запропонований спосіб зменшує кількість контекстів від 120 до 78.

Таблиця 21

Продуктивність кодування запропонованого способу за
coeff_abs_level_greater1_flag та coeff_abs_level_greater1_flag

Часто-та BD	НЕ всіх інтра			НЕ довільного доступу			НЕ низької затримки В		
	Y	U	V	Y	U	V	Y	U	V
Клас А	-								
	0,00 %	0,04 %	0,03 %	0,05 %	0,31 %	-0,35 %			
Клас В	0,01 %	0,04 %	0,03 %	0,01 %	0,03 %	-0,09 %			-
							0,00 %	-0,15 %	0,23 %
Клас С	-0,01 %	0,05 %	0,00 %	0,03 %	0,06 %	0,06 %			-
							0,00 %	0,23 %	0,23 %
Клас D	0,00 %	0,01 %	-0,03 %	0,01 %	0,22 %	0,04 %	-0,01 %	0,26 %	0,24 %
	-								
Клас Е	0,00 %	0,02 %	-0,03 %				0,09 %	-0,52 %	0,16 %
Всього	0,00 %	0,01 %	0,01 %	0,02 %	0,15 %	-0,09 %	0,01 %	-0,02 %	0,04 %
код. Т [%]									
декод. Т [%]									

- 5 Таблиця 21 перераховує кількість контекстів для всіх синтаксичних елементів, згаданих у попередніх розділах. Загальне зниження становить 56 контекстів.

Таблиця 22

Порівняння кількості контекстів у запропонованому способі та HM4.0

# контекстів	HM4.0	Запропонований спосіб
Pred_type	10	6
Merge_idx	4	1
Inter_pred_flag	4	3
Ref_idx_l0, ref_idx_l1, ref_idx_l2	3	2
cbf_cb, cbf_cr	10	5
coeff_abs_level_greater1_flag	60	36
coeff_abs_level_greater2_flag	60	42
Всього	151	95

- 10 На фіг. 2 представлена блок-схема, що ілюструє зразковий відеокодер 20, який може реалізувати способи, описані в даному розкритті. Відеокодер 20 може виконувати інтра- та інтер-кодування блоків відео в сегментах відео. Інтра-кодування спирається на просторове прогнозування для зменшення або усунення просторової надмірності у відео протягом визначеного відеокадру або зображення. Інтер-кодування ґрунтується на часовому прогнозуванні для зменшення або усунення часової надмірності у відео в суміжних кадрах або зображеннях відеопослідовності. Інтра-режим (режим I) може відноситися до будь-якого з декількох режимів просторового стиснення. Інтер-режими, такі як однонаправлене прогнозування (режим P) або двонаправлене прогнозування (режим B), можуть відноситися до будь-якого з різних режимів часового стиснення.

- 20 У прикладі на фіг. 2 відеокодер 20 включає в себе модуль 35 розділення, модуль 41 прогнозування, пам'ять 64 опорних зображень, суматор 50, модуль 52 перетворення, модуль 54 квантування і модуль 56 ентропійного кодування. Модуль 41 прогнозування включає в себе модуль 42 оцінки руху, модуль 44 компенсації руху і модуль 46 інтра-прогнозування. Для відновлення блоків відео відеокодер 20 також включає в себе модуль 58 оберненого квантування, модуль 60 оберненого перетворення і суматор 62. Фільтр усунення блочності (не показаний на фіг. 2) також може бути включений, щоб фільтрувати межі блоків для усунення артефактів блочності з відновленого відео. За бажанням фільтр усунення блочності, як

правило, фільтрує вихідний сигнал суматора 62. Додаткові фільтри контуру (в контурі або після контуру) також можуть бути використані додатково до фільтра усунення блочності.

Як показано на фіг. 2, відеокoder 20 приймає відеодані, і модуль 35 розділення розділяє дані на блоки відео. Таке розділення може також включати розділення на сегменти (слайси), мозаїчні елементи (плитки) або інші великі одиниці, а також розділення блока відео, наприклад, відповідно до структури квадродререва LCU ти CU. Відеокoder 20 звичайно ілюструє компоненти, які кодують блоки відео в сегменті відео, що підлягає кодуванню. Сегмент може бути розділений на декілька блоків відео (і, можливо, в набори блоків відео, які називаються плитками). Модуль 41 прогнозування може вибрати один з множини можливих режимів кодування, таких як один з множини режимів інтра-кодування або один з множини режимів інтер-кодування для поточного блока відео на основі результатів за помилками (наприклад, швидкість кодування і рівень спотворень). Модуль 41 прогнозування може надати в результаті інтра- або інтер-кодований блок на суматор 50 для формування залишкових даних блока і на суматор 62 для відновлення кодованого блока для використання як опорного зображення.

Модуль 46 інтра-прогнозування в модулі 41 прогнозування може виконувати кодування з інтра-прогнозуванням поточного блока відео по відношенню до одного або більше сусідніх блоків у тому самому кадрі або сегменті як поточний блок, що підлягає кодуванню, щоб забезпечити просторове стиснення. Модуль 42 оцінки руху і модуль 44 компенсації руху в модулі 41 прогнозування виконують кодування з інтер-прогнозуванням поточного блока відео по відношенню до одного або більше блоків прогнозування в одному або більше опорних зображень, щоб забезпечити часове стиснення.

Модуль 42 оцінки руху може бути виконаний з можливістю визначення режиму інтер-прогнозування для сегмента відео відповідно до зумовленого шаблону для відеопослідовності. Зумовлений шаблон може визначити сегменти відео в послідовності як P сегменти, B сегменти або GPB (узагальнені P/B) сегменти. Модуль 42 оцінки руху і модуль 44 компенсації руху можуть бути з високим ступенем інтеграції, але проілюстровані окремо для концептуальних цілей. Оцінка руху, здійснювана модулем 42 оцінки руху, є процесом генерації векторів руху, який оцінює рух для блоків відео. Вектор руху, наприклад, може вказувати зміщення PU блока відео всередині поточного відеокadру або зображення по відношенню до блока прогнозування в опорному зображенні.

Блок прогнозування являє собою блок, який знаходиться в точній відповідності з PU кодованого блока відео в значенні піксельної різниці, який може бути визначений сумою абсолютних різниць (SAD), сумою квадратичних різниць (SSD) або іншими різницевиими метриками. У деяких прикладах відеокoder 20 може обчислювати значення для субцілих піксельних позицій опорних зображень, що зберігаються в пам'яті 64 опорних зображень. Наприклад, відеокoder 20 може інтерполювати значення в одну чверть піксельних позицій, одну восьму піксельних позицій або інших дробових піксельних позицій опорного зображення. Таким чином, модуль 42 оцінки руху може виконувати пошук руху відносно повних піксельних позицій і дробових піксельних позицій і виводити вектор руху з дробовою піксельною точністю.

Модуль 42 оцінки руху обчислює вектор руху для PU блока відео в інтер-кодованому сегменті шляхом порівняння положення PU з положенням блока прогнозування опорного зображення. Опорне зображення може бути вибране з першого списку опорних зображень (список 0) або другого списку опорних зображень (Список 1), кожний з яких ідентифікує одне або більше опорних зображень, збережених в пам'яті 64 опорних зображень. Модуль 42 оцінки руху передає обчислений вектор руху на модуль 56 ентропійного кодування і модуль 44 компенсації руху.

