



ДЕРЖАВНА СЛУЖБА  
ІНТЕЛЕКТУАЛЬНОЇ  
ВЛАСНОСТІ  
УКРАЇНИ

УКРАЇНА

(19) **UA** (11) **114103** (13) **C2**  
(51) МПК (2017.01)  
**H04N 7/00**

## (12) ОПИС ДО ПАТЕНТУ НА ВИНАХІД

(21) Номер заявки:	<b>а 2014 11260</b>	(72) Винахідник(и):	<b>Чень Ін (US), Ван Є-Куй (US), Чжан Лі (US)</b>
(22) Дата подання заявки:	<b>14.03.2013</b>	(73) Власник(и):	<b>КВЕЛКОММ ІНКОРПОРЕЙТЕД,</b> Attn: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121-1714, United States of America (US)
(24) Дата, з якої є чинними права на винахід:	<b>25.04.2017</b>	(74) Представник:	<b>Мошинська Ніна Миколаївна, реєстр. №115</b>
(31) Номер попередньої заявки відповідно до Паризької конвенції:	<b>61/663,484, 61/746,476, 13/801,731, 61/611,959, 61/624,990, 61/658,344</b>	(56) Перелік документів, взятих до уваги експертизою:	<b>WO 2008047303 A2, 24.04.2008</b> I-K KIM et al. Restriction on motion vector scaling for Merge and AMVP. 7. JCT-VC MEETING, 98. MPEG meeting. 21.11.2011 - 30.11.2011, Geneva. (Joint Collaborative Team on Video Coding of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16 ). no. JCTVC-G551, XP 030110535 Wiegand T. et al. WD3: Working Draft 3 of High-Efficiency Video Coding. 29.03.2011, no. JCTVC-E603, XP030009014 Hannuksela M.M., Gopalakrishna S. M. AHG21: On reference picture list construction and reference picture marking 7. JCT-VC meeting, 98. MPEG meeting. 21.11.2011 - 30.11.2011, Geneva, (Joint Collaborative Team on Video Coding of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16 ), Nr: JCTVC-G643, XP 030110627
(32) Дата подання попередньої заявки відповідно до Паризької конвенції:	<b>22.06.2012, 27.12.2012, 13.03.2013, 16.03.2012, 16.04.2012, 11.06.2012</b>		
(33) Код держави-учасниці Паризької конвенції, до якої подано попередню заявку:	<b>US, US, US, US, US</b>		
(41) Публікація відомостей про заявку:	<b>26.01.2015, Бюл.№ 2</b>		
(46) Публікація відомостей про видачу патенту:	<b>25.04.2017, Бюл.№ 8</b>		
(86) Номер та дата подання міжнародної заявки, поданої відповідно до Договору РСТ	<b>PCT/US2013/031573, 14.03.2013</b>		

## (54) СИНТАКСИЧНІ РОЗШИРЕННЯ ВИСОКОГО РІВНЯ ДЛЯ ВИСОКОЕФЕКТИВНОГО ВІДЕОКОДУВАННЯ

### (57) Реферат:

В одному прикладі, пристрій містить у собі відеокодер, виконаний з можливістю кодувати значення рахунка порядку зображення (РОС) для першого зображення відеоданих, кодувати ідентифікатор зображення другої розмірності для першого зображення і кодувати, відповідно до базової специфікації відеокодування або розширення базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення. Відеокодер може містити відеокодер і відеодекодер. Ідентифікатор зображення другої розмірності може містити, наприклад, ідентифікатор виду, індекс порядку виду, ідентифікатор рівня або інший подібний ідентифікатор. Відеокодер може кодувати значення РОС і ідентифікатор зображення другої

UA 114103 C2

розмірності під час кодування вектора руху для блока другого зображення, наприклад, під час удосконаленого прогнозування вектора руху або кодування в режимі злиття.



Фіг. 12

Дана заявка заявляє пріоритет заявки на патент США №61/611,959, поданої 16 березня 2012 року, заявки на патент США №61/624,990, поданої 16 квітня 2012 року, заявки на патент США №61/658,344, поданої 11 червня 2012 року, заявки на патент США №61/663,484, поданої 22 червня 2012 року, і заявки на патент США №61/746,476, поданої 27 грудня 2012 року, зміст яких повністю включений в дану заявку за допомогою посилання.

Галузь техніки, до якої належить винахід

Дане розкриття стосується кодування відео.

Рівень техніки

Можливості цифрового відео можуть бути включені в широкий ряд пристроїв, включаючи цифрові телевізори, цифрові системи прямого мовлення, бездротові системи мовлення, персональні цифрові помічники (PDA), портативні або настільні комп'ютери, планшети, пристрої для читання електронних книг, цифрові камери, цифрові записуючі пристрої, цифрові медіапрогравачі, пристрої відеоігор, ігрові відеоприставки, стільникові або супутникові радіотелефони, так звані "смартфони", пристрої для відео телеконференції, пристрої потокового відео тощо. Цифрові відеопристрої здійснюють способи відеокодування, як, наприклад, описані в стандартах, визначених MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, частина 10, Просунуте кодування (AVC) відеосигналу, стандарт Високоєфективного Відеокодування (HEVC), що знаходиться в розробці в даний час, і розширення цих стандартів. Відеопристрої можуть передавати, приймати, кодувати, декодувати і/або зберігати цифрову відеоінформацію ефективніше шляхом здійснення подібних способів відеокодування.

Способи відеокодування містять у собі просторове (із внутрішнім (intra-) кодуванням) прогнозування і/або часове (із зовнішнім (inter-) кодуванням) прогнозування для скорочення або усунення надмірності, властивої для послідовностей відеокадрів. Для кодування відео на основі блоків, відеослайс (наприклад, відеокадр або частина відеокадру) може бути розділений на відеоблоки, які також називають деревоподібними блоками, елементами (одинацями) (CU) кодування і/або вузлами кодування. Відеоблоки у внутрішньо-кодованому (I) слайсі зображення кодуються з використанням просторового прогнозування відносно опорних відліків у сусідніх блоках у тому ж зображенні. Відеоблоки у зовнішньокодованому (P або B) слайсі можуть використовувати просторове прогнозування відносно опорних відліків у сусідніх блоках у тому ж зображенні або часове прогнозування відносно опорних відліків в інших опорних зображеннях. Зображення можна також називати кадрами, а опорні зображення можна називати опорними кадрами.

Просторове або часове прогнозування має своїм результатом прогнозуючий блок для блока, який повинен бути кодований. Залишкові дані являють собою різниці пікселів між вихідним блоком, який повинен бути кодований, і прогнозуючим блоком. Зовнішньокодований блок кодується відповідно до вектора руху, що вказує на блок опорних відліків, які формують прогнозуючий блок, і залишковими даними, які вказують різницю між закодованим блоком і прогнозуючим блоком. Внутрішньокодований блок кодується відповідно до режиму внутрішнього кодування і залишкових даних. Для додаткового стиснення залишкові дані можуть бути перетворені з області пікселів в область перетворення, що приведе до залишкових коефіцієнтів перетворення, які потім можуть бути квантовані. Квантовані коефіцієнти перетворення, початково розміщені в двовимірному масиві, можуть бути скановані для відтворення одновимірного вектора коефіцієнтів перетворення, і може бути застосоване ентропійне кодування для досягнення ще більшого стиснення.

Розкриття винаходу

Загалом, дане розкриття описує різні способи підтримки розширень стандартів кодування, як, наприклад, Стандарт Високоєфективного Відеокодування, що розвивається (HEVC), зі зміною тільки синтаксису високого рівня. Наприклад, дане розкриття описує способи як у базовій специфікації HEVC, так і в розширеннях HEVC багатовидового відеокодека і/або тривимірного (3D) відеокодека, де базовий вид порівняний з базовою специфікацією HEVC. Загалом, "базова специфікація відеокодування" може відповідати специфікації відеокодування, що використовується для кодування двовимірних, однорівневих відеоданих. Розширення базової специфікації відеокодування можуть розширювати можливості базової специфікації відеокодування, щоб дозволити 3D і/або багаторівневе відеокодування. Базова специфікація HEVC являє собою приклад базової специфікації відеокодування, у той час як розширення MVC і SVC до базової специфікації HEVC являють собою приклади розширень базової специфікації відеокодування.

В одному прикладі, спосіб включає в себе декодування значення рахунка порядку зображення (POC) для першого зображення відеоданих, декодування ідентифікатора

зображення другої розмірності для першого зображення і декодування, відповідно до базової специфікації відеокодування, другого зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення. Ідентифікатор зображення другої розмірності може додатково бути спрощений до типу зображення, наприклад, чи є зображення довгостроковим або короткостроковим зображенням, або чи має зображення, якщо воно опорне зображення, те ж значення рахунка порядку зображення (РОС), що і зображення, яке посиляється на нього. При формуванні кандидатів векторів руху із сусідніх блоків, кандидат може вважатися недоступним, коли він має ідентифікатор зображення другої розмірності, відмінний від такого у вектора руху, який повинен бути прогнозований, ідентифікатор зображення другої розмірності якого - це зображення, на яке вказує цей вектор руху, і ідентифіковане індексом цільового посилання.

В іншому прикладі, спосіб включає в себе кодування значення рахунка порядку зображення (РОС) для першого зображення відеоданих, кодування ідентифікатора зображення другої розмірності для першого зображення і кодування, відповідно до базової специфікації відеокодування, другого зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

В іншому прикладі, пристрій містить у собі відеодекодер, виконаний з можливістю декодувати значення рахунка порядку зображення (РОС) для першого зображення відеоданих, декодувати ідентифікатор зображення другої розмірності для першого зображення і декодувати, відповідно до базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

В іншому прикладі, пристрій містить у собі відеокодер, виконаний з можливістю кодувати значення рахунка порядку зображення (РОС) для першого зображення відеоданих, кодувати ідентифікатор зображення другої розмірності для першого зображення і кодувати, відповідно до базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

В іншому прикладі, пристрій містить у собі засіб декодування значення рахунка порядку зображення (РОС) для першого зображення відеоданих, засіб декодування ідентифікатора зображення другої розмірності для першого зображення і засіб декодування, відповідно до базової специфікації відеокодування, другого зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

В іншому прикладі, пристрій містить у собі засіб для кодування значення рахунка порядку зображення (РОС) для першого зображення відеоданих, засіб для кодування ідентифікатора зображення другої розмірності для першого зображення і засіб для кодування, відповідно до базової специфікації відеокодування, другого зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

В іншому прикладі, зчитуваний комп'ютером носій даних, на якому зберігаються інструкції, які, при їхньому виконанні, спонукають процесор декодувати значення рахунка порядку зображення (РОС) для першого зображення відеоданих, декодувати ідентифікатор зображення другої розмірності для першого зображення і декодувати, відповідно до базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

В іншому прикладі, зчитуваний комп'ютером носій даних, на якому зберігаються інструкції, які, при їхньому виконанні, спонукають процесор кодувати значення рахунка порядку зображення (РОС) для першого зображення відеоданих, кодувати ідентифікатор зображення другої розмірності для першого зображення і кодувати, відповідно до базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

Деталі одного або декількох прикладів вказані в супровідних кресленнях і наступному описі. Інші особливості, цілі і переваги будуть очевидні з опису і креслень, а також з формули винаходу.

Короткий опис креслень

Фіг. 1 є блоковою діаграмою, яка зображує приклад системи кодування і декодування відео, що може використовувати способи кодування відеоданих відповідно до синтаксичного розширення високого рівня стандарту відеокодування.

Фіг. 2 є блоковою діаграмою, яка зображує приклад відеокодера, що може здійснювати способи кодування відеоданих відповідно до синтаксичного розширення високого рівня стандарту відеокодування.

Фіг. 3 є блоковою діаграмою, яка зображує приклад відеодекодера, що може здійснювати способи кодування відеоданих відповідно до синтаксичного розширення високого рівня стандарту відеокодування.

Фіг. 4 є концептуальною діаграмою, яка зображує приклад моделі прогнозування MVC.

5 Фіг. 5-9 є концептуальними діаграмами, які зображують потенційні проблеми, які повинні бути переборені для досягнення тільки синтаксичного розширення високого рівня HEVC.

Фіг. 10 є концептуальною діаграмою, яка зображує приклад групи сусідніх блоків для поточного блока для використання в прогнозуванні вектора руху.

10 Фіг. 11 є блок-схемою, яка зображує приклад способу кодування відеоданих, відповідно до способів даного розкриття.

Фіг. 12 є блок-схемою, яка зображує приклад способу декодування відеоданих, відповідно до способів даного розкриття.

Детальний опис

15 Загалом, дане розкриття описує різні способи підтримки розширень стандартів кодування, як, наприклад, Стандарт Високоєфективного Відеокодування, що розвивається (HEVC), зі зміною тільки синтаксису (HLS) високого рівня. Наприклад, дане розкриття описує способи як у базовій специфікації HEVC, так і в розширеннях HEVC багатовидового Відеокодування (MVC) і/або кодування тривимірного відеосигналу (3DV), де базовий вид порівнянний з базовою специфікацією HEVC.

20 Дане розкриття описує конкретні способи, які забезпечують профіль тільки синтаксису високого рівня в специфікації розширення HEVC. Термін "міжвидовий" у контексті MVC\3DV може бути замінений на "міжрівневий" у контексті Масштабованого Відеокодування (SCV). Тобто, хоча опис цих способів первинно фокусується на "міжвидовому" кодуванні, ті ж або подібні ідеї можна застосувати до "міжрівневих" опорних зображень для розширення тільки HLS SVC HEVC.

25 Фіг. 1 є блоковою діаграмою, яка зображує приклад системи 10 кодування і декодування відео, що може використовувати способи кодування відеоданих відповідно тільки до синтаксичного розширення високого рівня стандарту відеокодування. Як зображено на Фіг. 1, система 10 містить у собі пристрій-джерело 12, що забезпечує закодовані відеодані, які повинні 30 бути декодовані пізніше пристроєм-адресатом 14. Більш конкретно, пристрій-джерело 12 забезпечує відеодані пристрою-адресату 14 по зчитуваному комп'ютером носію 16. Пристрій-джерело 12 і пристрій-адресат 14 можуть містити будь-який із широкого діапазону пристроїв, включаючи настільні комп'ютери, ноутбуки (тобто, портативні комп'ютери), планшетні комп'ютери, телевізійні приставки, телефонні трубки, як, наприклад, так звані "смартфони", так звані "смартпади", телевізори, камери, пристрої відображення, цифрові медіаплеєри, ігрові відеоприставки, пристрій потокового відео або тому подібне. У деяких випадках, пристрій-джерело 12 і пристрій-адресат 14 можуть бути обладнані для бездротового зв'язку.

35 Пристрій-адресат 14 може приймати закодовані відеодані, які повинні бути декодовані, по зчитуваному комп'ютером носію 16. Зчитуваний комп'ютером носій 16 може містити будь-який 40 тип носія або пристрою, здатного переміщувати закодовані відеодані від пристрою-джерела 12 до пристрою-адресата 14. В одному прикладі, зчитуваний комп'ютером носій 16 може містити носій зв'язку для дозволу пристрою-джерелу 12 передавати закодовані відеодані прямо пристрою-адресату 14 у реальному часі. Закодовані відеодані можуть бути модульовані відповідно до стандарту зв'язку, як, наприклад, протокол бездротового зв'язку, і передані 45 пристрою-адресату 14. Носій зв'язку може містити будь-який бездротовий або дротовий носій зв'язку, як, наприклад, радіочастотний (RF) спектр або одна або кілька фізичних ліній передачі. Носій зв'язку може формувати частину пакетної мережі, як, наприклад, локальна мережа, регіональна мережа або глобальна мережа, як, наприклад, Інтернет. Носій зв'язку може містити в собі роутери, перемикачі, базові станції або будь-яке інше обладнання, яке може бути 50 використане для посилення зв'язку від пристрою-джерела 12 до пристрою-адресата 14.

У деяких прикладах, закодовані дані можуть бути виведені від вихідного інтерфейсу 22 пристрою збереження. Подібним чином, доступ до закодованих даних може бути здійснений від пристрою збереження за допомогою вхідного інтерфейсу. Пристрій збереження може містити в 55 собі будь-який з різноманітних розподілених носіїв даних або носіїв даних локального доступу, як, наприклад, накопичувач на твердих дисках, диски Blu-ray, DVD, CD-ROM, флеш-пам'ять, енергозалежний або енергонезалежний запам'ятовуючий пристрій, або будь-які інші придатні цифрові носії даних для збереження закодованих відеоданих. У додатковому прикладі, пристрій збереження може відповідати файловому серверу або іншому проміжному пристрою збереження, який може зберігати закодоване відео, сформоване пристроєм-джерелом 12. 60 Пристрій-адресат 14 може здійснювати доступ до збережених відеоданих у пристрої

збереження по потоковій передачі або завантаженню даних. Файловий сервер може бути будь-яким типом сервера, здатним зберігати закодовані відеодані і передавати ці закодовані відеодані пристрою-адресату 14. Приклади файлових серверів містять у собі веб-сервер (наприклад, для веб-сайту), сервер FTP, пристрої мережевого сховища (NAS) даних або накопичувач на локальному диску. Пристрій-адресат 14 може здійснювати доступ до закодованих відеоданих по будь-якому стандартному з'єднанню даних, включаючи інтернет-з'єднання. Це може включати в себе бездротовий канал (наприклад, з'єднання Wi-Fi), дротове з'єднання (наприклад, DSL, кабельний модем і так далі) або їхню комбінацію, що підходить для доступу до закодованих відеоданих, які зберігаються на файловому сервері. Передача закодованих відеоданих від пристрою збереження може бути потоковою передачею, передачею завантаження або їх комбінацією.

Способи даного розкриття необов'язково обмежені бездротовими застосуваннями або налаштуваннями. Способи можуть бути застосовані до кодування відео для підтримки будь-якого з різноманітних мультимедійних застосувань, як, наприклад, ефірні телевізійні широкомовні передачі, кабельні телевізійні передачі, супутникові телевізійні передачі, поточкові передачі відеосигналу по інтернету, як, наприклад, динамічна адаптивна потокова передача по HTTP (DASH), цифровий відеосигнал, який закодований на носії даних, декодування цифрового відеосигналу, що зберігається на носії даних, або інші застосування. У деяких прикладах, система 10 може бути виконана з можливістю підтримувати односторонню або двосторонню передачу відеосигналу для підтримки таких застосувань, як потокова передача відеосигналу, відтворення відеосигналу, широкомовна передача відеосигналу і/або відеотелефонія.

У прикладі на Фіг. 1, пристрій-джерело 12 містить у собі джерело 18 відео, відеокодер 20 і вихідний інтерфейс 22. Пристрій-адресат 14 містить у собі вхідний інтерфейс 28, відеодекодер 30 і пристрій 32 відображення. Відповідно до даного розкриття, відеокодер 20 пристрою-джерела 12 може бути виконаний з можливістю застосовувати способи кодування відеоданих відповідно тільки до синтаксичного розширення високого рівня стандарту відеокодування. В інших прикладах, пристрій-джерело й пристрій-адресат можуть містити в собі інші компоненти або установки. Наприклад, пристрій-джерело 12 може приймати відеодані від зовнішнього джерела 18 відео, як, наприклад, зовнішня камера. Подібним чином, пристрій-адресат 14 може взаємодіяти з зовнішнім пристроєм відображення, замість того, щоб містити в собі інтегрований пристрій відображення.

Зображена система 10 на Фіг. 1 є лише прикладом. Способи кодування відеоданих відповідно тільки до синтаксичного розширення високого рівня стандарту відеокодування можуть бути виконані будь-яким цифровим пристроєм кодування і/або декодування відео. Хоча, загалом, способи даного винаходу виконуються пристроєм відеокодування, способи можуть також бути виконані відеокодером/декодером, як правило, який називається "CODEC". Більше того, способи даного розкриття можуть також бути виконані пристроєм попередньої обробки відеосигналу. Пристрій-джерело 12 і пристрій-адресат 14 є лише прикладами таких пристроїв кодування, у яких пристрій 12 формує закодовані відеодані для передачі пристрою-адресату 14. У деяких прикладах, пристрої 12, 14 можуть функціонувати по суті симетричним чином, так що кожен з пристроїв 12, 14 містить у собі компоненти кодування і декодування відео. Отже, система 10 може підтримувати односторонню або двосторонню передачу між відеопристроями 12, 14, наприклад, для потокового відео, відтворення відеосигналу, широкомовної передачі відеосигналу або відеотелефонії.

Джерело 18 відео пристрою-джерела 12 може містити в собі пристрій захоплення відеосигналу, як, наприклад, відеокамера, відеоархів, що містить раніше захоплене відео, і/або інтерфейс подачі відео для прийому відео від провайдера відеовмісту. Як додаткова альтернатива, джерело 18 відео може формувати дані комп'ютерної графіки як вихідне відео, або комбінацію відео в реальному часі, архівоване відео і сформоване комп'ютером відео. У деяких випадках, якщо джерелом 18 відео є відеокамера, то пристрій-джерело 12 і пристрій-адресат 14 можуть формувати так звані камерофони або відеофони. Однак, як згадано вище, способи, описані в даному винаході, можуть бути застосовні до кодування відео загалом і можуть бути застосовні в бездротових і/або дротових застосуваннях. У будь-якому випадку, захоплене, попередньо захоплене або сформоване комп'ютером відео може бути закодоване відеокодером 20. Закодована відеоінформація потім може бути виведена вихідним інтерфейсом 22 на зчитуваний комп'ютером носій 16.

Зчитуваний комп'ютером носій 16 може містити в собі енергозалежні носії, як, наприклад, бездротова широкомовна або дротова мережева передача, або носії даних (тобто, енергонезалежні носії даних), як, наприклад, накопичувач на твердих дисках, флеш-пам'ять, компакт-диск, цифровий відеодиск, диск Blu-ray або інші зчитувані комп'ютером носії даних. У

деяких прикладах, мережевий сервер (не зображений) може приймати закодовані відеодані від пристрою-джерела 12 і забезпечувати закодовані відеодані пристрою-адресату 14, наприклад, по мережевій передачі. Подібним чином, обчислювальний пристрій блока відтворення носія, як, наприклад, блока маркування диска, може приймати закодовані відеодані від пристрою-джерела 12 і виробляти диск, що містить закодовані відеодані. Отже, зчитуваний комп'ютером носій 16 можна розуміти як такий, що включає в себе один або декілька зчитуваних комп'ютером носіїв різних форм, у різних прикладах.

Вхідний інтерфейс 28 пристрою-адресата 14 приймає інформацію від зчитуваного комп'ютером носія даних 16. Інформація зчитуваного комп'ютером носія даних 16 може містити в собі синтаксичну інформацію, визначену відеокодером 20, що також використовується відеодекодером 30, що містить у собі синтаксичні елементи, які описують характеристики і/або обробні блоки й інші закодовані елементи, наприклад, GOP. Пристрій 32 відображення відображає закодовані відеодані користувачу і може містити будь-який з різноманітних пристроїв, як, наприклад, катодно-променева трубка (CRT), рідкокристалічний дисплей (LCD), плазмовий дисплей, дисплей на органічних світлодіодах (OLED) або інший тип пристрою відображення.

Відеокодер 20 і відеодекодер 30 можуть функціонувати відповідно до стандарту відеокодування, як, наприклад, стандарт Високоєфективного Відеокодування (HEVC), що знаходиться в розробці в даний час, і можуть відповідати тестовій моделі (HM) HEVC. Остання робоча версія HEVC, яка називається "Робоча версія 7 HEVC" або "WD7", описана в документі JCTVC-I1003, Бросс і інші, "Робоча версія 7 текстової специфікації високоєфективного Відеокодування (HEVC)", Спільна експертна група (JCT-VC) по кодуванню відео ITU-T SG16 WP3 і ISO/IEC JTC1/SC29/WG11, 9-а зустріч: Женева, Швейцарія, з 27 квітня 2012 року по 7 травня 2012 року, що, за станом на 22 червня 2012 року, можна завантажити з [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/9\\_Geneva/wg11/JCTVC-I1003-v3.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/9_Geneva/wg11/JCTVC-I1003-v3.zip). Як відзначено вище, дане розкриття містить у собі способи розширення HEVC з використанням синтаксису високого рівня. Відповідно, відеокодер 20 і відеодекодер 30 можуть функціонувати відповідно до версії HEVC, розширеної з використанням синтаксису високого рівня.

Альтернативно, відеокодер 20 і відеодекодер 30 можуть функціонувати відповідно до інших приватних або промислових стандартів, як, наприклад, стандарт ITU-T H.264, який альтернативно називається MPEG-4, Частина 10, Просунуте кодування (AVC) відеосигналу, або розширеннями таких стандартів. Знову ж, ці розширення можуть бути досягнуті з використанням синтаксису високого рівня. Способи даного винаходу, однак, не обмежені будь-яким конкретним стандартом кодування. Інші приклади стандартів відеокодування містять у собі MPEG-2 ITU-T і H.263. Хоча це не зображено на Фіг. 1, у деяких аспектах відеокодер 20 і відеодекодер 30 можуть бути інтегровані з аудіокодером і декодером і можуть містити в собі відповідні елементи MUX-DEMUX або інші технічні засоби або програмне забезпечення для керування кодуванням аудіо і відео в загальному потоці даних або окремих потоках даних. Якщо це застосовно, то елементи MUX-DEMUX можуть відповідати протоколу мультимплексування ITU H.223 або іншим протоколам, як, наприклад, протокол (UDP) користувацької дейтаграми.

Стандарт ITU-T H.264/MPEG-4 (AVC) був сформульований Експертною групою (VCEG) кодування відео ITU-T разом з Експертною групою (MPEG) по рухомому зображенню ISO/IEC як продукт колективного партнерства, відомого як Об'єднана команда (JVT) по кодуванню відео. У деяких аспектах, способи, описані в даному розкритті, можуть бути застосовані до пристроїв, що загалом відповідають стандарту H.264. Стандарт H.264 описаний у Рекомендації H.264 ITU-T, Просунуте відеокодування для загальних аудіовізуальних послуг, дослідницькою групою ITU-T, від березня 2005 року, і в даному документі може називатися стандарт H.264 або специфікація H.264, або стандарт H.264/AVC, або специфікація. Об'єднана команда (JVT) по кодуванню відео продовжує роботу над розширеннями H.264/MPEG-4 AVC.

Відеокодер 20 і відеодекодер 30 можуть бути здійснені як будь-яка з різноманітних придатних схем пристрою кодування, як, наприклад, один або кілька мікропроцесорів, цифрові сигнальні процесори (DSP), спеціалізовані інтегральні схеми (ASIC), програмовані користувачем вентильні матриці (FPGA), дискретна логіка, програмне забезпечення, технічні засоби, програмно-апаратні засоби або будь-яка їхня комбінація. Коли способи здійснені частково в програмному забезпеченні, пристрій може зберігати інструкції для програмного забезпечення на придатному, енергонезалежному зчитуваному комп'ютером носії і виконувати інструкції в технічних засобах з використанням одного або декількох процесорів для виконання способів даного розкриття. Кожний з відеокодера 20 і відеодекодера 30 може бути включений в один або кілька пристроїв кодування або пристроїв декодування, будь-який з яких може бути інтегрований як частина комбінованого кодера/декодера (CODEC) у відповідному пристрої.

JCT-VC розробляє стандарт HEVC. Зусилля по стандартизації HEVC основані на моделі пристрою відеокодування, яка розвивається, що називається Тестовою моделлю (НМ) HEVC. НМ надає кілька додаткових можливостей пристроїв відеокодування відносно існуючих пристроїв, відповідно, наприклад, до ITU-T H.264/AVC. Наприклад, у той час як H.264

5 забезпечує дев'ять режимів кодування з внутрішнім прогнозуванням, НМ може забезпечувати тридцять три режими кодування з внутрішнім прогнозуванням.

Загалом, робоча модель НМ описує, що відеокадр або зображення може бути розділений на послідовність деревоподібних блоків або найбільших елементів (LCU) кодування, що містять у собі відліки яскравості і кольоровості. Синтаксичні дані усередині бітового потоку можуть

10 визначати розмір LCU, що є найбільшим елементом кодування у сенсі числа пікселів. Слайс містить у собі число послідовних деревоподібних блоків у порядку кодування. Відеокадр або зображення може бути розділене на один або декілька слайсів. Кожен деревоподібний блок може бути розбитий на елементи (CU) кодування, відповідно до дерева квадратів. Загалом, структура даних дерева квадратів містить у собі один вузол на CU, з кореневим вузлом, що

15 відповідає деревоподібному блоку. Якщо CU розбитий на чотири під-CU, то вузол, що відповідає CU, містить у собі чотири кінцеві вузли, кожний з яких відповідає одному з під-CU.

Кожен вузол структури даних дерева квадратів може забезпечувати синтаксичні дані відповідному CU. Наприклад, вузол у дереві квадратів може містити в собі прапор розділення, який вказує, чи розділений CU, що відповідає вузлу, на під-CU. Синтаксичні елементи CU

20 можуть бути визначені рекурсивно і можуть залежати від того, чи розділений CU на під-CU. Якщо CU не розділений додатково, він називається кінцевим CU. У даному розкритті, чотири під-CU кінцевого CU будуть також називатися кінцевими CU, навіть якщо немає чіткого розділення вихідного кінцевого CU. Наприклад, якщо CU розміру 16×16 не розділений додатково, чотири під-CU 8×8 будуть також називатися кінцевими CU, хоча CU 16×16 ніколи не

25 був розділений.

CU має ту ж мету, що і макроблок стандарту H.264, за винятком того, що CU не має відмінності в розмірі. Наприклад, деревоподібний блок може бути розділений на чотири дочірні вузли (які також називаються під-CU), і кожен дочірній вузол, у свою чергу, може бути

30 батьківським вузлом і бути розділений на інші чотири дочірні вузли. Останній, не розділений дочірній вузол, який називається кінцевим вузлом дерева квадратів, містить вузол кодування, який також називається кінцевим CU. Синтаксичні дані, зв'язані з закодованим бітовим потоком, можуть визначати кількість разів, що дерево квадратів може бути розділене, що називається максимальною глибиною CU, і може також визначати мінімальний розмір вузлів кодування. Відповідно, бітовий потік може також визначати найменший елемент (SCU) кодування. Дане

35 розкриття використовує термін "блок" для посилання на будь-який з CU, PU або TU, у контексті HEVC, або подібні структури даних у контексті інших стандартів (наприклад, макроблоки і їхні підблоки в H.264/AVC).

CU містить у собі вузол кодування, елементи (PU) прогнозування й елементи (TU) перетворення, зв'язані з вузлом кодування. Розмір CU відповідає розміру вузла кодування і

40 повинен бути квадратним за формою. Розмір CU може варіюватися від 8×8 пікселів до розміру дерева квадратів, максимально 64×64 пікселів або більше. Кожен CU може містити один або декілька PU і один або декілька TU. Синтаксичні дані, зв'язані з CU, можуть описувати, наприклад, розділення CU на один або декілька PU. Режими розділення можуть відрізнятися залежно від того, чи є CU закодованим у режимі пропуску або прямому режимі, закодованим у

45 режимі внутрішнього прогнозування або закодованим у режимі зовнішнього прогнозування. PU може бути розділений так, що за формою він не квадратний. Синтаксичні дані, зв'язані з CU, можуть також описувати, наприклад, розділення CU на один або декілька TU, відповідно до дерева квадратів. TU може бути квадратної або неквадратної (наприклад, прямокутної) форми.

Стандарт HEVC допускає перетворення, відповідно до TU, що можуть відрізнятися для

50 різних CU. TU, як правило, мають розмір на основі розміру PU усередині заданого CU, визначеного для розділеного LCU, хоча це не завжди може бути так. TU, як правило, мають той же розмір або менший, ніж PU. У деяких прикладах, залишкові відліки, що відповідають CU, можуть бути підрозділені на менші елементи з використанням структури дерева квадратів, відомої як "залишкове дерево (RQT) квадратів". Кінцеві вузли RQT можуть називатися

55 елементами (TU) перетворення. Значення різниці пікселів, зв'язані з TU, можуть бути перетворені для відтворення коефіцієнтів перетворення, що можуть бути квантовані.

Кінцевий CU може містити в собі один або кілька елементів (PU) прогнозування. Загалом, PU являє собою просторову область, що відповідає усім або частині відповідних CU, і може містити в собі дані для витягування опорного відліку для PU. Більше того, PU містить у собі дані,

60 що стосуються прогнозування. Наприклад, коли PU закодований у внутрішньому режимі, дані



для PU можуть бути включені в залишкове дерево (RQT) квадратів, що може містити в собі дані, які описують режим внутрішнього прогнозування для TU, що відповідає PU. Як інший приклад, коли PU закодований у зовнішньому режимі, PU може містити в собі дані, які визначають один або кілька векторів руху для PU. Дані, що визначають вектор руху для PU, можуть описувати, наприклад, горизонтальну компоненту вектора руху, вертикальну компоненту вектора руху, розрізнення вектора руху (наприклад, з точністю до однієї четвертої пікселя або з точністю до однієї восьмої пікселя), опорне зображення, на яке указує вектор руху, і/або список опорних зображень (наприклад, Список 0, Список 1, або Список C) для вектора руху.

Кінцевий CU, що має один або декілька PU, може також містити в собі один або кілька елементів (TU) перетворення. Елементи перетворення можуть бути визначені з використанням RQT (який також називається структурою дерева квадратів TU), як описано вище. Наприклад, прапор розділення може вказувати, чи розділений кінцевий CU на чотири елементи перетворення. Потім, кожен елемент перетворення може бути додатково розділений на додаткові під-TU. Коли TU не розділений додатково, він може називатися кінцевим TU. Загалом, для внутрішнього кодування, усі кінцеві TU, що належать кінцевому CU, мають однаковий режим внутрішнього прогнозування. Тобто, однаковий режим внутрішнього прогнозування, загалом, застосовується для обчислення прогнозованих значень для всіх TU кінцевого CU. Для внутрішнього кодування, відеокoder може обчислювати залишкове значення для кожного кінцевого TU з використанням режиму внутрішнього прогнозування, як різницю між частиною CU, що відповідає TU, і вихідним блоком. TU необов'язково обмежений розміром PU. Таким чином, TU можуть бути більші або менші, ніж PU. Для внутрішнього кодування, PU може бути розташований разом з відповідним кінцевим TU для того ж CU. У деяких прикладах, максимальний розмір кінцевого TU може відповідати розміру відповідного кінцевого CU.

Більше того, TU кінцевих CU можуть також бути зв'язані з відповідними структурами даних дерева квадратів, які називаються залишковими деревами (RQT) квадратів. Тобто, кінцевий CU може містити в собі дерево квадратів, що вказує, як кінцевий CU розділений на TU. Кореневий вузол дерева квадратів TU, загалом, відповідає кінцевому CU, у той час як кореневий вузол дерева квадратів CU, загалом, відповідає деревоподібному блоку (або LCU). TU RQT, які не розділені, називаються кінцевими TU. Загалом, дане розкриття використовує терміни CU і TU для посилання на кінцевий CU і кінцевий TU, відповідно, якщо тільки не вказане інше.

Відеопослідовність, як правило, містить у собі ряд відеокадрів або зображень. Група (GOP) зображень, загалом, містить ряд з одного або декількох зображень. GOP може містити в собі синтаксичні дані в заголовку GOP, заголовку одного або декількох зображень або в іншому місці, що описує кількість зображень, включених у GOP. Кожен слайс зображення може містити в собі синтаксичні дані слайса, що описують режим кодування для відповідного слайса. Відеокoder 20, як правило, функціонує з відеоблоками усередині окремих відеослайсів, щоб закодувати відеодані. Відеоблок може відповідати вузлу кодування усередині CU. Відеоблоки можуть мати фіксовані або змінювані розміри і можуть відрізнятися в розмірах відповідно до конкретного стандарту кодування.

Як приклад, НМ підтримує прогнозування в різних розмірах PU. Припустимо, що розмір конкретного CU дорівнює  $2N \times 2N$ , тоді НМ підтримує внутрішнє прогнозування в розмірах PU  $2N \times 2N$  або  $N \times N$ , і зовнішнє прогнозування в симетричних розмірах PU  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$  або  $N \times N$ . НМ також підтримує асиметричне розділення для зовнішнього прогнозування в розмірах PU  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  і  $nR \times 2N$ . При асиметричному розділенні, один напрямок CU не розділяється, у той час як інший напрямок розділяється на 25 % і 75 %. Частина CU, що відповідає 25 % розділення, указується "n", за яким йде вказівка "Зверху", "Знизу", "Ліворуч" або "Праворуч". Таким чином, наприклад, " $2N \times nU$ " стосується  $2N \times 2N$  CU, який розділений горизонтально з  $2N \times 0,5N$  PU зверху і  $2N \times 1,5N$  PU знизу.

