



УКРАЇНА

(19) UA (11) 92368 (13) C2  
(51) МПК (2009)  
H04N 7/26

МІНІСТЕРСТВО ОСВІТИ  
І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ ДЕПАРТАМЕНТ  
ІНТЕЛЕКТУАЛЬНОЇ  
ВЛАСНОСТІ

## ОПИС ДО ПАТЕНТУ НА ВІНАХІД

(54) МЕТОДИКА МАСШТАБОВАНOSTI НА ОСНОВІ ІНФОРМАЦІЇ ВМІСТУ

1

2

(21) a200805485

(22) 27.09.2006

(24) 25.10.2010

(86) PCT/US2006/037948, 27.09.2006

(31) 60/721,416

(32) 27.09.2005

(33) US

(31) 60/727,640

(32) 17.10.2005

(33) US

(31) 60/727,643

(32) 17.10.2005

(33) US

(31) 60/727,644

(32) 17.10.2005

(33) US

(31) 60/730,145

(32) 24.10.2005

(33) US

(31) 60/789,048

(32) 03.04.2006

(33) US

(31) 60/789,377

(32) 04.04.2006

(33) US

(46) 25.10.2010, Бюл.№ 20, 2010 р.

(72) РАВІІНДРАН ВІДЖАЯЛАКШМІ Р., US, УОЛКЕР  
ГОРДОН КЕНТ, US, ТЯНЬ ТАО, US, БХАМІДІПАТІ  
ПХАНІКУМАР, US, ШИ ФАН, US, ЧЕНЬ ПЕЙСУН,  
US, СУБРАМАНІА СІТАРАМАН ГАНАПАТХІ, US,  
ОГУЗ СЕФУЛЛАХ ХАЛІТ, US

(73) КВЕЛКОММ ІНКОРПОРЕЙТЕД, US

(56) ХР 011014178; 03.2001

ХР 005312621; 04.2006

ХР 0045244564; 08.2004

US 2001019657 A1; 06.09.2001

(57) 1. Спосіб кодування мультимедійних даних,  
який включає етапи, на яких:

- класифікують вміст мультимедійних даних;
- секціонують мультимедійні дані у першу групу даних і другу групу даних, основуючись на класифікації вмісту, при цьому секціонування визначається відповідно до співвідношення смуги пропускання; і
- кодують мультимедійні дані у першій групі даних і у другій групі даних на основі вказаного секціонування, при цьому співвідношення кількості даних у першій групі даних і кількості даних у другій групі

даних відповідає співвідношенню смуги пропускання.

2. Спосіб за п. 1, в якому перша група даних містить коефіцієнт, а друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних.

3. Спосіб за п. 1, в якому згадане кодування включає етап, на якому визначають швидкість передачі бітів на основі класифікації вмісту мультимедійних даних і кодують мультимедійні дані на основі швидкості передачі бітів.

4. Спосіб за п. 1, в якому класифікація вмісту включає етап, на якому визначають складність мультимедійних даних.

5. Спосіб за п. 4, в якому складність містить часову складність або просторову складність.

6. Спосіб за п. 4, в якому складність містить часову складність і просторову складність.

7. Спосіб за п. 1, в якому кодування включає етап, на якому кодують мультимедійні дані так, щоб забезпечити декодування тільки першої групи даних або декодування першої групи даних і другої групи даних в одну комбіновану групу даних.

8. Спосіб за п. 2, в якому перше диференціальне уточнення вказує різницю між вибраним відеокадром і даними кадру, що впливають з декодування першої групи даних.

9. Спосіб за п. 1, в якому перша група даних - це базовий рівень, а друга група даних - це рівень покращення.

10. Спосіб за п. 9, який додатково включає етапи, на яких:

- вибирають коефіцієнт з одного з коефіцієнтів залишкової помилки вихідного базового рівня і коефіцієнта залишкової помилки вихідного рівня покращення; і

- обчислюють перше диференціальне уточнення на основі згаданого коефіцієнта і коефіцієнта залишкової помилки вихідного рівня покращення.

11. Спосіб за п. 1, в якому кодування додатково включає етап, на якому кодують інформацію заголовків макроблоків і інформацію вектора руху в першій групі даних.

12. Спосіб за п. 1, в якому кодування додатково включає етап, на якому квантують першу групу даних з першим розміром кроку і квантують другу групу даних з другим розміром кроку, при цьому перший розмір кроку і другий розмір кроку співвід-

(13) C2

(11) 92368

(19) UA

носяться за допомогою коефіцієнта масштабування.

13. Спосіб за п. 1, в якому кодування додатково включає етап, на якому визначають перший параметр квантування, що має перший розмір кроку квантування, для використання при кодуванні першої групи даних і визначають другий параметр квантування, що має другий розмір кроку квантування, для використання при кодуванні другої групи даних, при цьому перший і другий параметри квантування визначаються на основі інформації вмісту відомих даних кадру і при цьому згаданий перший розмір кроку квантування більш грубий, ніж згаданий другий розмір кроку квантування.

14. Спосіб за п. 1, в якому кодування включає етап, на якому кодують першу групу даних за допомогою I-кадрів і P-кадрів або будь-якої їхньої комбінації і кодують другу групу даних за допомогою I-кадрів, P-кадрів і B-кадрів або будь-якої їхньої комбінації.

15. Пристрій для кодування мультимедійних даних, який містить:

- засіб класифікації вмісту мультимедійних даних;
- засіб секціонування мультимедійних даних у першу групу даних і другу групу даних, оснований на класифікації вмісту, при цьому секціонування визначається відповідно до співвідношення смуги пропускання; і
- засіб кодування мультимедійних даних у першій групі даних і в другій групі даних на основі зазначеного секціонування, при цьому співвідношення кількості даних у першій групі даних і кількості даних у другій групі даних відповідає співвідношенню смуги пропускання.

16. Пристрій за п. 15, в якому перша група даних містить коефіцієнт, а друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних.

17. Пристрій за п. 15, в якому згаданий засіб для кодування містить засіб визначення швидкості передачі бітів на основі класифікації вмісту мультимедійних даних і кодування мультимедійних даних на основі швидкості передачі бітів.

18. Пристрій за п. 15, в якому згаданий засіб класифікації вмісту містить засіб визначення складності мультимедійних даних, при цьому вибрані мультимедійні дані кодуються на основі складності мультимедійних даних.

19. Пристрій за п. 18, в якому складність містить часову складність або просторову складність.

20. Пристрій за п. 18, в якому складність містить часову складність і просторову складність.

21. Пристрій за п. 15, в якому згаданий засіб кодування містить засіб кодування мультимедійних даних так, щоб забезпечити декодування тільки першої групи даних або декодування першої групи даних і другої групи даних в одну комбіновану групу даних.

22. Пристрій за п. 16, в якому перше диференціальне уточнення вказує різницю між вибраним відеокадром і даними кадру, що впливають з декодування першої групи даних.

23. Пристрій за п. 16, в якому перша група даних - це базовий рівень, а друга група даних - це рівень покращення.

24. Пристрій за п. 15, в якому засіб кодування містить засіб кодування інформації заголовків макроблоків і інформації вектора руху в першій групі даних.

25. Пристрій за п. 15, в якому згаданий засіб кодування додатково містить засіб квантування першої групи даних з першим розміром кроку і квантування другої групи даних з другим розміром кроку, при цьому перший розмір кроку і другий розмір кроку співвідносяться за допомогою коефіцієнта масштабування.

26. Пристрій за п. 15, в якому згаданий засіб кодування містить засіб визначення першого параметра квантування, що має перший розмір кроку, квантування, для використання при кодуванні першої групи даних і визначення другого параметра квантування, що має другий розмір кроку квантування, для використання при кодуванні другої групи даних, при цьому перший і другий параметри квантування визначаються на основі інформації вмісту відомих даних кадру і при цьому згаданий перший розмір кроку квантування більш грубий, ніж згаданий другий розмір кроку квантування.

27. Пристрій за п. 15, в якому згаданий засіб кодування містить:

- засіб кодування першої групи даних за допомогою I-кадрів і P-кадрів; і
- засіб кодування другої групи даних за допомогою I-кадрів, P-кадрів і B-кадрів.

28. Пристрій за п. 23, в якому згаданий засіб кодування містить:

- засіб вибору коефіцієнта з одного з коефіцієнта залишкової помилки вихідного базового рівня і коефіцієнта залишкової помилки вихідного рівня покращення; і
- засіб обчислення першого диференціального уточнення на основі згаданого коефіцієнта і коефіцієнта залишкової помилки вихідного рівня покращення.

29. Пристрій, сконфігурований так, щоб кодувати мультимедійні дані, який містить:

- модуль класифікації вмісту, сконфігурований так, щоб класифікувати вміст мультимедійних даних і надавати дані класифікації вмісту;
- модуль секціонування, який секціонує мультимедійні дані в першу групу даних і другу групу даних, оснований на класифікації вмісту, при цьому секціонування визначається відповідно до співвідношення смуги пропускання; і
- кодер, сконфігурований так, щоб кодувати мультимедійні дані в першій групі даних і в другій групі даних на основі вказаного секціонування, при цьому співвідношення кількості даних у першій групі даних і кількості даних у другій групі даних відповідає співвідношенню смуги пропускання.

30. Пристрій за п. 29, в якому перша група даних містить коефіцієнт, а друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних.

31. Пристрій за п. 29, в якому кодер містить компонент швидкості передачі бітів, сконфігурований так, щоб визначати призначення бітів на основі класифікації вмісту, при цьому компонент кодування додатково сконфігурований так, щоб кодувати вибрані мультимедійні дані за допомогою призначення бітів.

32. Пристрій за п. 29, в якому класифікація вмісту включає визначення складності мультимедійних даних, при цьому вибрані мультимедійні дані кодуються на основі складності мультимедійних даних.

33. Пристрій за п. 32, в якому складність містить часову складність або просторову складність.

34. Пристрій за п. 32, в якому складність містить часову складність і просторову складність.

35. Пристрій за п. 29, в якому кодування включає кодування мультимедійних даних так, щоб забезпечити декодування тільки першої групи даних або декодування першої групи даних і другої групи даних в одну комбіновану групу даних.

36. Пристрій за п. 30, в якому перше диференціальне уточнення вказує різницю між вибраним відеокadreом і даними кадру, що впливають з декодування першої групи даних.

37. Пристрій за п. 29, в якому перша група даних - це базовий рівень, а друга група даних - це рівень покращення.

38. Машиночитаний носій, що містить інструкції, які при виконанні інструктують машини:

- класифікувати вміст мультимедійних даних;
- секціонувати мультимедійні дані в першу групу даних і другу групу даних, основуючись на класифікації вмісту, при цьому секціонування визначається відповідно до співвідношення смуги пропускання; і

- кодувати мультимедійні дані в першій групі даних і в другій групі даних на основі класифікації вказаного секціонування, при цьому співвідношення кількості даних у першій групі даних і кількості даних у другій групі даних відповідає співвідношенню смуги пропускання.

39. Машиночитаний носій за п. 38, в якому перша група даних містить коефіцієнт і друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних.

40. Машиночитаний носій за п. 38, в якому інструкції, щоб кодувати, містять інструкції, щоб визначати призначення бітів на основі класифікації вмісту, при цьому компонент кодування додатково сконфігурований так, щоб кодувати вибрані мультимедійні дані за допомогою призначення бітів.

41. Машиночитаний носій за п. 38, в якому класифікація вмісту включає визначення складності мультимедійних даних, при цьому вибрані мультимедійні дані кодуються на основі складності мультимедійних даних.

42. Машиночитані інструкції за п. 41, в яких складність містить часову складність або просторову складність.

43. Машиночитані інструкції за п. 41, в яких складність містить часову складність і просторову складність.

44. Процесор, сконфігурований так, щоб:

- класифікувати вміст мультимедійних даних;
- секціонувати мультимедійні дані у першу групу даних і другу групу даних, основуючись на класифікації вмісту, при цьому секціонування визначається відповідно до співвідношення смуги пропускання; і

- кодувати мультимедійні дані в першій групі даних і в другій групі даних на основі класифікації вказаного секціонування, при цьому співвідношення кількості даних у першій групі даних і кількості даних у другій групі даних відповідає співвідношенню смуги пропускання.

45. Процесор за п. 44, в якому перша група даних містить коефіцієнт і друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних.

46. Процесор за п. 44, при цьому процесор додатково сконфігурований так, щоб визначати призначення бітів на основі класифікації вмісту, при цьому компонент кодування додатково сконфігурований так, щоб кодувати вибрані мультимедійні дані за допомогою призначення бітів.

47. Процесор за п. 44, при цьому процесор додатково сконфігурований так, щоб визначати складність мультимедійних даних, і при цьому класифікація вмісту основана на складності мультимедійних даних.

48. Процесор за п. 47, в якому складність містить часову складність або просторову складність.

49. Процесор за п. 47, в якому складність містить часову складність і просторову складність.

#### Заява про перевагу 35 U.S.C. §119

Дана Патентна заявка вимагає пріоритет (а) Попередньої патентної заявки номер 60/721416, озаглавленої "A VIDEO TRANSCODER FOR REAL-TIME STREAMING AND MOBILE BROADCAST APPLICATIONS", зареєстрованої 27 вересня 2005 року, (b) Попередньої патентної заявки номер 60/789377, озаглавленої "A VIDEO TRANSCODER FOR REAL-TIME STREAMING AND MOBILE BROADCAST APPLICATIONS", зареєстрованої 4 квітня 2006 року, (c) Попередньої заявки номер 60/727643, озаглавленої "METHOD AND APPARATUS FOR SPATIO-TEMPORAL DEINTERLACING AIDED BY MOTION COMPENSATION FOR FIELD-BASED VIDEO", зареєстрованої 17 жовтня 2005 року, (d) Попередньої заявки номер 60/727644, озаглавленої

"METHOD AND APPARATUS FOR SHOT DETECTION IN VIDEO STREAMING", зареєстрованої 17 жовтня 2005 року, (e) Попередньої заявки номер 60/727640, озаглавленої "A METHOD AND APPARATUS FOR USING AN ADAPTIVE GOP STRUCTURE IN VIDEO STREAMING", зареєстрованої 17 жовтня 2005 року, (f) Попередньої заявки номер 60/730145, озаглавленої "INVERSE TELECINE ALGORITHM BASED ON STATE MACHINE", зареєстрованої 24 жовтня 2005 року, і (g) Попередньої заявки номер 60/789048, озаглавленої "SPATIO-TEMPORAL DEINTERLACING AIDED BY MOTION COMPENSATION FOR FIELD-BASED MULTIMEDIA DATA", зареєстрованої 3 квітня 2006 року. Всі сім цих попередніх патентних заявок призначені їх правонаступнику і тим самим

повністю містяться в даному документі по посиланню.

Посилання на Заявки на патент, що одночасно знаходяться на розгляді

Дана Заявка на патент пов'язана з Патентною заявкою (США) № 11/373577, озаглавленою "CONTENT CLASSIFICATION FOR MULTIMEDIA PROCESSING", зареєстрованою 10 березня 2006 року, призначеною правонаступнику цієї заявки і таким чином явно міститься в даному документі по посиланню.

Рівень техніки

Галузь техніки, до якої належить винахід

Дана заявка направлена на пристрої і способи перекодування відео даних для потокової передачі в реальному часі, а більш конкретно, до перекодування відеоданих для потокової передачі в реальному часі в мобільному ширококомовному додатку.

Рівень техніки

Ефективне відеостиснення використовується у множині мультимедійних додатків, таких як потокова передача відео і відеотелефонія, внаслідок обмежених ресурсів смуги пропускання і непостійності доступної смуги пропускання. Конкретні стандарти кодування відео, такі як MPEG-4 (ISO/IEC), H.264 (ITU) або аналогічне кодування відео, надають високоефективне кодування, оптимально придатне для таких додатків, як безпроводна ширококомовна передача. Деякі мультимедійні дані, наприклад, цифрові телевізійні представлення, як правило, кодуються згідно з іншими стандартами, такими як MPEG-2. Відповідно, транскодери використовуються для того, щоб перекодувати або перетворювати мультимедійні дані, кодовані згідно з одним стандартом (наприклад, MPEG-2), в інший стандарт (наприклад, H.264) до безпроводної ширококомовної передачі.

Вдосконалені оптимізовані по швидкості кодеки можуть надавати переваги по стійкості до помилок, відновленню після помилок і масштабованості. Більше того, застосування інформації, визначеної з самих мультимедійних даних, також може надавати додаткові удосконалення по кодуванню, в тому числі стійкість до помилок, відновлення після помилок і масштабованість. Відповідно, існує потреба в транскодері, який надає високоефективну обробку і стиснення мультимедійних даних, який використовує інформацію, визначену з самих мультимедійних даних, і є масштабованим і стійким до помилок для застосування у множині додатків мультимедійних даних, в тому числі мобільній ширококомовній передачі потокової мультимедійної інформації.

Суть винаходу

Кожний з пристроїв і способів, що винаходяться, заснованого на вмісті перекодування, описаних і проілюстрованих, має декілька аспектів, жоден з яких не відповідає виключно за необхідні атрибути. Не обмежуючи об'єм цього винаходу, далі будуть стисло обговорені його найбільш помітні ознаки. Після розгляду цього обговорення, а більш точно, після прочитання розділу, озаглавленого "Докладний опис винаходу", стане зрозуміло, як ознаки даного керованого вмістом перекодування надають удосконалення в пристрої і способи обробки мультимедійних даних.

Аспекти, що винаходяться, описані в даному документі, належать до використання інформації вмісту для різних способів кодування мультимедійних даних і в різних модулях і компонентах кодера, наприклад, кодера, використовуваного в транскодері. Транскодер може керувати перекодуванням мультимедійних даних за допомогою інформації вмісту. Інформація вмісту може прийматися з іншого джерела, наприклад, метаданих, які приймаються разом з відео. Транскодер може бути сконфігурований для того, щоб формувати інформацію вмісту за допомогою множини різних операцій обробки. У деяких аспектах транскодер формує класифікацію вмісту мультимедійних даних, яка потім використовується в одному або більше процесах кодування. У деяких аспектах керований вмістом транскодер може визначати просторову і часову інформацію вмісту мультимедійних даних і використовувати інформацію вмісту для кодування з однаковою якістю з урахуванням вмісту по каналах і заснованого на класифікації стиснення/призначення бітів.

У деяких аспектах інформація вмісту (наприклад, метадані, показники вмісту і/або класифікація вмісту) мультимедійних даних одержується і обчислюється, а потім надається компонентам транскодера для використання в обробці мультимедійних даних для кодування. Наприклад, пре-процесор може використовувати визначену інформацію вмісту для виявлення зміни сцен, перетворення з відеоформату в фільм (IVTC), усунення черезрядкової розгортки, компенсації руху і приглушення шумів (наприклад, двовимірного вейвлет-перетворення) і просторово-часового ослаблення шумів, наприклад, видалення дефектів зображення, приглушення реверберацій, розблокування і/або зниження шумності. У деяких аспектах препроцесор також може використовувати інформацію вмісту для знижувальної дискретизації просторового розрізнення, наприклад, визначення відповідних "захисних" зон і зон "керування рухомими об'єктами" при знижувальній дискретизації зі стандартного розрізнення (SD) до чверті логічної матриці відеографіки (QVGA).

У деяких аспектах кодер включає в себе модуль класифікації вмісту, який сконфігурований так, щоб обчислювати інформацію вмісту. Кодер може використовувати класифікацію вмісту для керування швидкістю передачі в бітах (наприклад, призначення бітів) при визначенні параметрів квантування для кожного MB, для оцінки руху, наприклад, виконання колірної оцінки руху (ME), виконання прогнозу векторів руху (MV), масштабованості при наданні базового рівня і рівня поліпшення, а також для стійкості до помилок за допомогою використання класифікації вмісту так, щоб впливати на схеми ієрархії прогнозу і стійкості до помилок, в тому числі, наприклад, процеси адаптивного внутрішнього оновлення, вирівнювання границь і надання надмірних даних I-кадру в рівні поліпшення. У деяких аспектах транскодер використовує класифікацію вмісту у взаємодії з мультиплексором даних для підтримання оптимальної якості мультимедійних даних в каналах. У деяких аспектах кодер може використовувати інформацію класифікації вмісту для примусо-

вої періодичної появи І-кадрів в кодованих даних, щоб забезпечити швидке перемикавання каналів. Ці реалізації також можуть застосовувати І-блоки, які можуть потребуватися в кодованих даних для стійкості до помилок, з тим щоб перемикавання з довільним доступом і стійкість до помилок (на основі, наприклад, класифікації вмісту) могли ефективно комбінуватися за допомогою ієрархії прогнозу, щоб підвищити ефективність кодування із збільшенням запасу стійкості по помилках.

У одному аспекті спосіб обробки мультимедійних даних включає класифікацію вмісту мультимедійних даних і кодування мультимедійних даних в першій групі даних і у другій групі даних на основі класифікації вмісту, причому перша група даних містить коефіцієнт, а друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних. Кодування може включати в себе визначення швидкості передачі бітів на основі класифікації вмісту мультимедійних даних і кодування мультимедійних даних на основі швидкості передачі бітів. Класифікація вмісту може включати визначення складності мультимедійних даних, і при цьому вибрані мультимедійні дані кодуються на основі складності мультимедійних даних. Складність може містити часову складність або просторову складність, або часову складність і просторову складність. Кодування може включати в себе кодування мультимедійних даних так, щоб забезпечити декодування тільки першої групи даних або першої групи даних і другої групи даних в одну комбіновану групу даних. Перше диференціальне уточнення може вказувати різницю між вибраним відеокadreм і даними кадру, що впливають з декодування першої групи даних. Перша група даних може бути базовим рівнем, а друга група даних може бути рівнем поліпшення. Крім цього, спосіб може включати в себе вибір коефіцієнта з одного з коефіцієнта залишкової помилки вихідного базового рівня і коефіцієнта залишкової помилки вихідного рівня поліпшення і обчислення першого диференціального уточнення на основі коефіцієнта і коефіцієнта залишкової помилки вихідного рівня поліпшення. Кодування додатково може включати кодування інформації заголовків макроблоків і інформації вектора руху в першій групі даних. Кодування додатково може включати квантування першої групи даних з першим розміром кроку і квантування другої групи даних з другим розміром кроку, при цьому перший розмір кроку і другий розмір кроку співвідносяться за допомогою коефіцієнта масштабування. Кодування додатково може включати в себе визначення першого параметра квантування, що має перший розмір кроку квантування, для використання при кодуванні першої групи даних і визначення другого параметра квантування, що має другий розмір кроку квантування, для використання при кодуванні другої групи даних, при цьому перший і другий параметри квантування визначаються на основі інформації вмісту вибраних даних кадру і при цьому згаданий перший розмір кроку квантування більш грубий, ніж згаданий другий розмір кроку квантування. У іншому аспекті кодування включає в себе кодування першої групи даних за допомогою І-кадрів і Р-кадрів або будь-

якої їх комбінації і кодування другої групи даних за допомогою І-кадрів, Р-кадрів і В-кадрів або будь-якої їх комбінації.