Компенсація руху, здійснювана модулем 44 компенсації руху, може включати в себе витягання або генерацію блока прогнозування на основі вектора руху, що визначається оцінкою руху, можливо виконуючи інтерполяцію з субпіксельною точністю. Після прийому вектора руху для PU поточного блока відео, модуль 44 компенсації руху може визначити положення блока прогнозування, на який вказує вектор руху в одному зі списків опорних зображень. Відеокoder 20 формує залишковий блок відео шляхом віднімання піксельних значень блока прогнозування з піксельних значень поточного кодованого блока відео, утворюючи піксельні різниці значення. Піксельні різниці значення утворюють залишкові дані для блока і можуть включати різниці компоненти як яскравості, так і кольоровості. Суматор 50 являє собою компонент або компоненти, які виконують цю операцію віднімання. Модуль 44 компенсації руху також може генерувати синтаксичні елементи, асоційовані з блоками відео і сегментами відео, для використання відеокoderом 30 при декодуванні блоків відео сегментів відео.

Модуль 46 інтра-прогнозування може інтра-прогнозувати поточний блок, як альтернативу інтер-прогнозування, що виконується модулем 42 оцінки руху і модулем 44 компенсації руху, як

описано вище. Зокрема, модуль 46 інтра-прогнозування може визначити режим інтра-прогнозування, що використовується для кодування поточного блока. У деяких прикладах модуль 46 інтра-прогнозування може кодувати поточний блок, використовуючи різні режими інтра-прогнозування, наприклад, під час окремих проходів кодування, і модуль 46 інтра-прогнозування 46 (або модуль 40 вибору режиму в деяких прикладах) може вибрати відповідний режим інтра-прогнозування для використання з протестованих режимів. Наприклад, модуль 46 інтра-прогнозування може обчислювати значення співвідношення швидкість-спотворення шляхом аналізу співвідношення швидкість-спотворення для різних протестованих режимів інтра-прогнозування і вибрати режим інтра-прогнозування, що має кращі характеристики співвідношення швидкість-спотворення серед протестованих режимів. Аналіз швидкість-спотворення звичайно визначає величину спотворення (або помилку) між кодованим блоком і вихідним, некодованим блоком, який був закодований для одержання кодованого блока, а також швидкість передачі бітів (тобто кількість бітів), що використовується для генерації кодованого блока. Модуль 46 інтра-прогнозування може обчислювати відношення спотворень і швидкостей для різних кодованих блоків, щоб визначити, який режим інтра-прогнозування демонструє краще значення співвідношення швидкості-спотворення для блока.

У будь-якому випадку, після вибору режиму інтра-прогнозування для блока, модуль 46 інтра-прогнозування може надавати інформацію, яка вказує на вибраний режим інтра-прогнозування, на модуль 56 ентропійного кодування. Модуль 56 ентропійного кодування може кодувати інформацію, яка вказує вибраний режим інтра-прогнозування, відповідно до методів даного винаходу. Відеокодер 20 може включати в бітовий потік, що передається, дані конфігурації, які можуть включати в себе множину таблиць індексів режиму інтра- прогнозування і множину таблиць індексів модифікованого режиму інтра-прогнозування (які також називають таблицями відображення кодових слів), визначення контекстів кодування для різних блоків і вказівки найбільш імовірного режиму інтра-прогнозування, таблиці індексів режиму інтра-прогнозування і таблиць індексів модифікованого інтра-прогнозування, щоб використовувати для кожного з контекстів.

Після того як модуль 41 прогнозування генерує блок прогнозування для поточного блока відео за допомогою інтер-прогнозування або інтра-прогнозування, відеокодер 20 формує залишковий блок відео шляхом віднімання блока прогнозування з поточного блока відео. Залишкові відеодані в залишковому блоці можуть включатися в одну або більше TU і подаватися на модуль 52 перетворення. Модуль 52 перетворення перетворює залишкові відеодані в залишкові коефіцієнти перетворення, використовуючи перетворення, таке як дискретне косинусне перетворення (DCT) або концептуально подібне перетворення. Модуль 52 перетворення може перетворювати залишкові відеодані з піксельної ділянки в ділянці перетворення, наприклад частотну ділянку.

Модуль 52 перетворення може відправити одержані в результаті коефіцієнти перетворення на модуль 54 квантування. Модуль 54 квантування квантує коефіцієнти перетворення для подальшого зниження бітової швидкості. Процес квантування може зменшити бітову глибину, асоційовану з деякими або усіма коефіцієнтами. Ступінь квантування може бути модифікована шляхом налаштування параметра квантування. У деяких прикладах модуль 54 квантування може потім виконувати сканування матриці, що включає в себе квантовані коефіцієнти перетворення. Як альтернатива, модуль 56 ентропійного кодування може виконувати сканування. Як один приклад, методи кодування, описані в даному розкритті, можуть бути виконані повністю або частково модулем 56 ентропійного кодування. Однак аспекти даного опису не обмежені цим. Наприклад, методи кодування, описані в даному розкритті, можуть бути виконані компонентом відеокодера 20, не показаним на фіг. 2, таким як процесор або будь-який інший компонент. У деяких прикладах методи кодування даного розкриття можуть бути виконані одним з інших блоків або модулів, показаних на фіг. 2. У деяких інших прикладах методи кодування даного розкриття можуть бути виконані за допомогою комбінації блоків і модулів відеокодера 20. Таким чином, відеокодер 20 може бути сконфігурований для виконання зразкових методів, описаних у даному розкритті.

Після квантування модуль 56 ентропійного кодування кодує квантовані коефіцієнти перетворення. Наприклад, модуль 56 ентропійного кодування може виконувати контекстно-адаптивне кодування змінної довжини (CAVLC), контекстно-адаптивне бінарне арифметичне кодування (CABAC), основане на синтаксисі контекстно-адаптивне бінарне арифметичне кодування (SBAC), ентропійне кодування з розділенням інтервалу імовірності (PIPE) або іншу технологію ентропійного кодування або метод. Після ентропійного кодування модулем 56 ентропійного кодування кодований бітовий потік може передаватися на відеокодер 30 або архівуватися для подальшої передачі або витягання відеокодером 30. Блок 56 ентропійного

кодування також може ентропійно кодувати вектори руху та інші синтаксичні елементи для поточного кодованого сегмента відео.

В одному прикладі даного розкриття модуль 56 ентропійного кодування може бути сконфігурований, щоб визначати перший тип прогнозування для блока відеоданих у Р сегменті, представляти перший тип прогнозування як синтаксичний елемент типу прогнозування Р сегмента, визначати другий тип прогнозування для блока відеоданих у В сегменті, представляти другий тип прогнозування як синтаксичний елемент типу прогнозування В сегмента, визначати бінаризацію Р сегмента для синтаксичного елемента типу прогнозування Р сегмента, визначати бінаризацію В сегмента для синтаксичного елемента типу прогнозування В сегмента, причому синтаксичний елемент типу прогнозування Р сегмента і синтаксичний елемент типу прогнозування В сегмента визначаються з використанням тієї самої логіки бінаризації, і кодувати відеодані на основі бінаризації синтаксичного елемента типу прогнозування Р сегмента і синтаксичного елемента типу прогнозування В сегмента.

В іншому прикладі даного розкриття модуль 56 ентропійного кодування може бути сконфігурований, щоб визначати тип розділення для режиму прогнозування для блока відеоданих, кодувати структурний елемент типу розділення синтаксичного елемента типу прогнозування для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування з одним контекстом, причому один контекст є однаковим для будь-якого типу розділення, і кодувати структурний елемент типу розділення синтаксису типу прогнозування для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування в режимі обходу.

В іншому прикладі даного розкриття модуль 56 ентропійного кодування може бути сконфігурований, щоб кодувати прапор кодованого блока кольоровості C_b для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC), причому CABAC використовує набір контекстів, що включає в себе один або більше контекстів, і кодувати прапор кодованого блока кольоровості C_r з використанням CABAC, причому CAABC використовує той самий набір контекстів, що і для прапора кодованого блока кольоровості C_b . Відеокодер 20 і відеодекодер 30 можуть бути додатково сконфігуровані, щоб вибирати контекст з одного або більше контекстів на основі глибини перетворення модуля перетворення, асоційованого з блоком відеоданих.