У даному розкритті, " $N \times N$ " і " $N$  на  $N$ " можуть бути використані взаємозамінно для посилання на піксельні розміри відеоблока відносно вертикальних і горизонтальних розмірів, наприклад,  $16 \times 16$  пікселів або 16 на 16 пікселів. Загалом, блок  $16 \times 16$  має 16 пікселів у вертикальному напрямку ( $y=16$ ) і 16 пікселів у горизонтальному напрямку ( $x=16$ ). Подібним чином, блок  $N \times N$ , загалом, має  $N$  пікселів у вертикальному напрямку і  $N$  пікселів у горизонтальному напрямку, де  $N$  являє собою ненегативне ціле значення. Пікселі в блоці можуть бути упорядковані в ряди і стовпці. Більше того, блоки необов'язково повинні мати однакове число пікселів у горизонтальному напрямку, а також у вертикальному напрямку. Наприклад, блоки можуть містити  $N \times M$  пікселів, де  $M$  необов'язково дорівнює  $N$ .

Слідом за кодуванням із внутрішнім прогнозуванням або зовнішнім прогнозуванням з використанням PU CU, відеокoder 20 може обчислювати залишкові дані для TU CU. PU можуть містити синтаксичні дані, які описують спосіб або режим формування прогнозних піксельних

даних у просторовому домені (який також називається піксельним доменом), і TU можуть містити коефіцієнти в домені перетворення слідом за застосуванням перетворення, наприклад, дискретного косинусного перетворення (DCT), цілочисельного перетворення, хвильового перетворення або концептуально схожого перетворення до залишкових відеоданих. Залишкові дані можуть відповідати різницям пікселів між пікселями незакодованого зображення і значеннями прогнозування, що відповідають PU. Відеокoder 20 може формувати TU, що включають у себе залишкові дані для CU, а потім перетворювати TU для відтворення коефіцієнтів перетворення для CU.

Слідом за будь-якими перетвореннями для відтворення коефіцієнтів перетворення, відеокoder 20 може виконувати квантування коефіцієнтів перетворення. Квантування, загалом, стосується процесу, у якому коефіцієнти перетворення квантуються для можливого скорочення кількості даних, використовуваних для представлення коефіцієнтів, що забезпечує додатковий стиснення. Процес квантування може скоротити глибину бітів, зв'язану з деякими або всіма коефіцієнтами. Наприклад, значення  $n$ -бітів може бути округлене до значення  $m$ -бітів під час квантування, де  $n$  більше, ніж  $m$ .

Слідом за квантуванням, відеокoder може сканувати коефіцієнти перетворення, виробляючи одновимірний вектор із двовимірної матриці, що включає в себе квантовані коефіцієнти перетворення. Сканування може бути задумане так, щоб розмістити коефіцієнти з високою енергією ( $i$ , отже, з низькою частотою) у передній частині масиву і розмістити коефіцієнти з низькою енергією ( $i$ , отже, з високою частотою) у задній частині масиву. У деяких прикладах, відеокoder 20 може використовувати попередньо заданий порядок сканування для сканування квантованих коефіцієнтів перетворення для відтворення упорядкованого вектора, який можна ентропійно закодувати. В інших прикладах, відеокoder 20 може виконувати адаптивне сканування. Після сканування квантованих коефіцієнтів перетворення для формування одновимірного вектора, відеокoder 20 може ентропійно кодувати одновимірний вектор, наприклад, відповідно до контекстно-залежного адаптивного кодування зі змінною довжиною кодового слова (CAVLC), контекстно-залежним адаптивним двійковим арифметичним кодуванням (CABAC), синтаксичним контекстно-залежним адаптивним двійковим арифметичним кодуванням (SBAC), ентропійним кодуванням (PIPE) з розділенням інтервалів імовірності або іншою методологією ентропійного кодування. Відеокoder 20 може також ентропійно кодувати синтаксичні елементи, зв'язані з закодованими відеоданими для використання відеодекодером 30 при декодуванні відеоданих.

Для виконання CABAC, відеокoder 20 може призначати контекст усередині контекстної моделі символу, який повинен бути переданий. Контекст може стосуватися, наприклад, того, чи є сусідні значення символу ненульовими чи ні. Для виконання CAVLC, відеокoder 20 може вибирати змінну довжину кодового слова для символу, який повинен бути переданий. Кодові слова в VLC можуть бути побудовані так, що відносно коротші коди відповідають найбільш ймовірним символам, у той час як довгі коди відповідають менш ймовірним символам. Таким чином, за допомогою використання VLC можна домогтися збереження бітів у порівнянні, наприклад, з використанням кодових слів однакової довжини для кожного символу, який повинен бути переданий. Визначення імовірності може бути основане на контексті, призначеному символу.

Відеокoder 20 може додатково відправляти синтаксичні дані, як, наприклад, синтаксичні дані на основі блока, синтаксичні дані на основі кадру і синтаксичні дані на основі GOP, відеодекодеру 30, наприклад, у заголовку кадру, заголовку блока, заголовку слайса або заголовку GOP. Синтаксичні дані GOP можуть описувати число кадрів у відповідній GOP, і синтаксичні дані кадру можуть указувати режим кодування/прогнозування, використовуваний для кодування відповідного кадру.

Загалом, дане розкриття описує різні приклади рішень для дозволу тільки синтаксичного розширення високого рівня (HLS) стандарту відеокoduвання, як, наприклад, HEVC. Наприклад, ці способи можуть бути використані для розробки розширення тільки-HLS для профілю HEVC, як, наприклад, MVC або SVC. Різні приклади описані далі. Необхідно розуміти, що різні приклади описані окремо, а елементи будь-яких або всіх прикладів можуть бути скомбіновані в будь-якій комбінації.

У першому прикладі змін у поточній базовій специфікації HEVC немає. У розширенні HEVC, зображення (наприклад, компонента виду) може бути ідентифіковане по двох якостях: його значенню рахунка порядку зображення (POC) і ідентифікатору зображення другої розмірності, наприклад, значенню `view_id` (яке може ідентифікувати вид, у якому зображення представлено). Відеокoderу 20 може знадобитися вказати компоненту виду, яка повинна бути використана для міжвидового прогнозування, як довгострокове опорне зображення.

В другому прикладі, змін у поточній базовій специфікації HEVC немає. У розширенні HEVC, можуть бути застосовані наступні зміни. Зображення (наприклад, компонента виду) може бути ідентифіковане по двох якостях: значенню РОС і ідентифікатору зображення другої розмірності, наприклад, `view_id`. В другому прикладі, процес маркування додаткового зображення може бути виконаний відразу до кодування поточної компоненти виду, для маркування всіх міжвидових опорних зображень як довгострокових опорних зображень. Інший процес маркування зображення може бути виконаний відразу після кодування поточної компоненти виду для маркування кожного міжвидового опорного зображення як або довгострокового, короткострокового або "невикористовуваного для посилання", що є тим же, що його попередній статус маркування до того, як була закодована поточна компонента виду.

У третьому прикладі, способи другого приклада використовуються і доповнюються, як описано далі. На додаток до способів другого приклада, для кожного міжвидового опорного зображення, після того, як воно марковано як довгострокове опорне зображення, його значення РОС відображається в нове значення РОС, що не еквівалентно значенню РОС будь-якого існуючого опорного зображення. Після декодування поточної компоненти виду, для кожного міжвидового опорного зображення, його значення РОС відображається назад у вихідне значення РОС, яке дорівнює поточній компоненті виду. Наприклад, поточна компонента виду може належати виду 3 (припустимо, що ідентифікатор виду дорівнює індексу порядку виду) і може мати значення РОС, яке дорівнює 5. Два міжвидових опорних зображення можуть мати свої значення РОС (які дорівнюють 5), конвертовані, наприклад, у 1025 і 2053. Після декодування поточної компоненти виду, значення РОС міжвидових зображень можуть бути назад конвертовані в 5.

У четвертому прикладі, способи другого або третього прикладів можуть бути використані і доповнені, як описано далі. На додаток до способів першого приклада або другого приклада, як вказано вище, у базовій специфікації HEVC, додатковий прийом може бути використаний для заборони прогнозування між будь-яким вектором руху, що стосується короткострокового зображення, і іншим вектором руху, що стосується довгострокових зображень, особливо, під час удосконаленого прогнозування вектора руху (AMVP).

У п'ятому прикладі, у розширенні HEVC, зображення може бути ідентифіковане по двох якостях: значенню РОС і ідентифікатору зображення другої розмірності, наприклад, `view_id`. У базовій специфікації HEVC, одна або кілька наступних прив'язок можуть бути додані (окремо або в комбінації). В одному прикладі (який називається приклад 5.1), при ідентифікації опорного зображення під час AMVP і режиму злиття, ідентифікація зображення другої розмірності, наприклад, індекс порядку виду, може бути використана разом з РОС. У контексті двовимірного 2D декодування відео в базовій специфікації HEVC, ідентифікація зображення другої розмірності завжди може вважатися (установлюватися) такою, що дорівнює 0.

В іншому прикладі (приклад 5.2), прогнозування між часовим вектором руху і міжвидовим вектором руху забороняється під час AMVP (включаючи часове прогнозування вектора руху (TMVP)). Якість вектора руху може бути визначена за допомогою зв'язаного опорного індексу, що ідентифікує опорне зображення і те, як на опорне зображення посилається зображення, що містить вектор руху, наприклад, як на довгострокове опорне зображення, короткострокове опорне зображення, або міжвидове опорне зображення. В іншому прикладі (приклад 5.3), прогнозування між часовим короткостроковим вектором руху і часовим довгостроковим вектором руху може бути заборонене (наприклад, явно або неявно). В іншому прикладі (приклад 5.4), прогнозування між часовим короткостроковим вектором руху і часовим довгостроковим вектором руху може бути дозволене (наприклад, явно або неявно).

В іншому прикладі (приклад 5.5), прогнозування між векторами руху, що посилаються на два різні міжвидові опорні зображення, може бути заборонене (наприклад, явно або неявно). Два міжвидових опорних зображення можна розглядати як мають різні типи, якщо значення ідентифікаторів зображення другої розмірності для них є різними. В іншому прикладі (приклад 5.6), прогнозування між векторами руху, що посилаються на два різні міжвидові опорні зображення, може бути дозволене (наприклад, явно або неявно). В іншому прикладі (приклад 5.7), прогнозування між векторами руху, що посилаються на довгострокове зображення і міжвидове зображення, може бути дозволене (наприклад, явно або неявно). В іншому прикладі (приклад 5.8), прогнозування між векторами руху, що посилаються на довгострокове зображення і міжвидове зображення, може бути заборонене (наприклад, явно або неявно).

У будь-якому з вищевказаних прикладів, прогнозування між двома векторами руху, що посилаються на два різні часові короткострокові опорні зображення, може завжди бути дозволене, і масштабування з одного в інше на основі значень РОС може бути дозволене. Додатково або альтернативно, у будь-якому з вищевказаних прикладів, прогнозування між

векторами руху, що посилаються на два різні довгострокові зображення, може бути заборонене. Конкретні деталі різних прикладів, описаних вище, обговорюються більш докладно далі.

Загалом, дане розкриття посилається на "вектор руху" або "дані вектора руху" як включають у себе опорний індекс (тобто, покажчик на опорне зображення) і  $x$ - і  $y$ -координати самого вектора руху. Обидва вектор руху диспаратності і часовий вектор руху можуть, загалом, називатися "векторами руху". Опорне зображення, що відповідає опорному індексу, може називатися опорним зображенням, на яке посилається вектор руху. Якщо вектор руху посилається на опорне зображення іншого виду, він називається вектор руху диспаратності.

Часовий вектор руху може бути короткостроковим часовим вектором руху ("короткостроковий вектор руху") або довгостроковим часовим вектором руху ("довгостроковий вектор руху"). Наприклад, вектор руху є короткостроковим, якщо він посилається на короткострокове опорне зображення, у той час як вектор руху є довгостроковим, якщо він посилається на довгострокове опорне зображення. Необхідно відзначити, що якщо не вказане інше, вектор руху диспаратності і довгостроковий вектор руху, загалом, описують різні категорії векторів руху, наприклад, для міжвидового прогнозування і часового внутрішньовидового прогнозування, відповідно. Короткострокове і довгострокове опорні зображення являють собою приклади часових опорних зображень.

Відеокoder 20 і відеокoder 30 можуть бути виконані з можливістю ідентифікувати опорне зображення з буфера (DPB) декодованих зображень, який може бути здійснений як пам'ять опорних зображень. Процес ідентифікації опорного зображення з DPB може бути використаний у будь-якому із прикладів способів, описаних у даному розкритті. Процес ідентифікації опорного зображення з DPB може бути використаний для наступних цілей у специфікації розширення HEVC: створення групи опорних зображень, створення списку опорних зображень і/або маркування опорних зображень.

Компонента виду, компонента виду текстури, компонента виду глибини або масштабований рівень (наприклад, з конкретною комбінацією `dependency_id` і `quality_id`) можуть бути ідентифіковані за допомогою значення рахунка порядку зображення (POC) і інформації ідентифікації зображення другої розмірності. Інформація ідентифікації зображення другої розмірності може містити в собі одне або декілька з наступних: ID виду (`view_id`) у багатовидовому контексті; індекс порядку виду в багатовидовому контексті; у 3DV (багатовидовий із глибиною) контексті, комбінацію індексу порядку виду і `depth_flag` (який вказує, чи є поточний компонент виду текстурою або глибиною), наприклад, індекс порядку виду, помножений на два, плюс значення `depth_flag`; у контексті SVC, ID рівня (в оточенні масштабованого кодування, наприклад, у SVC на основі AVC, ID рівня може дорівнювати `dependency_id`, помноженому на 16, плюс `quality_id`); або загальний ID рівня (`layer_id`), наприклад, значення `reserved_one_5bits` мінус 1, причому `reserved_one_5bits` вказана в базовій специфікації HEVC. Необхідно відзначити, що загальний ID рівня може застосовуватися в змiшаному 3DV (багатовидовий із глибиною) і масштабованому сценаріях. Вищевказані приклади можуть застосовуватися до будь-якого багаторівневого кодеку, включаючи масштабований відеокoder, за допомогою, наприклад, розгляду кожного рівня як виду. Іншими словами, для багатовидового відеокoderування, різні види можуть бути розглянуті як окремі рівні.

У деяких сценаріях, базовий рівень або залежний вид можуть мати багато представлень, наприклад, унаслідок використання різних фільтрів підвищувальної дискретизації/згладжування, або унаслідок факту використання зображення із синтезованим видом для прогнозування; таким чином, у місцезоташуванні одного виду, можуть бути два зображення, готові для використання, де одне є звичайним відновленим залежним зображенням виду, а інше є зображенням із синтезованим видом, обидва мають однаковий `view_id` або індекс порядку виду. У такому випадку, може бути використана ідентифікація зображення третьої розмірності.

Відеокoder 20 і відеокoder 30 також можуть бути виконані з можливістю ідентифікувати опорне зображення зі списку опорних зображень. Буфер (DPB) опорних зображень може бути організований у списки опорних зображень, наприклад, `RefPicList0`, який містить у собі потенційні опорні зображення, що мають значення POC менше, ніж значення POC поточного зображення, і `RefPicList1`, що містить у собі потенційні опорні зображення, що мають значення POC більше, ніж значення POC поточного зображення. Способи ідентифікації опорного зображення зі списку опорних зображень використовуються як прив'язка для поточної базової специфікації HEVC. Визначені функції можуть бути використані багато разів відеокoderом або відеокoderом під час AMVP і режиму злиття.

Компонента виду, компонента виду текстури, компонента виду глибини або масштабований рівень (наприклад, з конкретною комбінацією `dependency_id` і `quality_id`) можуть бути ідентифіковані за допомогою значення POC і інформації ідентифікації зображення другої

розмірності, що може бути однією з наступних: індекс порядку виду в багатовидовому або 3DV контексті. Функція `viewOldx (pic)` повертає індекс порядку виду для виду, до якого належить зображення, ідентифіковане як "pic". Ця функція повертає 0 для будь-якої компоненти виду, компоненти виду текстури або компоненти виду глибини базового виду; ID виду (`view_id`); у контексті 3DV, комбінацію індексу порядку виду і `depth_flag` (яка вказує, чи є поточний компонент виду текстурою або глибиною): індекс порядку виду, помножений на 2, плюс значення `depth_flag`; у контексті SVC, ID рівня (в оточенні масштабованого кодування, наприклад, SVC на основі AVC, ID рівня може бути дорівнює `dependency_id`, помноженому на 16, плюс `quality_id`); або загального ID рівня (`layer_id`), наприклад, значення `reserved_one_5bits` мінус 1, причому `reserved_one_5bits` вказаний у базовій специфікації HEVC. Функція `layerId (pic)` повертає `layer_id` зображення `pic`. `LayerId (pic)` повертає 0 для будь-якої компоненти виду (текстури) базового виду. `LayerId (pic)` повертає 0 для будь-якого зображення (або представлення рівня) базового рівня SVC. Необхідно відзначити, що загальний ID рівня може застосовуватися в змішаному 3DV (багатовидовий із глибиною) і масштабованому сценаріях.

У деяких сценаріях, базовий рівень або залежний вид можуть мати множину представлень, наприклад, унаслідок використання різних фільтрів підвищувальної дискретизації/згладжування, або внаслідок факту використання зображення із синтезованим видом для прогнозування; таким чином, у місцезоташуванні одного виду, можуть бути два зображення, готових для використання: одне є звичайним відновленим залежним зображенням виду, а інше є зображенням із синтезованим видом, обидва мають однаковий `view_id` або індекс порядку виду. У такому випадку, може бути використана ідентифікація зображення третьої розмірності.

Одна або декілька з вищезгаданих ідентифікацій зображень другої розмірності і/або третьої розмірності може бути визначена за допомогою функції `AddPicId (pic)`.

Відеокoder 20 і відеокoder 30 можуть також бути виконані з можливістю ідентифікувати тип запису в списку опорних зображень. Це може бути використано як прив'язка для поточної базової специфікації HEVC. Будь-яка або всі функції, визначені нижче, можуть бути використані багато разів відеокoderом 20 і/або відеокoderом 30 під час AMVP і/або режиму злиття. Будь-яка або усі з наступних прикладів способів можуть бути використані для ідентифікації типу запису в списку опорних зображень. В одному прикладі, функція "`RefPicType (pic)`" повертає 0, якщо зображення `pic` є часовим опорним зображенням, і повертає 1, якщо зображення `pic` не є часовим опорним зображенням. В іншому прикладі, функція `RefPicType (pic)` повертає 0, якщо зображення має той же РОС, що і поточне зображення, і повертає 1, якщо зображення має РОС, відмінний від РОС поточного зображення.

В іншому прикладі, результатів прикладів, описаних вище, можна досягти за допомоги заміщення використання функції `RefPicType (pic)` просто на перевірку того, чи дорівнює РОС "pic" (аргумент функції) РОС поточного зображення. У деяких прикладах, міжвидове опорне зображення може бути промарковане як "невикористовуване для посилання". Міжвидове опорне зображення може бути промарковане як "невикористовуване для посилання". Для спрощення, таке зображення називається неопорним зображенням у базовій специфікації HEVC. У деяких прикладах, зображення, марковане як "використовуване для довгострокового посилання" або "використовуване для короткострокового посилання", може називатися опорним зображенням у базовій специфікації HEVC. У деяких прикладах, функція `RefPicType (pic)` повертає 0, якщо зображення марковане як "використовуване для довгострокового посилання" або "використовуване для короткострокового посилання", і повертає 1, якщо зображення марковане як "невикористовуване для посилання". Додатково, у деяких прикладах, у розширенні HEVC, компонента виду, відразу після її декодування, може бути маркована як "невикористовувана для посилання", незалежно від значення синтаксичного елемента `nal_ref_flag`.

Після того, як весь елемент доступу закодований, компоненти виду елемента доступу можуть бути марковані як "використовувані для короткострокового посилання" або "використовувані для довгострокового посилання", якщо `nal_ref_flag` справжній. Альтернативно, компонента виду може бути промаркована тільки як "використовувана для короткострокового посилання" або "використовувана для довгострокового посилання", якщо це включено в набір опорних зображень (RPS) наступного компонента виду в порядку декодування в тому ж виді, відразу після того, як витягнутий RPS для наступної компоненти виду. Додатково, у базовій специфікації HEVC, поточне зображення, відразу після його декодування, може бути марковане як "невикористовуване для посилання".

У деяких прикладах, `RefPicType (picX, refIdx, LX)` повертає значення `RefPicType (pic)` у той час, коли `picX` є поточним зображенням, причому `pic` є опорним зображенням з індексом `refIdx` зі списку `LX` опорних зображень зображення `picX`.

Відносно приклада, що згадується вище як "четвертий приклад", відеокoder 20 і відеокoder 30 можуть бути виконані з можливістю дозволяти прогнозування між довгостроковими опорними зображеннями без масштабування під час AMVP і TMVP. Відносно AMVP, відеокoder 20 і відеокoder 30 можуть бути виконані з можливістю виконувати модифікований процес витягування кандидатів предиктора вектора руху (MVP). Введення в процес можуть містити в собі місцерозташування ( $x_P$ ,  $y_P$ ) яскравості, що указує верхній лівий відлік яскравості поточного елемента прогнозування відносно верхнього лівого відліку поточного зображення, змінні, що вказують ширину і висоту елемента прогнозування для яскравості,  $nPSW$  і  $nPSH$ , і опорний індекс  $refIdxLX$  (де  $X$  дорівнює 0 або 1) розділення поточного елемента прогнозування. Виведення процесу можуть містити в собі (де  $N$  заміщений або на  $A$ , або на  $B$ , де  $A$  відповідає сусіднім кандидатам ліворуч, а  $B$  відповідає сусіднім кандидатам зверху, як зображено в прикладі на Фіг. 10) вектори  $mvLXN$  руху сусідніх елементів прогнозування і прапори  $availableFlagLXN$  доступності сусідніх елементів прогнозування. Змінна  $isScaledFlagLX$ , де  $X$  дорівнює 0 або 1, може вважатися такою, що дорівнює 1 або 0.

Відеокoder 20 і відеокoder 30 можуть витягати вектор  $mvLXA$  руху і прапор  $availableFlagLXA$  доступності в наступному порядку етапів в одному прикладі, де підкреслений текст являє собою зміни відносно HEVC WD7:

1. Нехай набір із двох місцерозташувань відліку буде  $(x_{A_k}, y_{A_k})$ , де  $k=0, 1$ , що вказує місцерозташування відліку, де  $x_{A_k}=x_P-1$ ,  $y_{A_0}=y_P+nPSH$  і  $y_{A_1}=y_{A_0}-MinPuSize$ . Набір місцерозташувань відліку  $(x_{A_k}, y_{A_k})$  являє собою місцерозташування відліку безпосередньо з лівої сторони лівої межі розділення і її продовженої лінії.

2. Нехай прапор  $availableFlagLXA$  доступності буде спочатку вважатися таким, що дорівнює 0, і обидві компоненти  $mvLXA$  вважатися такими, що дорівнюють 0.

3. Коли одна або декілька з наступних умов істинні, змінна  $isScaledFlagLX$  вважатися такою, що дорівнює 1, у даному прикладі.

- елемент прогнозування, що покриває місцерозташування  $(x_{A_0}, y_{A_0})$  яскравості доступний [Ed. (Ред.) (BB): Переписати з використанням  $MinCbAddrZS[]$  і процесу доступності для мінімальних блоків кодування], і  $PredMode$  не  $MODE\_INTRA$ .

- елемент прогнозування, що покриває місцерозташування  $(x_{A_1}, y_{A_1})$  яскравості доступний [Ed. (BB): Переписати з використанням  $MinCbAddrZS[]$  і процесу доступності для мінімальних блоків кодування], і  $PredMode$  не  $MODE\_INTRA$ .

4. Для  $(x_{A_k}, y_{A_k})$  від  $(x_{A_0}, y_{A_0})$  до  $(x_{A_1}, y_{A_1})$ , де  $x_{A_1}, y_{A_1}$ , наступне застосовується повторно доти, поки  $availableFlagLXA$  не дорівнює 1:

- Якщо елемент прогнозування, що покриває місцерозташування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням  $MinCbAddrZS[]$  і процесу доступності для мінімальних блоків кодування], і  $PredMode$  не  $MODE\_INTRA$ ,  $predFlagLX[x_{A_k}][y_{A_k}]$  дорівнює 1, і опорний індекс  $refIdxLX[x_{A_k}][y_{A_k}]$  дорівнює опорному індексу  $refIdxLX$  поточного елемента прогнозування, то  $availableFlagLXA$  вважається таким, що дорівнює 1, вектор  $mvLXA$  руху вважається таким, що дорівнює вектору  $mvLX[x_{A_k}][y_{A_k}]$  руху,  $refIdxA$  вважається таким, що дорівнює  $refIdxLX[x_{A_k}][y_{A_k}]$ , а  $ListA$  вважається таким, що дорівнює  $ListX$ .

- Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням  $MinCbAddrZS[]$  і процесу доступності для мінімальних блоків кодування], і  $PredMode$  не  $MODE\_INTRA$ ,  $predFlagLX[x_{A_k}][y_{A_k}]$  (де  $Y=X$ ) дорівнює 1, і  $PicOrderCnt(RefPicListY[refIdxLY[x_{A_k}][y_{A_k}]])$  дорівнює  $PicOrderCnt(RefPicListX[refIdxLX])$ , то  $availableFlagLXA$  вважається таким, що дорівнює 1, вектор  $mvLXA$  руху вважається таким, що дорівнює вектору  $mvLY[x_{A_k}][y_{A_k}]$  руху,  $refIdxA$  вважається таким, що дорівнює  $refIdxLY[x_{A_k}][y_{A_k}]$ , а  $ListA$  вважається таким, що дорівнює  $ListY$ , і  $mvLXA$  вважається таким, що дорівнює  $mvLXA$ .

5. Коли  $availableFlagLXA$  дорівнює 0, то для  $(x_{A_k}, y_{A_k})$  від  $(x_{A_0}, y_{A_0})$  до  $(x_{A_1}, y_{A_1})$ , де  $y_{A_1}=y_{A_0}-MinPuSize$ , наступне застосовується повторно, поки  $availableFlagLXA$  не дорівнює 1, у даному прикладі:

- Якщо елемент прогнозування, що покриває місцерозташування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням  $MinCbAddrZS[]$  і процесу доступності для мінімальних блоків кодування], і  $PredMode$  не  $MODE\_INTRA$ ,  $predFlagLX[x_{A_k}][y_{A_k}]$  дорівнює 1, і  $RefPicListX[refIdxLX]$  і  $refPicList[refIdxLX[x_{A_k}][y_{A_k}]]$  обидва є довгостроковими опорними зображеннями або обидва є короткостроковими опорними зображеннями, то  $availableFlagLXA$  вважається таким, що дорівнює 1, вектор  $mvLXA$  руху вважається таким, що дорівнює вектору  $mvLX[x_{A_k}][y_{A_k}]$  руху,  $refIdxA$  вважається таким, що дорівнює  $refIdxLX[x_{A_k}][y_{A_k}]$ , а  $ListA$  вважається таким, що дорівнює  $ListX$ .

- Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY[ $x_{A_k}$ ][ $y_{A_k}$ ] (де  $Y=X$ ) дорівнює 1, і RefPicListX[refldxLX] і RefPicListY[refldxLY[ $x_{A_k}$ ][ $y_{A_k}$ ]] обидва є довгостроковими опорними зображеннями або обидва є короткостроковими опорними зображеннями, то availableFlagLXA вважається таким, що дорівнює 1, а вектор mvLXA руху вважається таким, що дорівнює вектору mvLY[ $x_{A_k}$ ][ $y_{A_k}$ ] руху, refldxA вважається таким, що дорівнює refldxLY[ $x_{A_k}$ ][ $y_{A_k}$ ], а ListA вважається таким, що дорівнює ListY.

- Коли availableFlagLXA дорівнює 1, і обидва RefPicListA[refldxA] і RefPicListX[refldxLX] обидва є короткостроковими опорними зображеннями, то mvLXA витягають, як вказано нижче (де позначення 8-### стосується розділів поточної робочої версії HEVC, тобто, WD7).

$$tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6) \quad (8-127)$$

$$mvLXA = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * ((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) >> 8)) \quad (8-128),$$

де td і tb можуть бути витягнуті так

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refldxA}])) \quad (8-129)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])) \quad (8-130)$$

Відеокодер 20 і відеокодер 30 можуть виконані з можливістю витягати вектор mvLXB руху і прапор availableFlagLXB доступності в наступному порядку етапів в одному прикладі, де підкреслений текст являє собою зміни відносно HEVC WD7:

- Нехай набір із трьох місцерозташувань відліку буде  $(x_{B_k}, y_{B_k})$ , де  $k=0, 1, 2$ , що вказує місцерозташування відліку, де  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ ,  $x_{B_2} = x_P - \text{MinPuSize}$  і  $y_{B_k} = y_P - 1$ . Набір місцерозташувань відліку  $(x_{B_k}, y_{B_k})$  являє собою місцерозташування відліку безпосередньо з верхньої сторони межі розділення зверху і її продовженої лінії. [Ed. (BB): Визначити MinPuSize у SPS, але витягування повинне залежати від використання прапора AMP]
- Коли  $y_P - 1$  менше, ніж  $((y_C >> \text{Log2CtbSize}) << \text{Log2CtbSize})$ , то застосовується наступне.
$$x_{B_0} = (x_{B_0} >> 3) << 3 + ((x_{B_0} >> 3) \& 1) * 7 \quad (8-131),$$

$$x_{B_1} = (x_{B_1} >> 3) << 3 + ((x_{B_1} >> 3) \& 1) * 7 \quad (8-132),$$

$$x_{B_2} = (x_{B_2} >> 3) << 3 + ((x_{B_2} >> 3) \& 1) * 7 \quad (8-133).$$
- Нехай прапор availableFlagLXB доступності буде спочатку вважатися таким, що дорівнює 0, і обидві компоненти mvLXB вважатися такими, що дорівнюють 0.
- Для  $(x_{B_k}, y_{B_k})$  від  $(x_{B_0}, y_{B_0})$  до  $(x_{B_2}, y_{B_2})$ , де  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$  і  $x_{B_2} = x_P - \text{MinPuSize}$ , то наступне застосовується повторно доти, поки availableFlagLXB не дорівнює 1:

- Якщо елемент прогнозування, що покриває місцерозташування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[ $x_{B_k}$ ][ $y_{B_k}$ ] дорівнює 1, і опорний індекс refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ] дорівнює опорному індексу refldxLX поточного елемента прогнозування, то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLX[ $x_{B_k}$ ][ $y_{B_k}$ ] руху, refldxB вважається таким, що дорівнює refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ], а ListB вважається таким, що дорівнює ListX.

- Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY[ $x_{B_k}$ ][ $y_{B_k}$ ] (де  $Y=X$ ) дорівнює 1, і PicOrderCnt(RefPicListY[refldxLY[ $x_{B_k}$ ][ $y_{B_k}$ ]]) дорівнює PicOrderCnt(RefPicListX[refldxLX]), то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLY[ $x_{B_k}$ ][ $y_{B_k}$ ] руху, refldxB вважається таким, що дорівнює refldxLY[ $x_{B_k}$ ][ $y_{B_k}$ ], а ListB вважається таким, що дорівнює ListY.

- Коли isScaledFlagLX дорівнює 0 і availableFlagLXB дорівнює 1, то mvLXA вважається таким, що дорівнює mvLXB, refldxA вважається таким, що дорівнює refldxB, і availableFlagLXA вважається таким, що дорівнює 1.
- Коли isScaledFlagLX дорівнює 0, то availableFlagLXB вважається таким, що дорівнює 0 і для  $(x_{B_k}, y_{B_k})$  від  $(x_{B_0}, y_{B_0})$  до  $(x_{B_2}, y_{B_2})$ , де  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ , і  $x_{B_2} = x_P - \text{MinPuSize}$ , наступне застосовується повторно, поки availableFlagLXB не дорівнює 1:

- Якщо елемент прогнозування, що покриває місцерозташування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[ $x_{B_k}$ ][ $y_{B_k}$ ] дорівнює 1, і RefPicListX[refldxLX] і RefPicListX[refldxLX[ $x_{B_k}$ ][ $y_{B_k}$ ]] обидва є довгостроковими опорними зображеннями або обидва є короткостроковими опорними зображеннями, то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору

$mvLX[xB_k][yB_k]$  руху,  $refldxB$  вважається таким, що дорівнює  $refldxLX[xB_k][yB_k]$ , а  $ListB$  вважається таким, що дорівнює  $ListX$ .

- Інакше, якщо елемент прогнозування, що покриває місцерозташування ( $xB_k$ ,  $yB_k$ ) яскравості доступний [Ed. (BB): Переписати з використанням  $MinCbAddrZS[]$  і процесу доступності для мінімальних блоків кодування], і  $PredMode$  не  $MODE\_INTRA$ ,  $predFlagLY[xB_k][yB_k]$  (де  $Y \neq X$ ) дорівнює 1, і  $RefPicListX[refldxLX]$  і  $RefPicListY[refldxLY[xB_k][yB_k]]$  обидва є довгостроковими опорними зображеннями або обидва є короткостроковими опорними зображеннями, то  $availableFlagLXB$  вважається таким, що дорівнює 1, а вектор  $mvLXB$  руху вважається таким, що дорівнює вектору  $mvLY[xB_k][yB_k]$  руху,  $refldxB$  вважається таким, що дорівнює  $refldxLY[xB_k][yB_k]$ , а  $ListB$  вважається таким, що дорівнює  $ListY$ .

- Коли  $availableFlagLXB$  дорівнює 1,  $PicOrderCnt(RefPicListB[refldxB])$  не дорівнює  $PicOrderCnt(RefPicListX[refldxLX])$ , і  $RefPicListB[refldxB]$  і  $RefPicListX[refldxLX]$  обидва є короткостроковими опорними зображеннями, то  $mvLXB$  витягають, як вказано нижче (де позначення 8-### стосується розділів поточної робочої версії HEVC).

$tx = (16384 + (Abs(td) >> 1)) / td$  (8-134)  
 $DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) >> 6)$  (8-135)  
 $mvLXB = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvLXA) * ((Abs(DistScaleFactor * mvLXA) + 127) >> 8))$  (8-136),

де  $td$  і  $tb$  можуть бути витягнуті так

$td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicList[refldxB]))$  (8-137)

$tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refldxLX]))$  (8-138).

Відеокодер 20 і відеодекодер 30 можуть також бути виконані з можливістю виконувати модифікований процес витягування для часового прогнозування вектора руху (TMVP) для кодування векторів руху блоків сигналу яскравості. В одному прикладі, введення в цей процес містять у собі місцерозташування ( $xP$ ,  $yP$ ) яскравості, що вказує верхній лівий відлік яскравості поточного елемента прогнозування відносно верхнього лівого відліку поточного зображення, змінні, що вказують ширину і висоту елемента прогнозування для яскравості,  $nPSW$  і  $nPSH$ , і опорний індекс розділення  $refldxLX$  (де  $X$  дорівнює 0 або 1) поточного елемента прогнозування. Виведення цього процесу можуть містити в собі прогнозування  $mvLXCol$  вектора руху і прапор  $availableFlagLXCol$  доступності.

В одному прикладі, відеокодер 20 і відеодекодер 30 можуть виконувати функцію  $RefPicOrderCnt(picX, refldx, LX)$ , що повертає рахунок  $PicOrderCntVal$  порядку зображення опорного зображення з індексом  $refldx$  зі списку  $LX$  опорних зображень зображення  $picX$ . Ця функція може бути визначена, як вказано далі, де (8-141) і подібні посилання в цьому описі стосуються розділів HEVC WD7:

$RefPicOrderCnt(picX, refldx, LX) = PicOrderCnt(RefPicListX[refldx]$  зображення  $PicX$ ) (8-141)

Залежно від значень  $slice\_type$ ,  $collocated\_from\_10\_flag$  і  $collocated\_ref\_idx$ , змінна  $colPic$ , що визначає зображення, яке містить суміщене розділення, може бути витягнута наступним чином:

- Якщо  $slice\_type$  дорівнює  $B$  і  $collocated\_from\_10\_flag$  дорівнює 0, то змінна  $colPic$  визначає зображення, що містить суміщене розділення, як визначено  $RefPicList1[collocated\_ref\_idx]$ .

- Інакше, ( $slice\_type$  дорівнює  $B$  і  $collocated\_from\_10\_flag$  дорівнює 1 або  $slice\_type$  дорівнює  $P$ ), змінна  $colPic$  визначає зображення, що містить суміщене розділення, як вказано  $RefPicList0[collocated\_ref\_idx]$ .