У іншому аспекті пристрій кодування мультимедійних даних включає в себе засіб класифікації вмісту мультимедійних даних, засіб кодування мультимедійних даних в першій групі даних і у другій групі даних на основі класифікації вмісту, причому перша група даних містить коефіцієнт, а друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних. Засіб кодування може містити засіб для визначення швидкості передачі бітів на основі класифікації вмісту мультимедійних даних і кодування мультимедійних даних на основі швидкості передачі бітів. Засіб класифікації вмісту може включати в себе засіб визначення складності мультимедійних даних, і при цьому вибрані мультимедійні дані кодуються на основі складності мультимедійних даних, причому складність містить часову складність або просторову складність або часову складність і просторову складність. Засіб кодування може містити засіб для того, щоб забезпечити декодування тільки першої групи даних або першої групи даних і другої групи даних в одну комбіновану групу даних.

У іншому аспекті пристрій включає в себе модуль класифікації вмісту мультимедійних даних, сконфігурований так, щоб класифікувати вміст мультимедійних даних і надавати дані класифікації вмісту, і кодер, сконфігурований так, щоб кодувати мультимедійні дані в першій групі даних і у другій групі даних на основі класифікації вмісту, причому перша група даних містить коефіцієнт, а друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних. Кодер може включати в себе компонент швидкості передачі бітів, сконфігурований так, щоб визначати призначення бітів на основі класифікації вмісту, при цьому компонент кодування додатково сконфігурований так, щоб кодувати вибрані мультимедійні дані за допомогою призначення бітів.

У іншому аспекті машиночитаний носій містить інструкції, які при виконанні інструктують машині класифікувати вміст мультимедійних даних і кодувати мультимедійні дані в першій групі даних і у другій групі даних на основі класифікації вмісту, причому перша група даних містить коефіцієнт, а друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних.

У іншому аспекті процесор сконфігурований так, щоб класифікувати вміст мультимедійних даних і кодувати мультимедійні дані в першій групі даних і у другій групі даних на основі класифікації вмісту, причому перша група даних містить коефіцієнт, а друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних.

Короткий опис креслень

Фіг. 1А - це блок-схема мультимедійної ширококомовної системи, яка включає в себе транскодер для перекодування між різними відеоформатами.

Фіг. 1В - це блок-схема кодера, сконфігурованого так, щоб кодувати мультимедійні дані і нада-

вати кодовану першу групу даних і кодовану другу групу даних.

Фіг. 1С - це блок-схема процесора, сконфігурованого так, щоб кодувати мультимедійні дані.

Фіг. 2 - це блок-схема прикладу транскодера системи за Фіг. 1.

Фіг. 3 - це блок-схема послідовності операцій способу, яка ілюструє роботу аналізатора, що використовується в транскодері за Фіг. 2.

Фіг. 4 - це блок-схема послідовності операцій способу, яка ілюструє роботу декодера, що використовується в транскодері за Фіг. 2.

Фіг. 5 - це системна часова схема, яка ілюструє послідовність операцій, що виконуються за допомогою транскодера за Фіг. 2.

Фіг. 6 - це блок-схема, яка ілюструє послідовність операцій і функцій препроцесора, які можуть бути використані в транскодері за Фіг. 2.

Фіг. 7 - це блок-схема зразкового двохпрохідного кодера, який може бути використаний в транскодері за Фіг. 2.

Фіг. 8 ілюструє приклад класифікаційної діаграми, яка ілюструє один аспект того, як асоціативно зв'язувати значення текстури і руху з класифікацією вмісту.

Фіг. 9 - це блок-схема послідовності операцій способу, яка ілюструє зразкову операцію класифікації вмісту, наприклад, для використання в кодері за Фіг. 7.

Фіг. 10 - це блок-схема послідовності операцій способу, яка ілюструє операцію керування швидкості, наприклад, для використання в кодері за Фіг. 7.

Фіг. 11 - це блок-схема послідовності операцій способу, яка ілюструє роботу зразкового блока оцінки руху, наприклад, для використання в кодері за Фіг. 7.

Фіг. 12 - це блок-схема послідовності операцій способу, яка ілюструє роботу зразкової функції кодера для визначення режиму, наприклад, для використання в кодері за Фіг. 7.

Фіг. 13 - це блок-схема послідовності операцій способу, яка ілюструє зразкову операцію здійснення масштабованості для використання в кодері за Фіг. 7.

Фіг. 14 - це блок-схема послідовності операцій способу, яка ілюструє зразкову операцію здійснення потоку даних зі спотворенням залежно від швидкості, як відбувається, наприклад, в кодері за Фіг. 7.

Фіг. 15 - це графік, який ілюструє взаємозв'язок між складністю кодування, призначеними бітами і якістю візуального сприйняття людиною.

Фіг. 16 - це графік, який ілюструє нелінійну схему виявлення сцен.

Фіг. 17А - це блок-схема послідовності операцій способу, яка ілюструє обробку мультимедійних даних, які одержані, прийняті або іншим чином доступні.

Фіг. 17В - це блок-схема системи кодування мультимедіа.

Фіг. 18 - це схема, яка ілюструє процес усунення черезрядкової розгортки за допомогою оцінки/компенсації руху.

Фіг. 19 - це блок-схема мультимедійної системи зв'язку.

Фіг. 20 - це схема, яка ілюструє організацію відеопотоку бітів в рівні поліпшення і базовому рівні.

Фіг. 21 - це схема, яка ілюструє поєднання серій макроблоків з границями відеокадрів.

Фіг. 22 - це блок-схема, яка ілюструє ієрархію прогнозу.

Фіг. 23 - це блок-схема послідовності операцій, яка ілюструє спосіб кодування мультимедійних даних на основі інформації вмісту.

Фіг. 24 - це блок-схема послідовності операцій, яка ілюструє спосіб кодування мультимедійних даних, так щоб суміщати границі даних на основі рівня інформації вмісту.

Фіг. 25 - це графік, який ілюструє захищену область рухомих об'єктів і захищену область тайтлів кадру даних.

Фіг. 26 - це графік, який ілюструє захищену область рухомих об'єктів кадру даних.

Фіг. 27 - це блок-схема послідовності операцій, яка ілюструє процес кодування мультимедійних даних за допомогою адаптивного внутрішнього оновлення на основі інформації мультимедійного вмісту.

Фіг. 28 - це блок-схема послідовності операцій, яка ілюструє процес кодування мультимедійних даних за допомогою надмірних I-кадрів на основі інформації мультимедійного вмісту.

Фіг. 29 ілюструє вектори компенсації руху між поточним кадром і попереднім кадром  $MV_P$  і поточним кадром і наступним кадром  $MV_N$ .

Фіг. 30 - це блок-схема послідовності операцій способу, яка ілюструє виявлення кадрів.

Фіг. 31 - це блок-схема послідовності операцій способу, яка ілюструє кодування базового рівня і рівня поліпшення.

Фіг. 32 - це схематичне представлення, яке ілюструє кодування макроблока.

Фіг. 33 - це схематичне представлення, яке ілюструє модулі для кодування базового рівня і рівня поліпшення.

Фіг. 34 ілюструє приклад процесу в блоці вибору коефіцієнтів базового рівня і рівня поліпшення.

Фіг. 35 ілюструє ще один приклад процесу в блоці вибору коефіцієнтів базового рівня і рівня поліпшення.

Фіг. 36 ілюструє ще один приклад процесу в блоці вибору коефіцієнтів базового рівня і рівня поліпшення.

Фіг. 37 - це блок-схема послідовності операцій способу, яка ілюструє кодування мультимедійних даних на основі інформації вмісту.

Фіг. 38 - це схема, яка ілюструє можливі системні рішення в процесі перетворення з відеоформату в фільм.

Фіг. 39 ілюструє границі в макроблоці, які повинні бути відфільтровані за допомогою процесу розблокування.

Фіг. 40 - це схема, яка ілюструє процес просторово-часового усунення черезрядкової розгортки (деінтерлейсингу).

Фіг. 41 ілюструє приклад одновимірної монофазної повторної дискретизації.

Фіг. 42 - це блок-схема послідовності операцій способу, яка ілюструє приклад адаптивної GOP-структури в поточній передачі відео.

Потрібно зазначити, що аналогічні номери посилань означають аналогічні деталі по всіх представленнях на кресленнях відповідним чином.

#### Докладний опис винаходу

Подальший докладний опис направлений на визначені аспекти, що описуються в даній заявці. Проте, винахід може бути здійснений великою кількістю різних способів. Посилання в цьому докладному описі на "один аспект" або "асpekt" означає, що конкретна ознака, конструкція або характеристика, що описуються в зв'язку з аспектом, включені щонайменше в один аспект. Фрази "в одному аспекти", "згідно з одним аспектом" або "в деяких аспектах" в різних місцях даного докладного опису не обов'язково всі вказують на той же аспект, а також не є окремими або альтернативними аспектами, що взаємовключають інші аспекти. Більше того, описані різні ознаки, які можуть бути продемонстровані за допомогою деяких аспектів, але не за допомогою інших. Аналогічно, описані різні вимоги, які можуть бути вимогами для деяких аспектів, але не для інших аспектів.

Подальший опис включає в себе докладні відомості для того, щоб надати повне розуміння прикладів. Проте, фахівцям в даній галузі техніки потрібно розуміти, що приклади можуть бути використані на практиці, навіть якщо всі подробиці процесу або пристрою в прикладі або аспекті не пояснені або проілюстровані в даному документі. Наприклад, на блок-схемах можуть бути показані електричні компоненти, які не ілюструють всі електричні з'єднання або всі електричні елементи компонента, щоб не відволікати від розуміння прикладів непотрібними подробицями. У інших випадках, такі компоненти, інші структури і методики можуть бути детально показані, щоб додатково пояснити приклади.

Даний пристрій належить до пристроїв і способів керування кодуванням і перекодуванням за допомогою інформації вмісту кодованих мультимедійних даних. "Інформація вмісту" або "вміст" (мультимедійних даних) - це широкі терміни, які означають інформацію, пов'язану з вмістом мультимедійних даних, і можуть включати в себе, наприклад, метадані, показники, обчислені з мультимедійних даних, і інформацію, що належить до вмісту, асоціативно зв'язану з одним або більше показниками, наприклад, класифікацію вмісту. Інформація вмісту може бути надана в кодер або визначена за допомогою кодера, залежно від конкретного варіанта застосування. Інформація вмісту може бути використана для багатьох аспектів кодування мультимедійних даних, в тому числі виявлення зміни сцен, часової обробки, просторово-часового зниження шумів, знижувальної дискретизації, визначення швидкостей передачі бітів для квантування, масштабування стійкості до помилок, збереження оптимальної якості мультимедіа в ширококомовних каналах і швидкого перемикавання каналів. За допомогою одного або більше цих аспектів транскодер може керувати обробкою мультимедійних даних і формувати пов'язані з вмістом кодовані мультимедійні дані. Описи і креслення в даному документі, які описують аспекти перекодування, також можуть бути застосовані до аспектів кодування і аспектів декодування.

Пристрої і способи транскодера пов'язані з перекодуванням з одного формату в інший і детально описуються в даному документі як пов'язане з перекодуванням відео MPEG-2 в поліпшений масштабований формат H.264 для передачі по безпроводних каналах в мобільні пристрої, що ілюструють деякі аспекти. Проте, опис перекодування відео MPEG-2 в формат H.264 не виступає як обмеження галузі застосування винаходу, а є просто прикладом деяких аспектів винаходу. Розкриті пристрої і способи надають високоефективну архітектуру, яка підтримує стійке до помилок кодування за допомогою можливостей довільного доступу і розбивання на рівні і також може бути застосована до перекодування і/або кодування відеоформатів, відрізняючись від MPEG-2 і H.264.

"Мультимедійні дані" або просто "мультимедіа" при використанні в даному документі є широким терміном, який включає в себе відеодані (які можуть включати в себе аудіодані), аудіодані або і відеодані, і аудіодані. "Відеодані" або "відео" при використанні в даному документі є широким терміном, що належить до заснованих на кадрах або заснованих на полях даних, які включають в себе одне або більше зображення або пов'язані послідовності зображень, що містять текст, інформацію зображень і/або аудіоданих, і можуть бути використані для того, щоб означати мультимедійні дані (наприклад, терміни можуть бути використані взаємозамінно), якщо не вказано інше.

Нижче описуються приклади різних компонентів транскодера і приклади процесів, які можуть використовувати інформацію вмісту для кодування мультимедійних даних.

Фіг. 1A - це блок-схема, що ілюструє потік даних деяких аспектів системи 100 ширококомовної передачі мультимедійних даних. У системі 100 постачальник 106 мультимедійних даних передає кодовані мультимедійні дані 104 в транскодер 200. Кодовані мультимедійні дані 104 приймаються за допомогою транскодера 200, який обробляє мультимедійні дані 104 в "сирі" необроблені мультимедійні дані на етапі 110. Обробка на блоці 110 декодує і розбирає кодовані мультимедійні дані 104 і додатково обробляє мультимедійні дані, щоб підготувати їх до кодування в інший формат. Декодовані мультимедійні дані надаються на етап 112, де мультимедійні дані кодуються в попередньо визначений мультимедійний формат або стандарт. Після того як мультимедійні дані кодовані, на етапі 114 вони підготовлюються для передачі, наприклад, за допомогою безпроводної ширококомовної системи (наприклад, стільникової телефонної ширококомовної мережі або за допомогою іншої мережі зв'язку). У деяких аспектах прийняті мультимедійні дані 104 кодовані згідно зі стандартом MPEG-2. Після того, як перекодовані мультимедійні дані 104 декодовані, транскодер 200 кодує мультимедійні дані в стандарт H.264.

Фіг. 1B - це блок-схема транскодера 130, який може бути сконфігурований так, щоб виконувати обробку на етапах 110 і 112 Фіг. 1A. Транскодер 130 може бути сконфігурований так, щоб приймати мультимедійні дані, декодувати і розбирати мультимедійні дані на пакетовані елементарні потоки (наприклад, субтитри, аудіо, метадані, "необроб-

лене" відео, СС-дані і часові мітки представлення), кодувати їх в необхідний формат і надавати кодовані дані для додаткової обробки або передачі. Транскодер 130 може бути сконфігурований так, щоб надавати кодовані дані в дві або більше груп даних, наприклад, кодовану першу групу даних і кодовану другу групу даних, що згадується як багаторівневе кодування. У деяких прикладах аспектів різні групи даних (або рівні) в схемі багаторівневого кодування можуть кодуватися на різних рівнях якості і формуватися так, що дані, кодовані в першій групі даних, мають гіршу якість (наприклад, надають менший рівень візуальної якості при відображенні), ніж дані, кодовані у другій групі даних.

Фіг. 1С - це блок-схема процесора 140, який може бути сконфігурований так, щоб перекодувати мультимедійні дані, і може бути сконфігурований так, щоб виконувати частину або всю обробку, проілюстровану на етапах 110 і 112 за Фіг. 1А. Процесор 140 може включати в себе модулі 124а..п, виконувати один або більше процесів перекодування, описаних в даному документі, в тому числі декодування, розбір, попередню обробку і кодування, і використовувати інформацію вмісту для обробки. Процесор 140 також включає в себе внутрішній запам'ятовуючий пристрій 122 і може бути сконфігурований так, щоб обмінюватися даними із зовнішнім запам'ятовуючим пристроєм 120 напряму або непрямо за допомогою іншого пристрою. Процесор 140 також включає в себе модуль 126 зв'язку, сконфігурований так, щоб обмінюватися даними з одним або більш пристроями, зовнішніми для процесора 140, в тому числі приймати мультимедійні дані і надавати кодовані дані, такі як дані, кодовані в першій групі даних, і дані, кодовані у другій групі даних. У деяких прикладах аспектів різні групи даних (або рівні) в схемі багаторівневого кодування можуть кодуватися на різних рівнях якості і формуватися так, що дані, кодовані в першій групі даних, мають гіршу якість (наприклад, надають менший рівень візуальної якості при відображенні), ніж дані, кодовані у другій групі даних.

Транскодер 130 або його компоненти препроцесора 140 (сконфігуровані для перекодування), а також процеси, що містяться в ньому, можуть бути реалізовані за допомогою апаратних засобів, програмного забезпечення, мікропрограмного забезпечення, проміжного програмного забезпечення, мікрокоду або комбінації вищезазначеного. Наприклад, аналізатор, декодер, препроцесор або кодер можуть бути автономними компонентами, включеними як апаратні засоби, мікропрограмне забезпечення, проміжне програмне забезпечення в компонент іншого пристрою, або можуть бути реалізовані в мікрокоді або програмному забезпеченні, яке приводиться у виконання в процесорі, або як комбінація вищезазначеного. Коли реалізований в програмному забезпеченні, мікропрограмному забезпеченні, проміжному програмному забезпеченні або мікрокоді, програмний код або сегменти коду, які виконують процеси компенсації руху, класифікації кадрів і кодування, можуть бути збережені на машиночитаному носії, такому як носій зберігання. Сегмент коду може представляти процедуру, функцію, підпрограму, програму, стан-

дартну процедуру, вкладену процедуру, модуль, комплект програмного забезпечення, клас або будь-яке поєднання інструкцій, структур даних або операторів програми. Сегмент коду може бути пов'язаний з іншим сегментом коду або апаратною схемою за допомогою передачі і/або прийому інформації, даних, аргументів, параметрів або вмісту пам'яті.

Ілюстративний приклад архітектури транскодера

Фіг. 2 ілюструє блок-схему прикладу транскодера, який може бути використаний для транскодера 200, проілюстрованого в мультимедійній ширококомовній системі 100 за Фіг. 1. Транскодер 200 містить аналізатор/декодер 202, препроцесор 226, кодер 228 і синхронізуючий рівень 240, додатково описані нижче. Транскодер 200 сконфігурований так, щоб використовувати інформацію вмісту мультимедійних даних 104 для одного або більше аспектів процесу перекодування, як описано в даному документі. Інформація вмісту може бути одержана з джерела, зовнішнього для транскодера 200, через мультимедійні метадані або обчислена за допомогою транскодера, наприклад, за допомогою препроцесора 226 або кодера 228. Компоненти, показані на Фіг. 2, ілюструють компоненти, які можуть бути включені в транскодер, який використовує інформацію вмісту для одного або більше процесів перекодування. У конкретній реалізації один або більше компонентів транскодера 200 можуть бути виключені, або можуть бути включені додаткові компоненти. Додатково, частини транскодера і процесів перекодування описані для того, щоб надати можливість фахівцям в даній галузі техніки використати на практиці винахід, навіть якщо всі подробиці процесу або пристрою можуть не описуватися в даному документі.

Фіг. 5 ілюструє часову схему як графічну ілюстрацію часового взаємозв'язку роботи різних компонентів і/або процесів транскодера 200. Як показано на Фіг. 5, кодоване потокове відео 104 (кодовані мультимедійні дані), наприклад, відео MPEG-2, спочатку приймається в довільний нульовий момент часу (0) за допомогою аналізатора 205 (Фіг. 2). Далі відеопотік розбирається 501, демультіплексується 502 і декодується 503, наприклад, за допомогою аналізатора 205 в комбінації з декодером 214. Як проілюстровано, ці процеси можуть виконуватися паралельно з невеликим зміщенням за часом, щоб надати потокове виведення даних обробки в препроцесор 226 (Фіг. 2). У момент часу  $T_1$  504, після того як препроцесор 226 прийняв достатньо даних від декодера 214 для того, щоб почати виведення результатів обробки, етапи обробки, що залишилися, стають послідовними за характером, при цьому кодування 505 першого проходу, кодування 506 другого проходу і повторне кодування 507 виконуються послідовно після попередньої обробки до завершення повторного кодування в момент часу  $T_2$  508.

Транскодер 200, описаний в даному документі, може бути сконфігурований так, щоб перекодувати різні мультимедійні дані, і багато які з процесів застосовуються до того типу мультимедійних даних, який перекодується. Хоч деякі з прикладів, що надаються в даному документі, належать



до перекодування даних MPEG-2 в дані H.264, ці приклади не призначені для того, щоб обмежувати винахід цими даними. Аспекти кодування, що описуються нижче, можуть бути застосовані до перекодування будь-якого відповідного стандарту мультимедійних даних в інший відповідний стандарт мультимедійних даних.

#### Аналізатор/декодер

Знов посилаючись на Фіг. 2, аналізатор/декодер 202 приймає мультимедійні дані 104. Аналізатор/декодер 202 включає в себе аналізатор транспортних потоків ("аналізатор") 205, який приймає мультимедійні дані 104 і розбирає дані на елементарний потік (ES) відео 206, ES 208 аудіо, часові мітки представлення (PTS) 210 і інші дані, такі як субтитри 212. ES переносить один тип даних (відео або аудіо) з одного відео- або аудіокодера. Наприклад, ES відео містить відеодані для послідовності даних, що включають в себе заголовки послідовності і всі підчастини послідовності. Пакетований елементарний потік, або PES, складається з одного ES, який складений в пакети, кожний з яких в типовому варіанті починається з доданого заголовка пакета. PES-потік містить тільки один тип даних з одного джерела, наприклад, з одного відео- або аудіокодера. PES-пакети мають змінну довжину, не відповідну фіксованій довжині транспортних пакетів, і можуть бути набагато довшими, ніж транспортний пакет. Коли транспортні пакети сформовані з PES-потіку, PES-заголовки може бути поміщений в початок робочих даних транспортного пакета, відразу після заголовка транспортного пакета. Вміст PES-пакета, що залишився, заповнює робочі дані послідовних транспортних пакетів, поки PES-пакет не буде повністю використаний. Кінцевий транспортний пакет може бути заповнений до фіксованої довжини, наприклад, за допомогою доповнення байтами, наприклад, байтами = 0xFF (всі одиниці).