Модуль 58 оберненого квантування і модуль 60 оберненого перетворення застосовують обернене квантування та обернене перетворення, відповідно, для відновлення залишкового блока в піксельній ділянці для подальшого використання як опорного блока опорного зображення. Модуль 44 компенсації руху може обчислювати опорний блок шляхом підсумовування залишкового блока з блоком прогнозування одного з опорних зображень в одному зі списків опорних зображень. Модуль 44 компенсації руху може також застосовувати один або більше інтерполяційних фільтрів до відновленого залишкового блока для обчислення суб-цілих піксельних значень для використання в оцінці руху. Суматор 62 підсумовує відновлений залишковий блок із скомпенсованим за рухом блоком прогнозування, сформованим за допомогою модуля 44 компенсації руху, щоб одержати опорний блок для збереження в пам'яті 64 опорних зображень. Опорний блок може бути використаний модулем 42 оцінки руху і модулем 44 компенсації руху як опорний блок для інтер-прогнозування блока в подальшому відеокадрі або зображенні.

На фіг. 3 представлена блок-схема, що ілюструє зразковий відеодекодер 30, який може реалізувати способи, описані в даному розкритті. У прикладі на фіг. 3 відеодекодер 30 включає в себе модуль 80 ентропійного декодування, модуль 81 прогнозування, модуль 86 оберненого квантування, модуль 88 оберненого перетворення, суматор 90 і пам'ять 92 опорних зображень. Модуль 81 прогнозування включає в себе модуль 82 компенсації руху і модуль 84 інтра-прогнозування 84. Відеодекодер 30 може в деяких прикладах виконувати прохід декодування, загалом, оберненим чином до проходження кодування, описаного з посиланням на відеокодер 20 за фіг. 2.

Під час процесу декодування відеодекодер 30 приймає бітовий потік кодованого відео, який представляє блоки відеосегмента кодованого відео і асоційовані синтаксичні елементи, з відеокодера 20. Модуль 80 ентропійного декодування відеокодера 30 ентропійно декодує бітовий потік для генерації квантованих коефіцієнтів, векторів руху, а також інших синтаксичних елементів. Модуль 80 ентропійного декодування направляє вектори руху та інші синтаксичні елементи в модуль 81 прогнозування. Відеодекодер 30 може одержувати синтаксичні елементи на рівні сегмента відео і/або на рівні блока відео.

Як один приклад методи кодування, описані в даному розкритті, можуть бути виконані повністю або частково за допомогою модуля 80 ентропійного декодування. Однак аспекти

даного розкриття не обмежені цим. Наприклад, методи кодування, описані в даному розкритті, можуть бути виконані компонентом відеодекодера 30, не показаним на фіг. 3, таким, як процесор або будь-який інший компонент. У деяких прикладах методи кодування даного розкриття можуть бути виконані одним з інших блоків або модулів, показаних на фіг. 3. У ще
 5 деяких інших прикладах методи кодування даного розкриття можуть бути виконані за допомогою комбінації блоків і модулів відеодекодера 30. Таким чином, відеодекодер 30 може бути сконфігурований для виконання зразкових методів, описаних в даному розкритті.

В одному прикладі даного розкриття модуль 80 ентропійного декодування може бути сконфігурований, щоб відображати бінаризований синтаксичний елемент типу прогнозування Р сегмента на тип прогнозування з використанням відображення бінаризації для блока відеоданих у Р сегменті, відображати бінаризований синтаксичний елемент типу прогнозування В сегмента на тип прогнозування з використанням того самого відображення бінаризації для блока відеоданих у В сегменті і декодувати відеодані на основі відображених типів прогнозування.

В одному прикладі даного розкриття модуль 80 ентропійного декодування може бути
 15 сконфігурований, щоб приймати синтаксичний елемент типу прогнозування для блока відеоданих, який був закодований з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC), причому синтаксичний елемент типу прогнозування включає в себе структурний елемент типу розділення, що представляє тип розділення, і структурний елемент розміру розділення, що представляє розмір розділення, декодувати
 20 структурний елемент типу розділення синтаксичного елемента типу прогнозування з використанням контекстно-адаптивного бінарного арифметичного кодування з одним контекстом, причому один контекст є однаковим для будь-якого типу розділення, і декодувати структурний елемент розміру розділення синтаксичного елемента типу прогнозування для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування
 25 в режимі обходу.

В іншому прикладі даного розкриття модуль 80 ентропійного декодування може бути сконфігурований, щоб кодувати прапор кодованого блока кольоровості С_b для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC), причому CABAC використовує набір контекстів, що включає в себе один або більше контекстів, і
 30 кодувати прапор кодованого блока кольоровості С_r з використанням CABAC, причому CABAC використовує той самий набір контекстів, що і для прапора кодованого блока кольоровості С_b. Відеокодер 20 і відеодекодер 30 можуть бути додатково сконфігуровані, щоб вибирати контекст з одного або більше контекстів на основі глибини перетворення модуля перетворення, асоційованого з блоком відеоданих.

Коли сегмент відео кодований як інтра-кодований (I) сегмент, модуль 84 інтра-прогнозування модуля 81 прогнозування може генерувати дані прогнозування для блока відео поточного сегмента відео на основі сигналізованого режиму інтра-прогнозування і даних з раніше декодованих блоків поточного кадру або зображення. Коли відеокадр кодований як
 40 інтер-кодований (тобто В, Р або GPB) сегмент, модуль 82 компенсації руху модуля 81 прогнозування формує блоки прогнозування для блока відео поточного сегмента відео на основі векторів руху та інших синтаксичних елементів, одержаних з модуля 80 ентропійного декодування. Блоки прогнозування можуть бути сформовані з одного з опорних зображень в одному зі списків опорних зображень. Відеодекодер 30 може побудувати списки опорних кадрів, список 0 і список 1, використовуючи методи побудови за умовчанням на основі опорних
 45 зображень, що зберігаються в пам'яті 92 опорних зображень.

Модуль 82 компенсації руху визначає інформацію прогнозування для блока відео поточного сегмента відео шляхом аналізу векторів руху та інших синтаксичних елементів і використовує інформацію прогнозування для одержання блоків прогнозування для поточного декодованого блока відео. Наприклад, модуль 82 компенсації руху використовує деякі з одержаних
 50 синтаксичних елементів для визначення режиму прогнозування (наприклад, інтра- або інтер-прогнозування), що використовується для кодування блоків відеосегмента відео, типу сегмента інтер-прогнозування (наприклад, В сегмента, Р сегмента або GPB сегмента), структурної інформації для одного або більше списків опорних зображень для сегмента, векторів руху для кожного інтер-кодованого блока відеосегмента, станів інтер-прогнозування для кожного інтер-кодованого блока відеосегмента та іншої інформації для декодування блоків відео в поточному
 55 сегменті відео.

Модуль 82 компенсації руху також може виконувати інтерполяцію на основі інтерполяційних фільтрів. Модуль 82 компенсації руху може використовувати інтерполяційні фільтри, як використовується відеокодером 20 під час кодування блоків відео, для обчислення
 60 інтерпольованих значень для суб-цілих пікселів опорних блоків. У цьому випадку модуль 82

компенсації руху може визначити інтерполяційні фільтри, використані відеокодером 20, з прийнятих синтаксичних елементів та використовувати інтерполяційні фільтри для одержання блоків прогнозування.

Модуль 86 оберненого квантування обернено квантує, тобто деквантує, квантовані коефіцієнти перетворення, надані в потоці бітів і декодовані модулем 80 ентропійного декодування. Процес оберненого квантування може включати використання параметра квантування, обчислених відеокодером 20 для кожного блока відео в сегменті відео, щоб визначити ступінь квантування і аналогічно ступінь оберненого квантування, які повинні застосовуватися. Модуль 88 оберненого перетворення застосовує обернене перетворення, наприклад обернене DCT, обернене цілочисельне перетворення або концептуально подібний процес оберненого перетворення до коефіцієнтів перетворення для одержання залишкових блоків у піксельній ділянці.