Змінна  $colPu$  і її положення ( $xPCol$ ,  $yPCol$ ) можуть бути витягнуті з використанням наступного порядку етапів:

1. Змінна  $colPu$  може бути витягнута наступним чином

$yPRb = yP + nPSH$  (8-139)

- Якщо  $(yP >> Log2CtbSize)$  дорівнює  $(yPRb >> Log2CtbSize)$ , то горизонтальний компонент правого нижнього положення яскравості поточного елемента прогнозування визначається так

$xPRb = xP + nPSW$  (8-140)

і змінна  $colPu$  вважається такою, що дорівнює елементу прогнозування, що покриває модифіковане положення, задане  $((xPRb >> 4) << 4, (yPRb >> 4) << 4)$  усередині  $colPic$ .

- Інакше,  $((yP >> Log2CtbSize) \neq (yPRb >> Log2CtbSize))$ ,  $colPu$  маркується як "недоступна".

2. Коли  $colPU$  кодується в режимі внутрішнього прогнозування, або  $colPu$  промаркована як "недоступна", то застосовується наступне.

- Центральне положення яскравості поточного елемента прогнозування визначається так

$xPCtr = (xP + (nPSW >> 1))$  (8-141)

$yPCtr = (yP + (nPSH >> 1))$  (8-142)



- Змінна ColPu вважається такою, що дорівнює елементу прогнозування, що покриває модифіковане положення, задане  $((xPCtr > 4) < 4)$ ,  $(yPCtr > 4) < 4)$  у середині colPic.

$(xPCol, yPCol)$  вважається таким, що дорівнює верхньому лівому відліку яскравості colPu відносно верхнього лівого відліку яскравості colPic.

5 Функція LongTermRefPic (picX, refIdx, LX) може бути визначена наступним чином. Якщо опорне зображення з індексом refIdx зі списку LX опорних зображень зображення picX було промарковане як "використовуване для довгострокового посилання" у момент, коли picX було поточним зображенням, то LongTermRefPic (picX, refIdx, LX) повертає 1; інакше, LongTermRefPic (picX, refIdx, LX) повертає 0.

10 Змінні mvLXCol і availableFlagLXCol можуть бути витягнуті наступним чином, де підкреслений текст являє собою зміни відносно HEVC WD7:

- Якщо одна або кілька наступних умов істинні, то обидві компоненти mvLXCol вважаються такими, що дорівнюють 0, і availableFlagLXCol вважається таким, що дорівнює 0.

- colPU кодується в режимі внутрішнього прогнозування.

15 - colPu промаркована як "недоступна".

- pic\_temporal\_mvp\_enable\_flag дорівнює 0.

- Інакше, вектор mvCol руху, опорний індекс refIdxCol і ідентифікатор listCol опорного списку витягаються наступним чином.

20 - Якщо PredFlagL0[xPCol][yPCol] дорівнює 0, то mvCol, refIdxCol і listCol вважаються такими, що дорівнюють MvL1[xPCol][yPCol], RefIdxL1[xPCol][yPCol] і L1, відповідно.

- Інакше, (PredFlagL0[xPCol][yPCol] дорівнює 1), застосовується наступне.

- Якщо PredFlagL1[xPCol][yPCol] дорівнює 0, то mvCol, refIdxCol і listCol вважаються такими, що дорівнюють MvL0[xPCol][yPCol], RefIdxL0[xPCol][yPCol] і L0, відповідно.

- Інакше, (PredFlagL1[xPCol][yPCol] дорівнює 1), виконуються наступні призначення.

25 - Якщо PicOrderCnt(pic) кожного зображення pic у кожному списку опорних зображень менше або дорівнює PicOrderCntVal, то mvCol, refIdxCol і listCol вважаються такими, що дорівнюють MvLX[xPCol][yPCol], RefIdxLX[xPCol][yPCol] і LX, відповідно, де X є значенням X, для якого викликаний цей процес.

30 - Інакше, (PicOrderCnt(pic) щонайменше одного зображення pic у щонайменше одному списку опорних зображень, більше, ніж PicOrderCntVal, то mvCol, refIdxCol і listCol вважаються такими, що дорівнюють MvLN[xPCol][yPCol], RefIdxLN[xPCol][yPCol] і LN, відповідно, де N є значенням collocated\_from\_10\_flag.

- Якщо одна з наступних умов істинна, то змінна availableFlagLXCol вважається такою, що дорівнює 0:

35 - RefPicListX[refIdxLX] є довгостроковим опорним зображенням, і LongTermRefPic(colPic, refIdxCol, listCol) дорівнює 0;

- RefPicListX[refIdxLX] є короткостроковим опорним зображенням, і LongTermRefPic(colPic, refIdxCol, listCol) дорівнює 1;

40 - Інакше, змінна availableFlagLXCol вважається такою, що дорівнює 1, і застосовується наступне.

- Якщо RefPicListX[refIdxLX] є довгостроковим опорним зображенням, або LongTermRefPic(colPic, refIdxCol, listCol) дорівнює 1, або PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol) дорівнює PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]), то

45 mvLXCol = mvCol (8-143)

- Інакше, mvLXCol витягається як масштабована версія вектора mvCol руху, як визначено нижче

$tx = (16384 + (\text{Abs}(td) > 1)) / td$  (8-144)

$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) > 6)$  (8-145)

50  $mvLXCol = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvCol) * ((\text{Abs}(\text{DistScaleFactor} * mvCol) + 127) > 8))$  (8-146),

де td і tb можуть бути витягнуті так

$td = \text{Clip3}(-128, 127, \text{PicOrderCnt}(\text{colPic}) - \text{RefPicOrderCnt}(\text{colPic}, \text{refIdxCol}, \text{listCol}))$  (8-147)

$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}]))$  (8-148).

55 У вищевказаному прикладі, доступність суміщеного блока, використовуваного під час TMVP, також може залежати від типу опорного зображення (наприклад, чи є зображення довгостроковим або короткостроковим опорним зображенням) для суміщеного блока. Тобто, навіть коли правий нижній блок для TMVP доступний (після етапу 1 у підпункті), то нижній правий блок може додатково вважатися недоступним, якщо вектор руху в блоці посиляється на тип зображення (короткостроковий або довгостроковий), що відрізняється від типу цільового

60

опорного зображення. Подібним чином, центральний блок може додатково бути використаний для TMVP.

Наприклад, відеокoder 20 і відеокoder 30 можуть бути виконані з можливістю витягати предиктор вектора руху для вектора руху яскравості, відповідно до наступного докладного прикладу. Введення в цей процес, здійснювані відеокoderом 20 і відеокoderом 30, можуть містити в собі:

- місцезнаходження ( $x_P$ ,  $y_P$ ) яскравості, що визначає верхній лівий відлік яскравості поточного елемента прогнозування відносно верхнього лівого відліку поточного зображення,
- змінні, які визначають ширину і висоту елемента прогнозування для яскравості,  $nPSW$  і  $nPSH$ ,
- опорний індекс розділення  $refIdx_{LX}$  (де  $X$  дорівнює 0 або 1) поточного елемента прогнозування.

Виведення цього процесу можуть містити в собі:

- прогнозування  $mvLXCol$  вектора руху,
- прапор  $availableFlagLXCol$  доступності.

Функція  $RefPicOrderCnt(picX, refIdx, LX)$ , при виконанні її відеокoderом 20 і/або відеокoderом 30, може повертати рахунок  $PicOrderCntVal$  порядку зображення опорного зображення зі списку  $LX$  опорних зображень зображення  $picX$ . Приклад здійснення цієї функції визначений наступним чином:

$RefPicOrderCnt(picX, refIdx, LX) = PicOrderCnt(RefPicListX[refIdx] \text{ зображення } picX) \quad (8-141)$

Залежно від значень  $slice\_type$ ,  $collocated\_from\_10\_flag$  і  $collocated\_ref\_idx$ , змінна  $colPic$ , що визначає зображення, що містить суміщене розділення, може бути витягнута наступним чином:

- Якщо  $slice\_type$  дорівнює  $B$  і  $collocated\_from\_10\_flag$  дорівнює 0, то змінна  $colPic$  визначає зображення, що містить суміщене розділення, як визначено  $RefPicList1[collocated\_ref\_idx]$ .
- Інакше, ( $slice\_type$  дорівнює  $B$  і  $collocated\_from\_10\_flag$  дорівнює 1 або  $slice\_type$  дорівнює  $P$ ), змінна  $colPic$  визначає зображення, що містить суміщене розділення, як вказано  $RefPicList0[collocated\_ref\_idx]$ .

Відеокoder 20 і відеокoder 30 можуть витягати змінну  $colPu$  і її положення ( $xPCol$ ,  $yPCol$ ) з використанням наступного порядку етапів:

1. Відеокoder 20 і відеокoder 30 можуть витягати змінну  $colPu$  наступним чином:

$yPRb = yP + nPSH \quad (8-139)$

- Якщо  $(yP \gg \log_2 CtbSize)$  дорівнює  $(yPRb \gg \log_2 CtbSize)$ , то горизонтальний компонент правого нижнього положення яскравості поточного елемента прогнозування визначається так  $xPRb = xP + nPSW \quad (8-140)$

і змінна  $colPu$  вважається такою, що дорівнює елементу прогнозування, що покриває модифіковане положення, задане  $((xPRb \gg 4) \ll 4, (yPRb \gg 4) \ll 4)$  у середині  $colPic$ .

- Інакше,  $(yP \gg \log_2 CtbSize)$  не дорівнює  $(yPRb \gg \log_2 CtbSize)$ , то відеокoder 20 і відеокoder 30 можуть маркувати  $colPu$  як "недоступна".

2. Коли  $colPU$  кодується в режимі внутрішнього прогнозування, або  $colPu$  промаркована як "недоступна", то застосовується наступне, у даному прикладі:

- Центральне положення яскравості поточного елемента прогнозування визначається так

$xPCtr = (xP + (nPSW \gg 1)) \quad (8-141)$

$yPCtr = (yP + (nPSH \gg 1)) \quad (8-142)$

- Змінна  $colPu$  вважається такою, що дорівнює елементу прогнозування, що покриває модифіковане положення, задане  $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$  у середині  $colPic$ .

3. Відеокoder 20 і відеокoder 30 можуть установлювати ( $xPCol$ ,  $yPCol$ ) рівним верхньому лівому відліку яскравості  $colPu$  відносно верхнього лівого відліку яскравості  $colPic$ .

Функція  $LongTermRefPic(picX, refIdx, LX)$  може бути визначена наступним чином: Якщо опорне зображення з індексом  $refIdx$  зі списку  $LX$  опорних зображень зображення  $picX$  було промарковано як "використовуване для довгострокового посилання" у момент, коли  $picX$  було поточним зображенням, то  $LongTermRefPic(picX, refIdx, LX)$  повертає 1; інакше,  $LongTermRefPic(picX, refIdx, LX)$  повертає 0.

Відеокoder 20 і відеокoder 30 можуть витягати змінні  $mvLXCol$  і  $availableFlagLXCol$  наступним чином:

$availableFlagLXCol$  вважається таким, що дорівнює 0,  $numTestBlock$  дорівнює 0.

Якщо  $numTestBlock$  менше, ніж 2, і  $availableFlagLXCol$  дорівнює 0, то виконується наступний порядок.

$xPCtr = (xP + (nPSW \gg 1))$

$yPCtr = (yP + (nPSH \gg 1))$

- Якщо colPu покриває положення, задане  $((xPCtr >> 4) << 4, (yPCtr >> 4) << 4)$  у середині colPic, то numTestBlock вважається таким, що дорівнює 1;

- Інакше, якщо numTestBlock дорівнює 1, то colPu вважається таким, що дорівнює елементу прогнозування, що покриває модифіковане положення, задане  $((xPCtr >> 4) << 4, (yPCtr >> 4) << 4)$  у середині colPic, і  $(xPCol, yPCol)$  вважається таким, що дорівнює верхньому лівому відліку яскравості colPu відносно верхнього лівого відліку яскравості colPic.

5 numTestBlock++

- Якщо одна або кілька наступних умов істинні, то обидві компоненти mvLXCol вважаються такими, що дорівнюють 0, і availableFlagLXCol вважається таким, що дорівнює 0.

10 - colPU кодується в режимі внутрішнього прогнозування.

- colPu промаркована як "недоступна".

- pic\_temporal\_mvp\_enable\_flag дорівнює 0.

- Інакше, вектор mvCol руху, опорний індекс refIdxCol і ідентифікатор listCol опорного списку витягаються наступним чином.

15 - Якщо PredFlagL0[xPCol][yPCol] дорівнює 0, то mvCol, refIdxCol і listCol вважаються такими, що дорівнюють MvL1[xPCol][yPCol], RefIdxL1[xPCol][yPCol] і L1, відповідно.

- Інакше, (PredFlagL0[xPCol][yPCol] дорівнює 1), застосовується наступне.

- Якщо PredFlagL1[xPCol][yPCol] дорівнює 0, то mvCol, refIdxCol і listCol вважаються такими, що дорівнюють MvL0[xPCol][yPCol], RefIdxL0[xPCol][yPCol] і L0, відповідно.

20 - Інакше, (PredFlagL1[xPCol][yPCol] дорівнює 1), виконуються наступні призначення.

- Якщо PicOrderCnt(pic) кожного зображення pic у кожному списку опорних зображень менше або дорівнює PicOrderCntVal, то mvCol, refIdxCol і listCol вважаються такими, що дорівнюють MvLX[xPCol][yPCol], RefIdxLX[xPCol][yPCol] і LX, відповідно, де X є значенням X, для якого викликаний цей процес.

25 - Інакше, (PicOrderCnt(pic) щонайменше одного зображення pic у щонайменше одному списку опорних зображень більше, ніж PicOrderCntVal, то mvCol, refIdxCol і listCol вважаються такими, що дорівнюють MvLN[xPCol][yPCol], RefIdxLN[xPCol][yPCol] і LN, відповідно, де N є значенням collocated\_from\_10\_flag.

- Якщо одна з наступних умов істинна, то змінна availableFlagLXCol вважається такою, що дорівнює 0:

30 - RefPicListX[refIdxLX] є довгостроковим опорним зображенням, і LongTermRefPic(colPic, refIdxCol, listCol) дорівнює 0;

- RefPicListX[refIdxLX] є короткостроковим опорним зображенням, і LongTermRefPic(colPic, refIdxCol, listCol) дорівнює 1;

35 - Інакше, змінна availableFlagLXCol вважається такою, що дорівнює 1, і застосовується наступне.

- Якщо RefPicListX[refIdxLX] є довгостроковим опорним зображенням, або LongTermRefPic(colPic, refIdxCol, listCol) дорівнює 1, або PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol) дорівнює PicOrderCntVal-

40 PicOrderCnt(RefPicListX[refIdxLX]), то

mvLXCol=mvCol (8-143)

- Інакше, mvLXCol витягається як масштабована версія вектора mvCol руху, як визначено нижче

tx=(16384+(Abs(td)>>1))/td (8-144)

45 DistScaleFactor=Clip3(-4096, 4095, (tb\* tx+32)>>6) (8-145)

mvLXCol=Clip3(-8192, 8191.75, Sign(DistScaleFactor\* mvCol)\* ((Abs(DistScaleFactor\* mvCol)+127)>>8)) (8-146)

де td і tb можуть бути витягнуті так:

td=Clip3(-128, 127, PicOrderCnt(colPic)- RefPicOrderCnt(colPic, refIdxCol, listCol)) (8-147)

50 tb=Clip3(-128, 127, PicOrderCntVal-PicOrderCnt(RefPicListX[refIdxLX])) (8-148).

В альтернативному прикладі, довгостроковий вектор руху ніколи не прогнозується з іншого довгострокового вектора руху, якщо значення РОС опорних зображень неоднакові. Відеокодер 20 і відеокодер 30 можуть бути виконані відповідно до наступного процесу витягування кандидатів предиктора вектора руху, де підкреслений текст являє собою зміни відносно HEVC

55 WD7.

Змінна isScaledFlagLX, де X дорівнює 0 або 1, може вважатися рівною 0. Вектор mvLXA руху і прапор availableFlagLXA доступності витягаються в наступному порядку етапів, де багатокрапки являють собою той же текст, що й у поточній робочій версії HEVC, а підкреслення являє собою зміни відносно поточної робочої версії HEVC:

60 1. ...

2. ...  
3. ...  
4. ...

5. Коли availableFlagLXA дорівнює 0, для  $(x_{A_k}, y_{A_k})$  від  $(x_{A_0}, y_{A_0})$  до  $(x_{A_1}, y_{A_1})$ , де  $y_{A_1}=y_{A_0}-\text{MinPuSize}$ , наступне застосовується повторно доти, поки availableFlagLXA не дорівнює 1:

- Якщо елемент прогнозування, що покриває місцерозташування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[ $x_{A_k}$ ][ $y_{A_k}$ ] дорівнює 1, і RefPicListX[refIdxLX] і RefPicListX[refIdxLX[ $x_{A_k}$ ][ $y_{A_k}$ ]] обидва є довгостроковими опорними зображеннями з різними значеннями РОС або обидва є короткостроковими опорними зображеннями, то availableFlagLXA вважається таким, що дорівнює 1, вектор mvLXA руху вважається таким, що дорівнює вектору mvLX[ $x_{A_k}$ ][ $y_{A_k}$ ] руху, refIdxA вважається таким, що дорівнює refIdxLX[ $x_{A_k}$ ][ $y_{A_k}$ ], а ListA вважається таким, що дорівнює ListX.

- Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY[ $x_{A_k}$ ][ $y_{A_k}$ ] (де  $Y=X$ ) дорівнює 1, і RefPicListX[refIdxLX] і RefPicListY[refIdxLY[ $x_{A_k}$ ][ $y_{A_k}$ ]] обидва є довгостроковими опорними зображеннями з різними значеннями РОС або обидва є короткостроковими опорними зображеннями, то availableFlagLXA вважається таким, що дорівнює 1, а вектор mvLXA руху вважається таким, що дорівнює вектору mvLY[ $x_{A_k}$ ][ $y_{A_k}$ ] руху, refIdxA вважається таким, що дорівнює refIdxLY[ $x_{A_k}$ ][ $y_{A_k}$ ], а ListA вважається таким, що дорівнює ListY.

- Коли availableFlagLXA дорівнює 1, і RefpicListA[refIdxLA] і RefPicListX[refIdxLX] обидва є короткостроковими опорними зображеннями, то mvLXA витягають, як вказано нижче.

25 
$$Tx = (16384 + (\text{Abs}(td) > 1)) / td \quad (8-126)$$
  
$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) > 6) \quad (8-127)$$
  
$$mvLXA = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * ((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) > 8)) \quad (8-128),$$

де td і tb можуть бути витягнуті так

30 
$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refIdxA}])) \quad (8-129)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}])) \quad (8-130)$$

Відеокодер 20 і відеокодер 30 можуть витягати вектор mvLXB руху і прапор availableFlagLXB доступності з використанням наступного порядку етапів, де багатокрапки являють собою той же текст, що й у HEVC WD7:

35 1. ...  
2. ...  
3. ...  
4. ...  
5. ...

40 6... Коли isScaledFlagLX дорівнює 0, то availableFlagLXB вважається таким, що дорівнює 0 і для  $(x_{B_k}, y_{B_k})$  від  $(x_{B_0}, y_{B_0})$  до  $(x_{B_2}, y_{B_2})$ , де  $x_{B_0}=x_P+n\text{PSW}$ ,  $x_{B_1}=x_{B_0}-\text{MinPuSize}$ , і  $x_{B_2}=x_P-\text{MinPuSize}$ , наступне застосовується повторно, поки availableFlagLXB не дорівнює 1:

45 - Якщо елемент прогнозування, що покриває місцерозташування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[ $x_{B_k}$ ][ $y_{B_k}$ ] дорівнює 1, і RefPicListX[refIdxLX] і RefPicListX[refIdxLX[ $x_{B_k}$ ][ $y_{B_k}$ ]] обидва є довгостроковими опорними зображеннями з різними значеннями РОС або обидва є короткостроковими опорними зображеннями, то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLX[ $x_{B_k}$ ][ $y_{B_k}$ ] руху, refIdxB вважається таким, що дорівнює refIdxLX[ $x_{B_k}$ ][ $y_{B_k}$ ], а ListB вважається таким, що дорівнює ListX.

50 - Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY[ $x_{B_k}$ ][ $y_{B_k}$ ] (де  $Y=X$ ) дорівнює 1, і RefPicListX[refIdxLX] і RefPicListY[refIdxLY[ $x_{B_k}$ ][ $y_{B_k}$ ]] обидва є довгостроковими опорними зображеннями з різними значеннями РОС або обидва є короткостроковими опорними зображеннями, то availableFlagLXB вважається таким, що дорівнює 1, а вектор mvLXB руху вважається таким, що дорівнює вектору mvLY[ $x_{B_k}$ ][ $y_{B_k}$ ] руху, refIdxB вважається таким, що дорівнює refIdxLY[ $x_{B_k}$ ][ $y_{B_k}$ ], а ListB вважається таким, що дорівнює ListY.

- Коли availableFlagLXB дорівнює 1, PicOrderCnt(RefPicListB[refldxB]) не дорівнює PicOrderCnt(RefPicListX[refldXLX]), і RefPicListB[refldxB] і RefPicListX[refldXLX] обидва є короткостроковими опорними зображеннями, то mvLXB витягають, як вказано нижче.

$tx = (16384 + (\text{Abs}(td) \gg 1)) / td$  (8-134)

5  $\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6)$  (8-135)

$mvLXB = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * ((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) \gg 8))$  (8-136)

де td і tb можуть бути витягнуті так

$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicList}[\text{refldxB}]))$  (8-137)

10  $tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldXLX}]))$  (8-138)

Відеокодер 20 і відеодекодер 30 можуть бути виконані з можливістю витягати часові предиктори вектора руху яскравості, відповідно до наступного процесу, де підкреслений текст являє собою зміни відносно HEVC WD7. Змінні mvLXCol і availableFlagLXCol можуть бути витягнуті наступним чином:

15 - Якщо одна або кілька наступних умов істинні, ті обидві компоненти mvLXCol вважаються такими, що дорівнюють 0, і availableFlagLXCol вважається таким, що дорівнює 0.

- colPU кодується в режимі внутрішнього прогнозування.

- colPu промаркована як "недоступна".

- pic\_temporal\_mvp\_enable\_flag дорівнює 0.

20 - Інакше, вектор mvCol руху, опорний індекс refldxCol і ідентифікатор listCol опорного списку витягаються наступним чином.

- Якщо PredFlagL0[xPCol][yPCol] дорівнює 0, то mvCol, refldxCol і listCol вважаються такими, що дорівнюють MvL1[xPCol][yPCol], RefldxL1[xPCol][yPCol] і L1, відповідно.

- Інакше, (PredFlagL0[xPCol][yPCol] дорівнює 1), застосовується наступне.

25 - Якщо PredFlagL1[xPCol][yPCol] дорівнює 0, то mvCol, refldxCol і listCol вважаються такими, що дорівнюють MvL0[xPCol][yPCol], RefldxL0[xPCol][yPCol] і L0, відповідно.

- Інакше, (PredFlagL1[xPCol][yPCol] дорівнює 1), виконуються наступні призначення.

- Якщо PicOrderCnt(pic) кожного зображення піс у кожному списку опорних зображень менше або дорівнює PicOrderCntVal, то mvCol, refldxCol і listCol вважаються такими, що дорівнюють MvLX[xPCol][yPCol], RefldxLX[xPCol][yPCol] і LX, відповідно, де X є значенням X, для якого викликаний цей процес.

30 - Інакше, (PicOrderCnt(pic) щонайменше одного зображення піс у щонайменше одному списку опорних зображень, більше, ніж PicOrderCntVal, то mvCol, refldxCol і listCol вважаються такими, що дорівнюють MvLN[xPCol][yPCol], RefldxLN[xPCol][yPCol] і LN, відповідно, де N є значенням collocated\_from\_10\_flag.

35 - Якщо одна з наступних умов істинна, то змінна availableFlagLXCol вважається такою, що дорівнює 0:

- RefPicListX[refldXLX] є довгостроковим опорним зображенням, і LongTermRefPic(colPic, refldxCol, listCol) дорівнює 0;

40 - RefPicListX[refldXLX] є короткостроковим опорним зображенням, і LongTermRefPic(colPic, refldxCol, listCol) дорівнює 1;

- RefPicListX[refldXLX] є довгостроковим опорним зображенням, і LongTermRefPic(colPic, refldxCol, listCol) дорівнює 1, і RefPicOrderCnt(colPic, refldxCol, listCol) не дорівнює PicOrderCnt(RefPicListX[refldXLX]).

45 - Інакше, змінна availableFlagLXCol вважається такою, що дорівнює 1, і застосовується наступне.

- Якщо RefPicListX[refldXLX] є довгостроковим опорним зображенням, або LongTermRefPic(colPic, refldxCol, listCol) дорівнює 1, або PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refldxCol, listCol) дорівнює PicOrderCntVal-

50 PicOrderCnt(RefPicListX[refldXLX]), то

$mvLXCol = mvCol$  (8-143)

- Інакше, mvLXCol витягається як масштабована версія вектора mvCol руху, як визначено нижче

$tx = (16384 + (\text{Abs}(td) \gg 1)) / td$  (8-144)

55  $\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6)$  (8-145)

$mvLXCol = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvCol) * ((\text{Abs}(\text{DistScaleFactor} * mvCol) + 127) \gg 8))$  (8-146),

де td і tb можуть бути витягнуті так

$td = \text{Clip3}(-128, 127, \text{PicOrderCnt}(\text{colPic}) - \text{RefPicOrderCnt}(\text{colPic}, \text{refldxCol}, \text{listCol}))$  (8-147)

60  $tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldXLX}]))$  (8-148).

В іншому прикладі, міжвидове опорне зображення може бути марковане як "невикористовуване для посилання". Для простоти, таке зображення можна називати неопорне зображення в базовій специфікації HEVC, і зображення, марковане як "використовуване для довгострокового посилання" або "використовуване для короткострокового посилання" можна називати опорним зображенням у базовій специфікації HEVC. Терміни "опорне зображення" і "неопорне зображення" можна замінити на "зображення, марковане як "використовуване для посилання" і "зображення, марковане як "невикористовуване для посилання".

Функцію `UnusedRefPic(picX, refIdx, LX)` можна визначити наступним чином. Якщо опорне зображення з індексом `refIdx` зі списку `LX` опорних зображень зображення `picX` було марковане як "невикористовуване для посилання" у момент, коли `picX` було поточним зображенням, то `UnusedRefPic(picX, refIdx, LX)` повертає 1; інакше, `UnusedRefPic(picX, refIdx, LX)` повертає 0.

Відеокодер 20 і відеодекодер 30 можуть бути виконані з можливістю виконувати процес витягування кандидатів предиктора вектора руху наступним чином, де підкреслений текст являє собою зміни відносно HEVC WD7, а багатокрапки являють собою той же текст, що й у HEVC WD7:

Змінна `isScaledFlagLX`, де `X` дорівнює 0 або 1, може вважатися такою, що дорівнює 1 або 0.

Вектор `mvLXA` руху і прапор `availableFlagLXA` доступності можуть бути витягнуті в наступному порядку етапів:

1. ...

2. ...

3. ...

4. ...

5. Коли `availableFlagLXA` дорівнює 0, то для  $(x_{A_k}, y_{A_k})$  від  $(x_{A_0}, y_{A_0})$  до  $(x_{A_1}, y_{A_1})$ , де  $y_{A_1}=y_{A_0}-\text{MinPuSize}$ , наступне застосовується повторно, поки `availableFlagLXA` не дорівнює 1:

- Якщо елемент прогнозування, що покриває місцезонашування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням `MinCbAddrZS[]` і процесу доступності для мінімальних блоків кодування], і `PredMode` не `MODE_INTRA`, `predFlagLX[xAk][yAk]` дорівнює 1, і `RefPicListX[refIdxLX]` і `RefPicListX[refIdxLX[xAk][yAk]]` обидва є опорними зображеннями або обидва є неопорними зображеннями, то `availableFlagLXA` вважається таким, що дорівнює 1, вектор `mvLXA` руху вважається таким, що дорівнює вектору `mvLX[xAk][yAk]` руху, `refIdxA` вважається таким, що дорівнює `refIdxLX[xAk][yAk]`, а `ListA` вважається таким, що дорівнює `ListX`.

- Інакше, якщо елемент прогнозування, що покриває місцезонашування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням `MinCbAddrZS[]` і процесу доступності для мінімальних блоків кодування], і `PredMode` не `MODE_INTRA`, `predFlagLY[xAk][yAk]` (де  $Y=X$ ) дорівнює 1, і `RefPicListX[refIdxLX]` і `RefPicListY[refIdxLY[xAk][yAk]]` обидва є опорними зображеннями або обидва є неопорними зображеннями, то `availableFlagLXA` вважається таким, що дорівнює 1, а вектор `mvLXA` руху вважається таким, що дорівнює вектору `mvLY[xAk][yAk]` руху, `refIdxA` вважається таким, що дорівнює `refIdxLY[xAk][yAk]`, а `ListA` вважається таким, що дорівнює `ListY`.

- Коли `availableFlagLXA` дорівнює 1, і `RefPicListA[refIdxLA]` і `RefPicListX[refIdxLX]` обидва є короткостроковими опорними зображеннями, то `mvLXA` витягають, як вказано нижче.

$Tx = (16384 + (\text{Abs}(td) >> 1)) / td$  (8-126)

$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6)$  (8-127)

$mvLXA = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * ((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) >> 8))$  (8-128),

де `td` і `tb` можуть бути витягнуті так

$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refIdxA}]))$  (8-129)

$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}]))$  (8-130).

Відеокодер 20 і відеодекодер 30 можуть бути виконані з можливістю витягати вектор `mvLXB` руху і прапор `availableFlagLXB` доступності з використанням наступного порядку етапів, де підкреслений текст являє собою зміни відносно HEVC WD7:

1. ...

2. ...

3. ...

4. ...

5. ...

6... Коли `isScaledFlagLX` дорівнює 0, то `availableFlagLXB` вважається таким, що дорівнює 0 і для  $(x_{B_k}, y_{B_k})$  від  $(x_{B_0}, y_{B_0})$  до  $(x_{B_2}, y_{B_2})$ , де  $x_{B_0}=x_P+nPSW$ ,  $x_{B_1}=x_{B_0}-\text{MinPuSize}$ , і  $x_{B_2}=x_P-\text{MinPuSize}$ , наступне застосовується повторно, поки `availableFlagLXB` не дорівнює 1:

- Якщо елемент прогнозування, що покриває місцезонаштування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[xB<sub>k</sub>][yB<sub>k</sub>] дорівнює 1, і RefPicListX[refldxLX] і RefPicListX[refldxLX[xB<sub>k</sub>][yB<sub>k</sub>]] обидва є опорними зображеннями або

5 обидва є неопорними зображеннями, то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLX[xB<sub>k</sub>][yB<sub>k</sub>] руху, refldxB вважається таким, що дорівнює refldxLX[xB<sub>k</sub>][yB<sub>k</sub>], а ListB вважається таким, що дорівнює ListX.

- Інакше, якщо елемент прогнозування, що покриває місцезонаштування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY[xB<sub>k</sub>][yB<sub>k</sub>] (де Y=X) дорівнює 1, і RefPicListX[refldxLX] і RefPicListY[refldxLY[xB<sub>k</sub>][yB<sub>k</sub>]] обидва є опорними зображеннями або обидва є неопорними зображеннями, то availableFlagLXB вважається таким, що дорівнює 1, а вектор mvLXB руху вважається таким, що дорівнює вектору mvLY[xB<sub>k</sub>][yB<sub>k</sub>] руху, refldxB вважається таким, що дорівнює refldxLY[xB<sub>k</sub>][yB<sub>k</sub>], а ListB вважається

15 таким, що дорівнює ListY.

- Коли availableFlagLXB дорівнює 1, PicOrderCnt(RefPicListB[refldxB]) не дорівнює PicOrderCnt(RefPicListX[refldxLX]), і RefPicListB[refldxB] і RefPicListX[refldxLX] обидва є короткостроковими опорними зображеннями, то mvLXB витягають, як вказано нижче.

tx=(16384+(Abs(td)>>1))/td (8-134)

20 DistScaleFactor=Clip3(-4096, 4095, (tb\*tx+32)>>6) (8-135)

mvLXB=Clip3(-8192, 8191.75, Sign(DistScaleFactor\* mvLXA)\* ((Abs(DistScaleFactor\* mvLXA)+127)>>8)) (8-136),

де td і tb можуть бути витягнуті так

td=Clip3(-128, 127, PicOrderCntVal-PicOrderCnt(RefPicList[refldxB])) (8-137)

25 tb=Clip3(-128, 127, PicOrderCntVal-PicOrderCnt(RefPicListX[refldxLX])) (8-138)

Відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю витягати часові предиктори вектора руху яскравості наступним чином, де підкреслений текст являє собою зміни відносно HEVC WD7:

Змінні mvLXCol і availableFlagLXCol можуть бути витягнуті наступним чином.

30 - Якщо одна або кілька наступних умов істинні, ті обидві компоненти mvLXCol вважаються такими, що дорівнюють 0, і availableFlagLXCol вважається таким, що дорівнює 0.

- colPU кодується в режимі внутрішнього прогнозування.
- colPu промаркована як "недоступна".
- pic\_temporal\_mvp\_enable\_flag дорівнює 0.

35 - Інакше, вектор mvCol руху, опорний індекс refldxCol і ідентифікатор listCol опорного списку витягаються наступним чином.

- Якщо PredFlagL0[xPCol][yPCol] дорівнює 0, то mvCol, refldxCol і listCol вважаються такими, що дорівнюють MvL1[xPCol][yPCol], RefldxL1[xPCol][yPCol] і L1, відповідно.
- Інакше, (PredFlagL0[xPCol][yPCol] дорівнює 1), застосовується наступне.

40 - Якщо PredFlagL1[xPCol][yPCol] дорівнює 0, то mvCol, refldxCol і listCol вважаються такими, що дорівнюють MvL0[xPCol][yPCol], RefldxL0[xPCol][yPCol] і L0, відповідно.

- Інакше, (PredFlagL1[xPCol][yPCol] дорівнює 1), виконуються наступні призначення.
- Якщо PicOrderCnt(pic) кожного зображення піс у кожному списку опорних зображень менше або дорівнює PicOrderCntVal, то mvCol, refldxCol і listCol вважаються такими, що дорівнюють MvLX[xPCol][yPCol], RefldxLX[xPCol][yPCol] і LX, відповідно, де X є значенням X, для якого викликаний цей процес.

45 - Інакше, (PicOrderCnt(pic) щонайменше одного зображення піс у щонайменше одному списку опорних зображень, більше, ніж PicOrderCntVal, то mvCol, refldxCol і listCol вважаються такими, що дорівнюють MvLN[xPCol][yPCol], RefldxLN[xPCol][yPCol] і LN, відповідно, де N є значенням collocated\_from\_10\_flag.

50 - Якщо одна з наступних умов істинна, то змінна availableFlagLXCol вважається такою, що дорівнює 0:

- RefPicListX[refldxLX] є неопорним зображенням, і UnusedRefPic(colPic, refldxCol, listCol) дорівнює 0;

55 - RefPicListX[refldxLX] є опорним зображенням, і UnusedRefPic(colPic, refldxCol, listCol) дорівнює 1;

- Інакше, змінна availableFlagLXCol вважається такою, що дорівнює 1, і застосовується наступне.
- Якщо RefPicListX[refldxLX] є довгостроковим опорним зображенням, або

60 LongTermRefPic(colPic, refldxCol, listCol) дорівнює 1, або PicOrderCnt(colPic) -

RefPicOrderCnt(colPic, reldxCol, listCol) дорівнює PicOrderCntVal-  
PicOrderCnt(RefPicListX[reldxLX]), то  
mvLXCol=mvCol (8-143)

- Інакше, mvLXCol витягається як масштабована версія вектора mvCol руху, як визначено

5 нижче

tx=(16384+(Abs(td)>>1))/td (8-144)

DistScaleFactor=Clip3(-4096, 4095, (tb\* tx+32)>>6) (8-145)

mvLXCol=Clip3(-8192, 8191.75, Sign(DistScaleFactor\* mvCol)\* ((Abs(DistScaleFactor\*  
mvCol)+127)>>8)) (8-146),

10

де td і tb можуть бути витягнуті так

td=Clip3(-128, 127, PicOrderCnt(colPic) - RefPicOrderCnt(colPic, reldxCol, listCol)) (8-147)

tb=Clip3(-128, 127, PicOrderCntVal-PicOrderCnt(RefPicListX[reldxLX])) (8-148).

Відносно приклада, який називається вище "п'ятим прикладом", відеокодер 20 і  
відеокодер 30 можуть бути виконані з можливістю виконувати будь-який або усі з наступних  
15 способів. У даному прикладі, прогнозування між векторами руху, що посиляються на різні  
довгострокові опорні зображення, може бути заборонене, прогнозування між векторами руху,  
що посиляються на різні міжвидові опорні зображення, може бути заборонене, і прогнозування  
між векторами руху, що посиляються на міжвидове опорне зображення і довгострокове опорне  
зображення, може бути заборонене.