Аналізатор 205 передає ES 206 відео в декодер 214, який є частиною аналізатора/декодера 202, показаною тут. У інших конфігураціях аналізатор 205 і декодер 214 є окремими компонентами. PTS 210 відправляються в PTS-формуваць 215 транскодера, який може формувати окремі часові мітки представлення, конкретні для транскодера 200, для використання при компонуванні даних, які повинні бути відправлені з транскодера 200 в широкомовну систему. PTS-формуваць 215 транскодера може бути сконфігурований так, щоб надавати дані в синхронізуючий рівень 240 транскодера 200, щоб координувати синхронізацію широкомовної передачі даних.

Фіг. 3 ілюструє блок-схему послідовності операцій одного прикладу процесу 300, якого аналізатор 205 може додержуватися при розборі різних пакетованих елементарних потоків, описаних вище. Процес 300 починається на етапі 302, коли мультимедійні дані 104 приймаються від постачальника 106 вмісту (Фіг. 1). Процес 300 переходить до етапу 304, на якому виконується ініціалізація аналізатора 205. Ініціалізація може активуватися за допомогою незалежно сформованої команди 306 збирання даних. Наприклад, процес, який незалежний від аналізатора 205 і заснований на ТВ-розкладі, що приймається зовні, і інформації на-

стройки каналів, може формувати команду 306 збирання даних. Додатково, можуть бути введені дескриптори 308 буфера транспортного потоку (TS) реального часу, щоб допомагати в ініціалізації і для основної обробки.

Як проілюстровано на етапі 304, ініціалізація може включати в себе запит перевірки синтаксису команд, виконання обробки PSI/PSIP/SI першого проходу (конкретна для програми інформації/програма і протокол системної інформації/системна інформація), виконання обробки, конкретно пов'язаної або з командою збирання даних, або з перевіркою узгодженості PSI/PSIP/SI, призначення PES-буферів для кожного PES і задавання розподілу за часом (наприклад, для поєднання з необхідним моментом початку збирання даних). PES-буфери містять розібрані ES-дані і передають всі розібрані ES-дані у відповідні аудіодекодер 216, тестовий декодер 220, декодер 214 або PTS-формуваць 215 транскодера.

Після ініціалізації процес 300 переходить до етапу 310 для основної обробки мультимедійних даних 104, що приймаються. Обробка на етапі 310 може включати в себе фільтрацію цільових ідентифікаторів пакетів (PID), безперервний моніторинг і обробку PSI/PSIP/SI і процес розподілу за часом (наприклад, для досягнення необхідної тривалості збирання даних), з тим щоб мультимедійні дані, що надходять, передавалися у відповідні PES-буфери. Як результат обробки мультимедійних даних на етапі 310 формуються програмний дескриптор і індикатор PES-буфера "read", які взаємодіють з декодером 214 (Фіг. 2) так, як описано нижче.

Після етапу 310 процес 300 переходить до етапу 314, на якому виконується завершення операцій розбору, формування переривання таймера і звільнення PES-буферів після їх використання. Потрібно зазначити, що PES-буфери є для всіх релевантних елементарних потоків програми, згаданих в дескрипторі, наприклад, потоків аудіо, відео і субтитрів.

Знов посилаючись на Фіг. 2, аналізатор 205 відправляє ES 208 аудіо в аудіодекодер 216 для відповідності реалізації транскодера і надає кодований текст 216 в синхронізуючий рівень 240 і декодує аудіоінформацію. Інформація 212 субтитрів доставляється в текстовий декодер 220. Дані 218 кодованих субтитрів між кадрами (CC) з декодера 214 також надаються в текстовий кодер 220, який кодує інформацію 212 субтитрів і CC-дані 218 в форматі, що реалізується за допомогою транскодера 200.

Аналізатор/декодер 202 також включає в себе декодер 214, який приймає ES 206 відео. Декодер 214 може формувати метадані, асоціативно зв'язані з відеоданими, декодує кодований пакетований елементарний відеопотік в необроблене відео 224 (наприклад, в форматі стандартного розрізнення) і обробляє відеодані кодованих субтитрів між кадрами в ES-потік відео.

Фіг. 4 ілюструє блок-схему послідовності операцій, яка показує один приклад процесу 400 декодування, який може бути виконаний за допомогою декодера 214. Процес 400 починається з введення даних 206 елементарного відеопотоку на

етапі 402. Процес 400 переходить до етапу 404, на якому декодер ініціалізується. Ініціалізація може включати в себе ряд задач, в тому числі виявлення заголовка відеопослідовності (VSH), виконання VSH першого проходу, обробку відеопослідовності (VS) і дисплейного розширення VS (включаючи відеоформат, складові кольори і матричні коефіцієнти) і призначення буферів даних так, щоб належним чином буферизувати декодоване зображення, асоціативно зв'язані метадані і дані кодованих субтитрів між кадрами (CC). Додатково, інформація 406 "read" PES-буфера відео, що надається за допомогою аналізатора 205, вводиться (наприклад, яка може бути сформована за допомогою процесу 300 на етапі 310 за Фіг. 3).

Після ініціалізації на етапі 404 процес 400 переходить до етапу 408, на якому основна обробка ES відео виконується за допомогою декодера 214. Основна обробка включає в себе запитування інформації "read" PES-буфера відео або "інтерфейсу" для доступності нових даних, декодування ES відео, відновлення і збереження піксельних даних при синхронізації границь зображень, метаданих формування відео і аудіо-відео і збереження на границях зображень, і збереження CC-даних на границях зображень. Етап 410 результатів основної обробки 408 включає в себе формування дескрипторів послідовності, дескрипторів буфера декодованих зображень, дескрипторів буферів метаданих і дескрипторів буферів CC-даних.

Після основної обробки 408 процес 400 переходить до етапу 412, на якому він виконує процес завершення. Процес завершення може включати в себе визначення умов завершення, в тому числі відсутність нових даних, що з'являються протягом конкретного часу вище попередньо визначеного порога, виявлення кінцевого коду послідовності і/або виявлення явного сигналу завершення. Процес завершення додатково може включати в себе звільнення декодованого зображення, асоціативно зв'язаних метаданих і буферів CC-даних після їх використання за допомогою препроцесора, який описується нижче. Процес 400 завершується на етапі 414, де він може перейти в стан очікування ES відео, який повинен бути прийнятий як введення.

#### Препроцесор

Фіг. 2, і детальніше за Фіг. 6 ілюструють зразковий аспект препроцесора 226, який може використовувати інформацію вмісту для однієї або більше операцій попередньої обробки. Препроцесор 226 приймає метадані 222 і декодовані "необроблені" відеодані 224 від аналізатора/декодера 202. Препроцесор 226 сконфігурований так, щоб виконувати визначені типи обробки для відеоданих 224 і метаданих 222 і надавати оброблене мультимедіа (наприклад, опорні кадри базового рівня, опорні кадри рівня поліпшення, інформацію смуги пропускання, інформацію вмісту) і відео в кодер 228. Така попередня обробка мультимедійних даних дозволяє підвищувати візуальну чіткість, пригнічення накладення спектра і ефективність стиснення даних. Загалом, препроцесор 226 приймає відеопослідовності, що надаються за допомогою декодера 214 в аналізаторі/декодері 202, і перетворює відеопослідовності в прогресивні відеопос-

лідовності для додаткової обробки (наприклад, кодування) за допомогою кодера 228. У деяких аспектах препроцесор 226 може бути сконфігурований для множини операцій, в тому числі операцій перетворення з відеоформату в фільм, усунення черезрядкової розгортки, фільтрації (наприклад, видалення дефектів зображення, пригнічення реверберацій, розблокування і зниження шумності), зміни розміру (наприклад, знижувальної дискретизації просторового розрізнення зі стандартного розрізнення до чверті логічної матриці відеографіки (QVGA)) і формування GOP-структури (наприклад, обчислення формування таблиці складності, виявлення зміни сцен і виявлення затінення/спалаху).

Препроцесор 226 може використовувати метадані від декодера для того, щоб виконувати одну або більше операцій попередньої обробки. Метадані можуть включати в себе інформацію, пов'язану, описуючу або класифікуючу вміст мультимедійних даних ("інформацію вмісту"); зокрема, метадані можуть включати в себе класифікацію вмісту. У деяких аспектах метадані не включають в себе інформацію вмісту, необхідну для операцій кодування. У цих випадках препроцесор 226 може бути сконфігурований так, щоб визначати інформацію вмісту і використовувати інформацію вмісту для операцій попередньої обробки і/або надавати інформацію вмісту в інші компоненти транскодера 200, наприклад, декодер 228. У деяких аспектах препроцесор 226 може використовувати цю інформацію вмісту для того, щоб впливати на секціонування GOP, визначати відповідний тип фільтрації і/або визначати параметри кодування, які передаються в кодер.

Фіг. 6 показує ілюстративний приклад різних етапів обробки, які можуть бути включені в препроцесор 226, і ілюструє обробку, яка може бути виконана за допомогою препроцесора 226. У цьому прикладі препроцесор 226 приймає метадані і відео 222, 224 і надає вихідні дані 614, що містять (оброблені) метадані і відео, в кодер 228. У типовому варіанті передбачено три типи відео, яке може прийматися. По-перше, відео, що приймається, може бути прогресивним відео, де не потрібне усунення черезрядкової розгортки. По-друге, відеодані можуть бути відео, перетвореним з фільму у відеоформат, черезрядковим відео, перетвореним з кінопослідовностей в 24 кадри за секунду, в цьому випадку відео. По-третє, відео може бути черезрядковим відео, не перетвореним з фільму у відеоформат. Препроцесор 226 може обробляти ці типи відео так, як описано нижче.

На етапі 601 препроцесор 226 визначає, чи є прийняті відеодані 222, 224 прогресивним відео. У деяких випадках це може бути визначено з метаданих, якщо метадані містять цю інформацію, або за допомогою обробки самих відеоданих. Наприклад, процес перетворення з відеоформату в фільм, описаний нижче, дозволяє визначати, чи є відео 222, що приймається, прогресивним відео. Якщо є, процес переходить до етапу 607, на якому операції фільтрації (наприклад, зниження шумності) виконуються для відео, щоб зменшити шум, такий як білий гаусів шум. Якщо відеодані 222, 224 не є прогресивним відео, на етапі 601 процес пе-

реходить до етапу 604 до модуля 604 виявлення фаз.

Модуль 604 виявлення фаз розрізняє відео, яке спочатку перетворене з фільму у відеоформат, і яке починалося в стандартному широкоформатному форматі. Якщо прийняте рішення про те, що відео перетворене з фільму у відеоформат (шлях вирішення "ТАК", що виходить з модуля 604 виявлення фаз), перетворене з фільму у відеоформат відео повертається у вихідний формат при зворотному перетворенні 606 з відеоформату в фільм. Надмірні кадри ідентифікуються і усуваються, а поля, витягнуті з того ж відеокадру, переплітаються в повне зображення. Оскільки послідовність відновлених кінозображень фотографічно записана з регулярними інтервалами в 1/24 секунди, процес оцінки руху, виконуваний в модулі 612 секціонування GOP або декодері 228, є більш точним при використанні перетворених з відеоформату в фільм зображень, ніж перетворених з фільму у відеоформат даних, які мають нерегулярну часову базу.

У одному аспекті модуль 604 виявлення фаз ухвалює конкретні рішення після прийому відеокадру. Ці рішення включають в себе: (i) є поточне відео з виведення перетворення з фільму у відеоформат, а фаза просування на крок кадру 3:2 є однією з п'яти фаз  $P_0$ ,  $P_1$ ,  $P_2$ ,  $P_3$  і  $P_4$ , показаних на Фіг. 38, чи ні; і (ii) відео сформоване як традиційне NTSC. Це рішення позначається як фаза  $P_5$ . Дані рішення відображаються як виведення модуля 604 виявлення фаз, показаного на Фіг. 2. Шлях з модуля 604 виявлення фаз, помічений "ТАК", активує перетворення 606 з відеоформату в фільм, вказуючи те, що воно надане з коректною фазою просу-

вання на крок кадру, з тим щоб воно могло відсортувати поля, які сформовані з одного фотографічного зображення, і комбінувати їх. Шлях з модуля 604 виявлення фаз, помічений "НІ", аналогічно активує модуль 605 усунення черезрядкової розгортки, щоб розділити видимий NTSC-кадр на поля для оптимальної обробки. Модуль 604 виявлення фаз може безперервно аналізувати відеокадри, оскільки різні типи відео можуть прийматися в будь-який час. Як приклад, відео, відповідне NTSC-стандарту, може бути вставлене у відео як реклама. Після перетворення з відеоформату в фільм результуюче прогресивне відео відправляється в модуль зниження шумності (фільтр) 607, який може бути використаний для того, щоб знижувати білий гаусів шум.

Коли традиційне NTSC-відео розпізнається (шлях "НІ" з модуля 601 виявлення фаз), воно передається в модуль 605 усунення черезрядкової розгортки для стиснення. Модуль 605 усунення черезрядкової розгортки перетворює черезрядкові поля в прогресивне відео, і операції зниження шумності потім можуть бути виконані для прогресивного відео. Один ілюстративний приклад обробки усунення черезрядкової розгортки описаний нижче.

Традиційні аналогові відеопристрої, такі як телевізори, виконують рендеринг відео черезрядковим способом, тобто ці пристрої передають рядки розгортки з парними номерами (парне поле) і рядки розгортки з непарними номерами (непарне поле). З точки зору дискретизації сигналів, це еквівалентно просторово-часовій субдискретизації в шаблоні, описаному за допомогою:

$$F_{\langle x, y, n \rangle} = \begin{cases} \Theta_{\langle x, y, n \rangle} & \text{якщо } y \bmod 2 = 0 \\ \Theta_{\langle x, y, n \rangle} & \text{якщо } y \bmod 2 = 1 \\ \text{стирання,} & \text{в іншому випадку} \end{cases} \quad \begin{matrix} \text{для парних полів} \\ \text{для парних полів} \end{matrix} \quad (1)$$

де  $\Theta$  означає зображення вихідного кадру,  $F$  означає черезрядкове поле, а  $(x, y, n)$  представляє горизонтальну, вертикальну і часову позицію пікселя, відповідно.

Без втрати спільності можна передбачити, що  $n=0$  - це парне поле в даному описі, так що вищеведене рівняння 1 спрощується таким чином:

$$F_{\langle x, y, n \rangle} = \begin{cases} \Theta_{\langle x, y, n \rangle} & \text{якщо } y \bmod 2 = n \bmod 2 \\ \text{стирання,} & \text{в іншому випадку} \end{cases} \quad (2)$$

Оскільки проріджування не виконується в горизонтальному напрямі, шаблон субдискретизації може бути проілюстрований в наступній  $n$ -у координаті.

Мета модуля усунення черезрядкової розгортки полягає в тому, щоб перетворювати черезрядкове відео (послідовність полів) в нечерезрядкові прогресивні кадри (послідовність кадрів). Іншими словами, інтерполювати парні і непарні поля так, щоб "відновити" або сформувати повнокадрові зображення. Це може бути представлено за допомогою рівняння 3:

$$F_0_{\langle x, y, n \rangle} = \begin{cases} F_{\langle x, y, n \rangle} & \text{якщо } y \bmod 2 = n \bmod 2 \\ F_1_{\langle x, y, n \rangle} & \text{в іншому випадку} \end{cases} \quad (3)$$

де  $F_i$  представляє результати усунення черезрядкової розгортки для відсутніх пікселів.

Фіг. 40 - це блок-схема, яка ілюструє визначені аспекти модуля 605 усунення черезрядкової розгортки, який використовує  $W_{med}$ -фільтрацію і оцінку руху для того, щоб формувати прогресивний кадр з черезрядкових мультимедійних даних. Верхня частина Фіг. 40 показує таблицю 4002 інтенсивності руху, яка може бути сформована за допомогою інформації з поточного поля, двох попередніх полів (поля  $PP$  і поля  $P$ ) і двох подальших полів (поля  $Next$  і поля  $Next\ Next$ ). Таблиця 4002 інтенсивності руху категоризує або секціонує поточний кадр на два або більше різних рівня руху, і може бути сформована за допомогою просторово-часової фільтрації, описаної детальніше нижче. У деяких аспектах таблиця 4002 інтенсивності руху сформована так, щоб ідентифікувати статичні області, області повільного руху і області швидкого руху, як описано в посиланнях до рівнянь 4-8 нижче. Просторово-часовий фільтр, наприклад,

Wmed-фільтр 4004, фільтрує черезрядкові мультимедійні дані за допомогою критеріїв на основі таблиці інтенсивності руху і формує просторово-часовий підготовлений кадр з усуненою черезрядковою розгорткою. У деяких аспектах процес Wmed-фільтрації приводить до горизонтального сусідства у  $[-1, 1]$ , вертикального сусідства в  $[-3, 3]$  і часового сусідства в п'ять подальших полів, які представлені за допомогою п'яти полів (поле PP, поле P, поле Current, поле Next, поле Next Next), проілюстрованих на Фіг. 40, де  $Z^{-1}$  представляє затримку одного поля. Відносно поля Current, поле Next і поле P є полями без парності, а поле PP і поле Next Next є полями з парністю. "Сусідство", використовуване для просторово-часової фільтрації, означає просторове і часове розміщення полів і пікселів, фактично використовуваних в ході операції фільтрації, і може бути проілюстроване як "апертура", як показано, наприклад, на Фіг. 6 і 7.

Модуль 605 усунення черезрядкової розгортки також може включати в себе модуль 4006 зниження шумності (фільтр зниження шумності), сконфігурований так, щоб відфільтровувати просторово-часовий підготовлений кадр з усуненою черезрядковою розгорткою, сформований за допомогою Wmed-фільтра 4004. Зниження шумності просторово-часового підготовленого кадру з усуненою черезрядковою розгорткою робить подальший процес пошуку руху більш точним, особливо якщо початкова послідовність черезрядкових мультимедійних даних забруднена білим шумом. Він також може видаляти щонайменше частково перешкоди дискретизації між парними і непарними рядками в Wmed-зображенні. Модуль 4006 зниження шумності може бути реалізований як множина фільтрів, включаючи модуль зниження шумності на основі вейвлет-стиснення і вейвлет-фільтра Вінера. Модуль зниження шумності може бути використаний для того, щоб видаляти шум з варіанта Wmed-кадру, перш ніж він додатково обробляється за допомогою інформації компенсації руху, і може видаляти шум, який присутній в Wmed-кадрі, і зберігати наявність сигналу незалежно від частотного вмісту сигналу. Різні типи фільтрів зниження шумності можуть бути використані, в тому числі вейвлет-фільтри. Вейвлети - це клас функцій, використовуваних для того, щоб локалізувати даний сигнал як в просторовій області, так і в області масштабування. Фундаментальна ідея відносно вейвлетів полягає в тому, щоб аналізувати сигнал при різних масштабах або розрізненнях, з тим щоб незначні зміни у вейвлет-представленні формували відповідні невеликі зміни у вихідному сигналі.

Вейвлет-стиснення або вейвлет-фільтр Вінера також може бути застосований як модуль зниження шумності. Вейвлет-стиснення складається з вейвлет-перетворення сигналу шуму, за яким іде стиснення невеликих вейвлет-коефіцієнтів до нуля (або меншого значення) при залишанні незмінними великих коефіцієнтів. У завершення, зворотне перетворення виконується для того, щоб одержати оцінений сигнал.

Фільтрація зі зниженням шумності підвищує точність компенсації руху в шумних оточеннях. Зниження шумності з вейвлет-стисненням може приводити до стиснення в області вейвлет-

перетворення і в типовому варіанті включає три етапи: лінійне пряме вейвлет-перетворення, нелінійне зниження шумності зі стисненням і лінійне зворотне вейвлет-перетворення. Фільтр Вінера - це MSE-оптимальний лінійний фільтр, який може бути використаний для того, щоб поліпшувати зображення, погіршені за допомогою адитивного шуму і розмивання. Ці фільтри, загалом, відомі в даній галузі техніки і описуються, наприклад, в роботі "Ideal spatial adaptation by wavelet shrinkage", вказаній вище, і в роботі авторів S. P. Ghael, A. M. Sayeed і R. G. Baraniuk "Improvement Wavelet denoising via empirical Wiener filtering", Proceedings of SPIE, том 3169, стор. 389-399, Сан-Дієго, липень 1997 року, які явно повністю містяться в даному документі по посиланню.

У деяких аспектах фільтр зниження шумності заснований на аспекті біртогального кубічного В-сплайнового вейвлет-фільтра (4, 2). Один такий фільтр може бути заданий за допомогою виконання прямого і зворотного перетворення:

$$h(z) = \frac{3}{4} + \frac{1}{2} (z^{-1}) + \frac{1}{8} (z^{-1})^2 \quad (4)$$

(пряме перетворення)

i

$$g(z) = \frac{5}{4} z^{-1} - \frac{5}{32} (z^{-2}) + \frac{3}{8} (z^{-3}) - \frac{3}{32} (z^{-4}) \quad (5)$$

(зворотне перетворення)

Застосування фільтра зниження шумності дозволяє підвищити точність компенсації руху в шумному оточенні. Реалізації таких фільтрів додатково описані в роботі "Ideal spatial adaptation by wavelet shrinkage", автори D.L. Donoho і I.M. Johnstone, Biometrika, том 8, стор. 425-455, 1994 рік, які явно повністю містяться в даному документі по посиланню.

Нижня частина Фіг. 40 ілюструє аспект для визначення інформації руху (наприклад, варіантів векторів руху, оцінки руху, компенсації руху) черезрядкових мультимедійних даних. Зокрема, Фіг. 40 ілюструє схему оцінки руху і компенсації руху, яка використовується для того, щоб сформувати попередній прогресивний кадр після компенсації руху з вибраного кадру, і потім комбінується з попереднім Wmed-кадром, щоб сформувати "кінцевий" прогресивний кадр, показаний як поточний кадр 4014 з усуненою черезрядковою розгорткою. У деяких аспектах варіанти (або оцінки) векторів руху (MV) черезрядкових мультимедійних даних надаються в модуль усунення черезрядкової розгортки із зовнішніх модулів оцінки руху і використовуються для того, щоб надавати початкову точку для двоспрямованого модуля оцінки і компенсації руху (ME/MC) 4018. У деяких аспектах модуль 4022 вибору MV-варіантів використовує раніше визначені MV для сусідніх блоків для MV-варіантів оброблюваних блоків, наприклад, блоків в попередньому кадрі 4020 з усуненою черезрядковою розгорткою. Компенсація руху може здійснюватися двоспрямовано на основі попереднього кадру 70 з

усуненою черезрядковою розгорткою і наступного (наприклад, майбутнього) Wmed-кадру 4008. Поточний Wmed-кадр 4010 і поточний кадр 4016 після компенсації руху (МС) об'єднуються або комбінуються за допомогою модуля 4012 комбінування. Результуючий поточний кадр 4014 з усуненою черезрядковою розгорткою, тепер прогресивний кадр, надається зворотно в МЕ/МС 4018, щоб бути використаним як попередній кадр 4020 з усуненою черезрядковою розгорткою, а також передається за межі модуля 605 усунення черезрядкової розгортки для подальшої обробки.