Після того, як модуль 82 компенсації руху генерує блок прогнозування для поточного блока відео на основі векторів руху та інших синтаксичних елементів, відеодекодер 30 формує декодований блок відео шляхом підсумовування залишкових блоків з модуля 88 оберненого перетворення з відповідними блоками прогнозування, згенерованими модулем 82 компенсації руху. Суматор 90 являє собою компонент або компоненти, які виконують цю операцію підсумовування. За бажанням фільтр усунення блочності може також застосовуватися для фільтрації декодованих блоків, щоб видалити артефакти блочності. Інші фільтри контуру (в контурі кодування або після контуру кодування) також можуть бути використані для згладжування піксельних переходів або поліпшення іншим чином якості відео. Декодовані блоки відео в даному кадрі або зображенні потім зберігаються в пам'яті 92 опорних зображень, в якій зберігаються опорні кадри, що використовуються для подальшої компенсації руху. Пам'ять 92 опорних зображень також зберігає декодовані відео для подальшого представлення на пристрої відображення, такому як пристрій 32 відображення на фіг. 1.

На фіг. 6 представлена блок-схема, що ілюструє приклад способу кодування відео згідно з даним розкриттям. Спосіб за фіг. 6 може бути реалізований відеокодером 20. Відеокодер 20 може бути сконфігурований, щоб визначати перший тип прогнозування для блока відеоданих у Р сегменті (602) і представляти перший тип прогнозування як синтаксичний елемент типу прогнозування Р сегмента (604). Відеокодер 20 може бути додатково сконфігурований, щоб визначати другий тип прогнозування для блока відеоданих у В сегменті (606) і представляти другий тип прогнозування як синтаксичний елемент типу прогнозування В сегмента (608). Синтаксичний елемент типу прогнозування Р сегмента і синтаксичний елемент типу прогнозування В сегмента визначають режим прогнозування і тип розділення. Режим прогнозування може включати в себе один з інтер-прогнозування і інтра-прогнозування. Тип розділення може включати в себе одне із симетричних розділень і асиметричних розділень.

Відеокодер 20 може бути додатково сконфігурований, щоб визначати бінаризацію Р сегмента для синтаксичного елемента типу прогнозування Р сегмента (610) і визначати бінаризацію В сегмента для синтаксичного елемента типу прогнозування В сегмента, причому синтаксичний елемент типу прогнозування Р сегмента і синтаксичний елемент типу прогнозування В сегмента визначаються з використанням тієї самої логіки бінаризації (612). Відеокодер 20 може потім кодувати відеодані на основі бінаризацій синтаксичного елемента типу прогнозування Р сегмента і синтаксичного елемента типу прогнозування В сегмента (614).

Кодування відеоданих може містити бінаризацію синтаксичного елемента типу прогнозування Р сегмента з визначеними бінаризаціями Р сегмента, бінаризацію синтаксичного елемента типу прогнозування В сегмента з визначеними бінаризаціями В сегмента, застосування контекстно-адаптивного бінарного арифметичного кодування (CABAC) до бінаризованого синтаксичного елемента типу прогнозування Р сегмента і застосування контекстно-адаптивного бінарного арифметичного кодування (CABAC) до бінаризованого синтаксичного елемента типу прогнозування В сегмента.

На фіг. 7 представлена блок-схема, що ілюструє приклад способу декодування відео згідно з даним розкриттям. Спосіб за фіг. 7 може бути реалізований за допомогою відеодекодера 30. Відеодекодер 30 може бути сконфігурований для прийому підданого контекстно-адаптивному бінарному арифметичному кодуванню синтаксичного елемента типу прогнозування Р сегмента, який вказує тип прогнозування для блока відеоданих у Р сегменті (702), і прийому підданого контекстно-адаптивному бінарному арифметичному кодуванню синтаксичного елемента типу прогнозування В сегмента, який вказує тип прогнозування для блока відеоданих у В сегменті (704). Синтаксичний елемент типу прогнозування Р сегмента і синтаксичний елемент типу прогнозування В сегмента визначають режим прогнозування і тип розділення. Режим

прогнозування може включати в себе один з інтер-прогнозування та інтра- прогнозування. Тип розділення може включати в себе одне з симетричних розділень та асиметричних розділень.

Відеодекодер 30 може бути додатково сконфігурований, щоб декодувати синтаксичний елемент типу прогнозування Р сегмента для одержання бінаризованого синтаксичного елемента типу прогнозування Р сегмента (706) і декодувати синтаксичний елемент типу прогнозування В сегмента для одержання бінаризованого синтаксичного елемента типу прогнозування В сегмента (708). Відеодекодер 30 може бути додатково сконфігурований, щоб відображати бінаризований синтаксичний елемент типу прогнозування Р сегмента на тип прогнозування з використанням відображення бінаризації для блока відеоданих у Р сегменті (710), і щоб відображати бінаризований синтаксичний елемент типу прогнозування В сегмента на тип прогнозування з використанням того самого відображення бінаризації для блока відеоданих у В сегменті (712). Відеодекодер 30 може потім декодувати відеодані на основі відображених типів прогнозування (714).

На фіг. 8 представлена блок-схема послідовності операцій, що ілюструє приклад способу кодування відео згідно з даним розкриттям. Спосіб за фіг. 8 може бути реалізований відеокодером 20. Відеокодер 20 може бути сконфігурований, щоб визначати тип розділення для режиму прогнозування для блока відеоданих (802) і кодувати структурний елемент типу розділення синтаксичного елемента типу прогнозування для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC) з одним контекстом (804). Один контекст є однаковим для будь-якого типу розділення. В одному прикладі тип розділення є асиметричним розділенням, і структурний елемент типу розділення вказує, чи є асиметричне розділення вертикально розділеним або горизонтально розділеним. Наприклад, структурний елемент розміру розділення вказує, чи є перше розділення рівним одній чверті розміру блока відеоданих або перше розділення дорівнює трьом чвертям розміру блока відеоданих.

Відеокодер 20 може бути додатково сконфігурований, щоб кодувати структурний елемент розміру розділення синтаксичного елемента типу прогнозування для блока відеоданих з використанням CABAC в режимі обходу (806).

На фіг. 9 представлена блок-схема послідовності операцій, що ілюструє приклад способу декодування відео згідно з даним розкриттям. Спосіб за фіг. 9 може бути реалізований за допомогою відеодекодера 30. Відеодекодер 30 може бути сконфігурований, щоб приймати синтаксичний елемент типу прогнозування для блока відеоданих, які були закодовані з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC), причому синтаксичний елемент типу прогнозування включає в себе структурний елемент типу розділення, що представляє тип розділення, і структурний елемент розміру розділення, що представляє розмір розділення (902). В одному прикладі тип розділення є асиметричним розділенням, і структурний елемент типу розділення вказує, чи є асиметричне розділення вертикально розділеним або горизонтально розділеним. Наприклад, структурний елемент розміру розділення вказує, чи є перше розділення рівним одній чверті розміру блока відеоданих або перше розділення дорівнює трьом чвертям розміру блока відеоданих.

Відеодекодер 30 може бути додатково сконфігурований, щоб декодувати структурний елемент типу розділення синтаксичного елемента типу прогнозування з використанням CABAC з одним контекстом, причому один контекст є тим самим для будь-якого типу розділення (904), і декодувати структурний елемент розміру розділення синтаксичного елемента типу прогнозування з використанням CABAC в режимі обходу (906).