20

У даному прикладі, функція AddPicId(pic) повертає індекс порядку виду або ID рівня виду,  
або рівень, якому належить зображення піс. Таким чином, для будь-якого зображення "pic", що  
належить базовому виду або рівню, AddPicId(pic) повертає 0. У базовій специфікації HEVC,  
може застосовуватися наступне: функція AddPicId(pic) може бути визначена наступним чином:  
AddPicId(pic) повертає 0 (або reserved\_one\_5bits мінус 1). У даному прикладі, коли AddPicId(pic)  
25 не дорівнює 0, то зображення піс не є часовим опорним зображенням (тобто, ні  
короткостроковим опорним зображенням, ні довгостроковим опорним зображенням).  
AddPicId(picX, reldx, LX) може повертати AddPicId(pic), причому піс є опорним зображенням з  
індексом reldx зі списку LX опорних зображень зображення пісX.

30

Відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю виконувати процес  
витягування кандидатів предиктора вектора руху. Введення в процес можуть містити в собі  
місцерозташування (xP, yP) яскравості, що указує верхній лівий відлік яскравості поточного  
елемента прогнозування відносно верхнього лівого відліку поточного зображення, змінні, що  
вказують ширину і висоту елемента прогнозування для яскравості, nPSW і nPSH, і опорний  
індекс reldxLX (де X дорівнює 0 або 1) розділення поточного елемента прогнозування.  
35 Виведення цього процесу можуть містити в собі (де N заміщений або на A, або на B): вектори  
mvLXN руху сусідніх елементів прогнозування і прапори availableFlagLXN доступності сусідніх  
елементів прогнозування.

Змінна isScaledFlagLX, де X дорівнює 0 або 1, може вважатися такою, що дорівнює 1 або 0.  
Відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю витягати вектор mvLXA  
40 руху і прапор availableFlagLXA доступності з використанням наступного порядку етапів, де  
підкреслений текст являє собою зміни відносно HEVC WD7:

1. Нехай набір із двох місцерозташувань відліку буде (x<sub>A<sub>k</sub></sub>, y<sub>A<sub>k</sub></sub>), де k=0, 1, що вказує  
місцерозташування відліку, де x<sub>A<sub>k</sub></sub>=xP-1, y<sub>A<sub>0</sub></sub>=yP+nPSH і y<sub>A<sub>1</sub></sub>=y<sub>A<sub>0</sub></sub>-MinPuSize. Набір  
місцерозташувань відліку (x<sub>A<sub>k</sub></sub>, y<sub>A<sub>k</sub></sub>) являє собою місцерозташування відліку безпосередньо з  
45 лівої сторони лівої межі розділення і її продовженої лінії.

2. Нехай прапор availableFlagLXA доступності буде спочатку вважатися таким, що дорівнює  
0, і обидві компоненти mvLXA вважатися такими, що дорівнюють 0.

3. Коли одна або декілька з наступних умов істинні, змінна isScaledFlagLX вважається такою,  
що дорівнює 1:

50

- елемент прогнозування, що покриває місцерозташування (x<sub>A<sub>0</sub></sub>, y<sub>A<sub>0</sub></sub>) яскравості доступний  
[Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних  
блоків кодування], і PredMode не MODE\_INTRA,

- елемент прогнозування, що покриває місцерозташування (x<sub>A<sub>1</sub></sub>, y<sub>A<sub>1</sub></sub>) яскравості доступний  
[Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних  
55 блоків кодування], і PredMode не MODE\_INTRA.

4. Для (x<sub>A<sub>k</sub></sub>, y<sub>A<sub>k</sub></sub>) від (x<sub>A<sub>0</sub></sub>, y<sub>A<sub>0</sub></sub>) до (x<sub>A<sub>1</sub></sub>, y<sub>A<sub>1</sub></sub>), де y<sub>A<sub>1</sub></sub>=y<sub>A<sub>0</sub></sub>-MinPuSize, якщо availableFlagLXA  
дорівнює 0, то застосовується наступне:

60

- Якщо елемент прогнозування, що покриває місцерозташування (x<sub>A<sub>k</sub></sub>, y<sub>A<sub>k</sub></sub>) яскравості  
доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для  
мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[x<sub>A<sub>k</sub></sub>][y<sub>A<sub>k</sub></sub>] дорівнює 1, і



опорний індекс  $\text{refldxLX}[x_A][y_A]$  дорівнює опорному індексу  $\text{refldxLX}$  поточного елемента прогнозування, то  $\text{availableFlagLXA}$  вважається таким, що дорівнює 1, вектор  $\text{mvLXA}$  руху вважається таким, що дорівнює вектору  $\text{mvLX}[x_A][y_A]$  руху,  $\text{refldxA}$  вважається таким, що дорівнює  $\text{refldxLX}[x_A][y_A]$ , а  $\text{ListA}$  вважається таким, що дорівнює  $\text{ListX}$ .

5 - Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(x_A, y_A)$  яскравості доступний [Ed. (BB): Переписати з використанням  $\text{MinCbAddrZS}[]$  і процесу доступності для мінімальних блоків кодування], і  $\text{PredMode}$  не  $\text{MODE\_INTRA}$ ,  $\text{predFlagLY}[x_A][y_A]$  (де  $Y \neq X$ ) дорівнює 1,  $\text{AddPicId}(\text{RefPicListX}[\text{refldxLX}])$  дорівнює  $\text{AddPicId}(\text{RefPicListY}[\text{refldxLY}[x_A][y_A]])$ , і  $\text{PicOrderCnt}(\text{RefPicListY}[\text{refldxLY}[x_A][y_A]])$  дорівнює  $\text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])$ , то  $\text{availableFlagLXA}$  вважається таким, що дорівнює 1, вектор  $\text{mvLXA}$  руху вважається таким, що дорівнює вектору  $\text{mvLY}[x_A][y_A]$  руху,  $\text{refldxA}$  вважається таким, що дорівнює  $\text{refldxLY}[x_A][y_A]$ , а  $\text{ListA}$  вважається таким, що дорівнює  $\text{ListY}$ , і  $\text{mvLXA}$  вважається таким, що дорівнює  $\text{mvLXA}$ .

15 - Коли  $\text{availableFlagLXA}$  дорівнює 1, то  $\text{availableFlagLXA}$  вважається таким, що дорівнює 0, якщо одна або декілька з наступних умов істинна:

- Одне і тільки одне з  $\text{RefPicListX}[\text{refldxLX}]$  і  $\text{ListA}[\text{refldxA}]$  є довгостроковим опорним зображенням;

- Обидва  $\text{RefPicListX}[\text{refldxLX}]$  і  $\text{ListA}[\text{refldxA}]$  є довгостроковими опорними зображеннями, і  $\text{PicOrderCnt}(\text{ListA}[\text{refldxA}])$  не дорівнює  $\text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])$ .

20 5. Коли  $\text{availableFlagLXA}$  дорівнює 0, то для  $(x_A, y_A)$  від  $(x_{A_0}, y_{A_0})$  до  $(x_{A_1}, y_{A_1})$ , где  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ , якщо  $\text{availableFlagLXA}$  дорівнює 0, застосовується наступне:

25 - Якщо елемент прогнозування, що покриває місцерозташування  $(x_A, y_A)$  яскравості доступний [Ed. (BB): Переписати з використанням  $\text{MinCbAddrZS}[]$  і процесу доступності для мінімальних блоків кодування], і  $\text{PredMode}$  не  $\text{MODE\_INTRA}$ ,  $\text{predFlagLX}[x_A][y_A]$  дорівнює 1, і  $\text{AddPicId}(\text{RefPicListX}[\text{refldxLX}])$  дорівнює  $\text{AddPicId}(\text{RefPicListLX}[\text{refldxLX}[x_A][y_A]])$ , то  $\text{availableFlagLXA}$  вважається таким, що дорівнює 1, вектор  $\text{mvLXA}$  руху вважається таким, що дорівнює вектору  $\text{mvLX}[x_A][y_A]$  руху,  $\text{refldxA}$  вважається таким, що дорівнює  $\text{refldxLX}[x_A][y_A]$ , а  $\text{ListA}$  вважається таким, що дорівнює  $\text{ListX}$ .

30 - Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(x_A, y_A)$  яскравості доступний [Ed. (BB): Переписати з використанням  $\text{MinCbAddrZS}[]$  і процесу доступності для мінімальних блоків кодування], і  $\text{PredMode}$  не  $\text{MODE\_INTRA}$ ,  $\text{predFlagLY}[x_A][y_A]$  (де  $Y \neq X$ ) дорівнює 1, і  $\text{AddPicId}(\text{RefPicListX}[\text{refldxLX}])$  дорівнює  $\text{AddPicId}(\text{RefPicListLY}[\text{refldxLY}[x_A][y_A]])$ , то  $\text{availableFlagLXA}$  вважається таким, що дорівнює 1, а вектор  $\text{mvLXA}$  руху вважається таким, що дорівнює вектору  $\text{mvLY}[x_A][y_A]$  руху,  $\text{refldxA}$  вважається таким, що дорівнює  $\text{refldxLY}[x_A][y_A]$ , а  $\text{ListA}$  вважається таким, що дорівнює  $\text{ListY}$ .

35 - Коли  $\text{availableFlagLXA}$  дорівнює 1, то  $\text{availableFlagLXA}$  вважається таким, що дорівнює 0, якщо одна або декілька з наступних умов істинна:

- Одне і тільки одне з  $\text{RefPicListX}[\text{refldxLX}]$  і  $\text{ListA}[\text{refldxA}]$  є довгостроковим опорним зображенням;

40 - Обидва  $\text{RefPicListX}[\text{refldxLX}]$  і  $\text{ListA}[\text{refldxA}]$  є довгостроковими опорними зображеннями, і  $\text{PicOrderCnt}(\text{ListA}[\text{refldxA}])$  не дорівнює  $\text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])$ .

- Коли  $\text{availableFlagLXA}$  дорівнює 1, і обидва  $\text{RefPicListA}[\text{refldxA}]$  і  $\text{RefPicListX}[\text{refldxLX}]$  обидва є короткостроковими опорними зображеннями, то  $\text{mvLXA}$  витягають, як вказано нижче.

$$tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6) \quad (8-127)$$

$$\text{mvLXA} = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * \text{mvLXA}) * ((\text{Abs}(\text{DistScaleFactor} * \text{mvLXA}) + 127) >> 8)) \quad (8-128),$$

де  $td$  і  $tb$  можуть бути витягнуті так

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refldxA}])) \quad (8-129)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])) \quad (8-130).$$

Відеокoder 20 і відеокoder 30 можуть бути виконані з можливістю витягати вектор  $\text{mvLXB}$  руху і прапор  $\text{availableFlagLXB}$  доступності з використанням наступного порядку етапів, де підкреслений текст являє собою зміни відносно HEVC WD7:

55 1. Нехай набір із трьох місцерозташувань відліку буде  $(x_{B_k}, y_{B_k})$ , де  $k=0, 1, 2$ , що вказує місцерозташування відліку, де  $x_{B_0} = x_P + n\text{PSW}$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ ,  $x_{B_2} = x_P - \text{MinPuSize}$  і  $y_{B_k} = y_P - 1$ . Набір місцерозташувань відліку  $(x_{B_k}, y_{B_k})$  являє собою місцерозташування відліку безпосередньо з верхньої сторони межі розділення зверху і її продовженої лінії. [Ed. (BB): Визначити  $\text{MinPuSize}$  у SPS, але витягування повинне залежати від використання прапора AMP]

2. Коли  $y_P - 1$  менше, ніж  $((y_C >> \text{Log2CtbSize}) << \text{Log2CtbSize})$ , то застосовується наступне.

$$60 \quad x_{B_0} = (x_{B_0} >> 3) << 3 + ((x_{B_0} >> 3) \& 1) * 7 \quad (8-131)$$

$$xB_1 = (xB_1 >> 3) << 3 + ((xB_1 >> 3) \& 1) * 7 \quad (8-132)$$

$$xB_2 = (xB_2 >> 3) << 3 + ((xB_2 >> 3) \& 1) * 7 \quad (8-133)$$

3. Нехай прапор availableFlagLXB доступності буде спочатку вважатися таким, що дорівнює 0, і обидві компоненти mvLXB вважатися такими, що дорівнюють 0.

5 4. Для  $(xB_k, yB_k)$  від  $(xB_0, yB_0)$  до  $(xB_2, yB_2)$ , де  $xB_0 = xP + nPSW$ ,  $xB_1 = xB_0 - \text{MinPuSize}$  і  $xB_2 = xP - \text{MinPuSize}$ , якщо availableFlagLXB дорівнює 0, то застосовується наступне:

- Якщо елемент прогнозування, що покриває місцерозташування  $(xB_k, yB_k)$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[ $xB_k$ ][ $yB_k$ ] дорівнює 1, і опорний індекс refIdxLX[ $xB_k$ ][ $yB_k$ ] дорівнює опорному індексу refIdxLX поточного елемента прогнозування, то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLX[ $xB_k$ ][ $yB_k$ ] руху, refIdxB вважається таким, що дорівнює refIdxLX[ $xB_k$ ][ $yB_k$ ], а ListB вважається таким, що дорівнює ListX.

15 - Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(xB_k, yB_k)$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY[ $xB_k$ ][ $yB_k$ ] (де  $Y \neq X$ ) дорівнює 1, AddPicId(RefPicListX[refIdxLX]) дорівнює AddPicId(RefPicListLY[refIdxLY[ $xB_k$ ][ $yB_k$ ]]), і PicOrderCnt(RefPicListY[refIdxLY[ $xB_k$ ][ $yB_k$ ]]) дорівнює PicOrderCnt(RefPicListX[refIdxLX]), то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLY[ $xB_k$ ][ $yB_k$ ] руху, refIdxB вважається таким, що дорівнює refIdxLY[ $xB_k$ ][ $yB_k$ ], а ListB вважається таким, що дорівнює ListY.

- Коли availableFlagLXA дорівнює 1, то availableFlagLXA вважається таким, що дорівнює 0, якщо одна або декілька з наступних умов істинна:

25 - Одне і тільки одне з RefPicListX[refIdxLX] і ListB[refIdxB] є довгостроковим опорним зображенням.

- AddPicId(RefPicListX[refIdxLX]) не дорівнює AddPicId(ListB[refIdxB]).

- Обидва RefPicListX[refIdxLX] і ListB[refIdxB] є довгостроковими опорними зображеннями, і PicOrderCnt(ListB[refIdxB]) не дорівнює PicOrderCnt(RefPicListX[refIdxLX]).

30 5. Коли isScaledFlagLX дорівнює 0 і availableFlagLXB дорівнює 1, то mvLXA вважається таким, що дорівнює mvLXB, refIdxA вважається таким, що дорівнює refIdxB, і availableFlagLXA вважається таким, що дорівнює 1.

6. Коли isScaledFlagLX дорівнює 0, то availableFlagLXB вважається таким, що дорівнює 0 і для  $(xB_k, yB_k)$  від  $(xB_0, yB_0)$  до  $(xB_2, yB_2)$ , де  $xB_0 = xP + nPSW$ ,  $xB_1 = xB_0 - \text{MinPuSize}$ , і  $xB_2 = xP - \text{MinPuSize}$ , якщо availableFlagLXB дорівнює 0, то застосовується наступне:

35 - Якщо елемент прогнозування, що покриває місцерозташування  $(xB_k, yB_k)$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[ $xB_k$ ][ $yB_k$ ] дорівнює 1, і AddPicId(RefPicListX[refIdxLX]) дорівнює AddPicId(RefPicListX[refIdxLX][ $xB_k$ ][ $yB_k$ ]]), то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLX[ $xB_k$ ][ $yB_k$ ] руху, refIdxB вважається таким, що дорівнює refIdxLX[ $xB_k$ ][ $yB_k$ ], а ListB вважається таким, що дорівнює ListX.

40 - Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(xB_k, yB_k)$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY[ $xB_k$ ][ $yB_k$ ] (де  $Y \neq X$ ) дорівнює 1, і AddPicId(RefPicListX[refIdxLX]) дорівнює AddPicId(RefPicListY[refIdxLY[ $xB_k$ ][ $yB_k$ ]]), то availableFlagLXB вважається таким, що дорівнює 1, а вектор mvLXB руху вважається таким, що дорівнює вектору mvLY[ $xB_k$ ][ $yB_k$ ] руху, refIdxB вважається таким, що дорівнює refIdxLY[ $xB_k$ ][ $yB_k$ ], а ListB вважається таким, що дорівнює ListY.

50 - Коли availableFlagLXA дорівнює 1, то availableFlagLXA вважається таким, що дорівнює 0, якщо одна або декілька з наступних умов істинна:

- Одне і тільки одне з RefPicListX[refIdxLX] і ListB[refIdxB] є довгостроковим опорним зображенням.

- Обидва RefPicListX[refIdxLX] і ListB[refIdxB] є довгостроковими опорними зображеннями, і PicOrderCnt(ListB[refIdxB]) не дорівнює PicOrderCnt(RefPicListX[refIdxLX]).

55 - Коли availableFlagLXB дорівнює 1, PicOrderCnt(RefPicListB[refIdxB]) не дорівнює PicOrderCnt(RefPicListX[refIdxLX]), і RefPicListB[refIdxLB] і RefPicListX[refIdxLX] обидва є короткостроковими опорними зображеннями, то mvLXB витягають, як вказано нижче.

$$tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-134)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6) \quad (8-135)$$

$mvLXB = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvLXA) * ((Abs(DistScaleFactor * mvLXA) + 127) >> 8))$  (8-136),

де  $td$  і  $tb$  можуть бути витягнуті так

$td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicList[refIdxB]))$  (8-137)

5  $tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]))$  (8-138).

Відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю витягати часові предиктори вектора руху. Введення в цей процес можуть містити в собі місцерозташування ( $xP$ ,  $yP$ ) яскравості, що указує верхній лівий відлік яскравості поточного елемента прогнозування відносно верхнього лівого відліку поточного зображення, змінні, що вказують ширину і висоту елемента прогнозування для яскравості,  $nPSW$  і  $nPSH$ , і опорний індекс розділення  $refIdxLX$  (де  $X$  дорівнює 0 або 1) поточного елемента прогнозування. Виведення цього процесу можуть містити в собі прогнозування  $mvLXCol$  вектора руху і прапор  $availableFlagLXCol$  доступності.

Функція  $RefPicOrderCnt(picX, refIdx, LX)$ , що може повертати рахунок  $PicOrderCntVal$  порядку зображення опорного зображення з індексом  $refIdx$  зі списку  $LX$  опорних зображень зображення  $picX$ , може бути визначена наступним чином:

$RefPicOrderCnt(picX, refIdx, LX) = PicOrderCnt(RefPicListX[refIdx]$  зображення  $PicX$ ) (8-141)

Залежно від значень  $slice\_type$ ,  $collocated\_from\_10\_flag$  і  $collocated\_ref\_idx$ , відеокодер 20 і відеокодер 30 можуть витягати змінну  $colPic$ , що визначає зображення, яке містить суміщене розділення, наступним чином:

20 - Якщо  $slice\_type$  дорівнює  $B$  і  $collocated\_from\_10\_flag$  дорівнює 0, то змінна  $colPic$  визначає зображення, що містить суміщене розділення, як визначено  $RefPicList1[collocated\_ref\_idx]$ .

- Інакше, ( $slice\_type$  дорівнює  $B$  і  $collocated\_from\_10\_flag$  дорівнює 1 або  $slice\_type$  дорівнює  $P$ ), змінна  $colPic$  визначає зображення, що містить суміщене розділення, як вказано  $RefPicList0[collocated\_ref\_idx]$ .

25 Відеокодер 20 і відеокодер 30 можуть витягати змінну  $colPu$  і її положення ( $xPCol$ ,  $yPCol$ ) з використанням наступного порядку етапів, де підкреслений текст являє собою зміни відносно HEVC WD7:

1. Змінна  $colPu$  може бути витягнута наступним чином

$yPRb = yP + nPSH$  (8-139)

30 - Якщо  $(yP >> \log_2 CtbSize)$  дорівнює  $(yPRb >> \log_2 CtbSize)$ , то горизонтальний компонент правого нижнього положення яскравості поточного елемента прогнозування визначається так

$xPRB = xP + nPSW$  (8-140)

і змінна  $colPu$  вважається такою, що дорівнює елементу прогнозування, що покриває модифіковане положення, задане  $((xPRb >> 4) << 4, (yPRb >> 4) << 4)$  у середині  $colPic$ .

35 - Інакше,  $((yP >> \log_2 CtbSize)$  не дорівнює  $(yPRb >> \log_2 CtbSize)$ ),  $colPu$  маркується як "недоступна".

2. Коли  $colPU$  кодується в режимі внутрішнього прогнозування, або  $colPu$  промаркована як "недоступна", то застосовується наступне.

- Центральне положення яскравості поточного елемента прогнозування визначається так

40  $xPCtr = xP + (nPSW >> 1)$  (8-141)

$yPCtr = yP + (nPSH >> 1)$  (8-142)

- Змінна  $colPu$  вважається такою, що дорівнює елементу прогнозування, що покриває модифіковане положення, задане  $((xPCtr >> 4) << 4, (yPCtr >> 4) << 4)$  у середині  $colPic$ .

45 3.  $(xPCol, yPCol)$  вважається таким, що дорівнює верхньому лівому відліку яскравості  $colPu$  відносно верхнього лівого відліку яскравості  $colPic$ .

Функція  $LongTermRefPic(picX, refIdx, LX)$  визначена наступним чином. Якщо опорне зображення з індексом  $refIdx$  зі списку  $LX$  опорних зображень зображення  $picX$  було промарковане як "використовуване для довгострокового посилання" у момент, коли  $picX$  було поточним зображенням, то  $LongTermRefPic(picX, refIdx, LX)$  повертає 1; інакше,  $LongTermRefPic(picX, refIdx, LX)$  повертає 0.

50 Функція  $AddPicId(picX, refIdx, LX)$  повертає  $AddPicId(pic)$ , причому  $pic$  є опорним зображенням з індексом  $refIdx$  зі списку  $LX$  опорних зображень зображення  $picX$ .

Змінні  $mvLXCol$  і  $availableFlagLXCol$  витягнуті наступним чином.

55 - Якщо одна або кілька наступних умов істинні, ті обидві компоненти  $mvLXCol$  вважаються такими, що дорівнюють 0, і  $availableFlagLXCol$  вважається таким, що дорівнює 0.

-  $colPU$  кодується в режимі внутрішнього прогнозування.

-  $colPu$  промаркована як "недоступна".

-  $pic\_temporal\_mvp\_enable\_flag$  дорівнює 0.

60 - Інакше, вектор  $mvCol$  руху, опорний індекс  $refIdxCol$  і ідентифікатор  $listCol$  опорного списку витягаються наступним чином.

- Якщо  $PredFlagL0[xPCol][yPCol]$  дорівнює 0, то  $mvCol$ ,  $refIdxCol$  і  $listCol$  вважаються такими, що дорівнюють  $MvL1[xPCol][yPCol]$ ,  $RefIdxL1[xPCol][yPCol]$  і  $L1$ , відповідно.

- Інакше, ( $PredFlagL0[xPCol][yPCol]$  дорівнює 1), застосовується наступне.

- Якщо  $PredFlagL1[xPCol][yPCol]$  дорівнює 0, то  $mvCol$ ,  $refIdxCol$  і  $listCol$  вважаються такими, що дорівнюють  $MvL0[xPCol][yPCol]$ ,  $RefIdxL0[xPCol][yPCol]$  і  $L0$ , відповідно.

- Інакше, ( $PredFlagL1[xPCol][yPCol]$  дорівнює 1), виконуються наступні призначення.

- Якщо  $PicOrderCnt(pic)$  кожного зображення  $pic$  у кожному списку опорних зображень менше або дорівнює  $PicOrderCntVal$ , то  $mvCol$ ,  $refIdxCol$  і  $listCol$  вважаються такими, що дорівнюють  $MvLX[xPCol][yPCol]$ ,  $RefIdxLX[xPCol][yPCol]$  і  $LX$ , відповідно, де  $X$  є значенням  $X$ , для якого викликаний цей процес.

- Інакше, ( $PicOrderCnt(pic)$  щонайменше одного зображення  $pic$  у щонайменше одному списку опорних зображень, більше, ніж  $PicOrderCntVal$ , то  $mvCol$ ,  $refIdxCol$  і  $listCol$  вважаються такими, що дорівнюють  $MvLN[xPCol][yPCol]$ ,  $RefIdxLN[xPCol][yPCol]$  і  $LN$ , відповідно, де  $N$  є значенням  $collocated\_from\_10\_flag$ .

- Якщо одна з наступних умов істинна, то змінна  $availableFlagLXCol$  вважається такою, що дорівнює 0.

-  $AddPicId(RefPicListX[refIdxLX])$  не дорівнює  $AddPicId(colPic, refIdxCol, listCol)$ ;

-  $RefPicListX[refIdxLX]$  є короткостроковим опорним зображенням, і  $LongTermRefPic(colPic, refIdxCol, listCol)$  дорівнює 1;

-  $RefPicListX[refIdxLX]$  є довгостроковим опорним зображенням, і  $LongTermRefPic(colPic, refIdxCol, listCol)$  дорівнює 0;

-  $RefPicListLX[refIdxLX]$  є довгостроковим опорним зображенням, і  $LongTermRefPic(colPic, refIdxCol, listCol)$  дорівнює 1, і  $RefPicOrderCnt(colPic, refIdxCol, listCol)$  не дорівнює  $PicOrderCnt(RefPicListX[refIdxLX])$ .

- Інакше, змінна  $availableFlagLXCol$  вважається такою, що дорівнює 1, і застосовується наступне.

- Якщо  $RefPicListX[refIdxLX]$  є довгостроковим опорним зображенням, або  $LongTermRefPic(colPic, refIdxCol, listCol)$  дорівнює 1, або  $PicOrderCnt(colPic, RefPicOrderCnt(colPic, refIdxCol, listCol))$  дорівнює  $PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX])$ , то

$mvLXCol = mvCol$  (8-143)

- Інакше,  $mvLXCol$  витягається як масштабована версія вектора  $mvCol$  руху, як визначено нижче

$tx = (16384 + (Abs(td) >> 1)) / td$  (8-144)

$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) >> 6)$  (8-145)

$mvLXCol = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvCol) * ((Abs(DistScaleFactor * mvCol) + 127) >> 8))$  (8-146),

де  $td$  і  $tb$  можуть бути витягнуті так

$td = Clip3(-128, 127, PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol))$  (8-147)

$tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]))$  (8-148)

Відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю виконувати наступний процес витягування комбінованих бі-прогнозуючих кандидатів злиття. Введення в цей процес можуть містити в собі список  $mergeCandList$  кандидатів злиття, опорні індекси  $refIdxL0N$  і  $refIdxL1N$  кожного кандидата  $N$ , що знаходиться в  $mergeCandList$ , прапори  $predFlagL0N$  і  $predFlagL1N$  використання списку прогнозування кожного кандидата  $N$ , що знаходиться в  $mergeCandList$ , вектори  $mvL0N$  і  $mvL1N$  руху кожного кандидата  $N$ , що знаходиться в  $mergeCandList$ , число елементів  $numMergeCand$  усередині  $mergeCandList$ , і число елементів  $numOrigMergeCand$  усередині  $mergeCandList$  після процесу витягування просторових і часових кандидатів злиття. Виведення цього процесу можуть містити в собі список  $mergeCandList$  кандидатів злиття, число елементів  $numMergeCand$  усередині  $mergeCandList$ , індекси  $refIdxL0combCandk$  і  $refIdxL1combCandk$  кожного нового кандидата  $combCandk$ , доданого в  $mergeCandList$  під час виклику цього процесу, прапори  $predFlagL0combCandk$  і  $predFlagL1combCandk$  використання списку прогнозування кожного нового кандидата  $combCandk$ , що додається в  $mergeCandList$  під час виклику цього процесу, і вектори  $mvL0combCandk$  і  $mvL1combCandk$  руху кожного нового кандидата  $combCandk$ , що додається в  $mergeCandList$  під час виклику цього процесу.

Коли  $numOrigMergeCand$  більше, ніж 1, і менше, ніж  $MaxNumMergeCand$ , то змінна  $numInputMergeCand$  може вважатися такою, що дорівнює  $numMergeCand$ , змінні  $combIdx$  і  $combCnt$  можуть вважатися такими, що дорівнюють 0, змінна  $combStop$  може вважатися такою, що дорівнює FALSE (НЕПРАВДА), і наступні етапи можуть повторюватися доти, поки  $combStop$

не дорівнює TRUE (ПРАВДА) (де багатокрапки являють собою ті ж етапи, що й у HEVC WD7, а підкреслений текст являє собою зміни відносно HEVC WD7):

1. Змінні 10CandIdx і 11CandIdx витягають з використанням combIdx, як вказано в Таблиці 8-8.

2. Наступні призначення виконуються, коли 10Cand є кандидатом у положенні 10CandIdx, а 11Cand є кандидатом у положенні 11CandIdx у списку mergeCandList кандидатів злиття (10Cand=mergeCandList[10CandIdx], 11Cand=mergeCandList[11CandIdx]).

3. Коли усі з наступних умов істинні,

- predFlagL010Cand == 1

- predFlagL111Cand == 1

- AddPicId(RefPicListL0[refIdxL010Cand])!= AddPicId(RefPicListL1[refIdxL111Cand]) ||  
PicOrderCnt(RefPicList0[refIdxL010Cand])!= PicOrderCnt(RefPicList [refIdxL111Cand]) ||  
mvL010Cand!= mvL111Cand,  
то застосовується наступне.

- ...

4. ...

5. ...

Як альтернатива, прогнозування між двома довгостроковими опорними зображеннями може бути дозволене без масштабування, і прогнозування між двома міжвидовими опорними зображеннями може бути дозволене без масштабування. Відеокодер 20 і відеодекодер 30 можуть бути виконані з можливістю виконувати процес витягування кандидатів предиктора вектора руху наступним чином, де підкреслений текст являє собою зміни відносно HEVC WD7, а багатокрапки являють собою той же текст, що й у HEVC WD7:

Змінна isScaledFlagLX, де X дорівнює 0 або 1, може вважатися такою, що дорівнює 0.

Вектор mvLXA руху і прапор availableFlagLXA доступності можуть бути витягнуті в наступному порядку етапів:

1. ...

2. ...

3. ...

4. ...

5. Коли availableFlagLXA дорівнює 0, то для (xA<sub>k</sub>, yA<sub>k</sub>) від xA<sub>0</sub>, yA<sub>0</sub>) до (xA<sub>1</sub>, yA<sub>1</sub>), де yA<sub>1</sub>=yA<sub>0</sub>-MinPuSize, наступне застосовується повторно, поки availableFlagLXA не дорівнює 1, у даному прикладі:

- Якщо елемент прогнозування, що покриває місцерозташування (xA<sub>k</sub>, yA<sub>k</sub>) яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[xA<sub>k</sub>][yA<sub>k</sub>] дорівнює 1, і RefPicType(RefPicListX[refIdxLX]) дорівнює RefPicType(RefPicListX[refIdxLX[xA<sub>k</sub>][yA<sub>k</sub>]]), то availableFlagLXA вважається таким, що дорівнює 1, вектор mvLXA руху вважається таким, що дорівнює вектору mvLX[xA<sub>k</sub>][yA<sub>k</sub>] руху, refIdxA вважається таким, що дорівнює refIdxLX[xA<sub>k</sub>][yA<sub>k</sub>], а ListA вважається таким, що дорівнює ListX.

- Інакше, якщо елемент прогнозування, що покриває місцерозташування (xA<sub>k</sub>, yA<sub>k</sub>) яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY[xA<sub>k</sub>][yA<sub>k</sub>] (де Y=I/X) дорівнює 1, і RefPicType(RefPicListX[refIdxLX]) дорівнює RefPicType(RefPicListY[refIdxLY[xA<sub>k</sub>][yA<sub>k</sub>]]), то availableFlagLXA вважається таким, що дорівнює 1, а вектор mvLXA руху вважається таким, що дорівнює вектору mvLY[xA<sub>k</sub>][yA<sub>k</sub>] руху, refIdxA вважається таким, що дорівнює refIdxLY[xA<sub>k</sub>][yA<sub>k</sub>], а ListA вважається таким, що дорівнює ListY.

- Коли availableFlagLXA дорівнює 1, і RefPicListA[refIdxLA] і RefPicListX[refIdxLX] обидва є короткостроковими опорними зображеннями, то mvLXA витягають, як вказано нижче.

$$Tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-126)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6) \quad (8-127)$$

$$mvLXA = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * ((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) >> 8)) \quad (8-128),$$

де td і tb можуть бути витягнуті так

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListA}[\text{refIdxA}])) \quad (8-129)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}])) \quad (8-130).$$

Відеокодер 20 і відеодекодер 30 можуть витягати вектор mvLXB руху і прапор availableFlagLXB доступності з використанням наступного порядку етапів, де підкреслений текст являє собою зміни відносно HEVC WD7, а багатокрапки являють собою той же текст, що в HEVC WD7:

1. ...  
2. ...  
3. ...  
4. ...  
5. ...

6... Коли `isScaledFlagLX` дорівнює 0, то `availableFlagLXB` вважається таким, що дорівнює 0 і для  $(x_{B_k}, y_{B_k})$  від  $(x_{B_0}, y_{B_0})$  до  $(x_{B_2}, y_{B_2})$ , де  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - \text{MinPuSize}$ , і  $x_{B_2} = x_P - \text{MinPuSize}$ , то наступне застосовується повторно, поки `availableFlagLXB` не дорівнює 1:

- Якщо елемент прогнозування, що покриває місцерозташування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням `MinCbAddrZS[]` і процесу доступності для мінімальних блоків кодування], і `PredMode` не `MODE_INTRA`, `predFlagLX[xBk][yBk]` дорівнює 1, і `RefPicType(RefPicListX[refldxLX])` дорівнює `RefPicType(RefPicListX[refldxLX[xBk][yBk]])`, то `availableFlagLXB` вважається таким, що дорівнює 1, вектор `mvLXB` руху вважається таким, що дорівнює вектору `mvLX[xBk][yBk]` руху, `refldxB` вважається таким, що дорівнює `refldxLX[xBk][yBk]`, а `ListB` вважається таким, що дорівнює `ListX`.
- Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням `MinCbAddrZS[]` і процесу доступності для мінімальних блоків кодування], і `PredMode` не `MODE_INTRA`, `predFlagLY[xBk][yBk]` (де  $Y=X$ ) дорівнює 1, і `RefPicType(RefPicListX[refldxLX])` дорівнює `RefPicType(RefPicListY[refldxLY[xBk][yBk]])`, то `availableFlagLXB` вважається таким, що дорівнює 1, а вектор `mvLXB` руху вважається таким, що дорівнює вектору `mvLY[xBk][yBk]` руху, `refldxB` вважається таким, що дорівнює `refldxLY[xBk][yBk]`, а `ListB` вважається таким, що дорівнює `ListY`.
- Коли `availableFlagLXB` дорівнює 1, `PicOrderCnt(RefPicListB[refldxB])` не дорівнює `PicOrderCnt(RefPicListX[refldxLX])`, і обидва `RefPicListB[refldxB]` і `RefPicListX[refldxLX]` обидва є короткостроковими опорними зображеннями, то `mvLXB` витягають, як вказано нижче.  

$$tx = (16384 + (\text{Abs}(td) >> 1)) / td \quad (8-134)$$

$$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) >> 6) \quad (8-135)$$

$$mvLXB = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * mvLXA) * ((\text{Abs}(\text{DistScaleFactor} * mvLXA) + 127) >> 8)) \quad (8-136),$$
де `td` і `tb` можуть бути витягнуті так  

$$td = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicList}[\text{refldxB}])) \quad (8-137)$$

$$tb = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refldxLX}])) \quad (8-138).$$
Відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю виконувати процес витягування часового прогнозування вектора руху яскравості наступним чином.

Змінні `mvLXCol` і `availableFlagLXCol` можуть бути витягнуті наступним чином.