Можна виділити схеми прогнозу усунення черезрядкової розгортки, які містять міжпольову інтерполяцію, з внутрішньопольовою інтерполяцією за допомогою схеми усунення черезрядкової розгортки Wmed+МС. Іншими словами, просторово-часова Wmed-фільтрація може бути використана, насамперед, для цілей внутрішньопольової інтерполяції, тоді як міжпольова інтерполяція може виконуватися в ході компенсації руху. Це знижує пікове співвідношення "сигнал-шум" результату Wmed, але візуальна якість після того, як застосована компенсація руху, є більш задовільною, оскільки дефектні піксели з неточних рішень режиму внутрішньопольового прогнозу видаляються з процесу Wmed-фільтрації.

Після відповідної обробки перетворення з відеоформату в фільм або усунення черезрядкової розгортки на етапі 608 прогресивне відео обробляється для приглушення перешкод дискретизації і повторної дискретизації (наприклад, зміни розміру). У деяких аспектах повторної дискретизації модуль повторної дискретизації фази реалізований для зміни розміру зображень. У одному прикладі знижувальної дискретизації співвідношення між вихідним і зміненим зображенням може бути  $p/q$ , де  $p$  і  $q$  - це відносно прості числа. Загальне число фаз дорівнює  $p$ . Частота відсікання поліфазного фільтра в деяких аспектах дорівнює 0,6 для коефіцієнтів зміни розміру приблизно в 0,5. Частота відсікання не співпадає точно з коефіцієнтом зміни розміру, щоб підвищити високочастотну характеристику послідовності зміненого розміру. Це неминучо надає певні перешкоди дискретизації. Проте, добре відомо, що людське око віддає перевагу різким, але з невеликими перешкодами зображенням розмитим і не утримуючим перешкод зображенням.

Фіг. 41 ілюструє приклад поліфазної повторної дискретизації, що показує фази, якщо співвідношення зміни розміру становить  $3/4$ . Частота відсікання, проілюстрована на Фіг. 41, також становить  $3/4$ . Вихідні піксели проілюстровані на вищезгаданому кресленні з вертикальними осями. Синусоїдальна функція також намальована центрованою навколо осей, щоб представляти форму сигналу фільтра. Оскільки частота відсікання вибрана так, щоб бути такою ж, що і співвідношення повторної дискретизації, нулі синусоїдальної функції перекривають позицію пікселів після зміни розміру, проілюстрованої на Фіг. 41 за допомогою перетинів. Щоб знайти значення пікселя після зміни розміру, частка може бути підсумована з вихідних пікселів, як показано в наступному рівнянні:

$$v(x) = \sum_{i=-\infty}^{\infty} u_i(x) \text{sinc}(\pi f_c(x - x_i)) \quad (6)$$

де  $f_c$  - це частота відсікання. Вищезгаданий одновимірний поліфазний фільтр може бути застосований до горизонтального напрямку і вертикального напрямку.

Інший аспект повторної дискретизації (зміни розміру) враховує розгортку за межами екрана. У телевізійному сигналі NTSC зображення має 486 рядків розгортки, а в цифровому відео може мати 720 пікселів для кожного рядка розгортки. Проте, не все повне зображення видно на телевізорі внаслідок незбігу між розміром і форматом екрана. Частина зображення, яка не видна, називається розгорткою за межами екрана.

Щоб допомогти широкомовним станціям поміщувати корисну інформацію в область, видиму якомога більшою кількістю телевізорів, Суспільство інженерів кіно і телебачення (SMPTE) задало конкретні розміри кадру рухомих об'єктів, звані захищеною областю рухомих об'єктів і захищеною областю тайтлів. Див. керівні вказівки "SMPTE RP 27.3-1989 "Specifications for Safe Action and Safe Title Areas Test Pattern for Television Systems". Захищена область рухомих об'єктів задається за допомогою SMPTE як область, в якій повинні знаходитися всі суттєві рухомі об'єкти. Захищена область тайтлів задається як область, в якій вся корисна інформація може бути поміщена, щоб забезпечити видимість на більшості домашніх телевізійних приймачів.

Наприклад, посилаючись на Фіг. 25, захищена область 2510 рухомих об'єктів займає по центру 90% екрана, даючи 5% границю навколо. Захищена область 2505 тайтлів займає по центру 80% екрана, даючи 10% границі. Посилаючись тепер на Фіг. 26, оскільки захищена область тайтлів дуже мала, щоб додати додатковий вміст в зображення, деякі станції поміщують текст в захищену область рухомих об'єктів, яка знаходиться всередині білого прямокутного вікна 2615.

Звичайно чорні границі можуть бути видні в розгортці за межами екрана. Наприклад, на Фіг. 26 чорні границі відображаються в верхній стороні 2620 і нижній стороні 2625 зображення. Ці чорні границі можуть бути видалені в розгортці за межами екрана, оскільки відео H.264 застосовує розширення границь при оцінці руху. Розширені чорні границі дозволяють збільшувати залишок. Консервативно, границя може бути обрізана на 2% і потім виконана зміна розміру. Фільтри для зміни розміру можуть бути сформовані відповідним чином. Відсікання виконується для того, щоб видалити розгортку за межами екрана до поліфазної знижувальної дискретизації.

Знов посилаючись на Фіг. 6, прогресивне відео потім переходить до етапу 610, де виконуються операції розблокування і приглушення реверберацій. Два типи перешкод, "блокування" і "реверберації", як правило, виникають в додатках відеостиснення. Перешкоди зображення виникають, оскільки алгоритми стиснення ділять кожний кадр на блоки (наприклад, блоки  $8 \times 8$ ). Кожний блок відновлюється з деякими невеликими помилками, і

помилки по краях блока часто контрастують з помилками по краях сусідніх блоків, роблячи границі блоків видимими. На відміну від цього, перешкоди реверберацій з'являються як спотворення по краях елементів зображень. Перешкоди реверберації виникають, оскільки кодер відкидає дуже багато інформації при квантуванні високочастотних DCT-коефіцієнтів. У деяких ілюстративних прикладах, деблокування і приглушення реверберацій може використовувати фільтри нижніх частот з FIR (кінцевою імпульсною характеристикою), щоб приховувати ці видимі перешкоди.

У одному прикладі обробки деблокування фільтр деблокування може бути застосований до всіх країв блоків 4×4 кадру, за винятком країв на границі кадру і всіх країв, для яких процес фільтра деблокування відключений. Цей процес фільтрації повинен виконуватися на основі макроблоків після завершення процесу складання кадрів, причому всі макроблоки в кадрі обробляються в порядку збільшення адрес макроблоків. Для кожного макроблока вертикальні краї фільтруються першими, зліва направо, а потім горизонтальні краї фільтруються зверху вниз. Процес фільтра деблокування компоненти яскравості виконується по чотирьох краях з 16 вибірок, і процес фільтра деблокування для кожної компоненти кольоровості виконується по двох краях з 8 вибірок для горизонтального напрямку і для вертикального напрямку, як показано на Фіг. 39. Значення вибірки вище і зліва від поточного макроблока, які, можливо, вже модифіковані за допомогою операції процесу деблокування для попередніх макроблоків, повинні бути використані як входні дані процесу фільтра деблокування для поточного макроблока і можуть бути додатково модифіковані в ході фільтрації поточного макроблока. Значення вибірки, модифіковані в ході фільтрації вертикальних країв, можуть бути використані як входні дані для фільтрації горизонтальних країв для того ж макроблока. Процес деблокування може бути активований для компонент яскравості і кольоровості окремо.

У прикладі обробки приглушення реверберацій двовимірний фільтр може бути адаптивно застосований так, щоб згладжувати області біля країв. Крайові піксели піддаються невеликій фільтрації або не піддаються фільтрації, щоб уникнути розмивання.

#### Модуль секціонування GOP

Після деблокування і приглушення реверберацій прогресивне відео обробляється за допомогою модуля 612 секціонування GOP. Розміщення GOP може включати в себе виявлення зміни кадрів, формування таблиць складності (наприклад, часових, просторових таблиць смуги пропускання) і адаптивне секціонування GOP. Все вищезазначене описується далі.

#### А. Виявлення зміни сцен

Виявлення кадрів належить до визначення того, коли кадр в групі зображень (GOP) надає дані, які вказують те, що сталася зміна сцени. Загалом, в GOP кадри можуть не мати суттєвої зміни для будь-яких двох або трьох (або більшого числа) сусідніх кадрів, або можуть бути повільні зміни або швидкі зміни. Зрозуміло, ці класифікації зміни сцен за необхідності можуть бути додатково розбиті до

більшого рівня зміни залежно від варіанта застосування.

Виявлення зміни кадрів або сцен важливе для ефективного кодування відео. У типовому варіанті, коли GOP не змінюється значно, за I-кадром на початку GOP іде ряд предиктивних кадрів, що дозволяє значною мірою кодувати відео так, щоб подальше декодування і відображення відео було візуально прийнятним. Проте, коли сцена змінюється, різко або повільно, додаткові I-кадри і менше кодування з прогнозом (Р-кадри і В-кадри) може вимагатися, щоб надати згодом декодовані візуально прийнятні результати.

Системи і способи виявлення кадрів і кодування, які підвищують продуктивність існуючих систем кодування, описуються нижче. Ці аспекти можуть бути реалізовані в модулі 612 секціонування GOP препроцесора 226 (Фіг. 7) або включені в пристрій кодера, який може працювати з або без препроцесора. Дані аспекти використовують статистику (або показники), які включають в себе статистичні порівняння між сусідніми кадрами відеоданих, щоб визначати, якщо виникла різка зміна сцен, сцена повільно змінюється або є спалахи камери в сцені, які можуть особливо ускладнювати кодування відео. Статистика може бути одержана від препроцесора і потім відправлена в пристрій кодування або вона може бути сформована в пристрої кодування (наприклад, за допомогою процесора, сконфігурованого так, щоб виконувати компенсацію руху). Результуюча статистика допомагає в прийнятті рішення по виявленню зміни сцен. У системі, яка здійснює перекодування, часто є належний препроцесор або конфігурований процесор. Якщо препроцесор виконує усунення череззрядкової розгортки за допомогою компенсації руху, статистика компенсації руху доступна і готова до застосування. У таких системах алгоритм виявлення кадрів може трохи підвищити складність системи.

Ілюстративний приклад модуля виявлення кадрів, описаний в даному документі, повинен використовувати тільки статистику з попереднього кадру, поточного кадру і наступного кадру, а, отже, має дуже невелику затримку. Модуль виявлення кадрів розрізняє декілька різних типів подій кадрів, в тому числі різку зміну сцен, монтажний перехід і іншу повільну зміну сцен, а також спалах камери. За допомогою визначення різних типів подій кадрів з різними стратегіями в кодері ефективність кодування і візуальна якість поліпшуються.

Виявлення зміни сцен може бути використане для будь-якої системи кодування відео, щоб вона інтелектуально зберігала біти за допомогою вставки I-кадру з фіксованим інтервалом. У деяких аспектах інформація вмісту, одержана за допомогою препроцесора (наприклад, або яка міститься в метаданих, або обчислена за допомогою препроцесора 226), може бути використана для виявлення зміни сцен. Наприклад, залежно від інформації вмісту, порогові значення і інші критерії, описані нижче, можуть динамічно коректуватися для різних типів відеовмісту.

Кодування відео звичайно виконується для структурованої групи зображень (GOP). GOP звичайно починається з внутрішньо кодованого кадру (I-кадру), за яким іде послідовність з Р (прогнозую-

чих) або В (двоспрямованих) кадрів. У типовому варіанті І-кадр може зберігати всі дані, необхідні для того, щоб відображати кадр, В-кадр базується на даних в попередніх і подальших кадрах (наприклад, що містять тільки дані, змінні з попереднього кадру або відмінні від даних в наступному кадрі), а Р-кадр містить дані, які змінні з попереднього кадру. У більшості випадків між І-кадрами вставляються Р-кадри і В-кадри в кодованому відео. Відносно розміру (наприклад, числа бітів, використовуваних для того, щоб кодувати кадр), І-кадри в типовому варіанті набагато більші Р-кадрів, які, в свою чергу, більші В-кадрів. Для ефективної обробки кодування, передачі і декодування довжина GOP повинна бути достатньо великою для того, щоб знижувати фактичні втрати з великих І-кадрів, і достатньо маленькою для того, щоб протистояти незбігу між кодером і декодером або погіршенню якості каналу. Крім цього, макроблоки (МВ) в Р-кадрах можуть бути внутрішньо кодованими з тієї ж причини.

Виявлення зміни сцен може бути використане відеокодером для того, щоб визначати відповідну довжину GOP і вставляти І-кадри на основі довжини GOP замість вставки часто непотрібного І-кадру з фіксованим інтервалом. У практичній системі потокової передачі відео якість каналу зв'язку звичайно погіршується через бітові помилки або втрати пакетів. Те, де розміщуються І-кадри або І-МВ, може суттєво впливати на якість декодованого відео і зручність перегляду. Одна схема кодування полягає в тому, щоб використовувати внутрішньо кодовані кадри для зображень або частин зображень, які мають суттєву відмінність в порівнянні з спільно розміщеними попередніми зображеннями або частинами зображень. Звичайно ці зони не можуть ефективно і раціонально прогнозуватися при оцінці руху, і кодування може виконуватися більш ефективно, якщо ці зони виключені з методик міжкадрового кодування (наприклад, кодування за допомогою В-кадрів і Р-кадрів). У контексті погіршення якості каналу ці зони, ймовірно, зазнають поширення помилок, яке може бути зменшене або виключене (або практично виключене) за допомогою внутрішньоканалового кодування.

Частини відео GOP можуть бути класифіковані на дві або більше категорій, при цьому кожна зона може мати різні критерії внутрішньоканалового кодування, які можуть залежати від конкретної реалізації. Як приклад, відео може бути класифіковане на три категорії: різка зміна сцен, монтажний перехід і інші повільні зміни сцен, а також спалахи камери.

Різкі зміни сцен включають в себе кадри, які значно відрізняються від попереднього кадру, що звичайно викликаються роботою камери. Оскільки вміст цих кадрів відрізняється від вмісту попереднього кадру, кадри різкої зміни сцен повинні бути кодовані як І-кадри.

Монтажний перехід і інші повільні зміни сцен включають в себе повільне перемикавання сцен, що звичайно викликається за допомогою обчислювальної обробки зйомки камерою. Поступове змішування двох різних сцен може виглядати більш приємним для людських очей, але накладає складну задачу на кодування відео. Компенсація руху

не може ефективно зменшити швидкість передачі бітів цих кадрів, і більше число внутрішніх МВ може бути оновлене для цих кадрів.

Спалахи камери, або події спалаху камери, виникають, коли вміст кадру включає в себе спалахи камери. Ці спалахи мають відносно невелику тривалість (наприклад, один кадр) і є дуже яскравими, так що піксели в кадрі, що візуалізує спалахи, демонструють дуже високу яскравість відносно відповідної області в сусідньому кадрі. Спалахи камери зсувають яскравість зображення раптово і швидко. Звичайно тривалість спалаху камери менше тривалості часового маскування зорової системи людини (HVS), яка в типовому варіанті задається рівною 44 мс. Людські очі не чутливі до якості цих коротких сплесків яскравості, і тому вони можуть бути кодовані більш грубо. Оскільки кадри спалаху не можуть ефективно оброблятися за допомогою компенсації руху і вони є поганим варіантом прогнозу для майбутніх кадрів, грубе кодування цих кадрів не знижує ефективність кодування майбутніх кадрів. Сцени, класифіковані як спалахи, не повинні бути використані для того, щоб прогнозувати інші кадри внаслідок "штучної" високої яскравості, і інші кадри не можуть бути ефективно використані для того, щоб прогнозувати ці кадри, через ту ж причину. Після ідентифікації ці кадри можуть бути витягнуті, оскільки вони можуть потребувати відносно високого об'єму обробки. Один варіант полягає в тому, щоб видаляти спалахи камери і кодувати коефіцієнт DC на їх місці; це рішення є простим, обчислювально швидким і економить багато бітів.

Коли будь-які з вищенаведених категорій кадрів виявлені, оголошується подія кадру. Виявлення кадрів корисне не тільки для того, щоб підвищувати ефективність кодування, воно також може допомогти в ідентифікації пошуку і індексації відеовмісту. Один ілюстративний аспект процесу виявлення сцен описаний нижче. У цьому прикладі, процес виявлення кадрів спочатку обчислює інформацію, або показники, для вибраного кадру, що обробляється для виявлення кадрів. Показники можуть включати в себе інформацію з двовимірної обробки оцінки і компенсації руху для відео і інших заснованих на яскравості показників.

Щоб виконувати двовимірну оцінку/компенсацію руху, відеопослідовність повинна бути попередньо оброблена за допомогою модуля двоспрямованої компенсації руху, який зіставляє кожний блок  $8 \times 8$  поточного кадру з блоками в двох кадрах, найбільш близьких до сусідніх кадрів, один в минулому і один в майбутньому. Блок компенсації руху формує вектори руху і різниці показників для кожного блока. Фіг. 29 - це ілюстрація, яка демонструє приклад зіставлення пікселів поточного кадру С з минулим кадром Р і майбутнім (або наступним) кадром N і показує вектори руху для зіставлених пікселів (минулий вектор руху  $MV_P$  і майбутній вектор руху  $MV_N$ ). Загальний опис формування двоспрямованих векторів руху і пов'язаного кодування, загалом, наводиться нижче з посиланням на Фіг. 32.

Після визначення інформації двоспрямованого руху (наприклад, інформації руху, яка ідентифікує МВ (найбільш співпадаючі)) у відповідних сусідніх

кадрах, додаткові показники можуть бути сформовані (наприклад, за допомогою модуля компенсації руху в модулі 612 секціонування GOP або іншому відповідному компоненті) за допомогою різних порівнянь поточного кадру з наступним кадром і попереднім кадром. Модуль компенсації руху може формувати різницевий показник для кожного блоку. Різницевим показником може бути сума квадратів різниць (SSD) або сума абсолютної різниці (SAD). Без втрати спільності, тут як приклад використовується SAD.

Для кожного кадру співвідношення SAD, також згадуване як "коефіцієнт контрастності", обчислюється таким чином:

$$\gamma = \frac{\varepsilon + SAD_P}{\varepsilon + SAD_N} \quad (6)$$

де  $SAD_P$  і  $SAD_N$  - це сума абсолютних різниць прямого і зворотного різницевого показника, відповідно. Потрібно зазначити, що знаменник містить невелике позитивне число  $\varepsilon$ , щоб не допустити помилки ділення на нуль. Чисельник також містить  $\varepsilon$ , щоб компенсувати вплив одиниці в знаменнику. Наприклад, якщо попередній кадр, поточний кадр і наступний кадр ідентичні, пошук руху повинен давати  $SAD_P = SAD_N = 0$ . У цьому випадку вищезгадані формулатори обчислення  $\gamma = 1$  замість 0 або нескінченності.

Гістограма яскравості може бути обчислена для кожного кадру. У типовому варіанті мультимедійні зображення мають глибину яскравості (наприклад, число "елементів розрізнення") у вісім бітів. Глибина яскравості, використовувана для обчислення гістограми яскравості згідно з деякими аспектами, може бути задана рівною 16, щоб одержати гістограму. У інших аспектах глибина яскравості може бути задана рівною відповідному числу, яке може залежати від типу оброблюваних даних, доступної обчислювальної потужності або інших попередньо визначених критеріїв. У деяких аспектах глибина яскравості може задаватися динамічно на основі обчисленого або прийнятого показника, наприклад, вмісту даних.

Нижченаведене рівняння ілюструє один приклад обчислення різниці гістограм яскравості (лямбда):

$$\lambda = \frac{\sum_{i=1}^{16} |N_{Pi} - N_{Ci}|}{N} \quad (7)$$

де  $N_{Pi}$  - це число блоків в  $i$ -тому елементі розрізнення для попереднього кадру,  $N_{Ci}$  - це число блоків в  $i$ -тому елементі розрізнення для поточного кадру, а  $N$  - це загальне число блоків в кадрі. Якщо різниця гістограм яскравості попереднього і поточного кадру повністю несхожа (або неперетинна), то  $\lambda = 2$ .

Використовуючи цю інформацію, показник різниці кадрів ( $D$ ) обчислюється таким чином:

$$D = \frac{\gamma_C}{\gamma_P} + A\lambda \left( \lambda + 1 \right) \quad (8)$$

де  $A$  - це константа, вибрана за допомогою варіанта застосування,  $\gamma_C = \frac{\varepsilon + SAD_P}{\varepsilon + SAD_N}$  і

$$\gamma_P = \frac{\varepsilon + SAD_{PP}}{\varepsilon + SAD_C}.$$

Вибраний (поточний кадр) класифікується як кадр різкої зміни сцен, якщо різницевий показник кадрів задовольняє критерій, показаний в рівнянні 9:

$$D = \frac{\gamma_C}{\gamma_P} + A\lambda \left( \lambda + 1 \right) \geq T_1 \quad (9)$$

де  $A$  - це константа, вибрана за допомогою варіанта застосування, а  $T_1$  - це поріг.

У одному прикладі, який показує моделювання, задавання  $A=1$  і  $T_1=5$  надає хорошу продуктивність виявлення. Якщо поточний кадр є кадром різкої зміни сцен, то  $\gamma_C$  повинно бути великим, а  $\gamma_P$  повинно бути маленьким. Співвідношення  $\gamma_C/\gamma_P$  може бути використано замість тільки  $\gamma_C$ , з тим щоб показник був нормалізований до рівня активності.

Потрібно зазначити, що вищенаведений критерій використовує різницю гістограм яскравості лямбда ( $\lambda$ ) нелінійним способом. Фіг. 16 ілюструє, що  $\lambda^*(2\lambda+1)$  є опуклою функцією. Коли  $\lambda$  невелика (наприклад, близько нуля), це просто передспотворення. Чим більше стає  $\lambda$ , тим більше передспотворення надається за допомогою функції. За допомогою цього передспотворення для будь-якого  $\lambda$  більше 1,4, різка зміна сцен виявляється, якщо поріг  $T_1$  заданий рівним 5.