На фіг. 10 представлена блок-схема послідовності операцій, що ілюструє приклад способу кодування відео згідно з даним розкриттям. Спосіб за фіг. 10 може бути реалізований відеокодером 20 або відеодекодером. Для цілей фіг. 10 відеокодер 20 і відеодекодер 30 буде згадуватися в сукупності як пристрій кодування відео. Відповідно до методів за фіг. 10 пристрій кодування відео може бути сконфігурований, щоб кодувати прапор кодованого блока кольоровості С_b для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC), причому кодування прапора кодованого блока кольоровості С_b включає в себе використання набору контекстів, що включає в себе один або більше контекстів як частину CABAC (1002), а також кодувати прапор кодованого блока кольоровості С_r з використанням CABAC, причому кодування прапора кодованого блока кольоровості С_r включає в себе використання того самого набору контекстів, що і для прапора кодованого блока кольоровості С_b, як частину CABAC (1004). В одному прикладі набір контекстів включає в себе 5 контекстів.

В одному опційному прикладі даного розкриття пристрій кодування відео може бути додатково сконфігурований, щоб вибирати контекст з одного або більше контекстів на основі глибини перетворення одиниці перетворення, асоційованої з блоком відеоданих (1006).

При роботі як відеокoder пристрій кодування відео може бути додатково сконфігурований, щоб сигналізувати кодований прапор кодованого блока кольоровості C_b у кодованому бітовому потоці відео і сигналізувати кодований прапор кодованого блока кольоровості C_r в кодованому бітовому потоці відео. При роботі як відеодекодер пристрій кодування відео може бути додатково сконфігурований, щоб приймати кодований прапор кодованого блока кольоровості C_b в кодованому бітовому потоці відео, а також приймати кодований прапор кодованого блока кольоровості C_r в кодованому бітовому потоці відео.

В одному або більше прикладах описані функції можуть бути реалізовані в апаратних засобах, програмному забезпеченні, мікропрограмному забезпеченні або будь-якій їх комбінації. При реалізації в програмному забезпеченні функції можуть зберігатися або передаватися як одна або більше інструкцій або код на зчитуваному комп'ютером носії і виконуватися блоком обробки на основі апаратних засобів. Зчитувані комп'ютером носії можуть включати в себе зчитувані комп'ютером носії інформації, що відповідає матеріальному носію, такому як носії зберігання даних, або середовище зв'язку, включаючи будь-яке середовище, яке сприяє передачі комп'ютерної програми з одного місця в інше, наприклад, відповідно до протоколу зв'язку. Таким чином, зчитувані комп'ютером носії звичайно можуть відповідати (1) матеріальному зчитуваному комп'ютером носію зберігання даних, який є нечасовим (нетранзитивним), або (2) комунікаційному середовищу, такому як сигнал або несуча хвиля. Носії зберігання даних можуть бути будь-якими доступними носіями, до яких можуть одержувати доступ один або більше комп'ютерів або один або більше процесорів для витягання інструкцій, коду і/або структур даних для реалізації методів, описаних в даному розкритті. Комп'ютерний програмний продукт може включати в себе зчитуваний комп'ютером носій.

Як приклад, а не обмеження, такі зчитувані комп'ютером носії зберігання даних можуть включати в себе RAM, ROM, EEPROM, CD-ROM або інший оптичний диск, накопичувач на магнітних дисках або інші магнітні запам'ятовуючі пристрої, флеш-пам'ять або будь-який інший носій, який може використовуватися для зберігання бажаного програмного коду в формі інструкцій або структур даних і до якого може звертатися комп'ютер. Крім того, будь-яке з'єднання коректно називати зчитуванням комп'ютером носієм. Наприклад, якщо інструкції передаються від веб-сайта, сервера або іншого віддаленого джерела за допомогою коаксимального кабелю, волоконно-оптичного кабелю, виті пари, цифрової абонентської лінії (DSL) або бездротових технологій, таких як інфрачервона, радіо і мікрохвильова, то коаксимальний кабель, волоконно-оптичний кабель, вита пара, DSL або бездротові технології, такі як інфрачервона, радіочастотна і мікрохвильова, включаються у визначення носія. Потрібно розуміти, однак, що зчитувані комп'ютером носії даних і засоби зберігання даних не включають з'єднання, несучі хвилі, сигнали або інші транзитивні середовища зберігання, а замість цього направлені на нетранзитивні, матеріальні носії. Диски (disk і disc), як використовується тут, включають в себе компакт-диск (CD), лазерний диск, оптичний диск, цифровий універсальний диск (DVD), гнучкий диск і Blu-Ray Disc, де магнітні диски (disk) звичайно відтворюють дані магнітним способом, в той час як оптичні диски (disc) відтворюють дані оптично за допомогою лазерів. Комбінації вищеперерахованого також повинні бути включені в обсяг зчитуваних комп'ютером носіїв.

Інструкції можуть виконуватися одним або більше процесорами, такими як один або більше цифрових сигнальних процесорів (DSP), мікропроцесорів загального призначення, спеціалізованими інтегральними схемами (ASIC), програмованими логічними матрицями (FPGA) або іншими еквівалентними інтегрованими або дискретними логічними схемами. Відповідно термін "процесор", як він використаний тут, може відноситися до будь-якої з вказаних вище структур або будь-якої іншої структури, придатної для реалізації описаних тут методів. Крім того, в деяких аспектах функціональність, описана тут, може бути представлена в спеціалізованих апаратних засобах і/або програмних модулях, сконфігурованих для кодування і декодування або включена в комбінований кодек. Крім того, ці методи можуть бути повністю реалізовані в одній або більше схемах або логічних елементах.

Методи даного розкриття можуть бути реалізовані в широкому спектрі пристроїв або апаратів, включаючи бездротовий телефон, інтегральної схеми (IC) або набір мікросхем (наприклад, чіпсет). Різні компоненти, модулі або блоки описані в даному розкритті, щоб підкреслити функціональні аспекти пристроїв, сконфігурованих для виконання розкритих методів, але не обов'язково вимагає реалізації різними модулями апаратних засобів. Швидше, як описано вище, різні модулі можуть бути об'єднані в блоці апаратних засобів кодека або

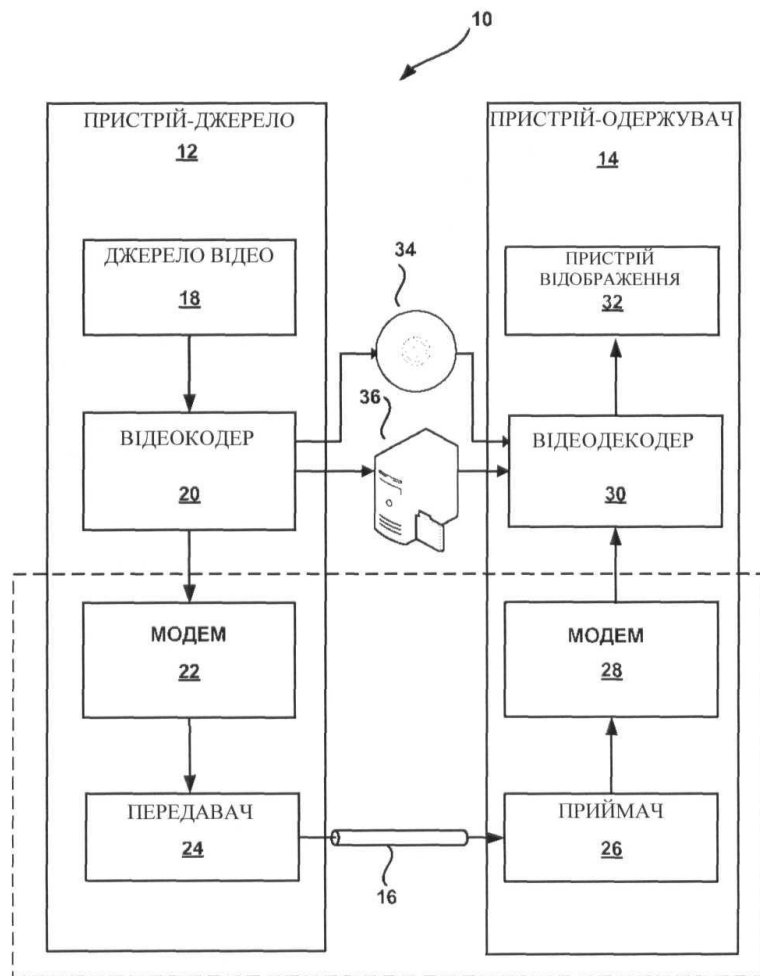
забезпечуються сукупністю взаємодіючих модулів апаратних засобів, в тому числі одного або більше процесорів, як описано вище, в поєднанні з відповідним програмним забезпеченням і/або програмно-апаратними засобами.