- Якщо одна або кілька наступних умов істинні, то обидві компоненти `mvLXCol` вважаються такими, що дорівнюють 0, і `availableFlagLXCol` вважається таким, що дорівнює 0.
- `colPU` кодується в режимі внутрішнього прогнозування.
- `colPu` промаркована як "недоступна".
- `pic_temporal_mvp_enable_flag` дорівнює 0.
- Інакше, вектор `mvCol` руху, опорний індекс `refldxCol` і ідентифікатор `listCol` опорного списку витягаються наступним чином.
- Якщо `PredFlagL0[xPCol][yPCol]` дорівнює 0, то `mvCol`, `refldxCol` і `listCol` вважаються такими, що дорівнюють `MvL1[xPCol][yPCol]`, `RefldxL1[xPCol][yPCol]` і `L1`, відповідно.
- Інакше, (`PredFlagL0[xPCol][yPCol]` дорівнює 1), застосовується наступне.
- Якщо `PredFlagL1[xPCol][yPCol]` дорівнює 0, то `mvCol`, `refldxCol` і `listCol` вважаються такими, що дорівнюють `MvL0[xPCol][yPCol]`, `RefldxL0[xPCol][yPCol]` і `L0`, відповідно.
- Інакше, (`PredFlagL1[xPCol][yPCol]` дорівнює 1), виконуються наступні призначення.
- Якщо `PicOrderCnt(pic)` кожного зображення `pic` у кожному списку опорних зображень менше або дорівнює `PicOrderCntVal`, то `mvCol`, `refldxCol` і `listCol` вважаються такими, що дорівнюють `MvLX[xPCol][yPCol]`, `RefldxLX[xPCol][yPCol]` і `LX`, відповідно, де `X` є значенням `X`, для якого викликаний цей процес.
- Інакше, (`PicOrderCnt(pic)` щонайменше одного зображення `pic` у щонайменше одному списку опорних зображень, більше, ніж `PicOrderCntVal`, то `mvCol`, `refldxCol` і `listCol` вважаються такими, що дорівнюють `MvLN[xPCol][yPCol]`, `RefldxLN[xPCol][yPCol]` і `LN`, відповідно, де `N` є значенням `collocated_from_10_flag`.
- Якщо `RefPicType(RefPicListX[refldxLX])` не дорівнює `RefPicType(colPic, refldxCol, listCol)`, то змінна `availableFlagLXCol` вважається такою, що дорівнює 0.
- Інакше, змінна `availableFlagLXCol` вважається такою, що дорівнює 1, і застосовується наступне.

- Якщо  $\text{RefPicListX}[\text{refIdxLX}]$  є довгостроковим опорним зображенням, або  $\text{LongTermRefPic}(\text{colPic}, \text{refIdxCol}, \text{listCol})$  дорівнює 1, або  $\text{PicOrderCnt}(\text{colPic}) - \text{RefPicOrderCnt}(\text{colPic}, \text{refIdxCol}, \text{listCol})$  дорівнює  $\text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}])$ , то

5  $\text{mvLXCol} = \text{mvCol}$  (8-143)

- Інакше,  $\text{mvLXCol}$  витягається як масштабована версія вектора  $\text{mvCol}$  руху, як визначено нижче

$\text{tx} = (16384 + (\text{Abs}(\text{td}) >> 1)) / \text{td}$  (8-144)

$\text{DistScaleFactor} = \text{Clip3}(-4096, 4095, (\text{tb} * \text{tx} + 32) >> 6)$  (8-145)

10  $\text{mvLXCol} = \text{Clip3}(-8192, 8191.75, \text{Sign}(\text{DistScaleFactor} * \text{mvCol}) * ((\text{Abs}(\text{DistScaleFactor} * \text{mvCol}) + 127) >> 8))$  (8-146),

де  $\text{td}$  і  $\text{tb}$  можуть бути витягнуті так

$\text{td} = \text{Clip3}(-128, 127, \text{PicOrderCnt}(\text{colPic}) - \text{RefPicOrderCnt}(\text{colPic}, \text{refIdxCol}, \text{listCol}))$  (8-147)

$\text{tb} = \text{Clip3}(-128, 127, \text{PicOrderCntVal} - \text{PicOrderCnt}(\text{RefPicListX}[\text{refIdxLX}]))$  (8-148).

15 Розділ 8.5.2.1.3 HEVC WD7 може залишатися тим же, для цілей даного прикладу.

В альтернативному прикладі, прогнозування між векторами руху, що посиляються на різні довгострокові опорні зображення, може бути заборонене, прогнозування між векторами руху, що посиляються на міжвидове опорне зображення і довгострокове опорне зображення, може бути заборонене, і прогнозування між векторами руху, що посиляються на різні міжвидові опорні зображення, може бути дозволене. У даному прикладі, відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю виконувати процес витягування кандидатів предиктора вектора руху, як описано нижче. Введення в процес можуть містити в собі місцерозташування ( $x_P$ ,  $y_P$ ) яскравості, що указує верхній лівий відлік яскравості поточного елемента прогнозування відносно верхнього лівого відліку поточного зображення, змінні, що вказують ширину і висоту елемента прогнозування для яскравості,  $nPSW$  і  $nPSH$ , і опорний індекс  $\text{refIdxLX}$  (де  $X$  дорівнює 0 або 1) розділення поточного елемента прогнозування. Виведення процесу можуть містити в собі (де  $N$  заміщений або на  $A$ , або на  $B$ ) вектори  $\text{mvLXN}$  руху сусідніх елементів прогнозування і прапори  $\text{availableFlagLXN}$  доступності сусідніх елементів прогнозування.

Змінна  $\text{isScaledFlagLX}$ , де  $X$  дорівнює 0 або 1, може вважатися такою, що дорівнює 1 або 0. Відеокодер 20 і відеокодер 30 можуть витягати вектор  $\text{mvLXA}$  руху і прапор  $\text{availableFlagLXA}$  доступності з використанням наступного порядку, де підкреслений текст являє собою зміни відносно HEVC WD7:

1. Нехай набір із двох місцерозташувань відліку буде  $(x_{A_k}, y_{A_k})$ , де  $k=0, 1$ , що вказує місцерозташування відліку, де  $x_{A_k} = x_P - 1$ ,  $y_{A_0} = y_P + nPSH$  і  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ . Набір місцерозташувань відліку  $(x_{A_k}, y_{A_k})$  являє собою місцерозташування відліку безпосередньо з лівої сторони лівої межі розділення і її продовженої лінії.

2. Нехай прапор  $\text{availableFlagLXA}$  доступності буде спочатку вважатися таким, що дорівнює 0, і обидві компоненти  $\text{mvLXA}$  вважатися такими, що дорівнюють 0.

3. Коли одна або декілька з наступних умов істинні, змінна  $\text{isScaledFlagLX}$  вважається такою, що дорівнює 1.

- елемент прогнозування, що покриває місцерозташування  $(x_{A_0}, y_{A_0})$  яскравості доступний [Ed. (BB): Переписати з використанням  $\text{MinCbAddrZS}[]$  і процесу доступності для мінімальних блоків кодування], і  $\text{PredMode}$  не  $\text{MODE\_INTRA}$ .

45 - елемент прогнозування, що покриває місцерозташування  $(x_{A_1}, y_{A_1})$  яскравості доступний [Ed. (BB): Переписати з використанням  $\text{MinCbAddrZS}[]$  і процесу доступності для мінімальних блоків кодування], і  $\text{PredMode}$  не  $\text{MODE\_INTRA}$ .

4. Для  $(x_{A_k}, y_{A_k})$  від  $(x_{A_0}, y_{A_0})$  до  $(x_{A_1}, y_{A_1})$ , де  $y_{A_1} = y_{A_0} - \text{MinPuSize}$ , якщо  $\text{availableFlagLXA}$  дорівнює 0, то застосовується наступне:

50 - Якщо елемент прогнозування, що покриває місцерозташування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням  $\text{MinCbAddrZS}[]$  і процесу доступності для мінімальних блоків кодування], і  $\text{PredMode}$  не  $\text{MODE\_INTRA}$ ,  $\text{predFlagLX}[x_{A_k}][y_{A_k}]$  дорівнює 1, і опорний індекс  $\text{refIdxLX}[x_{A_k}][y_{A_k}]$  дорівнює опорному індексу  $\text{refIdxLX}$  поточного елемента прогнозування, то  $\text{availableFlagLXA}$  вважається таким, що дорівнює 1, вектор  $\text{mvLXA}$  руху вважається таким, що дорівнює вектору  $\text{mvLX}[x_{A_k}][y_{A_k}]$  руху,  $\text{refIdxA}$  вважається таким, що дорівнює  $\text{refIdxLX}[x_{A_k}][y_{A_k}]$ , а  $\text{ListA}$  вважається таким, що дорівнює  $\text{ListX}$ .

55 - Інакше, якщо елемент прогнозування, що покриває місцерозташування  $(x_{A_k}, y_{A_k})$  яскравості доступний [Ed. (BB): Переписати з використанням  $\text{MinCbAddrZS}[]$  і процесу доступності для мінімальних блоків кодування], і  $\text{PredMode}$  не  $\text{MODE\_INTRA}$ ,  $\text{predFlagLY}[x_{A_k}][y_{A_k}]$  (де  $Y=X$ ) дорівнює 1,  $\text{AddPicId}(\text{RefPicListX}[\text{refIdxLX}])$  дорівнює  $\text{AddPicId}(\text{RefPicListY}[\text{refIdxLY}[x_{A_k}][y_{A_k}]])$ , і  $\text{PicOrderCnt}(\text{RefPicListY}[\text{refIdxLY}[x_{A_k}][y_{A_k}]])$  дорівнює

PicOrderCnt(RefPicListX[refIdxLX]), то availableFlagLXA вважається таким, що дорівнює 1, вектор mvLXA руху вважається таким, що дорівнює вектору mvLY[xA<sub>k</sub>][yA<sub>k</sub>] руху, refIdxA вважається таким, що дорівнює refIdxLY[xA<sub>k</sub>][yA<sub>k</sub>], а ListA вважається таким, що дорівнює ListY, і mvLXA вважається таким, що дорівнює mvLXA.

5 - Коли availableFlagLXA дорівнює 1, то availableFlagLXA вважається таким, що дорівнює 0, якщо вірно наступне

- Одне і тільки одне з RefPicListX[refIdxLX] і ListA[refIdxA] є довгостроковим опорним зображенням.

10 5. Коли availableFlagLXA дорівнює 0, то для (xA<sub>k</sub>, yA<sub>k</sub>) від (xA<sub>0</sub>, yA<sub>0</sub>) до (xA<sub>1</sub>, yA<sub>1</sub>), де yA<sub>1</sub>=yA<sub>0</sub>-MinPuDSize, якщо availableFlagLXA дорівнює 0, застосовується наступне:

- Якщо елемент прогнозування, що покриває місцерозташування (xA<sub>k</sub>, yA<sub>k</sub>) яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[xA<sub>k</sub>][yA<sub>k</sub>] дорівнює 1, і AddPicId(RefPicListLX[refIdxLX]) дорівнює AddPicId(RefPicListLX[refIdxLX[xA<sub>k</sub>][yA<sub>k</sub>]]), то availableFlagLXA вважається таким, що дорівнює 1, вектор mvLXA руху вважається таким, що дорівнює вектору mvLX[xA<sub>k</sub>][yA<sub>k</sub>] руху, refIdxA вважається таким, що дорівнює refIdxLX[xA<sub>k</sub>][yA<sub>k</sub>], а ListA вважається таким, що дорівнює ListX.

20 - Інакше, якщо елемент прогнозування, що покриває місцерозташування (xA<sub>k</sub>, yA<sub>k</sub>) яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY[xA<sub>k</sub>][yA<sub>k</sub>] (де Y=I/X) дорівнює 1, і AddPicId(RefPicListLY[refIdxLY[xA<sub>k</sub>][yA<sub>k</sub>]]) дорівнює AddPicId(RefPicListLY[refIdxLY[xA<sub>k</sub>][yA<sub>k</sub>]]), то availableFlagLXA вважається таким, що дорівнює 1, а вектор mvLXA руху вважається таким, що дорівнює вектору mvLY[xA<sub>k</sub>][yA<sub>k</sub>] руху, refIdxA вважається таким, що дорівнює refIdxLY[xA<sub>k</sub>][yA<sub>k</sub>], а ListA вважається таким, що дорівнює ListY.

25 - Коли availableFlagLXA дорівнює 1, то availableFlagLXA вважається таким, що дорівнює 0, якщо вірно наступне:

- Одне і тільки одне з RefPicListX[refIdxLX] і ListA[refIdxA] є довгостроковим опорним зображенням.

30 - Коли availableFlagLXA дорівнює 1, і обидва RefPicListA[refIdxLA] і RefPicListX[refIdxLX] обидва є короткостроковими опорними зображеннями, то mvLXA витягають, як вказано нижче.

$tx = (16384 + (Abs(td) >> 1)) / td$  (8-126)

$DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) >> 6)$  (8-127)

$mvLXA = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvLXA) * ((Abs(DistScaleFactor * mvLXA) + 127) >> 8))$  (8-128),

35 де td і tb можуть бути витягнуті так

$td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListA[refIdxA]))$  (8-129)

$tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refIdxLX]))$  (8-130)

Відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю витягати вектор mvLXB руху і прапор availableFlagLXB доступності з використанням наступного порядку етапів, де підкреслений текст являє собою зміни відносно HEVC WD7:

40 1. Нехай набір із трьох місцерозташувань відліку буде (xB<sub>k</sub>, yB<sub>k</sub>), де k=0, 1, 2, що вказує місцерозташування відліку, де xB<sub>0</sub>=xP+nPSW, xB<sub>1</sub>=xB<sub>0</sub>-MinPuDSize, xB<sub>2</sub>=xP-MinPuDSize і yB<sub>k</sub>=yP-1. Набір місцерозташувань відліку (xB<sub>k</sub>, yB<sub>k</sub>) являє собою місцерозташування відліку безпосередньо з верхньої сторони межі розділення зверху і її продовженої лінії. [Ed. (BB): Визначити MinPuDSize у SPS, але витягування повинне залежати від використання прапора AMP]

2. Коли yP-1 менше, ніж ((yC>>Log2CtbSize)<<Log2CtbSize), то застосовується наступне.

$xB_0 = (xB_0 >> 3) << 3 + ((xB_0 >> 3) \& 1) * 7$  (8-131)

$xB_1 = (xB_1 >> 3) << 3 + ((xB_1 >> 3) \& 1) * 7$  (8-132)

$xB_2 = (xB_2 >> 3) << 3 + ((xB_2 >> 3) \& 1) * 7$  (8-133)

50 3. Нехай прапор availableFlagLXB доступності буде спочатку вважатися таким, що дорівнює 0, і обидві компоненти mvLXB вважатися такими, що дорівнюють 0.

4. Для (xB<sub>k</sub>, yB<sub>k</sub>) від (xB<sub>0</sub>, yB<sub>0</sub>) до (xB<sub>2</sub>, yB<sub>2</sub>), де xB<sub>0</sub>=xP+nPSW, x<sub>1</sub>=x<sub>0</sub>-MinPuDSize і x<sub>2</sub>=xP-MinPuDSize, якщо availableFlagLXB дорівнює 0, то застосовується наступне:

55 - Якщо елемент прогнозування, що покриває місцерозташування (xB<sub>k</sub>, yB<sub>k</sub>) яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX[xB<sub>k</sub>][yB<sub>k</sub>] дорівнює 1, і опорний індекс refIdxLX[xB<sub>k</sub>][yB<sub>k</sub>] дорівнює опорному індексу refIdxLX поточного елемента прогнозування, то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLX[xB<sub>k</sub>][yB<sub>k</sub>] руху, refIdxB вважається таким, що дорівнює refIdxLX[xB<sub>k</sub>][yB<sub>k</sub>], а ListB вважається таким, що дорівнює ListX.



- Інакше, якщо елемент прогнозування, що покриває місцезонаштування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY $[x_{B_k}][y_{B_k}]$  (де  $Y \neq X$ ) дорівнює 1, AddPicId(RefPicListX[refldxLX]) дорівнює AddPicId(RefPicListLY[refldx $[x_{B_k}][y_{B_k}]$ ]), і PicOrderCnt(RefPicListY[refldxLY $[x_{B_k}][y_{B_k}]$ ]) дорівнює PicOrderCnt(RefPicListX[refldxLX]), то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLY $[x_{B_k}][y_{B_k}]$  руху, refldxB вважається таким, що дорівнює refldxLY $[x_{B_k}][y_{B_k}]$ , а ListB вважається таким, що дорівнює ListY.

- Коли availableFlagLXB дорівнює 1, то availableFlagLXB вважається таким, що дорівнює 0, якщо вірно наступне:

- Одне і тільки одне з RefPicListX[refldxLX] і ListB[refldxB] є довгостроковим опорним зображенням.

5. Коли isScaledFlagLX дорівнює 0 і availableFlagLXB дорівнює 1, то mvLXA вважається таким, що дорівнює mvLXB, refldxA вважається таким, що дорівнює refldxB, і availableFlagLXA вважається таким, що дорівнює 1.

6. Коли isScaledFlagLX дорівнює 0, то availableFlagLXB вважається таким, що дорівнює 0 і для  $(x_{B_k}, y_{B_k})$  від  $(x_{B_0}, y_{B_0})$  до  $(x_{B_2}, y_{B_2})$ , де  $x_{B_0} = x_P + nPSW$ ,  $x_{B_1} = x_{B_0} - MinPuSize$ , і  $x_{B_2} = x_P - MinPuSize$ , якщо availableFlagLXB дорівнює 0, то застосовується наступне:

- Якщо елемент прогнозування, що покриває місцезонаштування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLX $[x_{B_k}][y_{B_k}]$  дорівнює 1, і AddPicId(RefPicListX[refldxLX]) дорівнює AddPicId(RefPicListX[refldxLX $[x_{B_k}][y_{B_k}]$ ]), то availableFlagLXB вважається таким, що дорівнює 1, вектор mvLXB руху вважається таким, що дорівнює вектору mvLX $[x_{B_k}][y_{B_k}]$  руху, refldxB вважається таким, що дорівнює refldxLX $[x_{B_k}][y_{B_k}]$ , а ListB вважається таким, що дорівнює ListX.

- Інакше, якщо елемент прогнозування, що покриває місцезонаштування  $(x_{B_k}, y_{B_k})$  яскравості доступний [Ed. (BB): Переписати з використанням MinCbAddrZS[] і процесу доступності для мінімальних блоків кодування], і PredMode не MODE\_INTRA, predFlagLY $[x_{B_k}][y_{B_k}]$  (де  $Y \neq X$ ) дорівнює 1, і AddPicId(RefPicListX[refldxLX]) дорівнює AddPicId(RefPicListLY[refldxLY $[x_{B_k}][y_{B_k}]$ ]), то availableFlagLXB вважається таким, що дорівнює 1, а вектор mvLXB руху вважається таким, що дорівнює вектору mvLY $[x_{B_k}][y_{B_k}]$  руху, refldxB вважається таким, що дорівнює refldxLY $[x_{B_k}][y_{B_k}]$ , а ListB вважається таким, що дорівнює ListY.

- Коли availableFlagLXA дорівнює 1, то availableFlagLXA вважається таким, що дорівнює 0, якщо вірно наступне:

- Одне і тільки одне з RefPicListX[refldxLX] і ListB[refldxB] є довгостроковим опорним зображенням.

- Коли availableFlagLXB дорівнює 1, PicOrderCnt(RefPicListB[refldxB]) не дорівнює PicOrderCnt(RefPicListX[refldxLX]), і обидва RefPicListB[refldxB] і RefPicListX[refldxLX] обидва є короткостроковими опорними зображеннями, то mvLXB витягають, як вказано нижче.

40  $tx = (16384 + (Abs(td) >> 1)) / td$  (8-134)  
 $DistScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) >> 6)$  (8-135)  
 $mvLXB = Clip3(-8192, 8191.75, Sign(DistScaleFactor * mvLXA) * ((Abs(DistScaleFactor * mvLXA) + 127) >> 8))$  (8-136),

де  $td$  і  $tb$  можуть бути витягнуті так

45  $td = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicList[refldxB]))$  (8-137)

$tb = Clip3(-128, 127, PicOrderCntVal - PicOrderCnt(RefPicListX[refldxLX]))$  (8-138).

Відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю здійснювати процес витягування часового прогнозування вектора руху яскравості, як описано нижче. Введення в процес можуть містити в собі місцезонаштування  $(x_P, y_P)$  яскравості, що указує верхній лівий відлік яскравості поточного елемента прогнозування відносно верхнього лівого відліку поточного зображення, змінні, що вказують ширину і висоту елемента прогнозування для яскравості, nPSW і nPSH, і опорний індекс refldxLX (де  $X$  дорівнює 0 або 1) розділення поточного елемента прогнозування. Виведення процесу можуть містити в собі вектор mvLXCol руху і прапор availableFlagLXCol доступності.

55 Функція RefPicOrderCnt(picX, refldx, LX), в одному прикладі, повертає рахунок PicOrderCntVal порядку зображення опорного зображення з індексом refldx зі списку LX опорних зображень зображення picX і може бути визначена наступним чином:

$RefPicOrderCnt(picX, refldx, LX) = PicOrderCnt(RefPicListX[refldx] \text{ зображення } PicX)$  (8-141)

Залежно від значень `slice_type`, `collocated_from_10_flag` і `collocated_ref_idx`, відеокодер 20 і відеодекодер 30 можуть витягати змінну `colPic`, що визначає зображення, що містить суміщене розділення, наступним чином:

5 - Якщо `slice_type` дорівнює B і `collocated_from_10_flag` дорівнює 0, то змінна `colPic` визначає зображення, що містить суміщене розділення, як визначено `RefPicList1[collocated_ref_idx]`.

- Інакше, (`slice_type` дорівнює B і `collocated_from_10_flag` дорівнює 1 або `slice_type` дорівнює P), змінна `colPic` визначає зображення, що містить суміщене розділення, як вказано `RefPicList0[collocated_ref_idx]`.

10 Відеокодер 20 і відеодекодер 30 можуть витягати змінну `colPu` і її положення (`xPCol`, `yPCol`) з використанням наступного порядку етапів:

1. Змінна `colPu` витягається наступним чином

$yPRb = yP + nPSH$  (8-139)

15 - Якщо  $(yP \gg \text{Log2CtbSize})$  дорівнює  $(yPRb \gg \text{Log2CtbSize})$ , то горизонтальний компонент правого нижнього положення яскравості поточного елемента прогнозування визначається так

$xPRb = xP + nPSW$  (8-140)

і змінна `colPu` вважається такою, що дорівнює елементу прогнозування, що покриває модифіковане положення, задане  $((xPRb \gg 4) \ll 4, (yPRb \gg 4) \ll 4)$  усередині `colPic`.

- Інакше,  $((yP \gg \text{Log2CtbSize})$  не дорівнює  $(yPRb \gg \text{Log2CtbSize})$ ), `colPu` маркується як "недоступна".

20 2. Коли `colPU` кодується в режимі внутрішнього прогнозування, або `colPu` промаркована як "недоступна", то застосовується наступне.

- Центральне положення яскравості поточного елемента прогнозування визначається так

$xPCtr = (xP + (nPSW \gg 1))$  (8-141)

$yPCtr = (yP + (nPSH \gg 1))$  (8-142)

25 - Змінна `colPu` вважається такою, що дорівнює елементу прогнозування, що покриває модифіковане положення, задане  $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$  усередині `colPic`.

3.  $(xPCol, yPCol)$  вважається таким, що дорівнює верхньому лівому відліку яскравості `colPu` відносно верхнього лівого відліку яскравості `colPic`.

30 Функція `LongTermRefPic(picX, refIdx, LX)` може бути визначена наступним чином: якщо опорне зображення з індексом `refIdx` зі списку LX опорних зображень зображення `picX` було промарковане як "використовуване для довгострокового посилання" у момент, коли `picX` було поточним зображенням, то `LongTermRefPic(picX, refIdx, LX)` повертає 1; інакше, `LongTermRefPic(picX, refIdx, LX)` повертає 0.

35 Відеокодер 20 і відеодекодер 30 можуть здійснювати модифіковану версію функції "AddPicId()" HEVC. Наприклад, відеокодер 20 і відеодекодер 30 можуть здійснювати `AddPicId(picX, refIdx, LX)`, так що ця функція повертає `AddPicId(pic)`, причому "pic" є опорним зображенням з індексом `refIdx` зі списку LX опорних зображень зображення `picX`.

Відеокодер 20 і відеодекодер 30 можуть витягати змінні `mvLXCol` і `availableFlagLXCol` наступним чином, де підкреслений текст являє собою зміни відносно HEVC WD7:

40 - Якщо одна або кілька наступних умов істинні, ті обидві компоненти `mvLXCol` вважаються такими, що дорівнюють 0, і `availableFlagLXCol` вважається таким, що дорівнює 0.

- `colPU` кодується в режимі внутрішнього прогнозування.

- `colPu` промаркована як "недоступна".

- `pic_temporal_mvp_enable_flag` дорівнює 0.

45 - Інакше, вектор `mvCol` руху, опорний індекс `refIdxCol` і ідентифікатор `listCol` опорного списку витягаються наступним чином.

- Якщо `PredFlagL0[xPCol][yPCol]` дорівнює 0, то `mvCol`, `refIdxCol` і `listCol` вважаються такими, що дорівнюють `MvL1[xPCol][yPCol]`, `RefIdxL1[xPCol][yPCol]` і L1, відповідно.

- Інакше,  $(\text{PredFlagL0}[xPCol][yPCol])$  дорівнює 1, застосовується наступне.

50 - Якщо `PredFlagL1[xPCol][yPCol]` дорівнює 0, то `mvCol`, `refIdxCol` і `listCol` вважаються такими, що дорівнюють `MvL0[xPCol][yPCol]`, `RefIdxL0[xPCol][yPCol]` і L0, відповідно.

- Інакше,  $(\text{PredFlagL1}[xPCol][yPCol])$  дорівнює 1, виконуються наступні призначення.

55 - Якщо `PicOrderCnt(pic)` кожного зображення `pic` у кожному списку опорних зображень менше або дорівнює `PicOrderCntVal`, то `mvCol`, `refIdxCol` і `listCol` вважаються такими, що дорівнюють `MvLX[xPCol][yPCol]`, `RefIdxLX[xPCol][yPCol]` і LX, відповідно, де X є значенням X, для якого викликаний цей процес.

60 - Інакше,  $(\text{PicOrderCnt}(\text{pic}))$  щонайменше одного зображення `pic` у щонайменше одному списку опорних зображень, більше, ніж `PicOrderCntVal`, то `mvCol`, `refIdxCol` і `listCol` вважаються такими, що дорівнюють `MvLN[xPCol][yPCol]`, `RefIdxLN[xPCol][yPCol]` і LN, відповідно, де N є значенням `collocated_from_10_flag`.

- Якщо одна з наступних умов істинна, то змінна availableFlagLXCol вважається такою, що дорівнює 0:

- AddPicId(RefPicListX[refIdxLX]) не дорівнює AddPicId(colPic, refIdxCol, listCol);
- RefPicListX[refIdxLX] є короткостроковим опорним зображенням, і LongTermRefPic(colPic, refIdxCol, listCol) дорівнює 1;
- RefPicListX[refIdxLX] є довгостроковим опорним зображенням, і LongTermRefPic(colPic, refIdxCol, listCol) дорівнює 0;
- Інакше, змінна availableFlagLXCol вважається такою, що дорівнює 1, і застосовується наступне.

10 - Якщо RefPicListX[refIdxLX] є довгостроковим опорним зображенням, або LongTermRefPic(colPic, refIdxCol, listCol) дорівнює 1, або PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, listCol) дорівнює PicOrderCntVal-PicOrderCnt(RefPicListX[refIdxLX]), то

mvLXCol=mvCol (8-143)

15 - Інакше, mvLXCol витягається як масштабована версія вектора mvCol руху, як визначено нижче

tx=(16384+(Abs(td)>>1))/td (8-144)

DistScaleFactor=Clip3(-4096, 4095, (tb\* tx+32)>>6) (8-145)

mvLXCol=Clip3(-8192, 8191.75, Sign(DistScaleFactor\* mvCol)\* ((Abs(DistScaleFactor\* mvCol)+127)>>8)) (8-146),

20 де td і tb можуть бути витягнуті так

td=Clip3(-128, 127, PicOrderCnt(colPic)- RefPicOrderCnt(colPic, refIdxCol, listCol)) (8-147)

tb=Clip3(-128, 127, PicOrderCntVal-PicOrderCnt(RefPicListX[refIdxLX])) (8-148)

Відеокодер 20 і відеокодер 30 можуть бути виконані з можливістю виконувати процес витягування комбінованих бі-прогнозуючих кандидатів злиття. Введення в цей процес можуть містити в собі список mergeCandList кандидатів злиття, опорні індекси refIdxL0N і refIdxL1N кожного кандидата N, що знаходиться в mergeCandList, прапори predFlagL0N і predFlagL1N використання списку прогнозування кожного кандидата N, що знаходиться в mergeCandList, вектори mvL0N і mvL1N руху кожного кандидата N, що знаходиться в mergeCandList, число елементів numMergeCand усередині mergeCandList, і число елементів numOrigMergeCand усередині mergeCandList після процесу витягування просторових і часових кандидатів злиття. Виведення цього процесу можуть містити в собі список mergeCandList кандидатів злиття, число елементів numMergeCand усередині mergeCandList, індекси refIdxL0combCandk і refIdxL1combCandk кожного нового кандидата combCandk, доданого в mergeCandList під час виклику цього процесу, прапори predFlagL0combCandk і predFlagL1combCandk використання списку прогнозування кожного нового кандидата combCandk, що додається в mergeCandList під час виклику цього процесу, і вектори mvL0combCandk і mvL1combCandk руху кожного нового кандидата combCandk, що додається в mergeCandList під час виклику цього процесу.

Цей процес може бути визначений наступним чином, де підкреслений текст являє собою зміни відносно HEVC WD7, а багатокрапки являють собою той же текст, що й у HEVC WD7.

40 Коли numOrigMergeCand більше, ніж 1, і менше, ніж MaxNumMergeCand, то змінна numInputMergeCand може вважатися такою, що дорівнює numMergeCand, змінні combIdx і combCnt можуть вважатися такими, що дорівнюють 0, змінна combStop може вважатися такою, що дорівнює FALSE, і наступні етапи можуть повторюватися доти, поки combStop не дорівнює TRUE:

45 1. Змінні 10CandIdx і 11CandIdx витягають з використанням combIdx, як вказано в Таблиці 8-8.

2. Наступні призначення виконуються, коли 10Cand є кандидатом у положенні 10CandIdx, а 11Cand є кандидатом у положенні 11CandIdx у списку mergeCandList кандидатів злиття

50 (10Cand=mergeCandList[10CandIdx], 11Cand=mergeCandList[11CandIdx]).

3. Коли усі з наступних умов істинні,

- predFlagL010Cand == 1
- predFlagL111Cand == 1
- AddPicId(RefPicListL0[refIdxL010Cand])!= AddPicId(RefPicListL1[refIdxL111Cand]) ||
- 55 PicOrderCnt(RefPicList0[refIdxL010Cand])!= PicOrderCnt(RefPicList1 [refIdxL111Cand]) ||
- mvL010Cand!= mvL111Cand,
- то застосовується наступне.
- ...
- 4. ...
- 60 5. ...

У деяких прикладах, прогнозування між двома векторами руху, що посиляються на два різних довгострокових опорних зображення, може бути заборонено. В інших прикладах, прогнозування між двома векторами руху, що посиляються на два різні міжвидові опорні зображення, може бути заборонено.

Подібним чином, відеокодер 20 і відеодекодер 30 являють собою приклади відеокодера, виконаного з можливістю кодувати значення рахунка порядку зображення (РОС) для першого зображення відеоданих, кодувати ідентифікатор зображення другої розмірності для першого зображення і кодувати друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення. Кодування другого зображення на основі значення РОС і ідентифікатора зображення другої розмірності першого зображення може містити в собі ідентифікацію першого зображення з використанням значення РОС першого зображення й ідентифікатора зображення другої розмірності.

Більше того, як показано вище, кодування другого зображення може містити в собі дозвіл або заборону прогнозування вектора руху для вектора руху, що посиляється на перше зображення, на основі значення РОС і ідентифікатора зображення другої розмірності першого зображення, і значення РОС і ідентифікатора зображення другої розмірності опорного зображення, на яке посиляється кандидат предиктора вектора руху. Наприклад, якщо ідентифікатор зображення другої розмірності першого зображення вказує, що перше зображення є короткостроковим зображенням, і ідентифікатор зображення другої розмірності опорного зображення вказує, що опорне зображення є довгостроковим зображенням, то відеокодер 20 і відеодекодер 30 можуть забороняти прогнозування вектора руху між вектором руху, що посиляється на перше зображення, і вектором руху, що посиляється на опорне зображення.

Додатково, як показано вище, кодування другого зображення може містити в собі кодування вектора руху блока другого зображення, що посиляється на перше зображення, як відзначено вище. Таке кодування може бути основане на значенні РОС першого зображення в тому, що якщо предиктор вектора руху посиляється на опорне зображення, що має інше значення РОС, то відеокодер 20 і відеодекодер 30 можуть бути виконані з можливістю масштабувати предиктор вектора руху на основі різниць значень РОС між першим і другим зображеннями, і опорним зображенням і другим зображенням.

Відеокодер 20 і відеодекодер 30 можуть бути виконані з можливістю виконувати способи будь-якого або всіх прикладів, описаних вище, окремо або в комбінації. Відеокодер 20 і відеодекодер 30 можуть бути здійснені як будь-яка з множини придатних схем кодера або декодера, залежно від ситуації, як, наприклад, один або кілька мікропроцесорів, цифрових сигнальних процесорів (DSP), спеціалізованих інтегральних схем (ASIC), програмованих користувачем вентильних матриць (FPGA), схеми дискретної логіки, програмному забезпеченні, технічних засобах, програмно-апаратних засобах або будь-якій їхній комбінації. Кожний з відеокодера 20 і відеодекодера 30 може бути включений в один або декілька кодерів або декодерів, будь-який з яких може бути інтегрований як частина комбінованого відеокодера/декодера (CODEC). Пристрій, що включає в себе відеокодер 20 і/або відеодекодер 30, може містити інтегральну схему, мікропроцесор і/або пристрій бездротового зв'язку, як, наприклад, стільниковий телефон або планшетний комп'ютер.

Фіг. 2 є блоковою діаграмою, яка зображує приклад відеокодера 20, що може здійснювати способи кодування відеоданих відповідно тільки до синтаксичного розширення високого рівня стандарту відеокодування. Відеокодер 20 може виконувати внутрішнє і зовнішнє кодування відеоблоків усередині відеослайсів. Внутрішнє кодування покладається на просторове прогнозування для скорочення або усунення просторової надмірності у відеосигналі усередині заданого відеокадру або зображення. Зовнішнє кодування покладається на часове прогнозування для скорочення або усунення часової надмірності у відеосигналі усередині суміжних кадрів або зображень відеопослідовності. Внутрішній режим (I режим) може стосуватися будь-якого або декількох режимів кодування на основі просторового прогнозування. Зовнішні режими, як, наприклад, однонаправлене прогнозування (P режим) або двонаправлене прогнозування (B режим), можуть стосуватися будь-якого з декількох режимів кодування на основі часового прогнозування.

Як зображено на Фіг. 2, відеокодер 20 приймає поточний відеоблок усередині відеокадру, який повинен бути кодований. У прикладі на Фіг. 2, відеокодер 20 містить у собі елемент 40 вибору режиму, пам'ять 64 опорних зображень, суматор 50, елемент 52 обробки перетворення, елемент 54 квантування й елемент 56 ентропійного кодування. Елемент 40 вибору режиму, у свою чергу, містить у собі елемент 44 компенсації руху, елемент 42 оцінки руху, елемент 46 внутрішнього прогнозування й елемент 48 розділення. Для відновлення відеоблока, відеокодер

20 також містить у собі елемент 58 зворотного квантування, елемент 60 зворотного перетворення і суматор 62. Деблокуючий фільтр (не зображений на Фіг. 2) може також бути включений для фільтрації меж блока для усунення артефактів блоковості з відновленого відео. Якщо бажано, то деблокуючий фільтр, як правило, фільтрує вивід суматора 62. Додаткові

5 фільтри (у контурі або пост-контурі) можуть також бути використані додатково до деблокуючого фільтра. Такі фільтри не зображені для стислості, але, якщо бажано, то можуть фільтрувати вивід суматора 50 (як внутрішньоконтурний фільтр).

Під час процесу кодування, відеокодер 20 приймає відеокадр або слайс, який повинен бути кодований. Кадр або слайс може бути розділений на множину відеоблоків. Елемент 42 оцінки

10 руху й елемент 44 компенсації руху виконують кодування з зовнішнім прогнозуванням прийнятого відеоблока відносно одного або декількох блоків в одному або декількох опорних кадрах для забезпечення часового прогнозування. Елемент 44 компенсації руху може кодувати вектор руху відповідно до способів даного розкриття, наприклад, під час удосконаленого прогнозування вектора руху (AMVP), часового прогнозування вектора руху (TMVP) або

15 кодування в режимі злиття. Елемент 46 внутрішнього прогнозування може альтернативно виконувати кодування з внутрішнім прогнозуванням прийнятого відеоблока відносно одного або декількох сусідніх блоків у тому ж кадрі або слайсі, що і блок, який повинен бути кодований, для забезпечення просторового прогнозування. Відеокодер 20 може виконувати багато проходів кодування, наприклад, для вибору придатного режиму кодування для кожного блока відеоданих.