Поточний кадр визначається як монтажний перехід або повільна зміна сцен, якщо показник інтенсивності сцени  $D$  задовольняє критерій, показаний в рівнянні 5:

$$T_2 \leq D < T_1 \quad (10)$$

для визначеного числа безперервних кадрів, де  $T_1$  - це такий же поріг, що і використовуваний вище, а  $T_2$  - це інше значення порога.

Подія спалаху звичайно примушує гістограму яскравості зміститися до більш яскравої сторони. У камері цього ілюстративного аспекту статистика гістограми яскравості використовується для того, щоб визначати, чи містить поточний кадр спалахи камери. Процес виявлення кадрів може визначати, чи більше яскравість поточного кадру, ніж яскравість попереднього кадру, на визначений поріг  $T_3$ , і більше яскравість поточного кадру, ніж яскравість наступного кадру, на поріг  $T_3$ , як показано в рівняннях 11 і 12:

$$\bar{\gamma}_C - \bar{\gamma}_P \geq T_3 \quad (11)$$

$$\bar{\gamma}_C - \bar{\gamma}_N \geq T_3 \quad (12)$$

якщо вищезгаданий критерій не задовольняється, поточний кадр не класифікується як такий, що містить спалахи камери. Якщо критерій задо-



вольняється, процес виявлення кадрів визначає, чи більше зворотний різницеви показник  $SAD_P$  і прямий різницеви показник  $SAD_N$  визначеного порога  $T_4$ , як проілюстровано в рівняннях нижче:

$$SAD_P \geq T_4 \quad (13)$$

$$SAD_N \geq T_4 \quad (14)$$

де  $\bar{Y}_C$  - це середня яскравість поточного кадру,  $\bar{Y}_P$  - це середня яскравість попереднього кадру,  $\bar{Y}_N$  - це середня яскравість наступного кадру, а  $SAD_P$  і  $SAD_N$  - це прямий і зворотний різницеви показники, асоціативно зв'язані з поточним кадром.

Процес виявлення кадрів визначає події спалаху камери за допомогою визначення спочатку того, чи більше яскравість поточного кадру яскравості попереднього кадру і яскравості наступного кадру. Якщо ні, кадр не є подією спалаху камери; але якщо так, то це може бути. Процес виявлення кадрів в такому випадку може визначити, чи більше зворотний різницеви показник порога  $T_3$  і чи більше прямий різницеви показник порога  $T_4$ ; якщо обидві ці умови задоволені, процес виявлення кадрів класифікує поточний кадр як такий, що має спалахи камери. Якщо критерій не задовольняється, кадр не класифікується навіть як будь-який тип кадру або йому може бути надана класифікація за умовчанням, яка ідентифікує кодування, яке повинно бути виконане для кадру (наприклад, видалити кадр, кодувати як I-кадр).

Деякі зразкові значення  $T_1$ ,  $T_2$ ,  $T_3$  і  $T_4$  показані вище. У типовому варіанті ці порогові значення вибираються за допомогою тестування конкретної реалізації виявлення кадрів. У деяких аспектах одне або більше порогових значень  $T_1$ ,  $T_2$ ,  $T_3$  і  $T_4$  є попередньо визначеними, і ці значення включаються в класифікатор кадрів в пристрої кодування. У деяких аспектах одне або більше порогових значень  $T_1$ ,  $T_2$ ,  $T_3$  і  $T_4$  можуть бути задані в ході обробки (наприклад, динамічно) на основі викори-

стання інформації (наприклад, метаданих), що надається в класифікатор кадрів, або на основі інформації, обчисленої самим класифікатором кадрів.

Кодування відео за допомогою інформації виявлення кадрів в типовому варіанті виконується в кодері, але описане тут для повноти розкриття виявлення кадрів. Посилаючись на Фіг. 30, процес 301 кодування може використовувати інформацію виявлення кадрів для того, щоб кодувати відео на основі виявлених кадрів в послідовності кадрів. Процес 301 переходить до етапу 303 і перевіряє, чи класифікований поточний кадр як різка зміна сцени. Якщо так, на етапі 305 поточний кадр може бути кодований як I-кадр, і границя GOP може бути визначена. Якщо ні, процес 301 переходить до етапу 307; якщо поточний кадр класифікований як частина повільно змінюваної сцени, на етапі 309 поточний кадр і інші кадри в повільно змінюваній сцені можуть бути кодовані як кадри з прогнозом (наприклад, P-кадр або B-кадр). Процес 301 далі переходить до етапу 311, на якому він перевіряє, чи класифікований поточний кадр як сцена спалаху, що містить спалахи камери. Якщо так, на етапі 313 кадр може бути ідентифікований для спеціальної обробки, наприклад, для видалення або кодування коефіцієнта DC для кадру; якщо ні, класифікація поточного кадру не виконана, і поточний кадр може бути кодований відповідно до інших критеріїв, кодований як I-кадр або видалений.

У вищеописаному аспекті величина різниці між кадром, який повинен бути стиснений, і двома сусідніми кадрами вказується за допомогою різницевого показника кадрів  $D$ . Якщо значна величина односторонньої зміни яскравості виявлена, це вказує ефект монтажного переходу в кадрі. Чим більш короткочасним є монтажний перехід, тим більше посилення може бути досягнуте за допомогою використання B-кадрів. У деяких аспектах модифікований різницеви показник кадрів використовується так, як показано в рівнянні нижче:

$$D_1 = \begin{cases} -\alpha + 2\alpha \frac{|d_P - d_N|}{d_P + d_N} \times D, & \text{якщо } Y_P - \Delta \geq Y_C \geq Y_N + \Delta \text{ або } Y_P + \Delta \geq Y_C \geq Y_N - \Delta \\ D, & \text{в іншому випадку} \end{cases} \quad (15)$$

де  $d_P = |Y_C - Y_P|$  і  $d_N = |Y_C - Y_N|$  - це різниця компоненти яскравості між поточним кадром і попереднім кадром і різниця компоненти яскравості між поточним кадром і наступним кадром, відповідно,  $\Delta$  представляє константу, яка може бути визначена звичайним дослідним шляхом, оскільки вона може залежати від реалізації, а  $\alpha$  - це змінна зважування, що має значення від 0 до 1.

В. Формування таблиці смуги пропускання

Препроцесор 226 (Фіг. 6) також може бути сконфігурований так, щоб формувати таблицю смуги пропускання, яка може бути використана для кодування мультимедійних даних. У деяких аспектах модуль 712 класифікації вмісту в кодері 228 (Фіг. 7) замість нього формує таблицю смуги пропускання.

Візуальна якість людини  $V$  може бути функцією від складності кодування  $C$  і призначених бітів

В (також згадуваних як смуга пропускання). Фіг. 15 - це графік, який ілюструє даний взаємозв'язок. Потрібно зазначити, що показник складності кодування  $C$  розглядає просторові і часові частоти з точки зору бачення користувача. Для спотворень, більш чутливих до людських очей, значення складності відповідно вище. У типовому варіанті можна допустити, що  $V$  монотонно зменшується в  $C$  і монотонно збільшується в  $B$ .

Щоб досягнути постійної зорової якості, смуга пропускання ( $B_i$ ) призначається і-тому об'єкту (кадру або MB), який повинен бути кодований, що задовольняє критерії, виражені в нижченаведених двох рівняннях:

$$B_i = B(C_i, V) \quad (16)$$

$$B = \sum_i B_i \quad (17)$$

У двох вищенаведених рівняннях  $C_i$  - це складність кодування  $i$ -то об'єкта,  $B$  - це загальна доступна смуга пропускання, а  $V$  - це досягнута зорова якість для об'єкта. Візуальну якість людини важко сформулювати як рівняння. Отже, вищезгаданий набір рівнянь не є точно заданим. Проте, якщо передбачається, що тривимірна модель є безперервною у всіх змінних, співвідношення смуги пропускання ( $B/V$ ) може трактуватися як незмінне з сусідством пари ( $C, V$ ). Співвідношення смуги пропускання  $\beta_i$  задається в рівнянні, показаному нижче:

$$\beta_i = B_i/V \quad (18)$$

Призначення бітів в такому випадку може бути задане так, як виражено в наступних рівняннях:

$$\beta_i = \beta(C_i) \quad 1 = \sum_i \beta_i \text{ для } C_i, V \in [C_0, V_0] \quad (19)$$

де  $\delta$  вказує "сусідство".

На складність кодування впливає чутливість зорової системи людини, як просторова, так і часова. Зорова модель людини за Гродом є прикладом моделі, яка може бути використана для того, щоб задавати просторову складність. Ця модель враховує локальну просторову частоту і навколишнє освітлення. Результируючий показник називається  $D_{csat}$ . У точці попередньої обробки в процесі те, повинно зображення бути внутрішньо кодованим або зовнішньо кодованим, невідомо, і співвідношення смуги пропускання для обох випадків формуються. Біти призначаються згідно зі співвідношенням між  $\beta_{INTRA}$  різних відеооб'єктів. Для внутрішньо кодованих зображень співвідношення смуги пропускання виражається в наступному рівнянні:

$$\beta_{INTRA} = \beta_{0INTRA} \log_{10}(1 + \alpha_{INTRA} Y^2 D_{csat}) \quad (20)$$

У вищенаведеному рівнянні  $Y$  - це компонента середньої яскравості макроблока,  $\alpha_{INTRA}$  - це ваговий коефіцієнт для квадрата яскравості і члена  $D_{csat}$ , що іде за ним,  $\beta_{INTRA}$  - це коефіцієнт нормалізації, щоб гарантувати те, що  $1 = \sum_i \beta_i$ . Наприклад,

значення  $\alpha_{INTRA}=4$  досягає хорошої зорової якості. Інформація вмісту (наприклад, класифікація вмісту) може бути використана для того, щоб надавати  $\alpha_{INTRA}$  значення, яке відповідає необхідному рівню хорошої зорової якості для конкретного вмісту відео. У одному прикладі, якщо відеовміст містить широкомовну передачу новин з "говорючою головою", рівень зорової якості може бути заданий нижче, оскільки зображення інформації або відображувана частина екрана може вважатися меншої важливості, ніж аудіочастина, і менше бітів може бути виділено для того, щоб кодувати дані. У іншому прикладі, якщо відеовміст містить спортивний сюжет, інформація вмісту може бути використана так, щоб надавати  $\alpha_{INTRA}$  значення, яке відповідає більш високому рівню зорової якості, оскільки відображувані зображення можуть бути більш важливі для глядача, і, відповідно, більше число бітів виділяється для того, щоб кодувати дані.

Щоб зрозуміти це взаємовідношення, потрібно відмітити те, що смуга пропускання призначається логарифмічним чином згідно зі складністю кодування. Член квадрата яскравості  $Y$  відображає той факт, що коефіцієнти з великим модулем використовують більше бітів для того, щоб кодувати. Щоб представити логарифм з одержання негативних значень, одиниця додається до члена в дужках. Логарифми з іншими основами також можуть бути використані.

Часова складність визначається за допомогою вимірювання різницевого показника кадрів, який вимірює різницю між двома послідовними кадрами з урахуванням величини руху (наприклад, векторів руху) нарівні з різницеvim показником кадрів, таким як сума абсолютних різниць (SAD).

Призначення бітів для внутрішньо кодованих зображень може враховувати як просторову, так і часову складність. Це виражене нижче:

$$\beta_{INTER} = \beta_{0INTER} \log_{10} \left( 1 + \alpha_{INTER} \cdot SSD \cdot D_{csat} \exp \left( -\gamma \|MV_P + MV_N\|^2 \right) \right) \quad (21)$$

У вищенаведеному рівнянні  $MV_P$  і  $MV_N$  - це прямий і зворотний вектори руху для поточного MB (див. Фіг. 29). Потрібно зазначити, що  $Y^2$  в формулі внутрішньо кодованої смуги пропускання замінено сумою квадратів різниць (SSD).

Щоб зрозуміти роль  $\|MV_P + MV_N\|^2$  у вищенаведеному рівнянні, відмітимо наступні характеристики зорової системи людини: області, що піддаються плавному передбачуваному руху (невелике  $\|MV_P + MV_N\|^2$ ), привертають увагу і можуть бути відстежені оком і в типовому варіанті не можуть допускати більше спотворень, ніж стаціонарні області. Проте, області, що піддаються швидкому або непередбачуваному руху (велике

$\|MV_P + MV_N\|^2$ ), не можуть відстежувати і можуть допускати суттєве квантування. Досліди показали, що  $\alpha_{INTER}=1$ ,  $\gamma=0.001$  досягає хорошої зорової якості.

#### C. Адаптивне секціонування GOP

В іншому ілюстративному прикладі обробки, яка може бути виконана за допомогою препроцесора 226, модуль 612 секціонування GOP за Фіг. 6 також може адаптивно змінювати структуру групи зображень, кодованих разом, і описується з посиленням на приклад, що використовує MPEG2. Деякі більш старі стандарти стиснення відео (наприклад, MPEG2) не вимагають того, щоб GOP мала регулярну структуру, хоч вона може накладатися. Послідовність MPEG2 завжди починається з I-

кадру, тобто кадру, закодованого без посилання на попередні зображення. Формат MPEG2 GOP звичайно попередньо встановлюється в кодері за допомогою фіксування рознесення в GOP P- або прогнозних зображень, які ідуть після I-кадру. P-кадри - це зображення, які частково прогнозовані з попередніх I- або P-зображень. Кадри між початковим I-кадром і подальшими P-кадрами кодуються як B-кадри. B-кадр (B означає двоспрямований) може використовувати попередні і наступні I- або P-зображення окремо або одночасно як посилання. Число бітів, необхідних для того, щоб кодувати I-кадр, в середньому перевищує число бітів, необхідних для того, щоб кодувати P-кадр; аналогічно, число бітів, необхідних для того, щоб кодувати P-кадр, в середньому перевищує число бітів, необхідних для B-кадру. Пропущений кадр, якщо він використовується, не повинен вимагати бітів для свого представлення.

Концепція, що знаходиться в основі використання P- і B-кадрів і в більш нових алгоритмах стиснення, пропуск кадрів для того, щоб знизити швидкість передачі даних, необхідних для того, щоб представляти відео, є усуненням часової надмірності. Коли часова надмірність висока, тобто є незначна зміна від зображення до зображення, використання P-, B- або пропущених зображень ефективно представляє відеопотік, оскільки I- або P-зображення, декодовані раніше, використовуються надалі як посилання для того, щоб декодувати інші P- або B-зображення.

Адаптивне секціонування GOP засноване на адаптивному використанні цієї концепції. Різниця між кадрами квантується, і рішення, щоб представляти зображення за допомогою I, P, B або пропущеного кадру, автоматично приймається після того, як виконуються відповідні тести для квантованих різниць. Адаптивна структура має переваги, недоступні в фіксованій GOP-структурі. Фіксована структура повинна ігнорувати можливість того, що сталися невеликі зміни у вмісті; адаптивна процедура повинна представляти набагато більше B-кадрів, які повинні бути вставлені між кожним I- і P- або двома P-кадрами, тим самим зменшуючи число бітів, необхідних для того, щоб адаптивно представляти послідовність кадрів. Навпаки, коли зміна відеовмісту суттєва, ефективність P-кадрів значно знижується, оскільки різниця між передбаченими і опорними кадрами дуже велика. При цих умовах співпадаючі об'єкти можуть випадати із зон пошуку руху або схожість між співпадаючими об'єктами знижується внаслідок спотворення, що викликається за рахунок змін кута камери. У цій точці P-кадри або I- і її сусідній P-кадр повинні бути вибрані так, щоб бути ближче один до одного, і повинно бути вставлене менше число B-кадрів. Фіксована GOP може не виконувати це коректування.

У системі, розкритій тут, ці умови автоматично вимірюються. GOP-структура є гнучкою і зроблена так, щоб пристосовуватися до цих змін у вмісті. Система оцінює різницевий показник кадрів, який може розглядати як міру відстані між кадрами, з однаковими адитивними способами відстані. У концепції при умові кадрів  $F_1$ ,  $F_2$  і  $F_3$ , що мають міжкадрову відстань  $d_{12}$  і  $d_{23}$ , відстань між  $F_1$  і  $F_3$  береться рівною щонайменше  $d_{12}+d_{23}$ . Призначен-

ня кадрів робляться на основі цього аналогічного відстані показника.

Модуль секціонування GOP працює за допомогою призначення типів зображень кадрам по мірі того, як вони приймаються. Тип зображення вказує спосіб прогнозу, який може вимагатися при кодуванні кожного блока.

I-зображення кодуються без посилання на інші зображення. Оскільки вони автономні, вони надають точки доступу в потоці даних, де декодування може починатися. Тип кодування I призначається кадру, якщо відстань до попереднього кадру перевищує поріг зміни сцен.

P-зображення можуть використовувати попередні I- або P-зображення для прогнозу з компенсацією руху. Вони використовують блоки в попередніх полях або кадрах, які можуть бути зміщені від передбачуваного блока як основа для кодування. Після того як опорний кадр віднімається з блока, що розглядається, залишковий блок кодується, в типовому варіанті за допомогою дискретного косинусного перетворення, для усунення просторової надмірності. Типи кодування P призначаються кадру, якщо відстань між ним і останнім кадром, призначеним так, щоб бути P-кадром, перевищує другий поріг, який в типовому варіанті менше першого.

Зображення B-кадрів можуть використовувати попередні і наступні P- або I-зображення для компенсації руху, як описано нижче. Блок в B-зображенні може бути спрогнозованим прямо, зворотно або двоспрямовано; або він може бути внутрішньо кодований без посилання на інші кадри. У H.264 опорний блок може бути лінійною комбінацією максимум 32 блоків з максимум стільки кадрів. Якщо кадр не може бути призначений так, щоб бути I- або P-типу, він призначається так, щоб бути B-типу, якщо відстань від нього до проміжного попередника більше третього порога, який в типовому варіанті менше другого порога.

Якщо кадр не може бути призначений так, щоб стати кодованим B-кадром, йому призначається стан "пропуск кадру". Цей кадр може бути пропущений, оскільки він практично є копією попереднього кадру.

Оцінка показника, яка кількісно представляє відстань між сусідніми кадрами в порядку відображення, є першою частиною цієї обробки, яка виконується. Даний показник є відстанню, що посилюється на вищезгадане; з його допомогою кожний кадр може бути оцінений для належного типу. Таким чином, рознесення між I- і сусіднім P- або двома послідовними P-кадрами може бути змінним. Обчислення відстані починається за допомогою обробки відеокadrів із заснованим на блоках модулем компенсації руху, причому блок є базовою одиницею відеостиснення, що складається звичайно з  $16 \times 16$  пікселів, хоч можливі інші розміри блоків, наприклад,  $8 \times 8$ ,  $4 \times 4$  і  $8 \times 16$ . Для кадрів, що складають два поля з усуненою черезрядковою розгорткою компенсація руху може бути виконана на основі поля, при цьому пошук опорних блоків виконується швидше в полях, а не в кадрах. Для блока в першому полі поточного кадру прямий опорний блок виявляється в полях кадру, який іде за ним; аналогічно, зворотний опорний кадр вияв-

ляється в полях, які безпосередньо передують поточному полю. Поточні блоки збираються в стиснене поле. Процес продовжується з другим полем кадру. Два компенсованих поля комбінуються так, щоб формувати прямий і зворотний компенсований кадр.

Для кадрів, створених при перетворенні 606 з відеоформату в фільм, пошук опорних блоків виконується тільки на основі кадрів, оскільки присутні тільки відновлені кінокадри. Два опорних блоки і дві різниці, пряма і зворотна, знаходяться, приводячи також до прямого і зворотного компенсованого кадру. Отже, модуль компенсації руху формує вектори руху і показники різниці для кожного блока; але блок є частиною NTSC-поля у випадку виведення оброблюваного модуля 605 усунення черезрядкової розгортки і є частиною кінокадру, якщо виведення перетворення з відеоформату в фільм обробляється. Зазначимо, що різниці в показнику оцінюються між блоком в полі або кадрі, що розглядається, і блоком, який співпадає з ним, або в попередньому полі або кадрі, або в полі або кадрі, який іде відразу за ним, залежно від того, оцінюється пряма або зворотна різниця. Тільки значення яскравості попадають в це обчислення.

Таким чином, етап компенсації руху формує два набори різниць.

Вони знаходяться між блоками поточних значень яскравості і значень яскравості, взятих з кадрів, які знаходяться безпосередньо попереду і безпосередньо позаду поточного кадру у часі. Абсолютне значення кожної прямої і кожної зворотної різниці визначається для кожного пікселя, і кожне окремо підсумовується по всьому кадру. Обидва поля включаються в два підсумовування, коли обробляються NTSC-поля з усуненою черезрядковою розгорткою, які містять кадр. Таким чином,  $SAD_P$  і  $SAD_N$ , підсумовані абсолютні значення прямої і зворотної різниці, знаходяться.

Для кожного кадру співвідношення SAD обчислюється за допомогою взаємозв'язку:

$$\gamma = \frac{\varepsilon + SAD_P}{\varepsilon + SAD_N} \quad (22)$$

де  $SAD_P$  і  $SAD_N$  - це підсумовані абсолютні значення прямих і зворотних різниць, відповідно. Невелике позитивне число додається в чисельник  $\varepsilon$ , щоб не допустити помилки ділення на нуль. Аналогічний член  $\varepsilon$  додається в знаменник, зменшуючи чутливість  $\gamma$ , коли  $SAD_P$  або  $SAD_N$  близьке до нуля.

У альтернативному аспекті різницею може бути SSD, сума квадратів різниць, і SAD, сума абсолютних різниць, або SATD, в якій блоки значень пікселів перетворюються за допомогою застосування дискретного косинусного перетворення до них до того, як беруться різниці в елементах блоків. Суми оцінюються в області активного відео, хоч менша область може бути використана в інших аспектах.

Гістограма яскравості кожного кадру, як приймається (без компенсації руху), також обчислюється. Гістограма працює для коефіцієнта DC, тобто коефіцієнта (0,0) в матриці  $16 \times 16$  коефіцієнтів, яка

є результатом застосування двовимірного дискретного косинусного перетворення до блока значень яскравості, якщо доступні. Еквівалентно середнє значення із 256 значень яскравості в блоці  $16 \times 16$  може бути використане на гістограмі. Для зображень, глибина яскравості яких дорівнює восьми бітам, число елементів зображення задається рівним 16.

Наступний показник оцінює різницю гістограм:

$$\lambda = \frac{1}{N} \sum_{i=1}^{16} |N_{Pi} - N_{Ci}| \quad (23)$$

У вищенаведеному  $N_{Pi}$  - це число блоків з попереднього кадру в  $i$ -тому елементі розрізнення,  $N_{Ci}$  - це число блоків з поточного кадру, який належить  $i$ -тому елементу розрізнення, а  $N$  - це загальне число блоків в кадрі.