Різні приклади були описані. Ці та інші приклади входять в об'єм прикладеної формули винаходу.

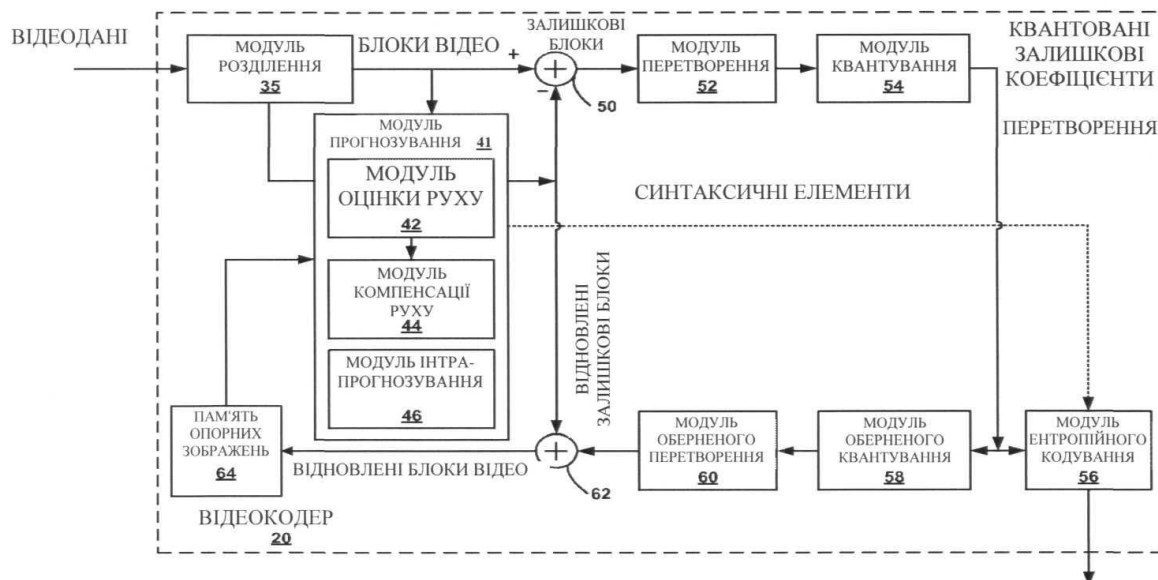
ФОРМУЛА ВИНАХОДУ

1. Спосіб кодування відеоданих, що включає:
 - 10 вибір контексту з одного або більше контекстів на основі глибини перетворення одиниці перетворення, асоційованої з блоком відеоданих;
 - кодування прапора кодованого блока кольоровості C_b для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC) і вибраного контексту, причому кодування прапора кодованого блока кольоровості C_b включає використання набору
 - 15 контекстів, що містить один або більше контекстів; і
 - кодування прапора кодованого блока кольоровості C_r з використанням CABAC, причому кодування прапора кодованого блока кольоровості C_r включає використання того самого набору контекстів, що і для прапора кодованого блока кольоровості C_b .
 2. Спосіб за п. 1, в якому набір контекстів містить 5 контекстів.
 - 20 3. Спосіб за п. 1, причому згаданий спосіб є способом кодування, при цьому спосіб додатково включає:
 - сигналізацію кодованого прапора кодованого блока кольоровості C_b в кодованому бітовому потоці відео; і
 - сигналізацію кодованого прапора кодованого блока кольоровості C_r в кодованому бітовому
 - 25 потоці відео.
 4. Спосіб за п. 1, причому згаданий спосіб є способом декодування, при цьому спосіб додатково включає:
 - прийом кодованого прапора кодованого блока кольоровості C_b в кодованому бітовому потоці відео; і
 - 30 прийом кодованого прапора кодованого блока кольоровості C_r в кодованому бітовому потоці відео.
 5. Пристрій, сконфігурований для кодування відеоданих, що містить: пристрій кодування відео, сконфігурований, щоб:
 - 35 вибирати контекст з одного або більше контекстів на основі глибини перетворення одиниці перетворення, асоційованої з блоком відеоданих;
 - кодувати прапор кодованого блока кольоровості C_b для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC) і вибраного контексту, причому кодування прапора кодованого блока кольоровості C_b включає використання набору контекстів, що включає в себе один або більше контекстів; і
 - 40 кодувати прапор кодованого блока кольоровості C_r з використанням CABAC, причому кодування прапора кодованого блока кольоровості C_r включає використання того самого набору контекстів, що і для прапора кодованого блока кольоровості C_b .
 6. Пристрій за п. 5, в якому набір контекстів включає в себе 5 контекстів.
 7. Пристрій за п. 5, в якому пристрій кодування відео є відеокодером, при цьому відеокодер
 - 45 додатково сконфігурований, щоб:
 - сигналізувати кодований прапор кодованого блока кольоровості C_b в кодованому бітовому потоці відео; і
 - сигналізувати кодований прапор кодованого блока кольоровості C_r в кодованому бітовому потоці відео.
 8. Пристрій за п. 5, в якому пристрій кодування відео є відеодекодером, при цьому відеодекодер
 - 50 додатково сконфігурований, щоб:
 - приймати кодований прапор кодованого блока кольоровості C_b в кодованому бітовому потоці відео; і
 - приймати кодований прапор кодованого блока кольоровості C_r в кодованому бітовому потоці
 - 55 відео.
 9. Пристрій, сконфігурований для кодування відеоданих, що містить:
 - засіб для вибору контексту з одного або більше контекстів на основі глибини перетворення одиниці перетворення, асоційованої з блоком відеоданих;
 - засіб для кодування прапора кодованого блока кольоровості C_b для блока відеоданих з
 - 60 використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC) і