20

Більше того, елемент 48 розділення може розділяти блоки відеоданих на підблоки, на основі оцінки попередніх схем розділення в попередніх проходах кодування. Наприклад, елемент 48 розділення може спочатку розділяти кадр або слайс на LCU і розділяти кожен LCU на під-CU на основі аналізу (наприклад, оптимізації швидкість передачі - спотворення) швидкість передачі -

25 спотворення. Елемент 40 вибору режиму може додатково виробляти структуру даних дерева квадратів, що вказує на розділення LCU на під-CU. Кінцевий вузол CU дерева квадратів може містити в собі один або декілька PU і один або декілька TU.

Елемент 40 вибору режиму може вибирати один або кілька режимів кодування, внутрішній або зовнішній, наприклад, на основі результатів помилки, і забезпечує блок, що вийшов, із

30 внутрішнім або зовнішнім кодуванням суматору 50 для формування даних залишкового блока і суматора 62 для відновлення закодованого блока для використання як опорного кадру. Елемент 40 вибору режиму також забезпечує синтаксичні елементи, як, наприклад, вектори руху, покажчики внутрішнього режиму, інформацію розділення й іншу синтаксичну інформацію, елементу 56 ентропійного кодування.

Елемент 42 оцінки руху й елемент 44 компенсації руху можуть бути високо інтегровані, але зображені окремо для концептуальних цілей. Оцінка руху, виконувана елементом 42 оцінки руху, є процесом формування векторів руху, що оцінюють рух для відеоблоків. Вектор руху, наприклад, може вказувати зміщення PU відеоблока усередині поточного відеокадру або зображення відносно прогнозуючого блока усередині опорного кадру (або іншого закодованого

40 елемента) відносно поточного блока, який кодують, усередині поточного кадру (або іншого закодованого елемента). Прогнозуючий блок є блоком, що близько збігається з блоком, який кодують, у частині різниці пікселів, що може бути визначена як сума (SAD) абсолютної різниці, сума (SSD) квадратної різниці, або інші метрики різниці. У деяких прикладах, відеокодер 20 може обчислювати значення подцілочисельних піксельних положень опорних зображень, що зберігаються в пам'яті 64 опорних зображень. Наприклад, відеокодер 20 може інтерполювати значення однієї четвертої піксельних положень, однієї восьмої піксельних положень або інших дробових піксельних положень опорного зображення. Отже, елемент 42 оцінки руху може виконувати пошук руху відносно цілочисельних піксельних положень і дробових піксельних положень і виводити вектор руху з дробовою піксельною точністю.

45

Елемент 42 оцінки руху обчислює вектор руху для PU відеоблока у зовнішньокодованому слайсі шляхом порівняння положення PU з положенням прогнозуючого блока опорного зображення. Опорне зображення може бути вибране з першого списку (Список 0) опорних зображень або із другого списку (Список 1) опорних зображень, кожний з яких ідентифікує одне або кілька опорних зображень, що зберігаються в пам'яті 64 опорних зображень. Елемент 42

50 оцінки руху відправляє обчислений вектор руху елементу 56 ентропійного кодування й елементу 44 компенсації руху.

Компенсація руху, виконувана елементом 44 компенсації руху, може залучати вибір або формування прогнозуючого блока на основі вектора руху, визначеного елементом 42 оцінки руху. Знову ж, елемент 42 оцінки руху й елемент 44 компенсації руху можуть функціонально

60 бути інтегровані, у деяких прикладах. Після прийому вектора руху для PU поточного відеоблока,

елемент 44 компенсації руху може розташовувати прогнозуючий блок, на який указує вектор руху, в одному зі списків опорних зображень. Суматор 50 формує залишковий відеоблок шляхом вирахування значень пікселів прогнозуючого блока зі значень пікселів поточного відеоблока, які кодують, формуючи значення піксельної різниці, як описано вище. Загалом, елемент 42 оцінки руху виконує оцінку руху відносно компонентів яскравості, а елемент 44 компенсації руху використовує вектори руху, обчислені на основі компонентів яскравості, для компонентів кольоровості і компоненту яскравості. Елемент 40 вибору режиму також може формувати синтаксичні елементи, пов'язані з відеоблоками і відеослайсом, для використання відеодекодером 30 при декодуванні відеоблоків відеослайса.

Елемент 46 внутрішнього прогнозування може виконувати внутрішнє прогнозування поточного блока, як альтернатива зовнішньому прогнозуванню, виконуваному елементом 42 оцінки руху й елементом 44 компенсації руху, як описано вище. Більш конкретно, елемент 46 внутрішнього прогнозування може визначати режим внутрішнього прогнозування для використання при кодуванні поточного блока. У деяких прикладах, елемент 46 внутрішнього прогнозування може кодувати поточний блок з використанням різних режимів внутрішнього прогнозування, наприклад, під час окремих проходів кодування, і елемент 46 внутрішнього прогнозування (або елемент 40 вибору режиму, у деяких прикладах) може вибирати придатний режим внутрішнього прогнозування для використання з протестованих режимів.

Наприклад, елемент 46 внутрішнього прогнозування може обчислювати значення швидкості передачі - спотворення з використанням аналізу швидкості передачі - спотворення для різних протестованих режимів внутрішнього прогнозування, і вибирати режим внутрішнього прогнозування, що має кращі характеристики швидкості передачі - спотворення серед протестованих режимів. Аналіз швидкості передачі - спотворення, загалом, визначає кількість спотворень (або помилку) між закодованим блоком і вихідним, незакодованим блоком, що був закодований для відтворення закодованого блока, а також швидкість проходження бітів (тобто, число бітів), використовувану для відтворення закодованого блока. Елемент 46 внутрішнього прогнозування може обчислювати співвідношення по спотвореннях і швидкостях передачі для різних закодованих блоків для визначення того, який з режимів внутрішнього прогнозування дає краще значення швидкості передачі - спотворення для блока.

Після вибору режиму внутрішнього прогнозування для блока, елемент 46 внутрішнього прогнозування може забезпечувати інформацію, що вказує вибраний режим внутрішнього прогнозування для блока, елементу 56 ентропійного кодування. Елемент 56 ентропійного кодування може кодувати інформацію, що вказує вибраний режим внутрішнього прогнозування. Відеокодер 20 може включати в передаваний бітовий потік дані конфігурації, що можуть містити в собі множину індексних таблиць режиму внутрішнього прогнозування і множину модифікованих індексних таблиць режиму внутрішнього прогнозування (які також називаються таблицями відображення кодових слів), визначення контекстів кодування для різних блоків і показники найбільш ймовірного режиму внутрішнього прогнозування, індексної таблиці режиму внутрішнього прогнозування і модифікованої індексної таблиці режиму внутрішнього прогнозування для використання для кожного з контекстів.

Відеокодер 20 формує залишковий відеоблок шляхом віднімання даних прогнозування від елемента 40 вибору режиму з вихідного відеоблока, який кодують. Суматор 50 являє собою компонент або компоненти, які виконують цю функцію вирахування. Елемент 52 обробки перетворення застосовує перетворення, як, наприклад, дискретне косинусне перетворення (DCT) або концептуально схоже перетворення, до залишкового блока, виробляючи відеоблок, що містить значення залишкових коефіцієнтів перетворення. Елемент 52 обробки перетворення може виконувати інші перетворення, які концептуально схожі на DCT. Вейвлет-перетворення, цілочисельне перетворення, перетворення піддіапазону або інші типи перетворень також можуть бути використані. У будь-якому випадку, елемент 52 обробки перетворення застосовує перетворення до залишкового блока, виробляючи блок залишкових коефіцієнтів перетворення. Перетворення може переводити залишкову інформацію з області піксельних значень в область перетворення, як, наприклад, частотна область. Елемент 52 обробки перетворення може відправляти коефіцієнти перетворення, що вийшли, елементу 54 квантування. Елемент 54 квантування квантує коефіцієнти перетворення для додаткового поліпшення швидкості передачі даних. Процес квантування може скоротити бітову глибину, пов'язану з деякими або всіма коефіцієнтами. Ступінь квантування може бути модифікований шляхом регулювання параметра квантування. У деяких прикладах, елемент 54 квантування потім може виконувати сканування матриці, що включає в себе квантовані коефіцієнти перетворення. Альтернативно, елемент 56 ентропійного кодування може виконувати сканування.

Слідом за квантуванням, елемент 56 ентропійного кодування ентропійно кодує квантовані коефіцієнти перетворення. Наприклад, елемент 56 ентропійного кодування може виконувати контекстно-залежне адаптивне кодування (CAVLC) зі змінною довжиною кодового слова, контекстно-залежне адаптивне двійкове арифметичне кодування (CABAC), синтаксичне контекстно-залежне адаптивне двійкове арифметичне кодування (SBAC), ентропійне кодування (PIPE) з розділенням інтервалів імовірності або інший спосіб ентропійного кодування. У випадку з контекстно-залежним ентропійним кодуванням, контекст може бути оснований на сусідніх блоках. Слідом за ентропійним кодуванням елементом 56 ентропійного кодування, закодований бітовий потік може бути переданий іншому пристрою (наприклад, відеодекодеру 30) або архівований для подальшої передачі або витягування.

Елемент 58 зворотного квантування й елемент 60 зворотного перетворення застосовують зворотне квантування і зворотне перетворення, відповідно, для відновлення залишкового блока в піксельній області, наприклад, для використання як опорного блока. Елемент 44 компенсації руху може обчислити опорний блок шляхом додавання залишкового блока до прогнозованого блока одного з кадрів пам'яті 64 опорних зображень. Елемент 44 компенсації руху може також застосовувати один або кілька фільтрів інтерполяції до відновленого залишкового блока для обчислення підцілочисельних піксельних значень для використання при оцінці руху. Суматор 62 додає відновлений залишковий блок до блока прогнозування з компенсованим рухом, який вироблений елементом 44 компенсації руху, для відтворення відновленого відеоблока для збереження в пам'яті 64 опорних зображень. Відновлений відеоблок може бути використаний елементом 42 оцінки руху й елементом 44 компенсації руху як опорний блок для кодування з зовнішнім прогнозуванням блока в наступному відеокадрі.

Відеокодер 20 може бути виконаний з можливістю виконувати будь-який або усі з різних прикладів способів, описаних відносно Фіг. 1, окремо або в комбінації. Наприклад, відповідно до способів даного розкриття, відеокодер 20 може кодувати зображення на основі значення рахунка порядку зображення (POC) опорного зображення й ідентифікатора другої розмірності опорного зображення. Тобто, відеокодер 20 може кодувати значення POC першого зображення (опорного зображення, у даному прикладі), а також ідентифікатор зображення другої розмірності для першого зображення. Ідентифікатор зображення другої розмірності може містити, наприклад, ідентифікатор виду для виду, що включає в себе перше зображення, індекс порядку виду для виду, що включає в себе перше зображення, комбінацію індексу порядку виду і прапора глибини, ідентифікатор рівня для рівня масштабованого відеокодування (SCV), що включає в себе перше зображення, і загальний ідентифікатор рівня.

Ідентифікатор другої розмірності може, додатково або альтернативно, містити значення, яке вказує, чи є перше зображення довгостроковим опорним зображенням або короткостроковим опорним зображенням. Альтернативно, окреме значення може вказувати, чи є перше зображення довгостроковим або короткостроковим опорним зображенням, додатково до значення POC і ідентифікатора зображення другої розмірності. У деяких прикладах, довгострокові і короткострокові вказівки для опорних зображень можуть вказувати, чи є опорні зображення часовими опорними зображеннями або міжвидовими опорними зображеннями. Наприклад, довгострокове опорне зображення може відповідати часовому опорному зображенню (тобто, опорному зображенню в тому ж рівні або виді), у той час як короткострокове опорне зображення може відповідати міжвидовому опорному зображенню. Як інший приклад, довгострокове опорне зображення може відповідати міжвидовому опорному зображенню, у той час як короткострокове опорне зображення може відповідати часовому опорному зображенню.

Подібним чином, відеокодер 20 може забороняти прогнозування вектора руху між векторами руху різних типів. "Типи" векторів руху можуть містити в собі, наприклад, часові вектори руху, що посилаються на часові опорні зображення (тобто, зображення в тому ж вигляді, що і поточне зображення, яке кодують), і вектори руху диспаратності, що посилаються на міжвидові опорні зображення (тобто, зображення у виді, відмінному від виду, що включає в себе поточне зображення). Як правило, міжвидові опорні зображення мають ті ж значення POC, що і поточне зображення. Тобто, як правило, міжвидові опорні зображення і поточне зображення знаходяться в тому самому елементі доступу. Відеокодер 20 може забороняти прогнозування векторів руху між векторами руху різних типів. Тобто, якщо поточний вектор руху поточного зображення є часовим вектором руху, то відеокодер 20 може забороняти прогнозування векторів руху відносно вектора руху диспаратності. Подібним чином, якщо поточний вектор руху є вектором руху диспаратності, то відеокодер 20 може забороняти прогнозування векторів руху відносно часового вектора руху. Відеокодер 20, інакше, може кодувати поточний вектор руху з використанням процесу кодування вектора руху, як, наприклад, просунуте прогнозування векторів руху (AMVP) або режим злиття.

У деяких прикладах, відеокодер 20 може бути виконаний з можливістю кодувати значення, яке вказує, чи є перше зображення (наприклад, компонента виду, у багатовидовому кодуванні відео) довгостроковим опорним зображенням, на основі, щонайменше частково, того, чи використовується перше зображення для міжвидового прогнозування. Наприклад, відеокодер 20 може кодувати синтаксичний елемент, який вказує, чи є перше зображення довгостроковим або короткостроковим опорним зображенням, у наборі параметрів послідовності (SPS), що відповідає послідовності, яка включає в себе перше зображення.

Додатково або альтернативно, відеокодер 20 може бути виконаний з можливістю маркувати міжвидові опорні зображення як довгострокові опорні зображення щонайменше тимчасово. Відеокодер 20 може додатково зберігати поточні статуси міжвидових опорних зображень, де статуси можуть містити одне з довгострокового опорного зображення, короткострокового опорного зображення і невикористовуваного для посилання. Таким чином, якщо перше зображення містить міжвидове зображення, то відеокодер 20 може маркувати перше зображення як довгострокове опорне зображення. Після кодування другого зображення відносно першого зображення, відеокодер 20 може відновлювати статус для міжвидового опорного зображення на основі збереженого статусу.

Додатково або альтернативно, відеокодер 20 може тимчасово призначати нові значення РОС міжвидовим опорним зображенням під час кодування другого зображення. Наприклад, відеокодер 20 може визначати набір значень РОС для поточних часових опорних зображень і призначати невикористані значення РОС міжвидовим опорним зображенням. Відеокодер 20 може також зберігати відповідні поточні значення РОС для кожного міжвидового опорного зображення. Після кодування другого зображення, відеокодер 20 може відновлювати збережені (тобто, вихідні) значення РОС для міжвидових опорних зображень. Тому що міжвидові опорні зображення, як правило, знаходяться в тому ж елементі доступу, що і друге зображення (тобто, зображення, що у даний момент кодується), у деяких прикладах, то відеокодер 20 може, навпаки, просто встановлювати значення РОС для міжвидових опорних зображень, які дорівнюють значенню РОС другого зображення, тобто, зображення, що у даний момент кодується, так що зберігати значення РОС необов'язково.

Подібним чином, відеокодер 20 на Фіг. 2 являє собою приклад відеокодера, виконаного з можливістю кодувати значення рахунку порядку зображення (РОС) для першого зображення відеоданих, кодувати ідентифікатор зображення другої розмірності для першого зображення, і кодувати, відповідно до базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення. Базова специфікація відеокодування може містити HEVC. Додатково, відеокодер 20 може бути виконаний з можливістю кодувати зображення відповідно до розширення базової специфікації відеокодування, наприклад, розширенням SVC або MVC HEVC. Таким чином, відеокодер 20 також являє собою приклад відеокодера, виконаного з можливістю кодувати значення рахунку порядку зображення (РОС) для першого зображення відеоданих, кодувати ідентифікатор зображення другої розмірності для першого зображення і кодувати, відповідно до розширення базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

Фіг. 3 є блоковою діаграмою, яка зображує приклад відеокодера 30, що може здійснювати способи кодування відеоданих відповідно тільки до синтаксичного розширення високого рівня стандарту відеокодування. У прикладі на Фіг. 3, відеокодер 30 містить у собі елемент 70 ентропійного декодування, елемент 72 компенсації руху, елемент 74 внутрішнього прогнозування, елемент 76 зворотного квантування, елемент 78 зворотного перетворення, пам'ять 82 опорних зображень і суматор 80. Відеокодер 30 може, у деяких прикладах, виконувати прохід декодування, загалом, протилежний проходу кодування, описаному відносно відеокодера 20 (Фіг. 2). Елемент 72 компенсації руху може формувати дані прогнозування на основі векторів руху, прийнятих від елемента 70 ентропійного декодування, у той час як елемент 74 внутрішнього прогнозування може формувати дані прогнозування на основі показників режиму внутрішнього прогнозування, прийнятих від елемента 70 ентропійного декодування.

Під час процесу декодування, відеокодер 30 приймає закодований потік двійкових відеоданих, що являє собою відеоблоки закодованого відеослайса і зв'язані синтаксичні елементи, від відеокодера 20. Елемент 70 ентропійного декодування відеокодера 30 ентропійно декодує бітовий потік для формування квантованих коефіцієнтів, векторів руху або показників режиму внутрішнього прогнозування й інших синтаксичних елементів. Елемент 70 ентропійного декодування направляє вектори руху й інші синтаксичні елементи елемента 72



компенсації руху. Відеодекодер 30 може приймати синтаксичні елементи на рівні відеослайса і/або рівні відеоблока.

Коли відеослайс закодований як внутрішньокодований (I) слайс, то елемент 74 внутрішнього прогнозування може формувати дані прогнозування для відеоблока поточного відеослайса на основі сигналізованого режиму внутрішнього прогнозування і даних від раніше декодованих блоків поточного кадру або зображення. Коли відеокадр закодований як зовнішньокодований (наприклад, B, P або GPB) слайс, то елемент 72 компенсації руху виробляє прогноуючі блоки для відеоблока поточного відеослайса на основі векторів руху й інших синтаксичних елементів, прийнятих від елемента 70 ентропійного декодування. Прогноуючі блоки можуть бути вироблені з одного з опорних зображень усередині одного зі списків опорних зображень. Відеодекодер 30 може створювати списки опорних кадрів, Список 0 і Список 1, використовуючи способи створення за умовчанням, на основі опорних зображень, що зберігаються в пам'яті 82 опорних зображень.

Відповідно до способів даного розкриття, елемент 70 ентропійного декодування може ентропійно декодувати закодовані дані, що представляють інформацію руху для поточного блока поточного зображення. Наприклад, відповідно до AMVP, елемент ентропійного декодування може декодувати значення різниці векторів руху (MVD) для поточного блока. Елемент 72 компенсації руху (або інший елемент відеокодера 30, як, наприклад, елемент 70 ентропійного кодування) може відновлювати вектор руху для поточного блока з використанням ентропійно декодованої інформації руху, як, наприклад, значення MVD. Наприклад, елемент 72 компенсації руху може визначати набір доступних предикторів вектора руху для поточного вектора руху, наприклад, на основі того, чи посилається поточний вектор руху на довгострокове опорне зображення або короткострокове опорне зображення (або часове або міжвидове опорне зображення), і того, чи посилається також група кандидатів опорних зображень на довгострокові або короткострокові опорні зображення (або часові або міжвидові опорні зображення).

Як описано вище, елемент 72 компенсації руху може визначати, що кандидати предикторів вектора руху різних типів недоступні для використання для прогнозування поточного вектора руху. Наприклад, коли поточний вектор руху є часовим вектором руху, то елемент 72 компенсації руху може визначати, що вектори руху диспаратності недоступні для використання як предикторів вектора руху для поточного вектора руху. Подібним чином, коли поточний вектор руху є вектором руху диспаратності, то елемент 72 компенсації руху може визначати, що часові вектори руху недоступні для використання як предикторів вектора руху для поточного вектора руху. У деяких прикладах, елемент 72 компенсації руху може також забороняти прогнозування векторів руху між довгостроковими і короткостроковими опорними зображеннями.

У випадку, коли поточний вектор руху є вектором руху диспаратності, то елемент 72 компенсації руху може також уникати масштабування предиктора вектора руху (що може, подібним чином, відповідати вектору руху диспаратності). Додатково або альтернативно, елемент 72 компенсації руху може призначати часове значення РОС міжвидовому опорному зображенню, на яке посилається предиктор вектора руху диспаратності, під час прогнозування вектора руху вектора руху диспаратності.

У будь-якому випадку, елемент 72 компенсації руху, або інший елемент відеокодера 30, може відтворювати вектор руху для поточного блока, наприклад, з використанням AMVP або режиму злиття. Елемент 72 компенсації руху визначає інформацію прогнозування для відеоблока поточного відеослайса шляхом синтаксичного аналізу векторів руху й інших синтаксичних елементів, і використовує інформацію прогнозування для відтворення прогноуючих блоків для поточного відеоблока, який кодують. Наприклад, елемент 72 компенсації руху використовує деякі з прийнятих синтаксичних елементів для визначення режиму прогнозування (наприклад, внутрішнє або зовнішнє прогнозування), використовуюваного для кодування відеоблоків відеослайса, типу слайса з зовнішнім прогнозуванням (наприклад, B слайс, P слайс або GPB слайс), інформації відновлення для одного або декількох списків опорних зображень для слайса, векторів руху для кожного зовнішньокодованого відеоблока слайса, статусу зовнішнього прогнозування для кожного зовнішньокодованого відеоблока слайса, і іншої інформації для декодування відеоблоків у поточному відеослайсі. Елемент 72 компенсації руху може кодувати вектор руху відповідно до способів даного розкриття, наприклад, під час удосконаленого прогнозування вектора руху (AMVP), часового прогнозування вектора руху (TMVP), або кодування в режимі злиття.

Елемент 72 компенсації руху може також виконувати інтерполяцію на основі фільтрів інтерполяції. Елемент 72 компенсації руху може використовувати фільтри так само, як і відеокодер 20 під час кодування відеоблоків, для обчислення інтерпольованих значень для подцілочисельних пікселів опорних блоків. У такому випадку, елемент 72 компенсації руху може

визначати фільтри інтерполяції, використані відеокодером 20, із прийнятих синтаксичних елементів, і використовувати фільтри інтерполяції для відтворення прогнозуючих блоків.

Елемент 76 зворотного квантування зворотно квантує, тобто, де-квантує, квантовані коефіцієнти перетворення, забезпечені в бітовому потоці і декодовані елементом 70 ентропійного декодування. Процес зворотного квантування може містити в собі використання параметра QPу квантування, обчисленого відеодекодером 30 для кожного відеоблока у відеослайсі, для визначення ступеня квантування і, подібним чином, ступеня зворотного квантування, який повинен бути застосований. Елемент 78 зворотного перетворення застосовує зворотне перетворення, наприклад, зворотне DCT, зворотне цілочисельне перетворення або концептуально схожий процес зворотного перетворення до коефіцієнтів перетворення, щоб виробити залишкові блоки в піксельній області.

Після того, як елемент 72 компенсації руху сформує прогнозуючий блок для поточного відеоблока на основі векторів руху й інших синтаксичних елементів, відеодекодер 30 формує декодований відеоблок шляхом підсумовування залишкових блоків від елемента 78 зворотного перетворення з відповідними прогнозуючими блоками, сформованими елементом 72 компенсації руху. Суматор 80 являє собою компонент або компоненти, які виконують цю функцію підсумовування. Якщо бажано, то деблокуючий фільтр також може бути застосований для фільтрації декодованих блоків, щоб усунути артефакти блоковості. Інші контурні фільтри (у контурі кодування або після контуру кодування) також можуть бути використані для згладжування піксельних переходів, або, інакше, поліпшення якості відео. Декодовані відеоблоки в заданому кадрі або зображенні потім зберігають у пам'яті 82 опорних зображень, що зберігає опорні зображення, використовувані в наступній компенсації руху. Пам'ять 82 опорних зображень також зберігає декодоване відео для подальшого представлення на пристрої відображення, як, наприклад, пристрій 32 відображення на Фіг. 1.

Подібним чином, відеодекодер 30 на Фіг. 3 являє собою приклад відеодекодера, виконаного з можливістю декодувати значення рахунка порядку зображення (POC) для першого зображення відеоданих і декодувати друге зображення на основі, щонайменше частково, значення POC і ідентифікатора зображення другої розмірності першого зображення. Базова специфікація відеокодування може містити HEVC. Додатково, відеодекодер 30 може бути виконаний з можливістю кодувати зображення відповідно до розширення базової специфікації відеокодування, наприклад, розширення SVC або MVC HEVC. Таким чином, відеодекодер 30 також являє собою приклад відеодекодера, виконаного з можливістю декодувати значення рахунка порядку зображення (POC) для першого зображення відеоданих, декодувати ідентифікатор зображення другої розмірності для першого зображення і декодувати, відповідно до розширення базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення POC і ідентифікатора зображення другої розмірності першого зображення.

Фіг. 4 являє собою концептуальну діаграму, яка зображує приклад моделі прогнозування MVC. Багатовидове відеокодування (MVC) є розширенням ITU-T H.264/AVC. Схожий спосіб може бути застосований до HEVC. У прикладі на Фіг. 4, зображені вісім видів (які мають ID виду від "S0" до "S7"), і для кожного виду зображені дванадцять часових місцерозташувань (від "T0" до "T11"). Тобто, кожен ряд на Фіг. 4 відповідає виду, а кожен стовпець указує часове місцерозташування.

Хоча MVC має так званий базовий вид, який декодується декодерами H.264/AVC, і стереовидова пара може бути підтримана також MVC, одна перевага MVC полягає в тому, що воно може підтримувати приклад, що використовує більше ніж два види, як введення 3D відео і декодує це 3D відео, представлене множиною видів. Засіб візуалізації клієнта, що має декодер MVC, може очікувати 3D відеоконтент із множиною видів.

Звичайна схема порядку декодування MVC називається кодування першого по часу. Елемент доступу може включати закодовані зображення усіх видів в один вивід за раз. Наприклад, кожне з зображень часу T0 може бути включене в спільний елемент доступу, кожне з зображень часу T1 може бути включене в другий, спільний елемент доступу, і так далі. Порядок декодування необов'язково є ідентичним виводу або порядку відображення.

Кадри, тобто, зображення, на Фіг. 4 вказані в перетинанні кожного ряду і кожного стовпця на Фіг. 4, з використанням затемненого блока, що включає в себе літеру, яка позначає, чи є відповідний кадр внутрішньокодованим (тобто, I-кадр), або зовнішньокодованим в одному напрямку (тобто, як P-кадр) або множині напрямків (тобто, як B-кадр). Загалом, прогнозування вказані стрілками, де кадр, на який вказано, використовує об'єкт, від якого вказано, як опора прогнозування. Наприклад, P-кадр виду S2 у часовому місцерозташуванні T0 прогнозується від I-кадру виду S0 у часовому місцерозташуванні T0.

При одновидовому кодуванні відео, кадри послідовності багатовидового відеокодування можуть бути закодовані з прогнозуванням відносно кадрів у різних часових місцєрозташуваннях. Наприклад, b-кадр виду S0 у часовому місцєрозташуванні T1 має стрілку, яка вказує на нього від I-кадру виду S0 у часовому місцєрозташуванні T1, що вказує, що b-кадр спрогнозований з I-кадру. Додатково, однак, у контексті багатовидового відеокодування, кадри можуть бути з міжвидовим прогнозуванням. Тобто, компонента виду може використовувати компоненти виду в інших видах для опори. У MVC, наприклад, міжвидове прогнозування реалізується так, начебто компонент виду в іншому виді є опорою зовнішнього прогнозування. Потенційні міжвидові опори сигналізуються в наборі параметрів послідовності (SPS) розширення MVC і можуть бути модифіковані за допомогою процесу створення списку опорних зображень, що дозволяє гнучкий порядок опор зовнішнього прогнозування або міжвидового прогнозування.

У розширенні MVC H.264/AVC, як приклад, міжвидове прогнозування підтримується компенсацією неспівпадаючого руху, що використовує синтаксис H.264/AVC компенсації руху, але дозволяє зображенню в іншому вигляді бути використаним як опорне зображення. Кодування двох видів може підтримуватися MVC, що, загалом, посиляється на стереоскопічні види. Одна з переваг MVC полягає в тому, що кодер MVC може брати більше ніж два види як введення 3D відеосигналу, а декодер MVC може декодувати такі багатовидові представлення. Таким чином, пристрій візуалізації з декодером MVC може очікувати 3D відеоконтент із більше ніж двома видами.

У MVC, міжвидове прогнозування (IVP) дозволене серед зображень у тому самому елементі доступу (тобто, з однаковим моментом часу). Елемент доступу, загалом, є елементом даних, що включає в себе усі компоненти виду (наприклад, всі елементи NAL) для спільного моменту часу. Таким чином, у MVC, міжвидове прогнозування дозволене серед зображень у тому самому елементі доступу. При кодуванні зображення в одному з небазових видів, зображення може бути додане до списку опорних зображень, якщо воно знаходиться в іншому вигляді, але усередині того ж моменту часу (наприклад, що має те ж саме значення POC, і, таким чином, що знаходиться в тому ж самому елементі доступу). Опорне зображення міжвидового прогнозування може знаходитися в будь-якому положенні списку опорних зображень, як і будь-яке опорне зображення зовнішнього прогнозування.

Як правило, створення списку опорних зображень для першого або другого списку опорних зображень зображення В містить у собі два етапи: ініціалізацію списку опорних зображень і переупорядкування (модифікацію) списку опорних зображень. Ініціалізація списку опорних зображень є явним механізмом, відповідно до якого відеокoder розташовує опорні зображення у пам'яті опорних зображень (також відомої як буфер декодованих зображень) у списку на основі порядку значень POC (рахунок порядку зображення, пов'язаний з порядком зображення).

Відеокoder може використовувати механізм переупорядкування списку опорних зображень для модифікації положення зображення, яке було додане до списку під час ініціалізації списку опорних зображень, у будь-яке нове положення, або для розташування будь-якого опорного зображення в пам'яті опорних зображень у будь-яке положення, навіть якщо зображення не належить ініціалізованому списку. Деякі зображення після переупорядкування (модифікації) списку опорних зображень можуть бути додані в додаткове положення в списку. Однак якщо положення зображення перевищує кількість активних опорних зображень у списку, то зображення не розглядається як запис фінального списку опорних зображень. Кількість активних опорних зображень може бути сигналізована в заголовку слайса для кожного списку. Після того, як створені списки опорних зображень (наприклад, RefPicList0 і RefPicList1, якщо доступно), опорний індекс списку опорних зображень може бути використаний для ідентифікації будь-якого опорного зображення, включеного до списку опорних зображень.

Для одержання часового предиктора вектора руху (TMVP), по-перше, необхідно ідентифікувати суміщене зображення. Якщо поточне зображення є В слайсом, то `collocated_from_10_flag` сигналізується в заголовку слайса для вказівки того, чи знаходиться суміщене зображення в RefPicList0 або RefPicList1. Після того, як список опорних зображень ідентифікований, `collocated_ref_idx`, сигналізований у заголовку слайса, використовується для ідентифікації зображення в зображенні в списку. Суміщений PU потім ідентифікується шляхом перевірки суміщеного зображення. Використовується або рух правого нижнього PU CU, що містить цей PU, або рух правого нижнього PU усередині центра PU CU, що містить цей PU. Коли вектори руху, ідентифіковані за допомогою вищевказаного процесу, використовуються для формування кандидата руху для AMVP або режиму злиття, їх необхідно масштабувати на основі часового місцєрозташування (відбитого за допомогою POC).

У HEVC, настроювання (SPS) параметра послідовності містить у собі прапор `sps_temporal_mvp_enable_flag`, і заголовок слайса містить у собі прапор

pic\_temporal\_mvp\_enable\_flag, коли sps\_temporal\_mvp\_enable\_flag дорівнює 1. Коли обидва pic\_temporal\_mvp\_enable\_flag і temporal\_id дорівнюють 0 для конкретного зображення, то ніякий вектор руху з зображень до цього конкретного зображення в порядку декодування не буде використаний як предиктор вектора руху при декодуванні конкретного зображення або зображення після конкретного зображення в порядку декодування.

В даний час, Експертна група (MPEG) по рухомому зображенню розробляє стандарт 3DV на основі HEVC, для якого частина зусиль по стандартизації також містить у собі стандартизацію багатовидового відеокодека на основі HEVC. Подібним чином, у HEVC на основі 3DV, дозволене міжвидове прогнозування на основі відновлених видових компонентів з різних видів.

AVC було розширене за допомогою багатовидового розширення таким чином, що розширення відповідає вимозі "тільки HLS" (тільки синтаксис високого рівня). Вимога "тільки HLS" гарантує, що виконуються тільки зміни синтаксису (HLS) високого рівня в багатовидовому відеокодуванні (MVC), так що ніякий модуль на рівні макроблока в AVC не потрібно переробляти, і він може бути повністю повторно використаний для MVC. Можливо, що вимога "тільки HLS" може бути виконана для розширення MVC/3DV HEVC, і також для розширення масштабованого відеокодування (SCV) HEVC, якщо багатоконтурне кодування вважається прийнятним.

Для дозволу міжвидового прогнозування, зміни HLS можуть бути зроблені з наступними цілями: ідентифікація зображення, де створення списку опорних зображень і маркування повинні бути здатними ідентифікувати зображення в конкретному виді.

Змін HLS недостатньо для відповідності вимозі "тільки HLS" у H.264/MVC, як і іншим обмеженням, і виконуються припущення, так що модулі кодування низького рівня ніколи не зустрінуться із ситуацією, наприклад, обробки масштабування з нульовим рухом. Такими обмеженнями, модифікаціями і припущеннями є:

Відміна часового прямого режиму, якщо суміщене зображення є міжвидовим (тільки) опорним зображенням.

Розгляд міжвидового (тільки) опорного зображення як не короткострокового: відносно просторового прямого.

Відміна схованого зваженого прогнозування.

Для відповідності вимозі "тільки HLS", такі модифікації в розширенні повинні бути тільки в синтаксисі високого рівня. Таким чином, не повинно бути модифікацій для синтаксичних елементів під заголовком слайса і ніяких змін процесу декодування рівня CU для специфікації розширення; наприклад, прогнозування вектора руху специфікації розширення HEVC повинне бути точно таким же, що й у базовій специфікації HEVC. Зміни HLS є нормативними змінами декодера специфікації розширення; однак, з погляду базової специфікації, такі зміни необов'язково повинні бути відомі і можуть бути інформативними.

Для дозволу функціональності, як, наприклад, ефективне міжвидове прогнозування, можуть бути здійснені модифікації в розширенні HEVC і базових специфікаціях. Зміни в базовій специфікації, що не впливають на звичайні процеси декодування або ефективність кодування базових декодерів HEVC, але націлені на дозвіл функціональності в специфікації розширення, називають прив'язками. У більшості випадків, вимога "тільки HLS" виконується з обома прив'язками в базовій специфікації і змінах HLS у специфікації розширення. Якщо прив'язки в базовій специфікації не визначені чітко, то конкретна бажана функціональність може бути не дозволена в специфікації розширення, або буде потрібно безліч модифікацій у специфікації розширення.

У тільки-HLS SVC, представлення базового рівня, можливо, після підвищуючої дискретизації і/або фільтрації, може бути занесене до списку опорних зображень поточного зображення поточного рівня. Таке зображення називають міжрівневе опорне зображення.

Різні модифікації можуть бути зроблені як у базовій специфікації, так і в специфікації розширення модифікації тільки-HLS HEVC. З урахуванням конкретної бажаної функціональності, на етапі, на якому структури базової специфікації і специфікації розширення можуть бути модифіковані, це є питанням компромісу між модифікацією базової специфікації і модифікацією специфікації розширення.

Фіг. 5-9 є концептуальними діаграмами, які зображують потенційні проблеми, які треба подолати для досягнення розширення тільки-HLS HEVC. Фіг. 5, наприклад, зображує приклад, у якому поточне зображення 100 містить у собі блоки, як, наприклад, блоки 102 і 104, прогнозовані з використанням різних способів прогнозування. Більш конкретно, поточне зображення 100 відповідає зображенню небазового виду, у той час як міжвидове опорне зображення 110 є зображенням базового виду. Блок 102 поточного зображення 100 є міжвидовим прогнозним відносно міжвидового опорного зображення 110 (з використанням

неспівпадаючого вектора 106 руху), у той час як блок 104 прогнозують з використанням зовнішнього прогнозування відносно короткострокового (ST) опорного зображення 112 того ж небазового виду (з використанням часового вектора 108 руху). Фіг. 5, отже, зображує приклад, у якому поточне зображення містить у собі сусідні блоки з часовим вектором руху (часовий вектор 108 руху) і міжвидовим вектором руху (який також називається вектор руху диспаратності, а саме, вектор 106 руху диспаратності).