Ці проміжні результати збираються з тим, щоб сформувати різницевий показник поточного кадру як:

$$D = \frac{\gamma_C}{\gamma_P} + \lambda \left( \frac{\gamma_C}{\gamma_P} + 1 \right) \quad (24)$$

де  $\gamma_C$  - це співвідношення SAD на основі поточного кадру, а  $\gamma_P$  - це співвідношення SAD на основі попереднього кадру. Якщо сцена має плавний рух, і її гістограма компоненти яскравості практично не змінюється, то  $D \sim 1$ . Якщо поточний кадр відображає різку зміну сцен, то  $\gamma_C$  повинно мати велике значення, а  $\gamma_P$  повинно мати невелике значення. Співвідношення  $\gamma_C/\gamma_P$  використовується замість тільки  $\gamma_C$ , з тим щоб показник був нормалізований до рівня активності контексту.

Фіг. 42 ілюструє процес призначення типів стиснення кадрам.  $D$ , поточна різниця кадрів, задана в рівнянні 19, є основою для рішень, що приймаються відносно призначення кадрів. Як вказує етап 4202 прийняття рішень, якщо кадр, що розглядається, є першим в послідовності, шлях вирішення, помічений "ТАК", іде до етапу 4206, тим самим оголошуючи те, що кадр є I-кадром. Накопичені різниці кадрів задаються рівними нулю на етапі 4208, і процес повертається (на етапі 4210) до початкового етапу. Якщо кадр, що розглядається, не є першим кадром в послідовності, шлях, помічений "НІ", іде з етапу 4202, де прийняте рішення, і на тестовому етапі 4204 поточна різниця кадрів тестується відносно порога зміни сцен. Якщо поточна різниця кадрів перевищує цей поріг, шлях вирішення, помічений "ТАК", іде з етапу 4206, знов приводячи до призначення I-кадру.

Якщо поточна різниця кадрів менше порога зміни сцен, шлях "НІ" іде до етапу 4212, де поточна різниця кадрів додається до накопиченої різниці кадрів. Продовжуючи проходити блок-схему послідовності операцій способом на етапі 4214 прийняття рішення, накопичена різниця кадрів порівнюється з порогом  $t$ , який, загалом, менше порога зміни сцен. Якщо накопичена різниця кадрів перевищує  $t$ , керування переходить до етапу 4216, і кадр призначається так, щоб бути P-кадром; накопичена різниця кадрів потім скидається до нуля на етапі

4218. Якщо накопичена різниця кадрів менше  $t$ , керування переходить від етапу 4214 до етапу 4220. Тут накопичена різниця кадрів порівнюється з  $x$ , яке менше  $t$ . Якщо поточна різниця кадрів менше  $t$ , кадр призначається як пропущений на етапі 4222, і потім процес повертається; якщо поточна різниця кадрів більше  $t$ , кадр призначається так, щоб бути В-кадром, на етапі 4226.

#### Кодер

Посилаючись знов на Фіг. 2, транскодер 200 включає в себе кодер 228, який приймає оброблені метадані і необроблене відео з препроцесора 226. Метадані можуть включати в себе будь-яку інформацію, початково прийняту у вихідному відео 104, і будь-яку інформацію, обчислену за допомогою препроцесора 226. Кодер 228 включає в себе кодер 230 першого проходу, кодер 232 другого проходу і повторний кодер 234. Кодер 228 також приймає вхідні дані з керування 231 транскодером, яке може надавати інформацію (наприклад, метадані, інформацію стійкості до помилок, інформацію вмісту, інформацію кодованої швидкості передачі бітів, інформацію балансу базового рівня і рівня поліпшення і інформацію квантування) з кодера 232 другого проходу в кодер 230 першого проходу, повторний кодер 234, а також препроцесор 226. Кодер 228 кодує відео, що приймається, за допомогою інформації вмісту, прийнятої від препроцесора 226, і/або інформації вмісту, яка формується за допомогою самого кодера 228, наприклад, за допомогою модуля 712 класифікації вмісту (Фіг. 7).

Фіг. 7 ілюструє блок-схему функціональних модулів, які можуть бути включені в зразковий двопротічний кодер, який може бути використаний для кодера 228, проілюстрованого на Фіг. 7, хоч Фіг. 7 і опис в даному документі не обов'язково дозволяють всю функціональність, яка може бути включена в кодер. Відповідно, визначені аспекти функціональних модулів описуються далі після нижченаведених опису кодування базового рівня і рівня поліпшення.

#### Кодування базового рівня і рівня поліпшення

Кодером 228 може бути масштабований SNR-кодер, який може кодувати необроблене відео і метадані з препроцесора 226 в першу групу кодованих даних, також згадувану в даному документі як базовий рівень, і одну або більше додаткових груп кодованих даних, також згадуваних в даному документі як рівні поліпшення. Алгоритм кодування формує коефіцієнти базового рівня і рівня поліпшення, які, коли декодовані, можуть бути комбіновані в декодері, коли обидва рівні доступні для декодування. Коли обидва рівні недоступні, кодування базового рівня дозволяє їх декодувати як один рівень.

Один аспект такого процесу багаторівневого кодування описаний в посиланні до Фіг. 31. На етапі 321 І-кадр кодується з повністю внутрішньо кодованими макроблоками (внутрішньо кодованими MB). У H.264 внутрішньо кодовані MB в І-кадрах кодуються з повністю використовуваним просторовим прогнозом, який надає значну величину поліпшення кодування. Передбачено два підрежими: внутрішній 4×4 і внутрішній 16×16. Якщо базовий рівень повинен використовувати перевагу поліпшення кодування, що надається за

допомогою просторового прогнозу, то базовий рівень повинен бути кодований і декодований перед кодуванням і декодуванням рівня поліпшення. Використовується двопротічне кодування і декодування І-кадрів. У базовому рівні параметр квантування базового рівня  $QP_b$  надає коефіцієнтам перетворення грубий розмір кроку квантування. Попіксельна різниця між вихідним кадром і відновленим кадром базового рівня кодується на рівні поліпшення. Рівень поліпшення використовує параметр квантування  $QP_e$ , який надає більш точний розмір кроку квантування. Засіб кодування, такий як кодер 228 за Фіг. 2, може виконувати кодування на етапі 321.

На етапі 323 кодер кодує дані базового рівня і дані рівня поліпшення для Р- і/або В-кадрів в оброблюваній GOP. Засіб кодування, такий як кодер 228, може виконувати кодування на етапі 323. На етапі 325 процес кодування перевіряє, чи є ще Р- або В-кадри, щоб кодувати. Засіб кодування, такий як масштабований SNR-кодер 228, може виконувати етап 325. Якщо ще Р- або В-кадри залишаються, етап 323 повторюється доти, поки не буде закінчене кодування всіх кадрів в GOP. Р- і В-кадри складаються із зовнішньо кодованих макроблоків (зовнішньо кодованих MB), хоч можуть бути внутрішньо кодовані MB в Р- і В-кадрах, як описано нижче.

Щоб декодер розрізняв дані базового рівня і рівня поліпшення, кодер 228 кодує службову інформацію, етап 327. Типи службової інформації включають в себе, наприклад, дані, що ідентифікують число рівнів, дані, що ідентифікують рівень як базовий рівень, дані, що ідентифікують рівень як рівень поліпшення, дані, що ідентифікують взаємозв'язки між рівнями (наприклад, рівень 2 є рівнем поліпшення для базового рівня 1 або рівень 3 є рівнем поліпшення для рівня поліпшення 2), або дані, що ідентифікують рівень як кінцевий рівень поліпшення в рядку рівнів поліпшення. Службова інформація може міститися в заголовках, пов'язаних з даними базового і/або рівня поліпшення, до яких вона належить, або міститися в окремих повідомленнях даних. Засіб кодування, такий як кодер 228 за Фіг. 2, може виконувати процес на етапі 327.

Щоб мати однорівневе кодування, коефіцієнти двох рівнів можуть бути комбіновані до зворотного квантування. Отже, коефіцієнти двох рівнів повинні бути сформовані інтерактивно, в іншому випадку це може вводити значний обсяг службової інформації. Одна причина збільшення службової інформації полягає в тому, що кодування базового рівня і кодування рівня поліпшення може використовувати різні часові посилення. Потрібний алгоритм для того, щоб формувати коефіцієнти базового рівня і рівня поліпшення, які можуть бути комбіновані в декодері до деквантування, коли обидва рівні доступні. У той же час, алгоритм повинен надавати прийнятне відео базового рівня, коли рівень поліпшення недоступний, або декодер ухвалює рішення декодувати рівень поліпшення через такі причини, як, наприклад, економія енергії. Подробиці ілюстративного прикладу такого процесу додатково описані нижче в контексті короткого

пояснення стандартного кодування з прогнозом безпосередньо далі.

P-кадри (або зовнішньо кодовані секції) можуть використовувати часову надмірність між зоною в поточному зображенні і найбільш співпадаючою зоною прогнозування в опорному зображенні. Місцеположення найбільш співпадаючої зони прогнозування в опорному зображенні може бути кодоване у векторі руху. Різниця між поточною зоною і найбільш співпадаючою опорною зоною прогнозування відома як залишкова помилка (або помилка прогнозу).

Фіг. 32 - це ілюстрація прикладу процесу складання P-кадрів, наприклад, в MPEG-4. Процес 331 є більш докладною ілюстрацією зразкового процесу, який може виконуватися на етапі 323 за Фіг. 31. Процес 331 включає в себе поточне зображення 333, складене з 5×5 макроблоків, при цьому число макроблоків в даному прикладі є довільним. Макроблок складений максимум з 16×16 пікселів. Піксели можуть бути задані як 8-бітове значення компоненти яскравості (Y) і два 8-бітових значення компоненти кольоровості (Cr і Cb). У MPEG компоненти Y, Cr і Cb можуть бути збережені в форматі 4:2:0, при цьому компоненти Cr і Cb піддані знижувальній дискретизації на 2 в напрямках X і Y. Отже, кожний макроблок повинен складатися з 256 компонент Y, 64 компонент Cr і 64 компонент Cb. Макроблок 335 поточного зображення 333 прогнозується з опорного зображення 337 в точці часу, відмінній від поточного зображення 333. У опорному зображенні 337 виконується пошук, щоб знайти найкращим чином співпадаючий макроблок 339, який є найближчим відносно значень Y, Cr і Cb до поточного макроблока, що кодується 335. Розміщення найкращим чином співпадаючого макроблока 339 в опорному зображенні 337 кодується у векторі 341 руху. Опорним зображенням 337 може бути I-кадр або P-кадр, який декодер повинен мати відновленим до складання поточного зображення 333. Найкращим чином співпадаючий макроблок 339 віднімається з поточного макроблока 335 (різниця для кожної з компонент Y, Cr і Cb обчислюється), приводячи до залишкової помилки 343. Залишкова помилка 343 кодується за допомогою двомірного дискретного косинусного перетворення (DCT) 345 і потім квантується 347. Квантування 347 може виконуватися для того, щоб надавати просторове стиснення, наприклад, за допомогою призначення меншого числа бітів високочастотним компонентам, при цьому призначаючи більше бітів низькочастотним компонентам. Квантовані коефіцієнти залишкової помилки 343 нарівні з ідентифікуючою інформацією вектора 341 руху і опорного зображення 333 є кодованою інформацією, що представляє поточний макроблок 335. Кодована інформація може бути збережена в запам'ятовуючому пристрої для майбутнього використання або оброблена, наприклад, для цілей корекції помилок або поліпшення зображення, або передана по мережі 349.

Кодовані квантовані коефіцієнти залишкової помилки 343 нарівні з кодованим вектором 341 руху можуть бути використані для того, щоб відновлювати поточний макроблок 335 в кодері для використання як частини опорного кадру для по-

дальшої оцінки і компенсації руху. Кодер може моделювати процедури декодера для даного відновлення P-кадрів. Моделювання в декодері приводить до того, що кодер і декодер працюють з одним опорним зображенням. Процес відновлення, виконуваний або в кодері для подальшого зовнішнього кодування, або в декодері, представлений тут. Відновлення P-кадру може бути почате після того, як опорний кадр (або частина зображення або кадру, який вказується посиленням) відновлений. Кодовані квантовані коефіцієнти деквантуються 351, а потім виконується двовимірне зворотне DCT, або IDCT, 353, приводячи до декодованої або відновленої залишкової помилки 355. Кодований вектор 341 руху декодується і використовується для того, щоб знаходити вже відновлений найбільш співпадаючий макроблок 357 у вже відновленому опорному зображенні 337. Відновлена залишкова помилка 355 потім додається до відновленого найбільш співпадаючого макроблока 357, щоб сформувати відновлений макроблок 359. Відновлений макроблок 359 може бути збережений в запам'ятовуючому пристрої, відображений незалежно або в зображенні з іншими відновленими макроблоками або оброблений додатково для поліпшення зображень.

B-кадри (або будь-яка секція, кодована з двоспрямованим прогнозом) можуть використовувати часову надмірність між зоною в поточному зображенні і найбільш співпадаючою зоною прогнозування в попередній зоні, найбільш співпадаючою зоною прогнозування в подальшій зоні. Подальша найбільш співпадаюча зона прогнозування і попередня найбільш співпадаюча зона прогнозування комбінуються, щоб сформувати двоспрямовану прогнозу зону. Різниця між зоною поточного зображення і найбільш співпадаючою опорною зоною прогнозування відома як залишкова помилка (або помилка прогнозу). Розміщення найбільш співпадаючої зони прогнозування в подальшому опорному зображенні найбільш співпадаючої зони прогнозування в попередньому опорному зображенні може бути кодоване в двох векторах руху.

Фіг. 33 ілюструє приклад процесу кодера для кодування коефіцієнтів базового рівня і рівня поліпшення, яке може бути виконане за допомогою кодера 228. Базовий рівень і рівень поліпшення кодуються, щоб надати масштабований потік бітів SNR. Фіг. 33 ілюструє приклад для кодування коефіцієнтів залишкової помилки між MB, як, наприклад, повинно бути виконано на етапі 323 за Фіг. 31. Проте, аналогічні способи можуть бути використані для того, щоб також кодувати коефіцієнти всередині MB. Засіб кодування, такий як компонент 228 кодера за Фіг. 2, може виконувати процес, проілюстрований на Фіг. 33 і етапі 323 на Фіг. 32. Вихідні (які повинні бути кодовані) відеодані 406 (відеодані містять інформацію компонентів яскравості і кольоровості в цьому прикладі) вводяться в контур 302 найбільш співпадаючого макроблока базового рівня і контур 365 найбільш співпадаючого макроблока рівня поліпшення. Призначення обох контурів 363 і 365 полягає в тому, щоб мінімізувати залишкову помилку, яка обчислюється в суматорах 367 і 369, відповідно. Контури 363 і 365 можуть здійснюватися паралельно, як

показано, або послідовно. Контури 363 і 365 включають в себе логіку для виконання пошуку в буферах 371 і 373, відповідно, які містять опорні кадри, щоб ідентифікувати найбільш співпадаючий макроблок, який мінімізує залишкову помилку між найбільш співпадаючим макроблоком і вихідними даними 361 (буфери 371 і 373 можуть бути одним і тим же буфером). Залишкові помилки контуру 363 і контуру 365 відрізняються, оскільки контур 363 базового рівня, загалом, використовує більш грубий розмір кроку квантування (більш високе значення QP), ніж контур 365 рівня поліпшення. Блоки 375 і 377 перетворення перетворюють залишкові помилки кожного контуру.

Перетворені коефіцієнти потім розбираються на коефіцієнти базового рівня і рівня поліпшення в модулі 379 вибору. Розбір модуля 379 вибору може виконуватися в декількох формах, як описано нижче. Одна загальна ознака методик розбору полягає в тому, що коефіцієнт рівня поліпшення,  $C'_{enh}$ , обчислюється так, що він є диференціальним уточненням до коефіцієнта базового рівня  $C'_{base}$ . Обчислення рівня поліпшення, щоб бути уточненням до базового рівня, дає можливість декодеру декодувати коефіцієнт базового рівня самостійно і мати прийнятне представлення зображення або комбінувати коефіцієнти базового рівня і рівня поліпшення і мати уточнене представлення зображення. Коефіцієнти, вибрані за допомогою модуля 379 вибору, потім квантуються за допомогою квантувачів 381 і 383. Квантовані коефіцієнти  $C'_{base}$  і  $C'_{enh}$  (обчислені з допомогою квантувачів 381 і

383, відповідно) можуть бути збережені в запам'ятовуючому пристрої або передані по мережі в декодер.

Щоб зіставити відновлення макроблока в декодері, деквантувач 385 деквантує коефіцієнти залишкової помилки базового рівня. Деквантовані коефіцієнти залишкової помилки зворотно перетворюються 387 і додаються 389 до найбільш співпадаючого макроблока, знайденого в буфері 371, приводячи до відновленого макроблока, який співпадає з тим, що повинно бути відновлено в декодері. Квантувач 383, деквантувач 391, зворотний перетворювач 393, суматор 397 і буфер 373 виконують в контурі 365 рівня поліпшення обчислення, аналогічні тим, які виконувалися в контурі 363 базового рівня. Крім того, суматор 393 використовується для того, щоб комбінувати зворотно квантовані коефіцієнти рівня поліпшення і базового рівня, використовувати при відновленні рівня поліпшення. Квантувач і деквантувач рівня поліпшення, загалом, використовують більш точний розмір кроку квантувача (менше QP), ніж базовий рівень.

Фіг. 34, 35 і 36 ілюструють приклади процесів модуля вибору коефіцієнтів базового рівня і рівня поліпшення, які можуть використовуватися в модулі 379 вибору за Фіг. 33. Засіб вибору, такий як кодер 228 за Фіг. 2, може виконувати процеси, проілюстровані на Фіг. 34, 35 і 36. Використовуючи Фіг. 34 як приклад, перетворені коефіцієнти розбираються на коефіцієнти базового і рівня поліпшення, як показано в наступних рівняннях:

$$C'_{base} = \begin{cases} 0, \text{ якщо } C_{base} \text{ і } C_{enh} & \text{протилежні за знаком} \\ \min(C_{base}, C_{enh}) & \text{в іншому випадку} \end{cases} \quad (25)$$

$$C'_{enh} = C_{enh} - Q_b^{-1}(Q_b(C'_{base})) \quad (26)$$

де функція  $\min$  може бути або математичним мінімумом, або мінімальною величиною з двох аргументів. Рівняння 25 проілюстроване як етап 401, а рівняння 26 проілюстроване як суматор 510 на Фіг. 34. У рівнянні 26  $Q_b$  означає квантувач 381 базового рівня, а  $Q_b^{-1}$  означає деквантувач 385 базового рівня. Рівняння 26 перетворює коефіцієнт

рівня поліпшення в диференціальне уточнення коефіцієнта базового рівня, обчисленого за допомогою рівняння 25.

Фіг. 35 - це ілюстрація ще одного прикладу модуля 379 вибору коефіцієнтів базового рівня і рівня поліпшення. У цьому прикладі рівняння (.), що міститься на етапі 405, представляє наступне:

$$C'_{base} = \begin{cases} 0, \text{ якщо } C_{base} \text{ і } C_{enh} & \text{протилежні за знаком} \\ \min(C_{base}, C_{enh}) & \text{в іншому випадку} \end{cases} \quad (27)$$

Суматор 407 обчислює коефіцієнт рівня поліпшення, як показано в наступних двох рівняннях:

$$C'_{enh} = C_{enh} - Q_b^{-1}(Q_b(C'_{base})) \quad (28)$$

де  $C_{base}$  задається за допомогою рівняння 27.

Фіг. 36 - це ілюстрація ще одного прикладу модуля 379 вибору коефіцієнтів базового рівня і рівня поліпшення. У цьому прикладі коефіцієнт базового рівня залишається незмінним, і рівень поліпшення дорівнює різниці між квантованим/деквантованим коефіцієнтом базового рівня і вихідним коефіцієнтом рівня поліпшення.

Крім коефіцієнтів залишкової помилки базового рівня і рівня поліпшення, декодеру потрібна інформація, яка ідентифікує те, як кодовані МВ.

Засіб кодування, такий як компонент 228 кодера за Фіг. 2, може кодувати службову інформацію, яка може включати в себе таблицю внутрішньо кодованих і зовнішньо кодованих частин, таку як, наприклад, таблиця МВ, в якій макроблоки (або субмакроблоки) ідентифікуються як внутрішньо кодовані або зовнішньо кодовані (також ідентифікуючи те, який тип зовнішнього кодування, в тому числі, наприклад, пряме, зворотне або двоспрямоване), і на які зовнішньо кодовані частини кадру(ів) є посилання. У зразковому аспекті таблиця МВ і коефіцієнти базового рівня кодуються а базовому рівні, а коефіцієнти рівня поліпшення кодуються в рівні поліпшення.

Р-кадри і В-кадри можуть містити внутрішньо кодовані МВ, а також між-МВ. Загальнопоширено

для гібридних відеокoderів використовувати оптимізацію співвідношення "швидкість передачі-спотворення" (RD), щоб ухвалювати рішення кодувати конкретні макроблоки в Р- або В-кадрах як внутрішньо кодовані МВ. Щоб мати однорівневе кодування, при якому внутрішньо кодовані МВ не залежать від між-МВ рівня поліпшення, все сусідні між-МВ не використовуються для просторового прогнозу внутрішньо кодованих МВ базового рівня. Щоб зберігати обчислювальну складність незмінною для декодування рівня поліпшення, для внутрішньо кодованих МВ в Р- або В-кадрі базового рівня уточнення на рівні поліпшення може бути пропущене.

Внутрішньо кодовані МВ в Р- або В-кадрах вимагають набагато більше бітів, ніж між МВ. З цієї причини внутрішньо кодовані МВ в Р- або В-кадрах можуть кодуватися тільки з якістю базового рівня при більш високому QP. Це приводить до певного погіршення якості відео, але дане погіршення повинно бути непомітним, якщо воно усувається в подальшому кадрі за допомогою коефіцієнтів між-МВ в базовому рівні і рівні поліпшення, як описано вище. Дві причини роблять це погіршення непомітним. Перша - це ознака зорової системи людини (HVS), а друга - це те, що між-МВ поліпшують внутрішні МВ. З об'єктами, які змінюють позицію з першого кадру на другий кадр, деякі піксели в першому кадрі не видні у другому кадрі (інформація, яка повинна бути прихована), а деякі піксели у другому кадрі видні для першого разу (неприхована інформація). Очі людини нечутливі до неприхованої і яка повинна бути прихованою візуальної інформації. Таким чином, для неприхованої інформації, навіть якщо вона кодована з більш низькою якістю, очі можуть не повідомити відмінність. Якщо та ж інформація залишається в наступному Р-кадрі, є висока імовірність того, що наступний Р-кадр на рівні поліпшення може поліпшити її, оскільки рівень поліпшення має менше QP.