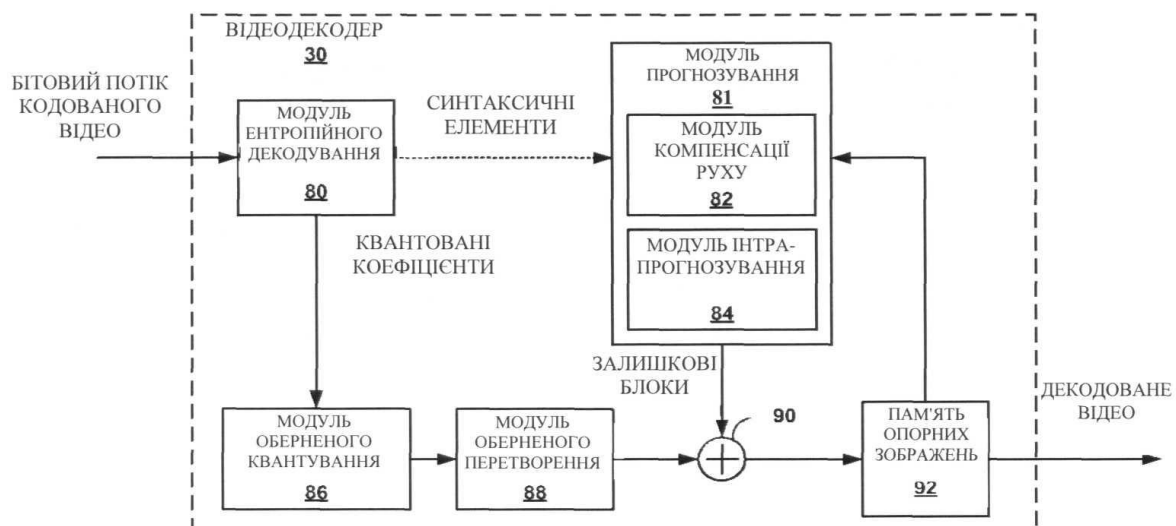
- вибраного контексту, причому кодування прапора кодованого блока кольоровості C_b включає використання набору контекстів, що містить один або більше контекстів; і
- засіб для кодування прапора кодованого блока кольоровості C_r з використанням CABAC, причому кодування прапора кодованого блока кольоровості C_r включає використання того
- 5 самого набору контекстів, що і для прапора кодованого блока кольоровості C_b .
10. Пристрій за п. 9, в якому набір контекстів включає в себе 5 контекстів.
11. Пристрій за п. 9, причому пристрій сконфігурований для кодування відеоданих, при цьому пристрій додатково містить:
- засіб для сигналізації кодованого прапора кодованого блока кольоровості C_b в кодованому
- 10 бітовому потоці відео; і
- засіб для сигналізації кодованого прапора кодованого блока кольоровості C_r в кодованому бітовому потоці відео.
12. Пристрій за п. 9, причому пристрій сконфігурований для декодування відеоданих, при цьому пристрій додатково містить:
- 15 засіб для прийому кодованого прапора кодованого блока кольоровості C_b в кодованому бітовому потоці відео; і
- засіб для прийому кодованого прапора кодованого блока кольоровості C_r в кодованому бітовому потоці відео.
13. Зчитуваний комп'ютером носій зберігання даних, який зберігає інструкції, які, при їх виконанні, спонукають один або більше процесорів, сконфігурованих, щоб кодувати відеодані:
- 20 вибирати контекст з одного або більше контекстів на основі глибини перетворення одиниці перетворення, асоційованої з блоком відеоданих;
- кодувати прапор кодованого блока кольоровості C_b для блока відеоданих з використанням контекстно-адаптивного бінарного арифметичного кодування (CABAC) і вибраного контексту,
- 25 причому кодування прапора кодованого блока кольоровості C_b включає використання набору контекстів, що містить один або більше контекстів; і
- кодувати прапор кодованого блока кольоровості C_r з використанням CABAC, причому кодування прапора кодованого блока кольоровості C_r включає використання того самого набору контекстів, що і для прапора кодованого блока кольоровості C_b .
- 30 14. Зчитуваний комп'ютером носій зберігання даних за п. 13, в якому набір контекстів містить 5 контекстів.
15. Зчитуваний комп'ютером носій зберігання даних за п. 13, причому один або більше процесорів сконфігурований, щоб кодувати відеодані, і при цьому інструкції додатково спонукають один або більше процесорів:
- 35 сигналізувати кодований прапор кодованого блока кольоровості C_b в кодованому бітовому потоці відео; і
- сигналізувати кодований прапор кодованого блока кольоровості C_r в кодованому бітовому потоці відео.
16. Зчитуваний комп'ютером носій зберігання даних за п. 13, причому один або більше процесорів сконфігуровані, щоб декодувати відеодані, і при цьому інструкції додатково спонукають один або більше процесорів:
- 40 приймати кодований прапор кодованого блока кольоровості C_b в кодованому бітовому потоці відео; і
- приймати кодований прапор кодованого блока кольоровості C_r в кодованому бітовому потоці
- 45 відео.