Дане розкриття визнає, що, у деяких прикладах, вектор руху диспаратності не повинен бути масштабований для прогнозування часового вектора руху. Додатково, дане розкриття також визнає, що, у деяких прикладах, часовий вектор руху не повинен бути масштабований для прогнозування вектора руху диспаратності. Дане розкриття також визнає, що повинно бути можливим відмінити прогнозування вектора руху диспаратності з часового короткострокового вектора руху, наприклад, під час AMVP, і відмінити прогнозування часового вектора руху з вектора руху диспаратності. Вектори руху диспаратності, як правило, відповідають локальній розбіжності того самого об'єкта в різних видах. Однак часові вектори руху, як правило, відповідають руху об'єкта. У НТМ, що є опорним програмним забезпеченням 3DV, прогнозування між векторами руху двох вищевказаних категорій заборонено.

Фіг. 6 зображує приклад, у якому поточне зображення містить у собі блоки, спрогнозовані з використанням міжвидових опорних зображень різних видів. Більш конкретно, у даному прикладі, міжвидове опорне зображення 120 є видом 0, а міжвидове опорне зображення 122 є видом 1. Поточне зображення 124 є видом 2. Поточне зображення 124 містить у собі блоки 126, 128, спрогнозовані з використанням міжвидового прогнозування з обох міжвидового опорного зображення 120 виду 0 і міжвидового опорного зображення 122 виду 1. Більш конкретно, у даному прикладі, блок 126 спрогнозований з міжвидового опорного зображення 122, у той час як блок 128 спрогнозований з міжвидового опорного зображення 120.

Блоки 162 і 128 спрогнозовані з використанням різних векторів руху диспаратності. Тобто, блок 126 спрогнозований з використанням вектора 130 руху диспаратності, що посилається на частину міжвидового опорного зображення 122, у той час як блок 128 спрогнозований з використанням вектора 132 руху диспаратності, що посилається на частину міжвидового опорного зображення 120. Відповідно, Фіг. 6 являє собою приклад, у якому поточне зображення містить у собі сусідні блоки з міжвидовими векторами руху, що посилаються на міжвидові опорні зображення різних видів.

Дане розкриття визнає, що повинно бути можливим ідентифікувати, чи відповідають два вектори руху диспаратності тому самому опорному зображенню. Коли запис у RefPicList0 і запис у RefPicList1 обох є міжвидовими опорними зображеннями, повинно бути можливим, під час AMVP, ідентифікувати, чи є ці два опорні зображення однаковими. Коли RefPicList (де "X" може являти собою значення 0 або 1, наприклад) містить два записи, що є міжвидовими опорними зображеннями, повинно бути можливим, під час AMVP, ідентифікувати, чи є ці два опорні зображення однаковими. Додатково, два записи з однаковим значенням РОС можуть бути неідентичними, наприклад, коли два записи відповідають різним видам, як зображено на Фіг. 6.

Фіг. 7 зображує приклад, у якому опорне зображення в небазовому виді містить у собі блоки, спрогнозовані з використанням міжвидового прогнозування відносно міжвидового опорного зображення в базовому виді і з використанням зовнішнього прогнозування відносно довгострокового (LT) опорного зображення в небазовому виді. Тобто, Фіг. 7 зображує приклад, у якому поточне зображення 140 містить у собі сусідні блоки 146, 148 з часовим вектором 152 руху (що посилається на довгострокове опорне зображення 144) і міжвидовим вектором 150 руху (що посилається на міжвидове опорне зображення 142). Міжвидовий вектор 150 руху можна також називати "вектор 150 руху диспаратності". Дане розкриття визнає, що повинно бути можливим відмінити прогнозування вектора руху диспаратності, як, наприклад, вектор 150 руху диспаратності, з часового довгострокового вектора руху, як, наприклад, часовий вектор 152 руху, і відмінити прогнозування часового довгострокового вектора руху з вектора руху диспаратності.

Фіг. 8 зображує приклад, у якому поточне зображення в небазовому виді містить у собі блоки, що спрогнозовані з використанням зовнішнього прогнозування, як з довгострокового (LT) опорного зображення, так і короткострокового (ST) опорного зображення, небазового виду. Тобто, Фіг. 8 зображує приклад, у якому поточне зображення 160 містить у собі сусідні блоки 166, 168 з часовими довгостроковим і короткостроковим векторами руху. Більш конкретно, блок 166 спрогнозований з використанням часового вектора 170 руху, що посилається на довгострокове опорне зображення 162, у той час як блок 168 спрогнозований з використанням часового вектора 172 руху, що посилається на короткострокове опорне зображення 164. Отже, часовий вектор 170 руху можна називати довгостроковим вектором руху або довгостроковим

часовим вектором руху, у той час як часовий вектор 172 руху можна називати короткостроковим вектором руху або короткостроковим часовим вектором руху. Дане розкриття визнає, що повинно бути можливим відмінити прогнозування між часовими короткостроковими векторами руху і часових довгострокових векторів руху, наприклад, під час AMVP.

Фіг. 9 зображує приклад, у якому поточне зображення в небазовому виді містить у собі блоки, що спрогнозовані з використанням зовнішнього прогнозування, де блоки спрогнозовані відносно різних довгострокових (LT) опорних зображень небазового виду. Тобто, Фіг. 9 зображує приклад, у якому поточне зображення 180 містить у собі сусідні блоки 186, 188 з часовими векторами 190, 192 руху, що посилаються на довгострокові зображення 184, 182, відповідно. Більш конкретно, у даному прикладі, блок 186 спрогнозований з використанням часового вектора 190 руху, що посилається на частину довгострокового опорного зображення 184, у той час як блок 188 спрогнозований з використанням часового вектора 192 руху, що посилається на частину довгострокового опорного зображення 182. Дане розкриття визнає, що повинно бути можливим дозволяти і/або забороняти прогнозування часових довгострокових векторів руху під час AMVP.

Фіг. 10 є концептуальною діаграмою, яка зображує приклад групи сусідніх блоків для поточного блока. Більш конкретно, у даному прикладі, поточний блок має сусідні блоки ліворуч, позначені A0 і A1, і сусідні блоки B0, B1 і B2 зверху. Поточний блок може бути закодований з використанням зовнішнього прогнозування, наприклад, часового прогнозування або міжвидового прогнозування. Таким чином, відеокодер, як, наприклад, відеокодер 20 або відеодекодер 30, може кодувати поточний блок з використанням вектора руху. Більше того, відеокодер може кодувати вектор руху. У різних прикладах, відеокодер може кодувати вектор руху для поточного блока з використанням способів, описаних вище, наприклад, для удосконаленого прогнозування вектора руху (AMVP), часового прогнозування (TMVP) вектора руху, або режиму злиття. Предиктор TMVP може відповідати вектору руху для блока, який суміщений з поточним блоком у раніше закодованому зображенні.

Вектор руху одного або декількох сусідніх блоків A0, A1, B0, B1 і B2 можуть мати типи, відмінні від вектора руху, використовуваного для кодування поточного блока. Наприклад, поточний блок може бути закодований з використанням довгострокового вектора руху, у той час як один або кілька блоків A0, A1, B0, B1 і B2 можуть бути закодовані з використанням короткострокового вектора руху. Як інший приклад, поточний блок може бути закодований з використанням короткострокового вектора руху, у той час як один або кілька блоків A0, A1, B0, B1 і B2 можуть бути закодовані з використанням довгострокового вектора руху. Як інший приклад, поточний блок може бути закодований з використанням вектора руху диспаратності, у той час як один або кілька блоків A0, A1, B0, B1 і B2 можуть бути закодовані з використанням часового вектора руху. Як ще один приклад, поточний блок може бути закодований з використанням часового вектора руху, у той час як один або кілька блоків A0, A1, B0, B1 і B2 можуть бути закодовані з використанням вектора руху диспаратності. У таких випадках, як пояснено вище, відеокодер, як, наприклад, відеокодер 20 або відеодекодер 30, може забороняти прогнозування вектора руху між векторами руху різних типів.

Приклад на Фіг. 10 зображує кандидатів просторового предиктора вектора руху. Однак необхідно розуміти, що кандидати часового предиктора вектора руху можуть також бути розглянуті для часового прогнозування вектора руху (TMVP). Такі кандидати TMVP можуть відповідати інформації руху для суміщених блоків у раніше закодованих зображеннях, тобто, блоків, які суміщені з блоком, позначеним "поточний блок" на Фіг. 10. Додатково, відповідно до способів даного розкриття, кандидат TMVP може бути розглянутий як недоступний для використання як предиктора вектора руху, коли інформація руху кандидата TMVP і вектор руху поточного блока вказують на зображення різних типів, наприклад, короткострокове і довгострокове опорні зображення.

Фіг. 11 є блок-схемою, яка зображує приклад способу кодування відеоданих відповідно до способів даного розкриття. Етапи в прикладі способу на Фіг. 11 можуть, альтернативно, бути виконані в іншому порядку, або істотно паралельно, у деяких прикладах. Подібним чином, конкретні етапи можуть бути опущені, і/або інші етапи можуть бути додані. Хоча в описі це виконує відеокодер 20, необхідно розуміти, що інші пристрої відеокодування можуть бути виконані з можливістю виконувати істотно схожий спосіб.

У даному прикладі, відеокодер 20 кодує значення рахунка порядку зображення (POC) опорних зображень для поточного зображення (200). Наприклад, відеокодер 20 може кодувати значення POC або дані, що являють собою значення POC (як, наприклад, молодші біти (LSB)), для конкретних опорних зображень у структурі даних набору параметрів послідовності (SPS) для послідовності, що включає в себе поточне зображення. Відеокодер 20 може також,

додатково, або альтернативно, кодувати значення POC для одного або декількох опорних зображень у заголовку слайса поточного зображення. У деяких прикладах, відеокодер 20 може кодувати дані, що являють собою значення POC довгострокових опорних зображень, у SPS, і значення POC короткострокових опорних зображень, у заголовку слайса. Відеокодер 20 може також кодувати значення POC міжвидових опорних зображень, наприклад, у SPS, заголовку слайса або в іншому місці. Загалом, значення POC міжвидових опорних зображень є тими ж, що і значення POC поточного зображення, яке кодують.

Відеокодер 20 також може кодувати ідентифікатори другої розмірності опорних зображень (202). Ідентифікатори другої розмірності можуть містити в собі один або кілька ідентифікаторів виду для видів, що включають у себе опорні зображення, індекси порядку виду для видів, що включають у себе опорні зображення, комбінацію індексів порядку виду і прапорів глибини, ідентифікатори рівня для рівнів масштабованого відеокодування (SCV), що включають у себе опорні зображення, і/або загальні ідентифікатори рівня. Таким чином, комбінація значення POC для опорного зображення й ідентифікатор другої розмірності для опорного зображення можуть бути використані для ідентифікації опорного зображення.

Відеокодер 20 може додатково виконувати пошук руху для поточного блока поточного зображення. Тобто, елемент 42 оцінки руху може шукати опорні зображення для опорного блока, що найбільш близько збігається з поточним блоком. Це може привести до інформації руху, що включає в себе вектор руху, що посиляється на опорний блок, а також опорне зображення, у якому знаходиться опорний блок. Таким чином, елемент 44 компенсації руху відеокодера 20 може прогнозувати поточний блок з використанням вектора руху, що вказує на одне з опорних зображень (204).

Відеокодер 20 також може кодувати вектор руху, наприклад, з використанням удосконаленого прогнозування вектора руху (AMVP), часового прогнозування вектора руху (TMVP), або режиму злиття. Більш конкретно, відеокодер 20 може визначати набір доступних кандидатів предиктора вектора руху (206). Наприклад, з посиланням на Фіг. 10, відеокодер 20 може визначати, чи доступні вектори руху для сусідніх блоків A0, A1, B0, B1 і B2. Більш конкретно, відповідно до способів даного розкриття, відеокодер 20 може визначати, що вектор руху одного із сусідніх блоків недоступний, коли вектор руху сусіднього блока має тип, відмінний від вектора руху поточного блока. Подібним чином, відеокодер 20 може визначати, чи посиляється вектор руху для кандидата часового предиктора вектора руху на інший тип опорного зображення, ніж вектор руху для поточного блоку у визначенні того, чи доступний кандидат TMVP для використання як предиктора кодування вектора руху поточного блока.

Як пояснено вище, приклади різних типів векторів руху містять у собі довгострокові вектори руху, короткострокові вектори руху, часові вектори руху, і вектори руху диспаратності. Таким чином, відеокодер 20 може визначати тип вектора руху поточного блока, а також типи векторів руху сусідніх блоків, і визначати, що вектори руху сусідніх блоків типів, відмінних від типу поточного вектора руху поточного блока, недоступні для використання як предикторів вектора руху для поточного вектора руху. Для визначення типів, відеокодер 20 може посилатися на значення POC опорних зображень, на які посиляються кандидати векторів руху, значення POC опорного зображення, на яке посиляється поточний вектор руху, ідентифікатори другої розмірності опорних зображень, на які посиляються кандидати векторів руху, і/або ідентифікатор другої розмірності опорного зображення, на який посиляється поточний вектор руху.

Далі, відеокодер 20 може вибирати один з доступних кандидатів предиктора вектора руху із сусіднього блока (який може містити в собі так званий суміщений блок у раніше закодованому зображенні і/або відповідний блок у зображенні іншого виду) як предиктора вектора руху для поточного вектора руху (208). Відеокодер 20 може потім кодувати поточний вектор руху з використанням вибраного предиктора вектора руху (210).

Додатково, відеокодер 20 може обчислювати залишковий блок для поточного блока (212). Як пояснено вище відносно Фіг. 2, суматор 50 може обчислювати піксельні різниці між вихідним, незакодованим блоком і прогнозованим блоком, сформованим елементом 44 компенсації руху. Елемент 52 обробки перетворення, елемент 54 квантування й елемент 56 ентропійного кодування можуть потім, відповідно, перетворювати, квантувати і сканувати залишковий блок (214). Більш конкретно, елемент 52 обробки перетворення може перетворювати залишковий блок для відтворення блока коефіцієнтів перетворення, елемент 54 квантування можуть квантувати коефіцієнти перетворення й елемент 56 ентропійного кодування може сканувати квантовані коефіцієнти перетворення. Елемент 56 ентропійного кодування може потім ентропійно кодувати квантовані коефіцієнти перетворення і закодовану інформацію вектора руху (216).

Подібним чином, спосіб на Фіг. 11 являє собою приклад способу, що включає в себе кодування значення рахунка порядку зображення (POC) для першого зображення відеоданих, кодування ідентифікатора зображення другої розмірності для першого зображення і кодування, відповідно до базової специфікації відеокодування (або розширенням базової специфікації відеокодування), другого зображення на основі, щонайменше частково, значення POC і ідентифікатора зображення другої розмірності першого зображення. Додатково, спосіб може включати в себе заборону прогнозування вектора руху між першим вектором руху першого блока другого зображення, причому перший вектор руху посиляється на короткострокове опорне зображення, і другим вектором руху другого блока другого зображення, причому другий вектор руху посиляється на довгострокове опорне зображення. Додатково або альтернативно, спосіб може включати в себе заборону прогнозування вектора руху між першим вектором руху першого блока другого зображення, причому перший вектор руху посиляється на міжвидове опорне зображення, і другим вектором руху другого блока другого зображення, причому другий вектор руху посиляється на часове опорне зображення.

Фіг. 12 є блок-схемою, яка зображує приклад способу декодування відеоданих відповідно до способів даного розкриття. Етапи в прикладі способу на Фіг. 12 можуть, альтернативно, бути виконані в іншому порядку, або істотно паралельно, у деяких прикладах. Подібним чином, конкретні етапи можуть бути опущені, і/або інші етапи можуть бути додані. Хоча в описі це виконує відеодекодер 30, необхідно розуміти, що інші пристрої декодування відео можуть бути виконані з можливістю виконувати істотно схожий спосіб.

У даному прикладі, відеодекодер 30 декодує значення рахунка порядку зображення (POC) опорних зображень для поточного зображення (230). Наприклад, відеодекодер 30 може декодувати значення POC або дані, що являють собою значення POC (як, наприклад, молодші біти (LSB)), для конкретних опорних зображень у структурі даних набору параметрів послідовності (SPS) для послідовності, що включає в себе поточне зображення. Відеодекодер 30 може відновлювати значення POC з декодованих LSB для значень POC шляхом приєднання LSB до відповідних MSB, витягнутих, наприклад, з раніше декодованого повного значення POC. Відеодекодер 30 може також, додатково або альтернативно, декодувати значення POC для одного або декількох опорних зображень у заголовку поточного слайса поточного зображення. У деяких прикладах, відеодекодер 30 може декодувати дані, що являють собою значення POC довгострокових опорних зображень, у SPS, і значення POC короткострокових опорних зображень, у заголовку слайса. Відеодекодер 30 може також декодувати значення POC міжвидових опорних зображень, наприклад, у SPS, заголовку слайса або в іншому місці. Загалом, значення POC міжвидових опорних зображень є тими ж, що і значення POC поточного зображення, яке кодується.

Відеодекодер 30 також можуть декодувати ідентифікатори другої розмірності опорних зображень (232). Ідентифікатори другої розмірності можуть містити в собі один або кілька ідентифікаторів виду для видів, що включають у себе опорні зображення, індекси порядку виду для видів, що включають у себе опорні зображення, комбінацію індексів порядку виду і прапорів глибини, ідентифікатори рівня для рівнів масштабованого Відеокодування (SCV), що включають у себе опорні зображення, і/або загальні ідентифікатори рівня. Таким чином, комбінація значення POC для опорного зображення і двовимірний ідентифікатор для опорного зображення можуть бути використані для ідентифікації опорного зображення. Таким чином, для ідентифікації опорного зображення, інформація руху може містити в собі значення POC і ідентифікатор зображення другої розмірності для опорного зображення.

Відеодекодер 30 також може декодувати вектор руху для поточного блока поточного зображення. Більш конкретно, відеодекодер 30 може визначати набір доступних кандидатів предиктора вектора руху (234). Наприклад, з посиланням на Фіг. 10, відеодекодер 30 може визначати, чи доступні вектори руху для сусідніх блоків A0, A1, B0, B1 і B2. Більш конкретно, відповідно до способів даного розкриття, відеодекодер 30 може визначати, що вектор руху одного з цих сусідніх блоків недоступний, коли вектор руху сусіднього блока має тип, відмінний від вектора руху поточного блока. Подібним чином, відеодекодер 30 може визначати, чи посиляється вектор руху для кандидата часового предиктора вектора руху на інший тип опорного зображення, ніж вектор руху для поточного блока у визначенні того, чи доступний кандидат TMVP для використання як предиктора кодування вектора руху поточного блока.

Як пояснено вище, приклади різних типів векторів руху містять у собі довгострокові вектори руху, короткострокові вектори руху, часові вектори руху, і вектори руху диспаратності. Таким чином, відеодекодер 30 може визначати тип вектора руху поточного блока, а також типи векторів руху сусідніх блоків, і визначати, що вектори руху сусідніх блоків типів, відмінних від типу поточного вектора руху поточного блока, недоступні для використання як предикторів



вектора руху для поточного вектора руху. Для визначення типів, відеодекодер 30 може посилається на значення РОС опорних зображень, на які посилаються кандидати векторів руху, значення РОС опорного зображення, на яке посилається поточний вектор руху, ідентифікатори другої розмірності опорних зображень, на які посилаються кандидати векторів руху, і/або ідентифікатор другої розмірності опорного зображення, на який посилається поточний вектор руху.

Далі, відеодекодер 30 може вибирати один з доступних кандидатів предиктора вектора руху із сусіднього блока (який може містити в собі так званий суміщений блок у раніше закодованому зображенні і/або відповідний блок у зображенні іншого виду) як предиктор вектора руху для поточного вектора руху (236). Відеодекодер 30 може потім декодувати поточний вектор руху з використанням вибраного предиктора вектора руху (238). Наприклад, з використанням AMVP, відеодекодер 30 може декодувати значення різниці (MVD) векторів руху для поточного вектора руху, потім застосовувати значення MVD до вибраного передбачувача вектора руху. Тобто, відеодекодер 30 може додавати x-компоненту значення MVD до x-компоненти вибраного предиктора вектора руху, і у-компоненту значення MVD до у-компоненти вибраного предиктора вектора руху.

Елемент 72 компенсації руху відеодекодера 30 може потім прогнозувати поточний блок з використанням вектора руху, що вказує на одне з опорних зображень (240). Тобто, на додаток до самого вектора руху, відеодекодер 30 може декодувати інформацію ідентифікації опорного зображення для блока, якому відповідає вектор руху, як, наприклад, значення РОС і значення ідентифікатора другої розмірності. Таким чином, відеодекодер 30 може визначати опорне зображення, на яке вказує вектор руху, з використанням значення РОС і значення ідентифікатора другої розмірності. Відповідно, елемент 72 компенсації руху може формувати прогнозований блок для поточного блока з використанням вектора руху й інформації ідентифікації опорного зображення, тобто, значення РОС і значення ідентифікатора другої розмірності.

Елемент 70 ентропійного декодування можуть додатково декодувати квантовані коефіцієнти перетворення для залишкового блока, що відповідає поточному блоку (242). Елемент 70 ентропійного декодування, елемент 76 зворотного квантування й елемент 78 зворотного перетворення, відповідно, зворотно сканують, квантують і перетворюють квантовані коефіцієнти перетворення для відтворення залишкового блока (244). Суматор 80 відеодекодера 30 може потім комбінувати (тобто, додавати на піксельній основі) прогнозований блок і залишковий блок для відтворення поточного блока (246).

Подібним чином, спосіб на Фіг. 12 являє собою приклад способу, що включає в себе декодування значення рахунку порядку зображення (РОС) для першого зображення відеоданих, декодування ідентифікатора зображення другої розмірності для першого зображення і декодування, відповідно до базової специфікації відеокодування (або розширенням базової специфікації відеокодування), другого зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення. Додатково, спосіб може включати в себе заборону прогнозування вектора руху між першим вектором руху першого блока другого зображення, причому перший вектор руху посилається на короткострокове опорне зображення, і другим вектором руху другого блока другого зображення, причому другий вектор руху посилається на довгострокове опорне зображення. Додатково або альтернативно, спосіб може включати в себе заборону прогнозування вектора руху між першим вектором руху першого блока другого зображення, причому перший вектор руху посилається на міжвидове опорне зображення, і другим вектором руху другого блока другого зображення, причому другий вектор руху посилається на часове опорне зображення.

Необхідно визнати, що, залежно від приклада, конкретні дії або події будь-якого зі способів, описаних у даному документі, можуть бути виконані в іншій послідовності, можуть бути додані, з'єднані або пропущені (наприклад, не всі описані дії або події необхідні для здійснення способів). Більше того, у конкретних прикладах, дії або події можуть бути виконані паралельно, а не послідовно, наприклад, за допомогою багатопотокової обробки, обробки переривань або множини процесорів.

В одному або декількох прикладах, описані функції можуть бути здійснені в технічних засобах, програмному забезпеченні, програмно-апаратних засобах або їхній комбінації. При здійсненні в програмному забезпеченні, функції можуть зберігатися або передаватися як одна або декілька команд або код на зчитуваному комп'ютером носії, і виконуватися елементом обробки технічних засобів. Зчитувані комп'ютером носії можуть містити в собі зчитувані комп'ютером носії даних, що відповідають фізичному носію, як, наприклад, засобу збереження даних, або засобам зв'язку, що включають у себе будь-який носій, що полегшує передачу

комп'ютерної програми з одного місця в інше, наприклад, відповідно до протоколу зв'язку. Таким чином, зчитувані комп'ютером носії, загалом, можуть відповідати (1) фізичним зчитуванням комп'ютером носіям даних, які є енергонезалежними, або (2) засобам зв'язку, як, наприклад, сигнал або несуча хвиля. Засоби збереження даних можуть бути будь-якими доступними носіями, доступ до яких може бути здійснений одним або декількома комп'ютерами або одним або декількома процесорами для витягування команд, коду і/або структур даних для здійснення способів, описаних у даному розкритті. Комп'ютерний програмний продукт може містити в собі зчитуваний комп'ютером носій.

Для прикладу, але не обмеження, такі зчитувані комп'ютером носії даних можуть містити RAM, ROM, EEPROM, CD-ROM або інші сховища на оптичних дисках, сховища на магнітних дисках, або інші магнітні пристрої збереження, флеш-пам'ять, або будь-який інший носій, що може бути використаний для збереження бажаного програмного коду у формі команд або структур даних, і доступ до якого може бути здійснений комп'ютером. Також, будь-яке з'єднання правильно називати зчитуваний комп'ютером носій. Наприклад, якщо інструкції передаються з веб-сайта, сервера або іншого віддаленого джерела з використанням коаксіального кабелю, оптоволоконного кабелю, виті пари, цифрової абонентської лінії (DSL), або бездротових технологій, як, наприклад, інфрачервона, радіо і мікрохвильова, то коаксіальний кабель, оптоволоконний кабель, вита пара, цифрова абонентська лінія (DSL), або бездротові технології, як, наприклад, інфрачервона, радіо і мікрохвильова, включені у визначення носія. Необхідно розуміти, однак, що зчитувані комп'ютером носії даних і засоби збереження даних не містять у собі з'єднання, несучі хвилі, сигнали й інші енергозалежні носії, але, навпаки, спрямовані на енергонезалежні, фізичні носії даних. Диск [disk] і диск [disc], як використано в даному документі, містить у собі компактний диск (CD), лазерний диск, оптичний диск, універсальний цифровий диск (DVD), гнучкий диск і диск Blu-ray, де диски [disks] звичайно відтворюють дані магнітним чином, у той час як диски [discs] відтворюють дані оптично за допомогою лазерів. Комбінації вищевказаного також повинні бути включені в обсяг зчитуваних комп'ютером носіїв.

Інструкції можуть бути виконані одним або декількома процесорами, як, наприклад, один або кілька цифрових сигнальних процесорів (DSP), мікропроцесорів загального призначення, спеціалізованих інтегральних схем (ASIC), програмованих користувачем вентилятих матриць (FPGA) або інших еквівалентних інтегральних або дискретних логічних схем. Відповідно, термін "процесор", як використано в даному документі, може стосуватися будь-якої з вищевказаних структур або будь-якої іншої структури, що підходить для здійснення способів, описаних у даному документі. Додатково, у деяких аспектах, функціональність, описана в даному документі, може бути забезпечена усередині закріплених технічних засобів і/або модулів програмного забезпечення, виконаних з можливістю кодувати і декодувати, або убудованих у комбінований кодек. Також, способи можуть бути повністю здійснені в одній або декількох схемах або логічних елементах.

Способи даного розкриття можуть бути здійснені в різноманітних пристроях або приладах, що включають у себе бездротову трубку, інтегральну схему (IC) або групу IC (наприклад, комплект мікросхем). Різні компоненти, модулі або елементи описані в даному розкритті для акценту на функціональних аспектах пристроїв, виконаних з можливістю виконувати розкриті способи, але необов'язково потребуючі реалізації в різних елементах технічних засобів. Швидше навпаки, як описано вище, різні елементи можуть бути скомбіновані в елементі технічних засобів кодека або забезпечені в сукупності взаємодіючих елементів технічних засобів, включаючи один або кілька процесорів, як описано вище, разом із придатним програмним забезпеченням і/або програмно-апаратними засобами.

#### ФОРМУЛА ВИНАХОДУ

1. Спосіб декодування відеоданих, який включає:  
 декодування даних другого зображення, що стосуються значення рахунка порядку зображення (РОС) для першого зображення відеоданих;  
 декодування даних другого зображення, що стосуються ідентифікатора зображення другої розмірності для першого зображення;  
 декодування, відповідно до базової специфікації відеокодування, другого зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення; і  
 заборону прогнозування вектора руху між першим вектором руху першого блока другого зображення і другим вектором руху другого блока другого зображення, при цьому перший

вектор руху посилається на короткострокове опорне зображення, і при цьому другий вектор руху посилається на довгострокове опорне зображення.

2. Спосіб за п. 1, у якому кодування другого зображення включає: ідентифікацію першого зображення з використанням значення РОС і ідентифікатора зображення другої розмірності; і декодування щонайменше частини другого зображення відносно першого зображення.

3. Спосіб за п. 2, у якому ідентифікація першого зображення включає ідентифікацію першого зображення під час декодування вектора руху для блока другого зображення, при цьому кодування вектора руху містить кодування вектора руху згідно з щонайменше одним з удосконаленого прогнозування вектора руху (AMVP), часового прогнозування вектора руху (TMVP) і режиму злиття.

4. Спосіб за п. 1, який додатково включає:

дозвіл прогнозування між першим короткостроковим вектором руху другого зображення і другим короткостроковим вектором руху другого зображення; і

масштабування щонайменше одного з першого короткострокового вектора руху і другого короткострокового вектора руху на основі значення РОС для першого короткострокового опорного зображення, на яке посилається перший короткостроковий вектор руху, і значення РОС для другого короткострокового опорного зображення, на яке посилається другий короткостроковий вектор руху.

5. Спосіб за п. 1, який додатково включає декодування значення, яке вказує, чи містить третє зображення відеоданих довгострокове опорне зображення, при цьому значення, яке вказує, чи містить третє зображення довгострокове опорне зображення, додатково вказує, чи використовується третє зображення для міжвидового прогнозування.

6. Спосіб за п. 1, який додатково включає декодування, відповідно до розширення базової специфікації відеокодування, третього зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

7. Спосіб за п. 6, який додатково включає, до декодування третього зображення, маркування всіх міжвидових опорних зображень, у тому числі перше зображення, як довгострокових опорних зображень.

8. Спосіб за п. 7, який додатково включає:

збереження статусу для кожного з міжвидових опорних зображень для третього зображення, при цьому статус містить одне з довгострокового опорного зображення, короткострокового опорного зображення і невикористовуваного для посилення, до маркування міжвидових опорних зображень як довгострокових опорних зображень, при цьому міжвидові опорні зображення містять у собі перше зображення; і, після кодування другого зображення, встановлення нових статусів для кожного з міжвидових опорних зображень на основі збережених статусів.

9. Спосіб за п. 6, у якому базова специфікація відеокодування містить базову специфікацію Високоєфективного Відеокодування (HEVC), і в якому розширення базової специфікації відеокодування містить одне з розширення Масштабованого Відеокодування (SCV) для базової специфікації HEVC і розширення Багатовидового Відеокодування (MVC) для базової специфікації HEVC.

10. Спосіб за п. 6, у якому ідентифікатор зображення другої розмірності містить щонайменше один з ідентифікатора виду для виду, що включає в себе перше зображення, індексу порядку виду для виду, що включає в себе перше зображення, комбінації індексу порядку виду і прапора глибини, ідентифікатора рівня для рівня масштабованого відеокодування (SCV), що включає в себе перше зображення, і загального ідентифікатора рівня.

11. Спосіб за п. 6, який додатково включає, після декодування третього зображення, маркування кожного міжвидового опорного зображення як одного з довгострокового опорного зображення, короткострокового опорного зображення і невикористовуваного для посилення.

12. Спосіб за п. 11, який додатково включає:

після маркування міжвидового опорного зображення як довгострокового опорного зображення, призначення міжвидовому опорному зображенню нового значення РОС, що у даний момент є невикористовуваним; і,

після декодування другого зображення, відновлення вихідного значення РОС для міжвидового опорного зображення.

13. Спосіб за п. 12, у якому вихідне значення РОС містить значення РОС першого зображення.

14. Спосіб кодування відеоданих, який включає:

кодування даних другого зображення, що стосуються значення рахунка порядку зображення (РОС) для першого зображення відеоданих;

кодування даних другого зображення, що стосуються ідентифікатора зображення другої розмірності для першого зображення;

кодування, відповідно до базової специфікації відеокодування, другого зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення; і

заборону прогнозування вектора руху між першим вектором руху першого блока другого зображення, при цьому перший вектор руху посиляється на короткострокове опорне зображення, і другим вектором руху другого блока другого зображення, при цьому другий вектор руху посиляється на довгострокове опорне зображення.

15. Спосіб за п. 14, який додатково включає:

ідентифікацію першого зображення з використанням значення РОС і ідентифікатора зображення другої розмірності; і

кодування щонайменше частини другого зображення відносно першого зображення.

16. Спосіб за п. 15, у якому ідентифікація першого зображення містить ідентифікацію першого зображення під час кодування вектора руху для блока другого зображення, при цьому кодування вектора руху містить кодування вектора руху згідно з щонайменше одним з удосконаленого прогнозування вектора руху (AMVP), часового прогнозування вектора руху (TMVP) і режиму злиття.

17. Спосіб за п. 14, який додатково включає:

дозвіл прогнозування між першим короткостроковим вектором руху другого зображення і другим короткостроковим вектором руху другого зображення; і

масштабування щонайменше одного з першого короткострокового вектора руху і другого короткострокового вектора руху на основі значення РОС для першого короткострокового опорного зображення, на яке посиляється перший короткостроковий вектор руху, і значення РОС для другого короткострокового опорного зображення, на яке посиляється другий короткостроковий вектор руху.

18. Спосіб за п. 14, у якому ідентифікатор зображення другої розмірності містить щонайменше один з ідентифікатора виду для виду, що включає в себе перше зображення, індексу порядку виду для виду, що включає в себе перше зображення, комбінації індексу порядку виду і прапора глибини, ідентифікатора рівня для рівня масштабованого відеокодування (SCV), що включає в себе перше зображення, і загального ідентифікатора рівня.

19. Спосіб за п. 14, який додатково включає кодування значення, яке вказує, чи містить третє зображення відеоданих довгострокове опорне зображення, при цьому значення, яке вказує, чи містить третє зображення довгострокове опорне зображення, додатково вказує, чи використовується третє зображення для міжвидового прогнозування.

20. Спосіб за п. 14, який додатково включає кодування, відповідно до розширення базової специфікації відеокодування, третього зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

21. Спосіб за п. 20, який додатково включає, до кодування третього зображення, маркування всіх міжвидових опорних зображень як довгострокових опорних зображень.

22. Спосіб за п. 21, який додатково включає:

збереження статусу для кожного з міжвидових опорних зображень, при цьому статус містить одне з довгострокового опорного зображення, короткострокового опорного зображення і невикористовуваного для посилення, до маркування міжвидових опорних зображень як довгострокових опорних зображень; і,

після кодування другого зображення, встановлення нових статусів для кожного з міжвидових опорних зображень на основі збережених статусів.

23. Спосіб за п. 20, у якому базова специфікація відеокодування містить базову специфікацію Високоєфективного Відеокодування (HEVC), і в якому розширення базової специфікації відеокодування містить одне з розширення Масштабованого Відеокодування (SCV) для базової специфікації HEVC і розширення Багатовидового Відеокодування (MVC) для базової специфікації HEVC.

24. Спосіб за п. 14, який додатково включає, після кодування третього зображення, маркування кожного міжвидового опорного зображення як одного з довгострокового опорного зображення, короткострокового опорного зображення і невикористовуваного для посилення.

25. Спосіб за п. 24, який додатково включає:

після маркування міжвидового опорного зображення як довгострокового опорного зображення, призначення міжвидовому опорному зображенню нового значення РОС, що у даний момент є невикористовуваним; і,

після кодування другого зображення, відновлення вихідного значення РОС для міжвидового опорного зображення.

26. Спосіб за п. 25, у якому вихідне значення РОС містить значення РОС другого зображення.

27. Пристрій декодування відеоданих, який містить відеодекодер, виконаний з можливістю декодувати дані другого зображення, що стосуються значення рахунка порядку зображення (РОС) для першого зображення відеоданих, декодувати дані другого зображення, що стосуються ідентифікатора зображення другої розмірності для першого зображення, декодувати, відповідно до базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення, і забороняти прогнозування вектора руху між першим вектором руху першого блока другого зображення, при цьому перший вектор руху посиляється на короткострокове опорне зображення, і другим вектором руху другого блока другого зображення, при цьому другий вектор руху посиляється на довгострокове опорне зображення.

28. Пристрій за п. 27, у якому відеодекодер виконаний з можливістю ідентифікувати перше зображення з використанням значення РОС і ідентифікатора зображення другої розмірності і декодувати щонайменше частину другого зображення відносно першого зображення.