Інша стандартна методика, яка вводить внутрішньо кодовані МВ в Р- або В-кадри, відома як внутрішнє оновлення. У цьому випадку деякі МВ кодуються як внутрішньо кодовані МВ, навіть якщо стандартна оптимізація R-D повинна диктувати те, що вони повинні бути зовнішньо кодованими МВ. Ці внутрішньо кодовані МВ, що містяться в базовому рівні, можуть бути кодовані за допомогою  $QP_b$  або  $QP_e$ . Якщо  $QP_e$  використовується для базового рівня, то уточнення не потрібне на рівні поліпшення. Якщо  $QP_b$  використовується для базового рівня, то уточнення може потребуватися, інакше на рівні поліпшення буде помітним погіршення якості. Оскільки зовнішнє кодування більш ефективне, ніж внутрішнє кодування, відносно ефективності кодування, ці уточнення на рівні поліпшення внутрішньо кодуються. Таким чином, коефіцієнти базового рівня не використовуються для рівня поліпшення. Отже, якість підвищується на рівні поліпшення без введення нових операцій.

В-кадри, як правило, використовуються на рівні поліпшення внаслідок високої якості стиснення, яке вони пропонують. Проте, В-кадри можуть не мати опорних внутрішньо кодованих МВ Р-кадру. Якщо піксели В-кадру повинні бути кодовані з якіс-

тю рівня поліпшення, це може потребувати дуже великого числа бітів внаслідок більш низької якості внутрішньо кодованих МВ Р-кадру, як описано вище. За допомогою використання переваги якості HVS, як описано вище, МВ В-кадру можуть бути кодовані з більш низькою якістю при посиленні на внутрішньо кодовані МВ Р-кадрів більш низької якості.

Інший граничний випадок внутрішньо кодованих МВ в Р- або В-кадрах - це коли всі МВ в Р- або В-кадрі кодовані у внутрішньому режимі внаслідок наявності зміни сцен у відео, що кодується. У цьому випадку весь кадр може бути кодований з якістю базового рівня і без уточнення на рівні поліпшення. Якщо зміна сцен відбувається в В-кадрі і за умови, що В-кадри кодуються тільки на рівні поліпшення, то В-кадр може бути кодований з якістю базового рівня або просто відкинтий. Якщо зміна сцен відбувається в Р-кадрі, зміни можуть не потребуватися, але Р-кадр може бути відкинтий або кодований з якістю базового рівня. Масштабоване рівневе кодування додатково описане в Патентній заявці (США), що знаходиться одночасно на розгляді, номер [Адвокатська виписка номер 050078], яка озаглавлена "Scalable Video Coding With Two Layer Encoding And Single Layer Decoding" і належить її правонаступнику, і повністю міститься по посиланню в даному документі.

Частина першого проходу кодера

Фіг. 7 показує ілюстративний приклад кодера 228 за Фіг. 2. Показані етапи ілюструють різну обробку в кодері, яка може бути включена в кодер 228. У цьому прикладі кодер 228 включає в себе частину 706 першого проходу вище розділової лінії 704 і частину 706 другого проходу (включаючи функціональність кодера 232 другого проходу і повторного кодера 234 на Фіг. 2) під лінією 704.

Кодер 228 приймає метадані і необроблене відео від препроцесора 226. Метадані можуть включати в себе будь-які метадані, прийняті або обчислені за допомогою препроцесора 226, в тому числі метадані, пов'язані з інформацією вмісту відео. Частина 702 першого проходу кодера 228 ілюструє зразкові процеси, які можуть бути включені в кодування 702 першого проходу, яке описується нижче відносно функціональності. Фахівці в даній галузі техніки повинні знати, що ця функціональність може бути здійснена в різних формах (наприклад, апаратні засоби, програмне забезпечення, мікропрограмне забезпечення або комбінація вищезгаданого).

Фіг. 7 ілюструє модуль адаптивного внутрішнього оновлення (AIR). AIR-модуль 710 надає вхідні дані в модуль 708 створення екземплярів І-кадрів, який створює екземпляр І-кадру на основі метаданих. Частина 702 першого проходу також може включати в себе модуль 712 класифікації вмісту, сконфігурований так, щоб приймати метадані і відео і визначати інформацію вмісту, пов'язану з відео. Інформація вмісту може бути надана в модуль 714 призначення бітів керування швидкістю передачі, який також приймає метадані і відео. Модуль 714 призначення бітів керування визначає інформацію керування призначенням бітів і надає її в модуль 715 визначення режиму. Інформація вмісту і відео може надаватися в модуль 716 внут-



рішньої моделі (спотворень), який надає інформацію спотворень внутрішнього кодування в модуль 715 визначення режиму і модуль 718 масштабованості співвідношення "швидкість передачі-спотворення" для базового і рівня поліпшення. Відео- і метадані надаються в модуль 720 оцінки руху (спотворень), який надає інформацію спотворень зовнішнього кодування в модуль 718 масштабованості співвідношення "швидкість передачі-спотворення" для базового рівня і рівня поліпшення. Модуль 718 масштабованості співвідношення "швидкість передачі-спотворення" для базового рівня і рівня поліпшення визначає інформацію масштабованості співвідношення "швидкість передачі-спотворення" за допомогою оцінок спотворення з модуля 720 оцінки руху і модуля 716 спотворень внутрішньої моделі, яка надається в модуль 715 визначення режиму. Модуль 715 визначення режиму також приймає вхідні дані з модуля 722 упорядкування серій макроблоків/МВ. Модуль 722 упорядкування серій макроблоків/МВ приймає вхідні дані з модуля 740 стійкості до помилок (показаного в частині 706 другого проходу) і надає інформацію по поєднанню незалежно кодованих частин відео (серій макроблоків) з границями одиниць доступу для стійкості до помилок в модуль 715 визначення режиму. Модуль 715 визначення режиму визначає інформацію режиму кодування на основі вхідних даних і надає "оптимальний" режим кодування в частину 706 другого проходу. Додаткове ілюстративне пояснення деяких прикладів такого кодування в частині 702 першого проходу наводиться далі.

Як указано вище, модуль 712 класифікації вмісту приймає метадані і необроблене відео, що надається за допомогою препроцесора 226. У деяких прикладах препроцесор 226 обчислює інформацію вмісту з мультимедійних даних і надає інформацію вмісту в модуль 712 класифікації вмісту (наприклад, в метаданих), який може використовувати інформацію вмісту для того, щоб визначати класифікацію вмісту для мультимедійних даних. У деяких інших аспектах модуль 712 класифікації вмісту сконфігурований так, щоб визначати різну інформацію вмісту з мультимедійних даних, і також може бути сконфігурований так, щоб визначати класифікацію вмісту.

Модуль 712 класифікації вмісту може бути сконфігурований так, щоб визначати різну класифікацію вмісту для відео, що має різні типи вмісту. Різна класифікація вмісту може приводити до різних параметрів, використовуваних в аспектах кодування мультимедійних даних, наприклад, визначення швидкості передачі бітів (наприклад, призначення бітів) для визначення параметрів квантування, оцінка руху, масштабованість, стійкість до помилок, підтримання оптимальної якості мультимедійних даних в каналах, а також для схем швидкого перемикавання каналів (наприклад, періодична активація І-кадрів, щоб забезпечити швидке перемикавання каналів). Згідно з одним прикладом, кодер 228 сконфігурований так, щоб визначати оптимізацію співвідношення "швидкість передачі-спотворення" (R-D) і призначення швидкості передачі бітів на основі класифікації вмісту. Визначення класифікації вмісту дає можливість

мультимедійним даним бути стисливими для заданого рівня якості згідно з необхідною швидкістю передачі бітів на основі класифікації вмісту. Крім того, за допомогою класифікації вмісту мультимедійних даних (наприклад, визначення класифікації вмісту на основі зорової системи людини) результуюча перцепційна якість мультимедійних даних, що передаються, на дисплеї приймального пристрою залежить від відеовмісту.

Як приклад процедури, яку проходить модуль 712 класифікації вмісту для того, щоб класифікувати вміст, Фіг. 9 ілюструє процес 900, що ілюструє зразковий процес, за допомогою якого може працювати модуль 712 класифікації вмісту. Як показано, процес 900 починається на етапі 902 введення, де модуль 712 класифікації вмісту приймає необроблені мультимедійні дані і метадані. Далі процес 900 переходить до етапу 904, на якому модуль 712 класифікації вмісту визначає просторову інформацію і часову інформацію мультимедійних даних. У деяких аспектах просторова і часова інформація визначаються за допомогою просторового і часового маскування (наприклад, фільтрації). Просторова і часова інформація може бути визначена на основі метаданих, які включають в себе дані зміни сцен і згладжування векторів руху (MV). Процес 900 далі переходить до етапу 912, який виконує оцінки просторової складності, часової складності і чутливості. Процес 900 потім переходить до етапу 916, на якому вміст мультимедійних даних класифікується на основі результатів визначених просторових, часових даних і даних чутливості, на етапах 904 і 912. Крім того, на етапі 916 конкретна крива співвідношення "швидкість передачі-спотворення" (R-D) може бути вибрана, і/або дані кривої R-D можуть бути оновлені. Процес 900 далі переходить до етапу 918 виведення, при цьому виведення може включати в себе таблицю складності-спотворення або значення, що вказує просторову або часову діяльність (наприклад, класифікацію вмісту) і/або вибрану криву R-D. Посилаючись знов на Фіг. 7, модуль 712 класифікації вмісту надає виведення в модуль 714 призначення бітів керування швидкістю передачі, модуль внутрішньої моделі 716 (спотворень), а також в модуль 708 створення екземплярів І-кадрів, описаний вище.

Інформація вмісту

Модуль 712 класифікації вмісту може бути сконфігурований так, щоб обчислювати різноманітність інформації вмісту з мультимедійних даних, в тому числі різноманітність пов'язаних з вмістом показників, включаючи просторову складність, часову складність, значення коефіцієнта контрастності, стандартні відхилення і різниці показників кадрів, додатково описані нижче.

Модуль 712 класифікації вмісту може бути сконфігурований так, щоб визначати просторову складність і часову складність мультимедійних даних, а також асоціативно зв'язувати значення текстури з просторовою складністю і значення руху з часовою складністю. Модуль 712 класифікації вмісту приймає попередньо оброблену інформацію вмісту, пов'язану з вмістом мультимедійних даних, що кодуються з препроцесора 226, або альтернативно, препроцесор 226 може бути скон-

фігурований так, щоб обчислювати інформацію вмісту. Як описано вище, інформація вмісту може включати в себе, наприклад, одне або більше значень  $D_{csat}$ , значення коефіцієнта контрастності, вектори руху (MV) і суму абсолютних різниць (SAD).

Загалом, мультимедійні дані включають в себе одну або більше послідовностей зображень, або кадри. Кожний кадр може бути розбитий на блоки пікселів для обробки. Просторова складність - це широкий термін, який, загалом, описує міру рівня просторових деталей в кадрі. Сцени з переважно плоскими або незмінюваними або мало змінюваними областями компонента яскравості і кольоровості мають низьку просторову складність. Просторова складність асоціативно зв'язана з текстурою відеоданих. Просторова складність, в даному аспекті, заснована на показнику зорової чутливості людини, званому  $D_{csat}$ , який обчислюється для кожного блока як функція від локальної просторової частоти і навколишнього освітлення. Фахівці в даній галузі техніки знають методики використання шаблонів просторової частоти і характеристик освітлення і контрастності візуальних зображень, щоб використовувати перевагу зорової системи людини. Ряд показників чутливості відомий для використання переваг перспективних обмежень зорової системи людини і може бути використаний зі способом, описаним в даному документі.

Часова складність - це широкий термін, який використовується для того, щоб, загалом, описувати міру рівня руху в мультимедійних даних, що указуються посиланням між кадрами в послідовності кадрів. Сцени (наприклад, послідовності кадрів відеоданих) з незначним рухом або без руху мають низьку часову складність. Часова складність може бути обчислена для кожного макроблока і може бути заснована на значенні  $D_{csat}$ , векторах руху і сумі абсолютних піксельних різниць між одним кадром і іншим кадром (наприклад, опорним кадром).

Різницеви показник кадрів надає міру різниці між двома послідовними кадрами з урахуванням руху (наприклад, вектора руху або MV) нарівні із залишковою енергією, представленою як сума абсолютної різниці (SAD) між прогнозним параметром і поточним макроблоком. Різниця кадрів також надає міру двоспрямованої і односпрямованої ефективності прогнозу.

Один приклад різницевого показника кадрів, заснованого на інформації руху, яка приймається від препроцесора, що потенційно виконує усунен-

ня черезрядкової розгортки з компенсацією руху, такий. Модуль усунення черезрядкової розгортки виконує двоспрямовану оцінку руху, і тим самим двоспрямований вектор руху і інформація SAD доступні. Різниця кадрів, що представляється за допомогою SAD\_MV для кожного макроблока, може бути витягнута таким чином:

$$SAD\_MV = \log_{10}[SAD * \exp(-\min(1, MV))] \quad (29)$$

де  $MV = \text{Square\_root}(MV_x^2 + MV_y^2)$ ,  $SAD = \min(SAD_N, SAD_P)$ , де  $SAD_N$  - це SAD, обчислена із зворотного опорного кадру, а  $SAD_P$  - це SAD, обчислена з прямого опорного кадру.

Інший підхід оцінки різниці кадрів описаний вище з посиланням на рівняння 6-8. Співвідношення SAD (або коефіцієнт контрастності)  $\gamma$  може бути обчислене так, як описано вище в рівнянні 6. Гістограма яскравості кожного кадру також може бути визначена, при цьому різниці гістограми  $\lambda$  обчислюється за допомогою рівняння 7. Різницеви показник кадрів D може бути обчислений так, як показано в рівнянні 8.

У одному ілюстративному прикладі коефіцієнт контрастності і різницеви показник кадрів використовуються таким чином для того, щоб одержати класифікацію відеовмісту, яка дозволяє надійно прогнозувати ознаки в даній відеопослідовності. Хоч описується в даному документі як виконуване в кодері 228, препроцесор 226 також може бути сконфігурований для того, щоб визначати класифікацію вмісту (або іншу інформацію вмісту) і передавати класифікацію вмісту в кодер 228 за допомогою метаданих. Процес, описаний в прикладі нижче, класифікує вміст на вісім можливих класів, аналогічно класифікації, одержаній з аналізу на основі кривих R-D. Процес класифікації виводить значення в діапазоні від 0 до 1 для кожного суперкадру на основі складності сцени і числа подій зміни сцен в цьому суперкадрі. Модуль класифікації вмісту в препроцесорі може виконувати наступні етапи (1)-(5) для кожного суперкадру, щоб одержати показник класифікації вмісту із значень контрастності кадрів і різниці кадрів.

1. Обчислення середньої контрастності кадрів і відхилення контрастності кадрів із значень контрастності макроблоків.

2. Нормалізація значень контрастності кадрів і різниці кадрів за допомогою значень, одержаних з моделювання, тобто 40 і 5, відповідно.

3. Обчислення показника класифікації вмісту за допомогою, наприклад, узагальненого рівняння:

$$CCMetric = CCW1 * I\_Frame\_Contrast\_Mean + CCW2 * Frame\_Difference\_Mean - CCW3 * I\_Contrast\_Deviation^2 * \exp(CCW4 * Frame\_Difference\_Deviation^2) \quad (30)$$

де CCW1, CCW2, CCW3 і CCW4 - це вагові коефіцієнти. У цьому прикладі значення вибираються рівними 0,2 для CCW1, 0,9 для CCW2, 0,1 для CCW3 і -0,00009 для CCW4.

4. Визначення числа зміни сцен в суперкадрі. Загалом, суперкадр означає групу зображень або кадрів, яка може бути відображена в конкретному періоді часу. У типовому варіанті період часу

складає одну секунду. У деяких аспектах суперкадр містить 30 кадрів (для відео 30 кадрів в секунду). У інших аспектах суперкадр містить 24 кадри (для відео 24 кадру в секунду). Залежно від числа змін сцен один з наступних випадків приводиться у виконання.

(а) Немає зміни сцен. Коли немає зміни сцен в суперкадрі, показник повністю залежить тільки від

значень різниці кадрів, як показано в наступному

рівнянні:

$$CCMetric = (CCW2 + (CCW1/2) * Frame\_Difference\_Mean + CCW3 - (CCW1/2) * 1 * \exp(CCW4 * Frame\_Difference\_Deviation^2)) \quad (31)$$

(b) Одна зміна сцен. Коли є кадр з однією зміною сцен, що спостерігається в суперкадрі, рівняння за умовчанням повинно бути використано

не для того, щоб обчислювати показник, як показано нижче:

$$CCMetric = CCW1 * I\_Frame\_Contrast\_Mean + CCW2 * Frame\_Difference\_Mean - CCW3 * I\_Contrast\_Deviation^2 * \exp(CCW4 * Frame\_Difference\_Deviation^2) \quad (32)$$

(c) Дві зміни сцен. Коли спостерігається, що є саме більше 2 зміни сцен в даному суперкадрі, останньому суперкадру призначається більший ваговий коефіцієнт, ніж першому, оскільки пер-

ший в будь-якому випадку повинен бути оновлений за допомогою другого швидко, як показано в наступному рівнянні:

$$CCMetric = 0.1 * I\_Frame\_Contrast\_Mean1 + CCW1 * I\_Frame\_Contrast\_Mean2 + (CCW2 - 0.1) * Frame\_Difference\_Mean - CCW3 * I\_Contrast\_Deviation1^2 * I\_Contrast\_Deviation2^2 * \exp(CCW4 * Frame\_Difference\_Deviation^2) \quad (33)$$

(d) Три або більше змін сцен. Якщо в даному суперкадрі спостерігається більше 3 I-кадрів (скажімо, N), останньому I-кадру надається біль-

ший ваговий коефіцієнт, а всім іншим I-кадрам надається ваговий коефіцієнт 0,05, як показано в наступному рівнянні:

$$CCMetric = 0.05 * I\_Frame\_Contrast\_Mean_{(1...N-1)} + CCW1 * I\_Frame\_Contrast\_Mean(N) + (CCW2 - (0.05 * (N-1))) * Frame\_Difference\_Mean - CCW3 * I\_Contrast\_Deviation(N)^2 * I\_Contrast\_Deviation_{(1...N-1)}^2 * \exp(CCW4 * Frame\_Difference\_Deviation^2) \quad (34)$$

5. Коректування може бути використане для показника у випадку сцен з незначним рухом, коли середня різниця кадрів менше 0,05. Зсув (CCOFFSET) 0,33 повинен бути доданий в CCMetric.

Модуль 712 класифікації вмісту використовує значення  $D_{csat}$ , вектори руху і суму абсолютних різниць, щоб визначати значення, що вказує просторову складність для макроблока (або вказаний об'єм відеоданих). Часова складність визначається за допомогою вимірювання показника різниці кадрів (різниці між двома послідовними кадрами з урахуванням об'єму руху, з векторами руху і сумою абсолютних різниць між кадрами).

У деяких аспектах модуль 712 класифікації вмісту може бути сконфігурований так, щоб формувати таблицю смуги пропускання. Наприклад, формування таблиці смуги пропускання може виконуватися за допомогою модуля 712 класифікації вмісту, якщо препроцесор 226 не формує таблицю смуги пропускання.

Визначення значень текстури і руху

Для кожного макроблока в мультимедійних даних модуль 712 класифікації вмісту асоціативно зв'язує значення текстури з просторовою складністю і значення руху з часовою складністю. Значення текстури належить до значень свічення мультимедійних даних, при цьому низьке значення текстури вказує невеликі зміни в значеннях свічення сусідніх пікселів даних, а високе значення текстури вказує значні зміни в значеннях свічення сусідніх пікселів даних. Після того як значення текстури і руху обчислені, модуль 712 класифікації

вмісту визначає класифікацію вмісту за допомогою розгляду інформації руху і текстури. Модуль 712 класифікації вмісту асоціативно зв'язує текстуру для відеоданих, що класифікуються, з відносним значенням текстури, наприклад, "низька" текстура, "середня" текстура або "висока" текстура, яка, загалом, вказує складність значень яскравості макроблоків. Крім того, модуль 712 класифікації вмісту асоціативно зв'язує значення руху, обчислене для відеоданих, що класифікуються, з відносним значенням руху, наприклад, "низький" рух, "середній" рух або "високий" рух, яке, загалом, вказує об'єм руху макроблоків. У альтернативних аспектах більше або менше число категорій для руху і текстури може бути використане. Далі показник класифікації вмісту визначається з урахуванням асоціативно зв'язаних значень текстури і руху.

Фіг. 8 ілюструє приклад класифікаційної діаграми, яка ілюструє те, як значення текстури і руху асоціативно зв'язуються з класифікацією вмісту. Фахівці в даній галузі техніки<sup>1</sup> знайомі з множиною способів реалізації такої класифікаційної діаграми, наприклад, таблицею пошуку або базою даних. Класифікаційна діаграма формується на основі попередньо визначених оцінок вмісту відеоданих. Щоб визначити класифікацію відеоданих, значення текстури "низька", "середня" або "висока" (на осі x) перехресно посилається на значення руху "низький", "середній" або "високий" (на осі y). Класифікація вмісту, вказана в перетинному прямокутнику, призначається відеоданим. Наприклад, значення текстури "висока" і значення руху "середній" приводить до класифікації сім (7). Фіг. 8 ілюструє різні

комбінації відносних значень текстури і руху, які асоціативно зв'язані з вісьмома різними класифікаціями вмісту в даному прикладі. У деяких інших аспектах більше або менше число класифікацій може бути використане. Додатковий опис ілюстративного аспекту класифікації вмісту розкритий в Патентній заявці (США), що знаходиться одночасно на розгляді, номер 11/373577, озаглавлений "Content Classification For Multimedia Processing", зареєстрований 10 березня 2006 року, призначений правонаступнику цієї заявки і таким чином явно включений в даний документ по посиланню.