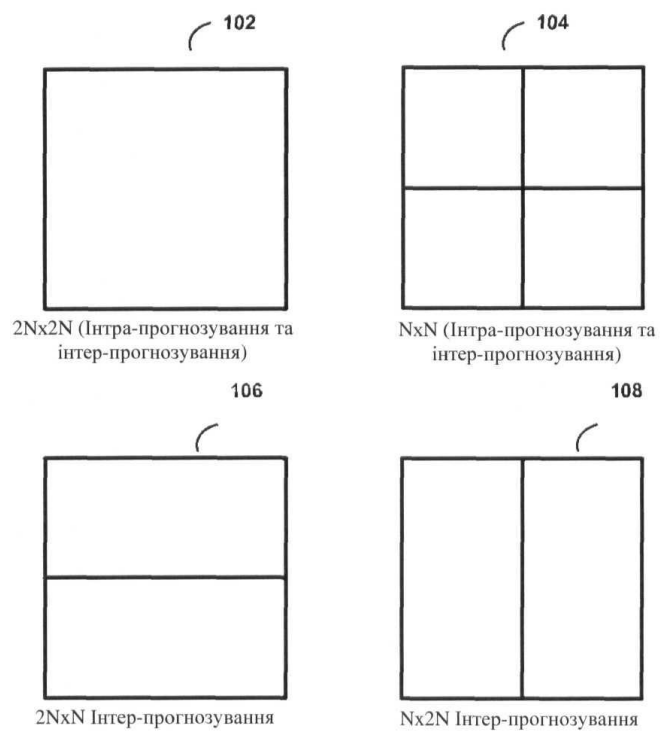
Фиг. 1



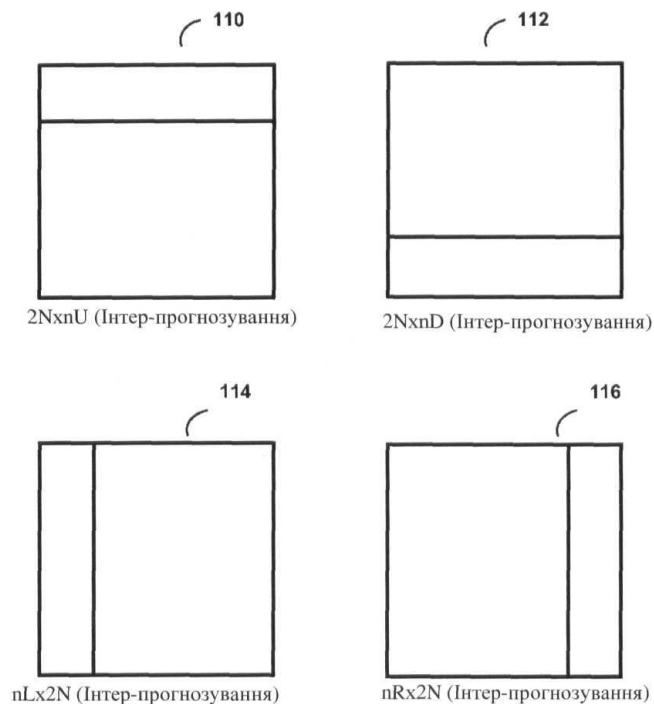
Фиг. 2



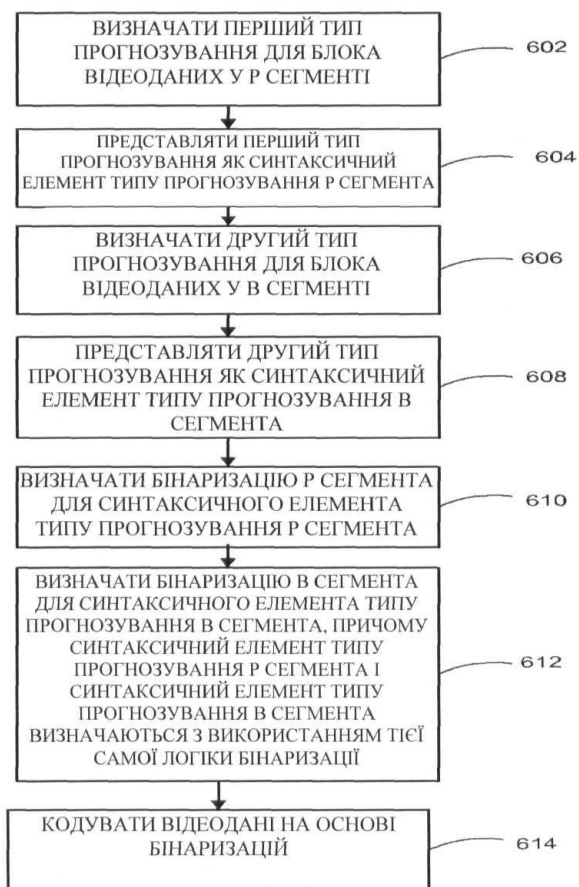
Фіг. 3



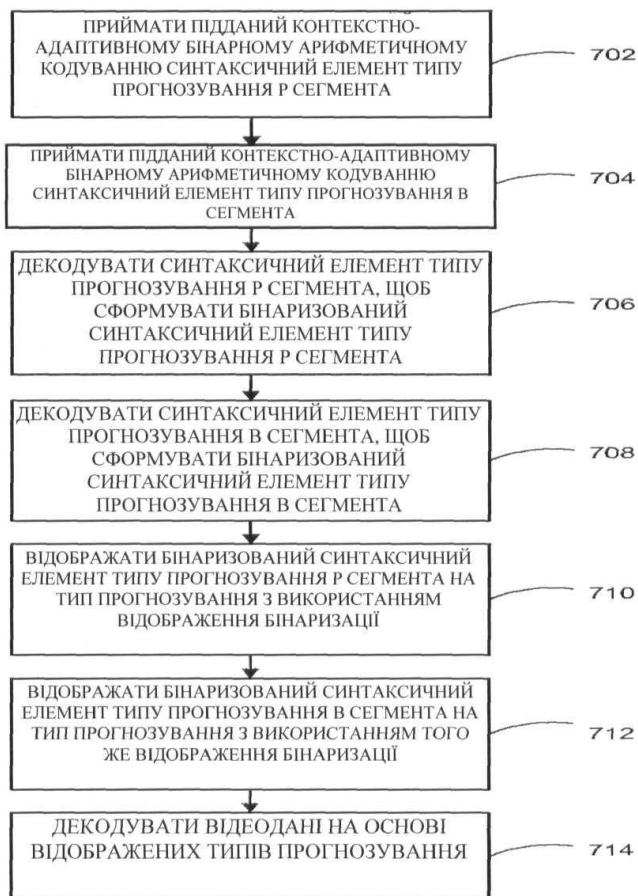
Фіг. 4



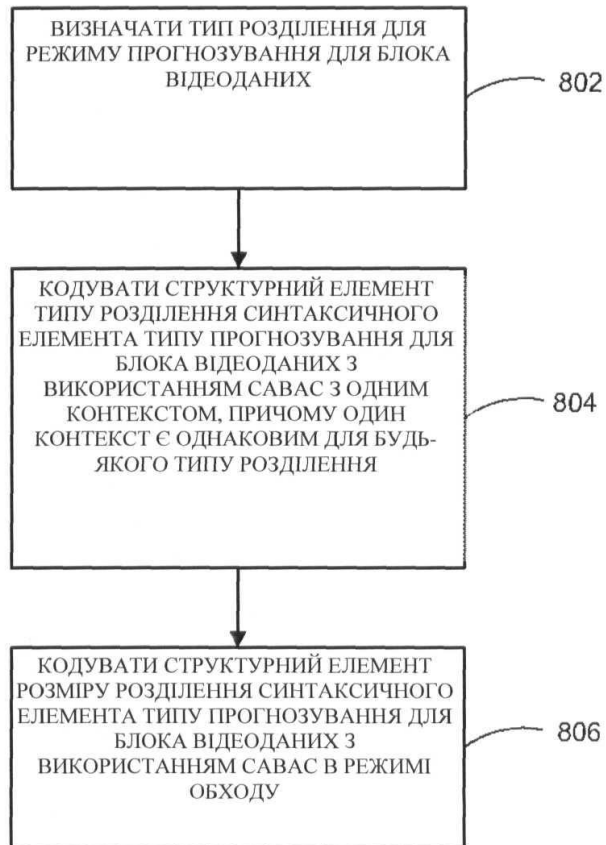
Фіг. 5



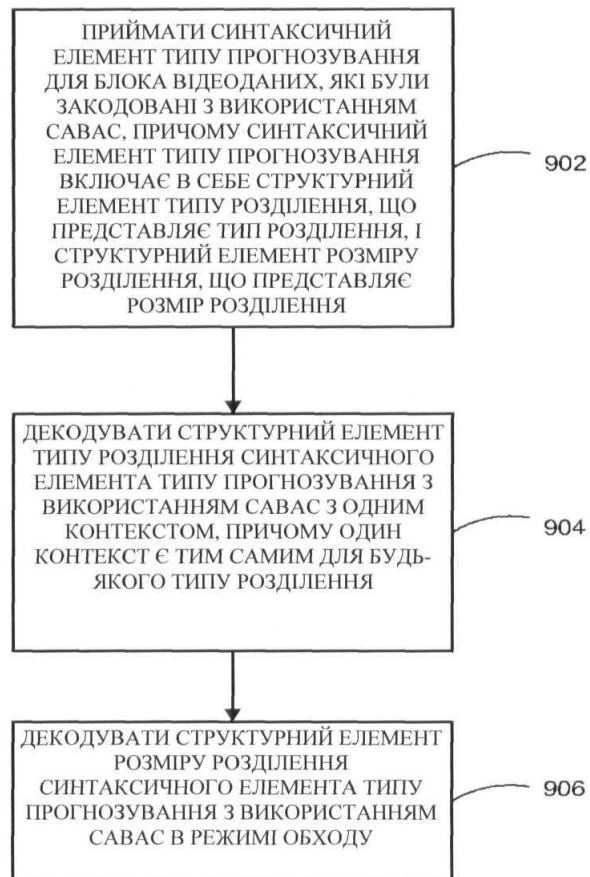
Фіг. 6



Фіг. 7



Фіг. 8



Фіг. 9

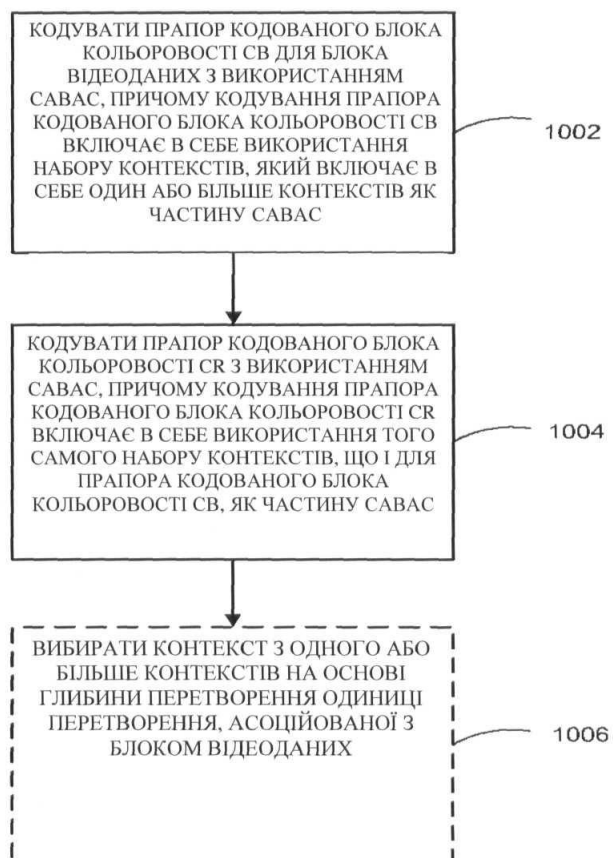


Fig. 10