29. Пристрій за п. 28, у якому відеодекодер виконаний з можливістю ідентифікувати перше зображення під час кодування вектора руху для блока другого зображення, і в якому відеодекодер виконаний з можливістю декодувати вектор руху згідно з щонайменше одним з удосконаленого прогнозування вектора руху (AMVP), часового прогнозування вектора руху (TMVP) і режиму злиття.

30. Пристрій за п. 27, у якому відеодекодер виконаний з можливістю дозволяти прогнозування між першим короткостроковим вектором руху другого зображення і другим короткостроковим вектором руху другого зображення і масштабувати щонайменше один з першого короткострокового вектора руху і другого короткострокового вектора руху на основі значення РОС для першого короткострокового опорного зображення, на яке посиляється перший короткостроковий вектор руху, і значення РОС для другого короткострокового опорного зображення, на яке посиляється другий короткостроковий вектор руху.

31. Пристрій за п. 27, у якому ідентифікатор зображення другої розмірності містить щонайменше один з ідентифікатора виду для виду, що включає в себе перше зображення, індексу порядку виду для виду, що включає в себе перше зображення, комбінації індексу порядку виду і прапора глибини, ідентифікатора рівня для рівня масштабованого відеокодування (SCV), що включає в себе перше зображення, і загального ідентифікатора рівня.

32. Пристрій за п. 27, у якому відеодекодер виконаний з можливістю декодувати значення, яке вказує, чи містить третє зображення відеоданих довгострокове опорне зображення, при цьому значення, яке вказує, чи містить третє зображення довгострокове опорне зображення, додатково вказує, чи використовується третє зображення для міжвидового прогнозування.

33. Пристрій за п. 27, у якому відеодекодер додатково виконаний з можливістю декодувати, відповідно до розширення базової специфікації відеокодування, третє зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення.

34. Пристрій за п. 33, у якому відеодекодер виконаний з можливістю маркувати всі міжвидові опорні зображення для третього зображення, у тому числі перше зображення, як довгострокові опорні зображення, до декодування третього зображення, зберігати статус для кожного з міжвидових опорних зображень, при цьому статус містить одне з довгострокового опорного зображення, короткострокового опорного зображення і невикористовуваного для посилення, до маркування міжвидових опорних зображень як довгострокових опорних зображень, і, після декодування третього зображення, установлювати нові статуси для кожного з міжвидових опорних зображень на основі збережених статусів.

35. Пристрій за п. 33, у якому відеодекодер додатково виконаний з можливістю маркувати кожне міжвидове опорне зображення для третього зображення, у тому числі перше зображення, як одне з довгострокового опорного зображення, короткострокового опорного зображення і невикористовуваного для посилення, після декодування третього зображення, призначати кожному з міжвидових опорних зображень нове значення РОС, що у даний момент є невикористовуваним, після маркування міжвидового опорного зображення як довгострокового опорного зображення, і відновлювати вихідне значення РОС для міжвидового опорного зображення, після кодування другого зображення.

36. Пристрій за п. 27, у якому пристрій містить щонайменше одне з: інтегральної схеми;

мікропроцесора; і

бездротового пристрою зв'язку, що містить у собі відеодекодер.

37. Пристрій кодування відеоданих, який містить відеокодер, виконаний з можливістю кодувати дані другого зображення, що стосуються значення рахунка порядку зображення (POC) для першого зображення відеоданих, кодувати дані другого зображення, що стосуються ідентифікатора зображення другої розмірності для першого зображення, кодувати, відповідно до базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення POC і ідентифікатора зображення другої розмірності першого зображення, і забороняти прогнозування вектора руху між першим вектором руху першого блока другого зображення, при цьому перший вектор руху посиляється на короткострокове опорне зображення, і другим вектором руху другого блока другого зображення, при цьому другий вектор руху посиляється на довгострокове опорне зображення.

38. Пристрій за п. 37, у якому відеокодер виконаний з можливістю ідентифікувати перше зображення з використанням значення POC і ідентифікатора зображення другої розмірності і кодувати щонайменше частину другого зображення відносно першого зображення.

39. Пристрій за п. 38, у якому відеокодер виконаний з можливістю ідентифікувати перше зображення під час кодування вектора руху для блока другого зображення, і в якому відеокодер виконаний з можливістю кодувати вектор руху згідно з щонайменше одним з удосконаленого прогнозування вектора руху (AMVP), часового прогнозування вектора руху (TMVP) і режиму злиття.

40. Пристрій за п. 37, у якому відеокодер виконаний з можливістю дозволяти прогнозування між першим короткостроковим вектором руху другого зображення і другим короткостроковим вектором руху другого зображення і масштабувати щонайменше один з першого короткострокового вектора руху і другого короткострокового вектора руху на основі значення POC для першого короткострокового опорного зображення, на яке посиляється перший короткостроковий вектор руху, і значення POC для другого короткострокового опорного зображення, на яке посиляється другий короткостроковий вектор руху.

41. Пристрій за п. 37, у якому ідентифікатор зображення другої розмірності містить щонайменше один з ідентифікатора виду для виду, що включає в себе перше зображення, індексу порядку виду для виду, що включає в себе перше зображення, комбінації індексу порядку виду і прапора глибини, ідентифікатора рівня для рівня масштабованого відеокодування (SCV), що включає в себе перше зображення, і загального ідентифікатора рівня.

42. Пристрій за п. 37, у якому відеокодер виконаний з можливістю кодувати значення, яке вказує, чи містить третє зображення відеоданих довгострокове опорне зображення, при цьому значення, яке вказує, чи містить третє зображення довгострокове опорне зображення, додатково вказує, чи використовується третє зображення для міжвидового прогнозування.

43. Пристрій за п. 37, у якому відеокодер додатково виконаний з можливістю кодувати, відповідно до розширення базової специфікації відеокодування, третє зображення на основі, щонайменше частково, значення POC і ідентифікатора зображення другої розмірності першого зображення.

44. Пристрій за п. 43, у якому відеокодер виконаний з можливістю маркувати всі міжвидові опорні зображення для третього зображення, що включають у себе перше зображення, як довгострокові опорні зображення, до кодування третього зображення, зберігати статус для кожного з міжвидових опорних зображень, при цьому статус містить одне з довгострокового опорного зображення, короткострокового опорного зображення і невикористовуваного для посилання, до маркування міжвидових опорних зображень як довгострокових опорних зображень, і, після кодування третього зображення, установлювати нові статуси для кожного з міжвидових опорних зображень на основі збережених статусів.

45. Пристрій за п. 43, у якому відеокодер додатково виконаний з можливістю маркувати кожне міжвидове опорне зображення для третього зображення, у тому числі перше зображення, як одне з довгострокового опорного зображення, короткострокового опорного зображення і невикористовуваного для посилання, після кодування третього зображення, призначати кожному з міжвидових опорних зображень нове значення POC, що у даний момент є невикористовуваним, після маркування міжвидового опорного зображення як довгострокового опорного зображення, і відновлювати вихідне значення POC для міжвидового опорного зображення, після кодування другого зображення.

46. Пристрій кодування відеоданих, який містить: засіб для кодування даних другого зображення, які стосуються значення рахунка порядку зображення (POC) для першого зображення відеоданих;

засіб для кодування даних другого зображення, що стосуються ідентифікатора зображення другої розмірності для першого зображення;

засіб для кодування, відповідно до базової специфікації відеокодування, другого зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення; і

- 5 засіб для заборони прогнозування вектора руху між першим вектором руху першого блока другого зображення, при цьому перший вектор руху посиляється на короткострокове опорне зображення, і другим вектором руху другого блока другого зображення, при цьому другий вектор руху посиляється на довгострокове опорне зображення.

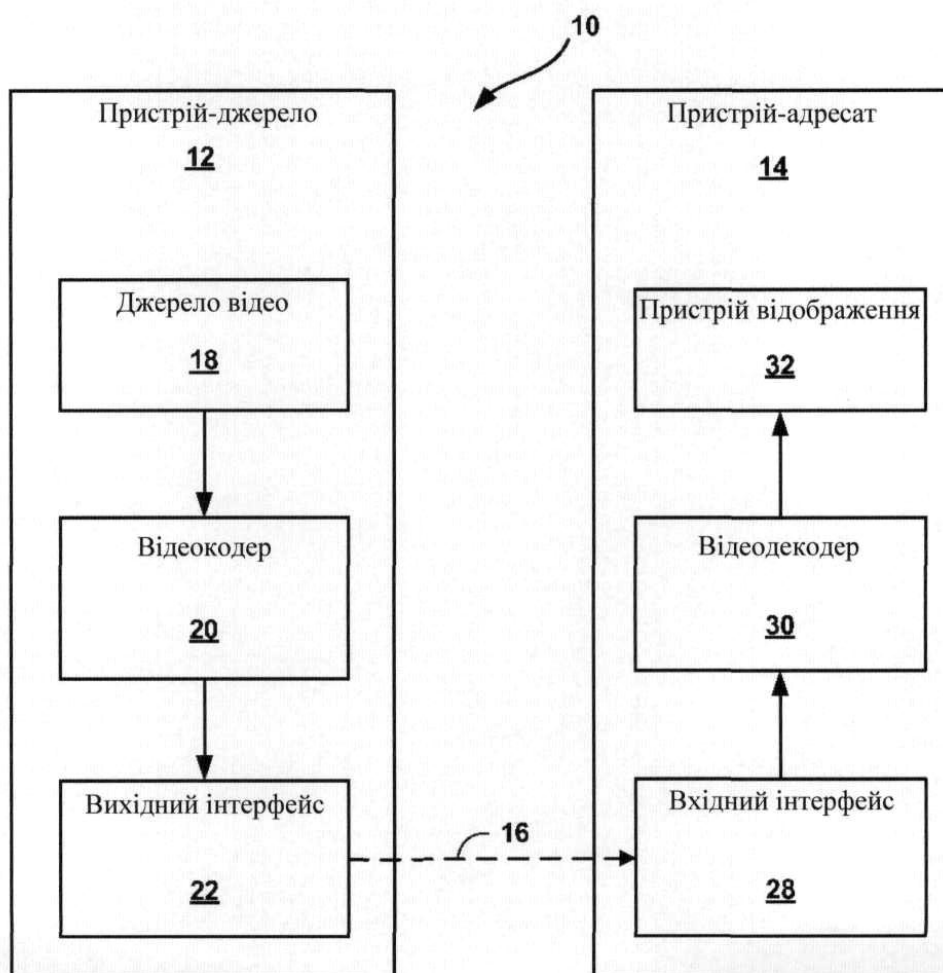
47. Зчитуваний комп'ютером носій даних, на якому зберігаються інструкції, що, при їхньому виконанні, спонукають процесор:

- 10 декодувати дані другого зображення, що стосуються значення рахунка порядку зображення (РОС) для першого зображення відеоданих;

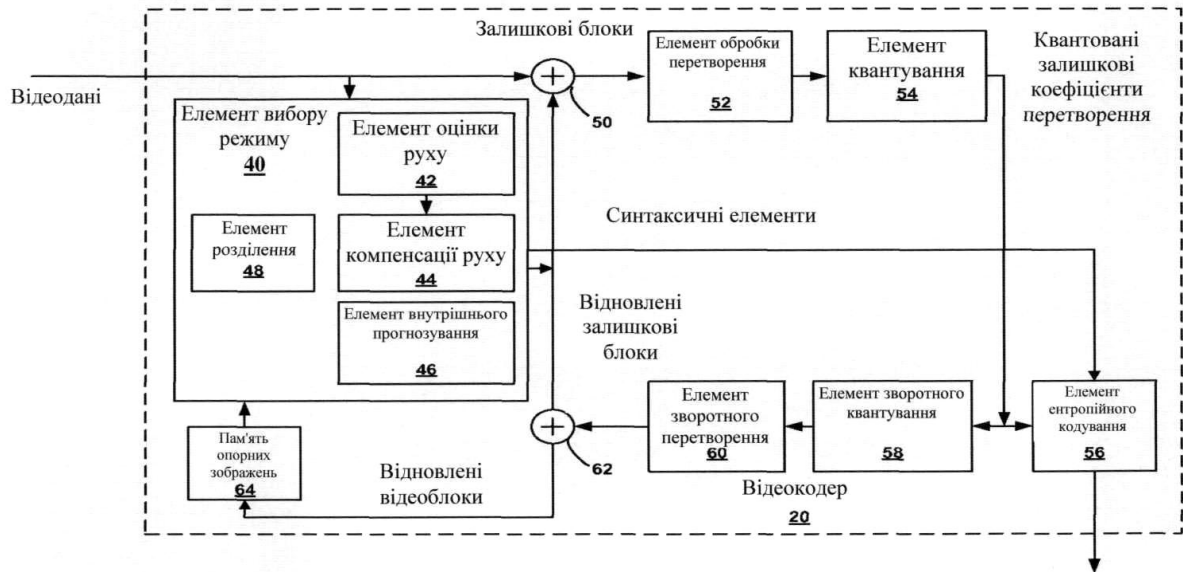
декодувати дані другого зображення, що стосуються ідентифікатора зображення другої розмірності для першого зображення;

- 15 декодувати, відповідно до базової специфікації відеокодування, друге зображення на основі, щонайменше частково, значення РОС і ідентифікатора зображення другої розмірності першого зображення; і

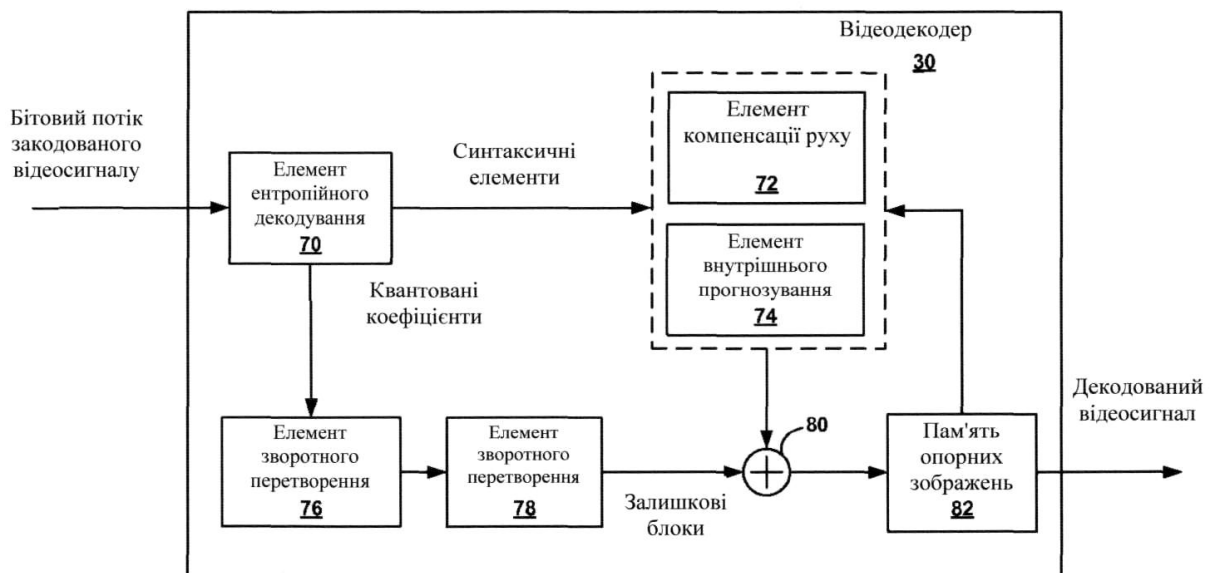
- 20 забороняти прогнозування вектора руху між першим вектором руху першого блока другого зображення, при цьому перший вектор руху посиляється на короткострокове опорне зображення, і другим вектором руху другого блока другого зображення, при цьому другий вектор руху посиляється на довгострокове опорне зображення.



Фіг. 1

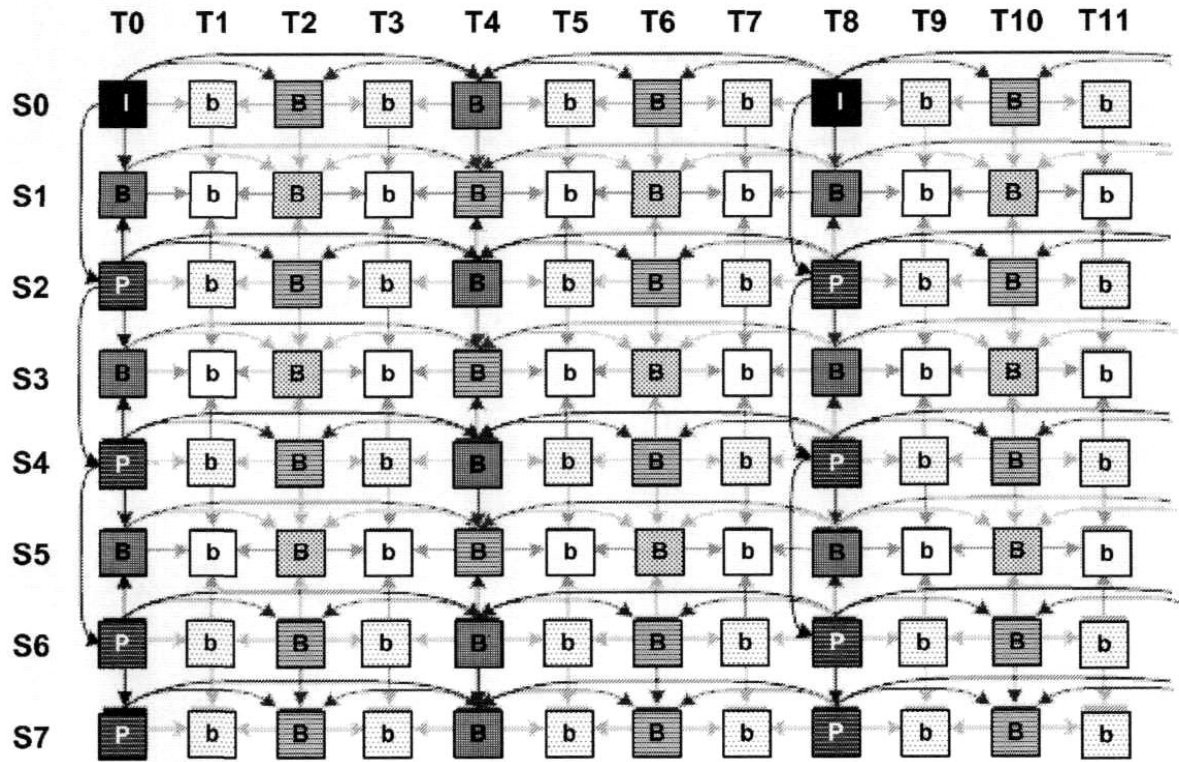


Фіг. 2

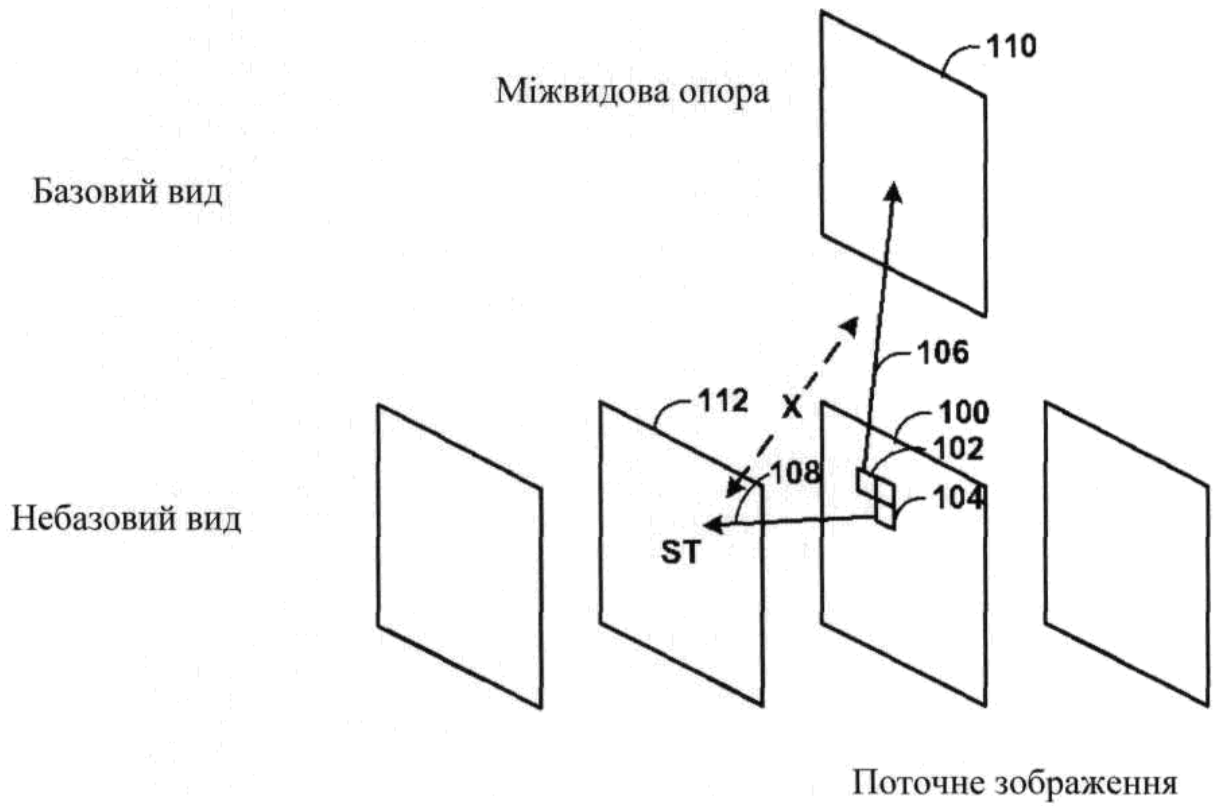


Фіг. 3

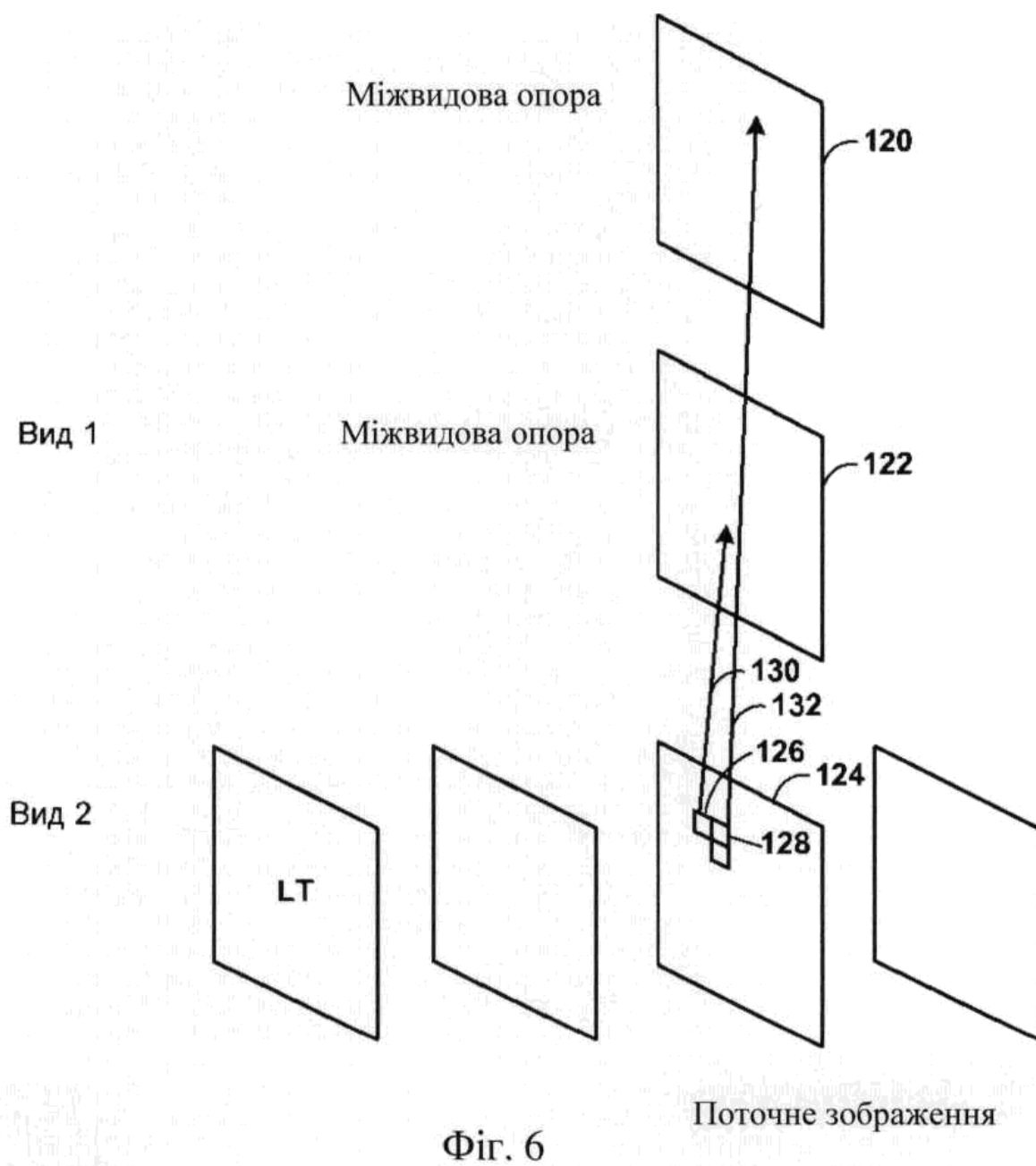


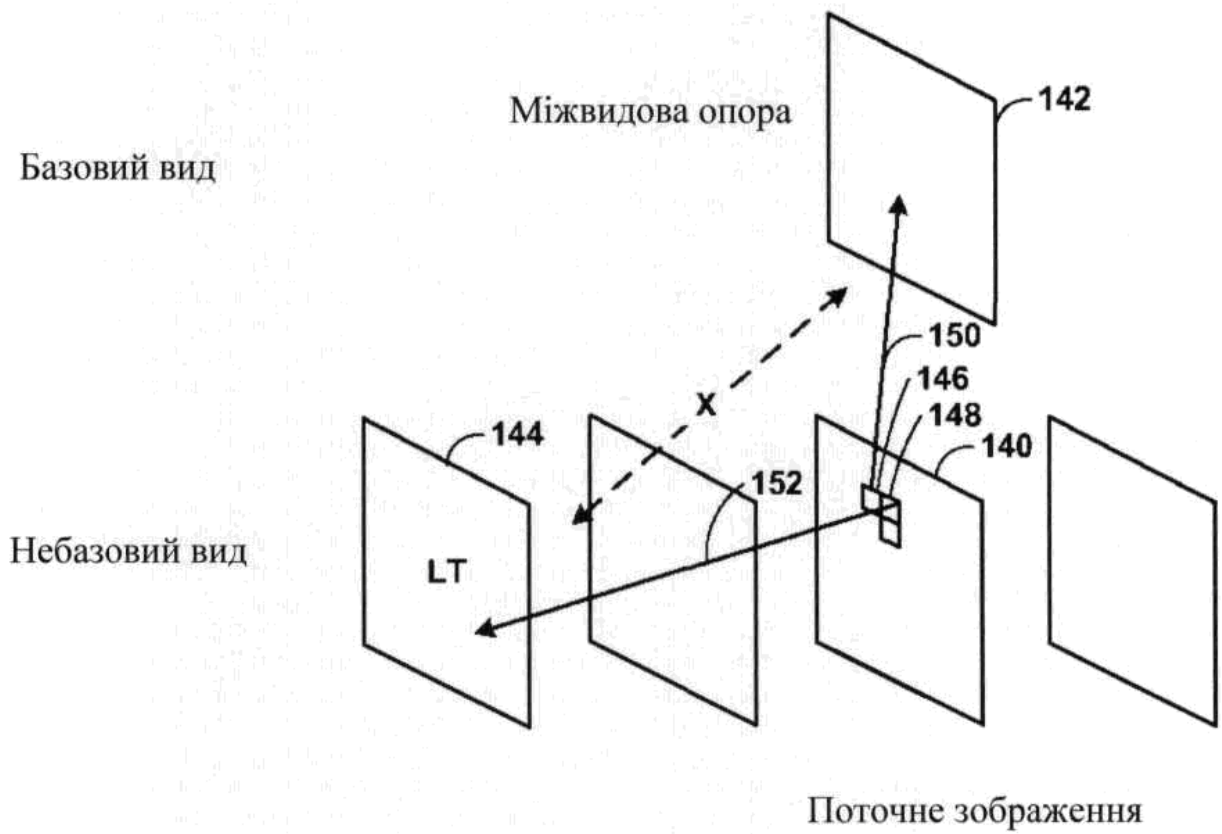


Фіг. 4

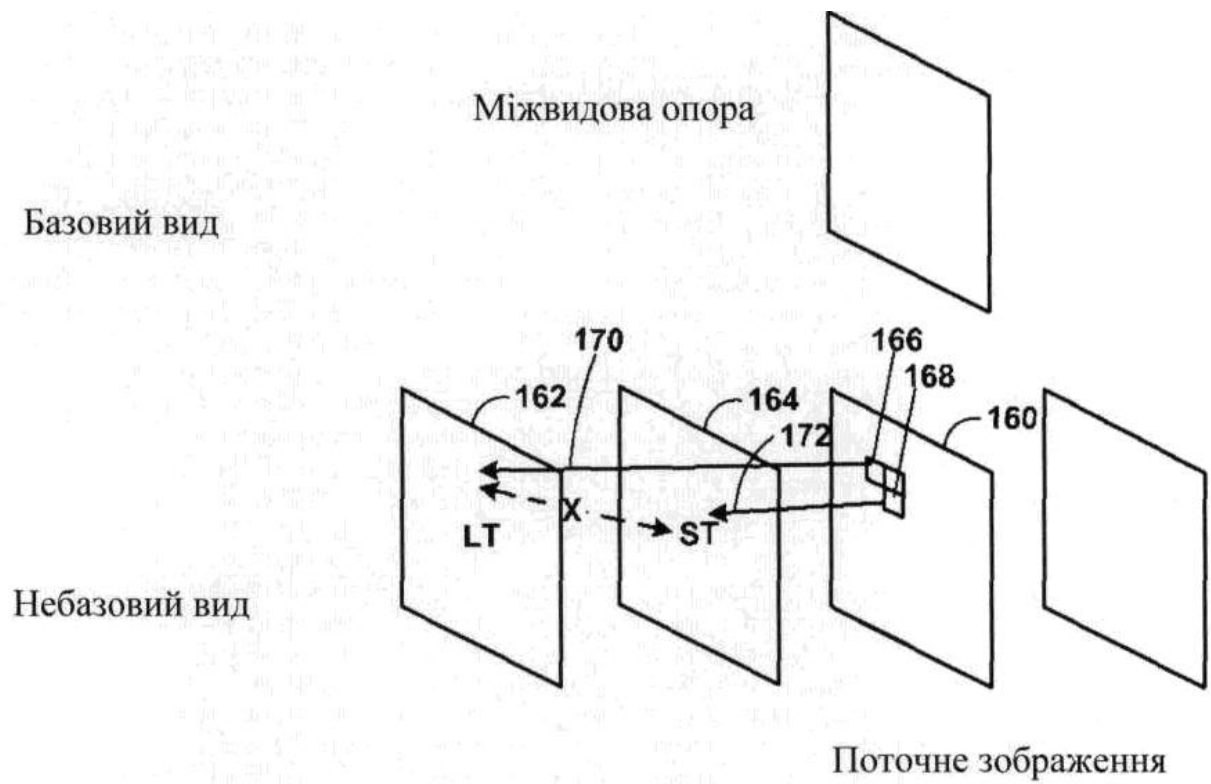


Фіг. 5

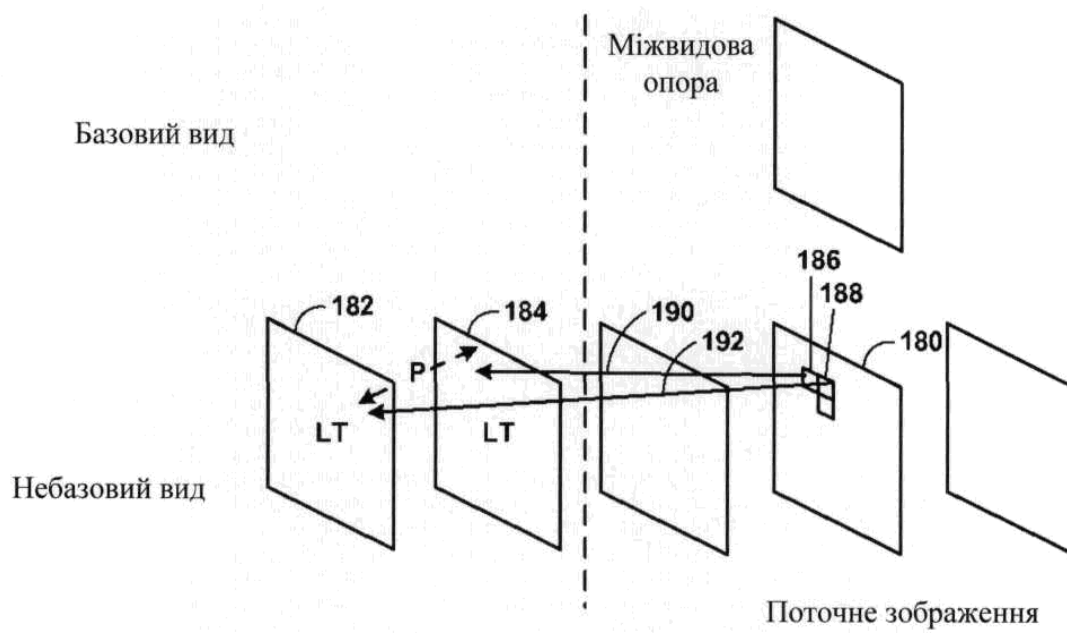




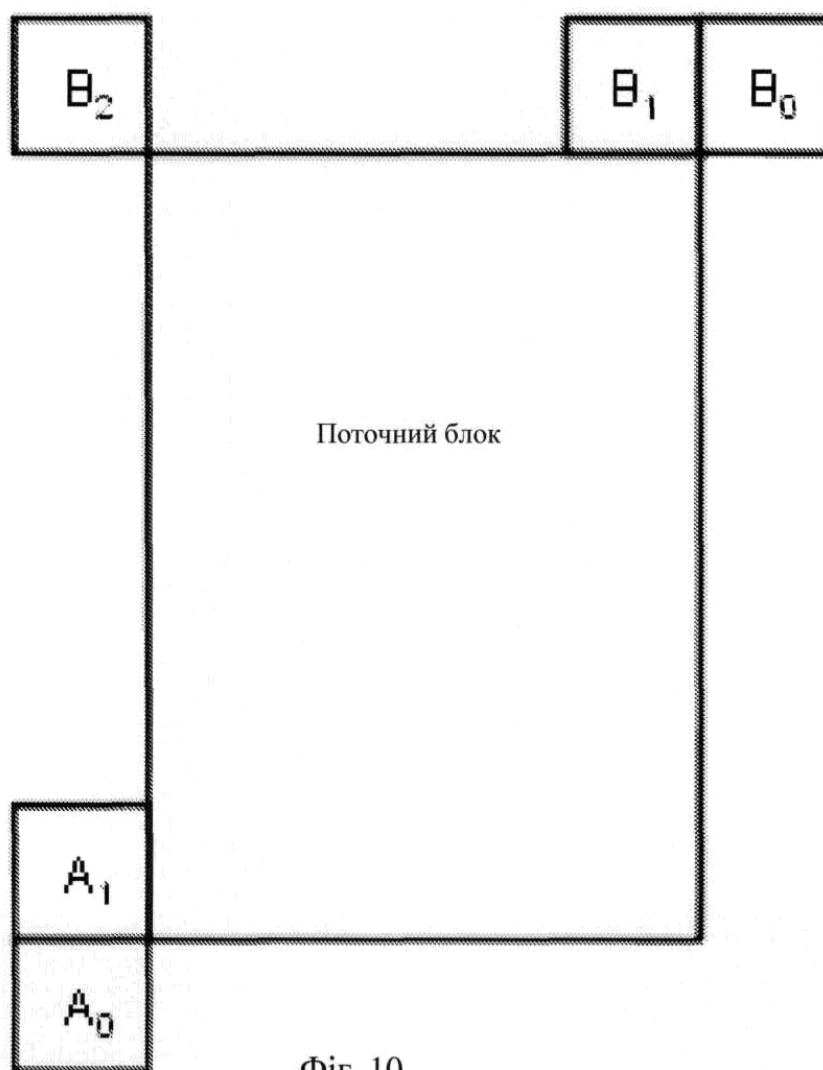
Фіг. 7



Фіг. 8



Фіг. 9



Фіг. 10



Фіг. 11



Фіг. 12

Комп'ютерна верстка Г. Паяльніков

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

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