Призначення бітів керування швидкістю передачі

Як описано в даному документі, класифікація вмісту мультимедійних даних може бути використана в алгоритмах кодування для того, щоб ефективно поліпшувати бітове керування при збереженні постійною якості відео, що сприймається. Наприклад, показник класифікації може бути використаний в алгоритмах для виявлення зміни сцен, керування призначенням швидкості передачі бітів при кодуванні і підвищувального перетворення швидкості передачі кадрів (FRUC). Системи компресора/декомпресора (кодек) і алгоритми обробки цифрових сигналів, як правило, використовуються при обміні відеоданими і можуть бути сконфігуровані так, щоб економити смугу пропускання, але є компроміс між якістю і економією смуги пропускання. Найкращі кодеки надають найбільшу економію смуги пропускання при створенні найменшого погіршення якості відео.

У одному ілюстративному прикладі модуль 714 призначення бітів керування швидкістю передачі використовує класифікацію вмісту для того, щоб визначати швидкість передачі бітів (наприклад, число бітів, що призначаються для кодування мультимедійних даних) і зберігає швидкість передачі бітів в запам'ятовуючому пристрої для використання за допомогою інших процесів і компонентів кодера 228. Швидкість передачі бітів, визначена з класифікації відеоданих, допомагає економити смугу пропускання при наданні мультимедійних даних з узгодженим рівнем якості. У одному аспекті різна швидкість передачі бітів може бути асоціативно зв'язана з кожною з восьми різних класифікацій вмісту, і потім ця швидкість передачі бітів використовується для того, щоб кодувати мультимедійні дані. Результуючий ефект полягає в тому, що, хоч різним класифікаціям мультимедійних даних призначається різне число бітів для кодування, якість, що сприймається, аналогічна або узгоджена при перегляді на дисплеї.

Загалом, мультимедійні дані з більш високою класифікацією вмісту вказують більш високий рівень руху і/або текстури, і їм призначається більше бітів при кодуванні. Мультимедійним даним з більш низькою класифікацією (що вказує менше текстури і руху) призначається менше число бітів. Для мультимедійних даних конкретної класифікації бітів швидкість передачі бітів може бути визначена на основі вибраного цільового рівня якості, що сприймається, для перегляду мультимедійних даних. Визначення якості мультимедійних даних може бути визначене за допомогою перегляду і упорядкування людьми мультимедійних даних. У

деяких альтернативних аспектах оцінки якості мультимедійних даних можуть виконуватися за допомогою автоматичних тестових систем за допомогою, наприклад, алгоритмів співвідношення сигнал-шум. У одному аспекті набір стандартних рівнів якості (наприклад, п'ять) і відповідна швидкість передачі бітів, необхідна для того, щоб досягнути кожного конкретного рівня якості, попередньо визначаються для мультимедійних даних кожної класифікації вмісту. Щоб визначити набір рівнів якості, мультимедійні дані конкретної класифікації вмісту можуть бути оцінені за допомогою формування середньої експертної оцінки (MOS), яка надає числовий індикатор якості, що візуально сприймається, мультимедійних даних, коли вони кодуються за допомогою конкретної швидкості передачі бітів. MOS може бути виражена як одне число в діапазоні від 1 до 5, де 1 - це найменша якість, що сприймається, а 5 - це найбільша якість, що сприймається. У інших аспектах MOS може мати більше п'яти або менше п'яти рівнів якості, і різні описи кожного рівня якості можуть бути використані.

Визначення якості мультимедійних даних може бути виконане за допомогою перегляду і упорядкування людьми мультимедійних даних. У деяких альтернативних аспектах оцінки якості мультимедійних даних можуть виконуватися за допомогою автоматичних тестових систем за допомогою, наприклад, алгоритмів співвідношення сигнал-шум. У одному аспекті набір стандартних рівнів якості (наприклад, п'ять) і відповідна швидкість передачі бітів, необхідна для того, щоб досягнути кожного конкретного рівня якості, попередньо визначаються для мультимедійних даних кожної класифікації вмісту.

Знання взаємозв'язку між рівнем якості, що візуально сприймається, і швидкістю передачі в бітах мультимедійних даних визначеної класифікації вмісту може бути визначене за допомогою викликання цільового (наприклад, необхідного) рівня якості. Цільовий рівень якості, що використовується для того, щоб визначати швидкість передачі бітів, може бути попередньо вибраний, вибраний за допомогою користувача, вибраний за допомогою автоматичного процесу або напівавтоматичного процесу, що вимагає введення від користувача або з іншого процесу, або вибраний динамічно за допомогою пристрою або системи кодування на основі попередньо визначених критеріїв. Цільовий рівень якості може бути вибраний, наприклад, на основі типу додатків кодування або типу клієнтського пристрою, який приймає мультимедійні дані.

У проілюстрованому прикладі на Фіг. 7 модуль 714 призначення бітів керування швидкістю передачі приймає і дані з модуля 712 класифікації вмісту, і метадані безпосередньо з препроцесора 226. Модуль 714 призначення бітів керування швидкістю передачі розміщується в частині першого проходу кодера 228, а модуль 738 точної настройки керування швидкістю передачі розміщується в частині 706 другого проходу. Цей аспект двопрохідного керування швидкістю передачі сконфігурований так, що перший прохід (модуль 714 призначення бітів керування швидкістю передачі) виконує контекстно-адаптивне призначення бітів з попере-

джувальним переглядом на один суперкадр (наприклад, при меті довгострокової середньої швидкості передачі бітів в 256 кбіт/с) і обмежує пікову швидкість, а другий прохід (модуль 738 точної настройки керування швидкістю передачі) уточнює результати першого проходу для дворівневої масштабованості і виконує адаптацію швидкості. Керування швидкістю працює на чотирьох рівнях: (1) рівень GOP - керує розподілом бітів I-, P-, B- і F-кадрів так, щоб зробити їх неоднорідними в рамках GOP; (2) рівень суперкадрів - керує жорсткими межами на максимальний розмір суперкадру; (3) рівень кадрів - керує вимогами по бітах згідно з просторовою і часовою складністю кадрів мультимедійних даних, яка заснована на інформації вмісту (наприклад, класифікації вмісту); і (4) рівень макроблоків - керує призначенням бітів макроблоків на основі таблиць просторової і часової складності, які засновані на інформації вмісту (наприклад, класифікації вмісту).

Зразкова блок-схема послідовності операцій роботи модуля 714 керування швидкістю передачі проілюстрована на Фіг. 10. Як показано на Фіг. 10, процес 1000 починається на етапі 1002 введення. Модуль 714 керування швидкістю передачі приймає різні вхідні дані, не всі з яких обов'язково проілюстровані на Фіг. 7. Наприклад, вхідна інформація може включати в себе метадані з препроцесора 226, цільову швидкість передачі бітів, розмір буфера кодера (або, як еквівалент, максимальний час затримки для керування швидкістю передачі), початкову затримку керування швидкістю передачі і інформацію частоти кадрів. Додаткова вхідна інформація може включати в себе вхідні дані на рівні групи зображень (GOP), в тому числі, наприклад, максимальний розмір суперкадру, довжину і розподіл P/B-кадрів в GOP (включаючи інформацію зміни сцен), необхідне компонування базового рівня і рівня поліпшення, показник спотворення залежно від складності для зображень в GOP для наступних 30 кадрів. Інша вхідна інформація включає в себе вхідні дані на рівні зображень, в тому числі таблицю спотворення залежно від складності для поточного зображення (прийнятого з модуля 712 класифікації вмісту), параметри квантування (QP) і розрив бітів за останні 30 кадрів (попадання у вікно передачі змінної тривалості). Нарешті, вхідна інформація на рівні макроблоків (MB) включає в себе, наприклад, середню абсолютну різницю (MAD) спільно розміщених макроблоків (MB) в опорному зображенні і шаблон кодованих блоків (CBP) макроблоків після квантування (незалежно від того, пропущені чи ні).

Після введення даних на етапі 1002 процес 1000 переходить до етапу 1004 для ініціалізації кодування потоку бітів. Паралельно виконується ініціалізація 1006 буфера. Потім ініціалізується GOP, як показано на етапі 1008, при цьому призначення 1010 бітів GOP приймається як частина ініціалізації. Після ініціалізації GOP послідовність операцій переходить до етапу 1012, де ініціалізується серія макроблоків. Ця ініціалізація включає в себе оновлення бітів заголовків, як показано за допомогою етапу 1014. Після того, як ініціалізація на етапах 1004, 1008 і 1012 виконана, керування швидкістю (RC) для базової одиниці або макроб-

лока (MB) виконується так, як показано за допомогою етапу 1016. Як частина визначення при керуванні швидкістю макроблока на етапі 1016 приймаються вхідні дані за допомогою інтерфейсів в кодері 228. Ці вхідні дані можуть включати в себе призначення 1018 бітів макроблока (MB), оновлення параметрів 1020 квадратичної моделі і оновлення параметрів 1022 медіанного абсолютного відхилення від медіани (MAD, стійка оцінка дисперсії). Потім процес 1000 переходить до етапу 1024 для приведення у виконання операцій після кодування одного зображення 1024. Ця процедура включає в себе прийом оновлення параметрів буфера, як показано за допомогою етапу 1026. Після цього процес 1000 переходить до етапу 1028 виведення, де модуль 714 керування швидкістю передачі виводить параметри квантування QP для кожного макроблока MB, які повинні бути використані за допомогою модуля 715 визначення режиму, як показано на Фіг. 7.

#### Оцінка руху

Модуль 720 оцінки руху приймає вхідні дані з метаданих і необробленого відео з препроцесора 226 і надає вихідні дані, які можуть включати в себе розмір блока, показники спотворення векторів руху і ідентифікатори опорних кадрів, в модуль 715 визначення режиму. Фіг. 11 ілюструє зразкову роботу модуля 720 оцінки руху. Як показано, процес 1100 починається з введення 1102. На рівні кадрів модуль 720 приймає введення ідентифікатора опорного кадру і векторів руху. На рівні макроблоків введення 1102 включає в себе вхідні піксели і піксели опорних кадрів. Процес 1100 переходить до етапу 1104, на якому виконується колірна оцінка руху (ME) і прогноз векторів руху. Щоб виконати цей процес, приймаються різні вхідні дані, в тому числі вектори руху MPEG-2 і вектори руху компоненти яскравості MV 1106, згладжування 1108 векторів руху і непричинні вектори 1110 руху. Потім процес 1100 переходить до етапу 1112, на якому виконується алгоритм або методологія пошуку векторів руху, наприклад, способи шестикутного або ромбоподібного пошуку. Вхідні дані в процес на етапі 1112 можуть включати в себе суму абсолютних різниць (SAD), суму квадратів різниць (SSD) і/або інші показники, як показано за допомогою етапу 1114. Після того як пошук векторів руху виконаний, процес 1100 переходить до етапу 1116 завершення, на якому виконується обробка завершення. Після цього процес 100 завершується на етапі 1118 виведення, який надає виведення розміру блока, вектора руху (MV), показників спотворення і ідентифікаторів опорних кадрів. Масштабованість R-D для базового рівня і рівня поліпшення Фіг. 13 ілюструє зразкову блок-схему послідовності операцій процесу 1300 масштабованості, який може бути виконаний за допомогою модуля 718 масштабованості R-D. Процес 1300 починається на початковому етапі 1302 і переходить до етапу 1304, на якому модуль 718 масштабованості R-D приймає вхідні дані з модуля 720 оцінки руху і виконує оцінку руху. Оцінка руху базується на введенні опорних кадрів базового рівня, опорних кадрів рівня поліпшення і вихідного кадру, який повинен бути кодований, як указано за допомогою етапу 1306. Ця

інформація може бути обчислена за допомогою модуля 612 секціонування GOP і передана в модуль 718 масштабованості R-D за допомогою, наприклад, метаданих. Процес 1300 переходить до етапу 1308, щоб визначити інформацію масштабованості даних базового рівня і рівня поліпшення. Кодування базового рівня потім виконується, як показано на етапі 1310, за яким іде кодування рівня поліпшення на етапі 1312. Кодування рівня поліпшення може використовувати результати кодування базового рівня для міжрівневого прогнозу як вхідні дані, як проілюстровано за допомогою етапу 1314, таким чином, за часом воно виконується після кодування базового рівня. Це додатково описано в Патентній заявці (США), що знаходиться одночасно на розгляді, номер [Адвокатська виписка номер 050078], озаглавленій "Scalable Video Coding With Two Layer Encoding And Single Layer Decoding". Після того як кодування завершено, процес 1300 завершується на етапі 1316.

Упорядкування серій макроблоків/макроблоків

Частина 702 першого проходу також включає в себе модуль 722 упорядкування серій макроблоків/макроблоків, який приймає вхідні дані з модуля 740 стійкості до помилок в частині другого проходу і надає інформацію поєднання серій макроблоків в модуль 715 визначення режиму. Серії макроблоків - це ділянки незалежно декодованих (декодованих по ентропії) кодованих відеоданих. Одиниці доступу (AU) - це кодовані відеокадри, кожний з яких містить набір NAL-одиниць, які завжди містять рівно одне первинно кодоване зображення. Крім первинно кодованого зображення, одиниця доступу також може містити одне або більше надмірно кодованих зображень або інші NAL-одиниці, що не містять серій макроблоків або сегменти даних серій макроблоків в кодованому зображенні. Декодування одиниці доступу завжди приводить до декодованого зображення.

Кадри можуть бути мультиплексованими з часовим розділенням каналів блоками пакетів фізичного рівня (званими TDM-капсулою), які надають найбільше часове рознесення. Суперкадр відповідає одній одиниці часу (наприклад, 1 секунді) і містить чотири кадри. Поєднання границь серій макроблоків і AU з границями кадрів у часовій області приводить до найбільш ефективного розділення і локалізації пошкоджених даних. У ході глибокого загасання велика частина суміжних даних в TDM-капсулі зачіпається за допомогою помилок. Внаслідок часового рознесення TDM-капсули, що залишилися, мають високу імовірність бути непошкодженими. Непошкоджені дані можуть бути використані для того, щоб відновлювати і приховувати втрачені дані від зачепленої TDM-капсули. Аналогічна логіка застосовується до мультиплексування в частотній області (FDM), де частотне рознесення досягається за допомогою розділення в частотних піднесучих, які модулюють символи даних. Крім того, аналогічна логіка застосовується до просторового (за допомогою розділення в антенах передавальних і приймальних пристроїв) і інших форм рознесення, що часто застосовуються в безпроводних мережах.

Щоб суміщати серії макроблоків і AU з кадрами, створення кодових блоків зовнішнього коду

(FEC) і інкапсуляцію MAC-рівня повинні також поєднуватися. Фіг. 20 ілюструє організацію кодованих відеоданих або потоку бітів відео в серії макроблоків і AU. Кодоване відео може складатися в одному або більше потоках бітів, наприклад, в потоці бітів базового рівня і потоці бітів рівня поліпшення, де застосовується багаторівневе відеокодування.

Потік бітів відео містить AU, як проілюстровано на Фіг. 20 за допомогою кадру 1' 2005, кадру 3' 2010 і кадру M' 2015. AU містять серії макроблоків даних, як проілюстровано за допомогою серії макроблоків 1 2020, серії макроблоків 2 2025 і серії макроблоків N 2030. Кожний початок серії макроблоків ідентифікується за допомогою початкового коду і надається в мережну адаптацію. Загалом, I-кадри або внутрішньо кодовані AU є великими, а за ними ідуть P-кадри або кадри з прямим прогнозом, за якими ідуть B-кадри. Кодування AU на декілька серій макроблоків накладає суттєві додаткові витрати відносно кодованої швидкості передачі бітів, оскільки просторовий прогноз в серіях макроблоків обмежений, і заголовки макроблоків також сприяють додатковим витратам. Оскільки границі серій макроблоків є точками повторної синхронізації, обмеження суміжних пакетів фізичного рівня серіями макроблоків керує помилками, оскільки, коли PLP пошкоджений, помилка обмежується серією макроблоків в PLP, тоді як, якщо PLP містив декілька серій макроблоків або частини декількох серій макроблоків, помилка повинна вплинути на всі серії макроблоків або частини серій макроблоків в PLP.

Оскільки I-кадри в типовому варіанті є великими, наприклад, порядку десятків кбіт/с, додаткова службова інформація, зумовлена декількома серіями макроблоків, не є значною частиною загального розміру I-кадру або загальної швидкості передачі бітів. Крім того, наявність більшого числа серій макроблоків у внутрішньо кодованому AU забезпечує більш оптимальну і більш часту повторну синхронізацію і більш ефективне маскування просторових помилок. Крім цього, I-кадри переносять найбільш важливу інформацію в потоці бітів відео, оскільки P- і B-кадри прогнозуються з I-кадрів. I-кадри також виступають як точки довільного доступу для виявлення каналу.

Посилаючись тепер на Фіг. 21, ретельне поєднання I-кадрів з границями кадрів і також серій макроблоків з I AU з границями кадрів надає найбільш ефективне керування помилками, захист від помилок (оскільки, якщо одна серія макроблоків, яка належала кадру 1 2105, втрачена, серії макроблоків, які належать кадру 2 2110, мають високу імовірність бути непошкодженими, оскільки кадр 2 2110 має значне часове розділення від кадру 1 2105), відновлення після помилок може виконуватися за допомогою повторної синхронізації і маскування помилок.

Оскільки P-кадри в типовому варіанті мають розмір порядку декількох кбіт/с, поєднання серій макроблоків P-кадру і цілого числа P-кадрів з границями кадрів надає стійкість до помилок без небажаної втрати ефективності (через аналогічні причини, що і для I-кадрів). Часове маскування помилок може використовуватися в таких аспек-

тах. Альтернативно, розосередження послідовних Р-кадрів так, щоб вони надходили в різних кадрах, надає додане часове рознесення в Р-кадрах, яке може бути зумовлене тим, що часове маскуванню засноване на векторах руху і даних з раніше відновлених І- або Р-кадрів. В-кадри можуть бути від надзвичайно малих (сотні бітів) до помірно великих (декілька тисяч бітів). Отже, поєднання цілого числа В-кадрів з границями кадрів потрібно для того, щоб досягати стійкості до помилок без небажаної втрати ефективності.

Модуль визначення режиму

Фіг. 12 ілюструє деякі приклади роботи модуля 715 визначення режиму. Як показано, процес 1200 починається на етапі 1202 введення. У одному ілюстративному прикладі різна інформація, що вводиться в модуль 715 визначення режиму, включає в себе тип серії макроблоків, режим Intra 4×4cost, режим Intra 16×16cost, режим IntraUV 8×8cost, режим IntraY 16×16, режим IntraUV, дані векторів руху (MVD), параметри квантування (QP), SpPredMB4×4Y, SpPredMB16×16Y, SpPredMB8×8U, SpPredMB8×8V, прапор "швидкість передачі-спотворення", необроблені YMB-піксели, необроблені UMB-піксели і необроблені VMB-піксели. Далі процес 1200 переходить до етапу 1204 ініціалізації кодування, яка може бути ініційована за допомогою вхідного сигналу або інтерфейсу, що направляє ініціалізацію кодера, як указано за допомогою етапу 1206. Ініціалізація може включати в себе задавання дозволених режимів (включаючи пропуск, напрямки), задавання вагових коефіцієнтів режиму (за необхідності за умовчанням всі вагові коефіцієнти рівні для всіх режимів) і задавання буферів. Після ініціалізації процес 1200 переходить до етапу 1208, де виконується основна обробка для визначення режиму, яка включає в себе: обчислення витрат режиму макроблоків (MB) для кожного дозволеного режиму, зважування витрат кожного режиму макроблоків MB за допомогою вагового коефіцієнта і вибір режиму мінімальних витрат режиму MB. Вхідні дані, залучені до цих операцій, включають в себе оцінку руху (наприклад, MVD і прогнозу) і просторовий прогноз (наприклад, внутрішні витрати і прогнози), як показано за допомогою етапів 1210 і 1212. З модулем 715 визначення режиму взаємодіє кодування по ентропії на етапі 1214, яке, крім іншого, підвищує міру стиснення. Процес 1200 переходить до етапу 1216, де буфери оновлюються так, щоб передавати інформацію в частину 706 другого проходу кодера. На завершення, процес 1200 переходить до етапу 1218, на якому "оптимальний" режим кодування може бути повідомлений в частину 706 другого проходу кодера.

Частина другого проходу кодера

Посилаючись знов на Фіг. 7, частина 706 другого проходу кодера 228 включає в себе модуль 232 кодера другого проходу для виконання другого проходу кодування. Кодер 232 другого проходу приймає виведення з модуля 715 визначення режиму. Кодер 232 другого проходу включає в себе модуль 726 MC/квантування з перетворенням і зигзагоподібний кодер (ZZ)/кодер 728 по ентропії. Результати кодера 232 другого проходу виводяться в модуль 730 масштабування і модуль 731 упа-

ковки пакета бітів, який виводить кодований базовий і рівень поліпшення для передачі за допомогою транскодера 200 за допомогою синхронізуючого рівня 240 (проілюстрованого на Фіг. 2). Як показано на Фіг. 2, потрібно зазначити, що базовий і рівень поліпшення з кодера 232 другого проходу і повторного кодера 234 асемблюються за допомогою синхронізуючого рівня 240 в пакетовані PES 242, що включають в себе базовий рівень і рівні поліпшення, PES 244 даних (наприклад, CC і інших текстових даних), і PES 246 аудіо. Потрібно зазначити, що аудіодекодер 236 приймає декодовану аудіоінформацію 218 і, в свою чергу, кодує інформацію і виводить інформацію 238 в синхронізуючий рівень 240.

Повторний кодер

Посилаючись знов на Фіг. 7, частина 706 другого проходу кодера також включає в себе повторний кодер 234, який відповідає повторному кодеру 234 на Фіг. 2. Повторний кодер 234 також приймає виведення частини 702 першого проходу і включає в себе частини MC/квантування 726 з перетворенням і ZZ/кодування 728 по ентропії. Додатково, модуль 730 масштабованості виконує виведення в повторний кодер 234. Повторний кодер 234 виводить результуючий базовий і рівень поліпшення з повторного кодування в модуль 731 упаковки пакета бітів для передачі в модуль синхронізації (наприклад, синхронізуючий рівень 240, показаний на Фіг. 2). Приклад кодера 228 на Фіг. 7 також включає в себе модуль 738 точної настройки керування швидкістю передачі, який надає зворотний зв'язок по упаковці потоку бітів в модуль 234 MC/квантування з перетворенням в кодері 232 другого проходу і модуль 736 ZZ/кодування по ентропії в повторному кодері 234, щоб допомогти точно настроїти кодування другого проходу (наприклад, підвищити ефективність стиснення).

Модуль стійкості до помилок

Приклад кодера 228, проілюстрований на Фіг. 7, також включає в себе модуль 740 стійкості до помилок в частині 706 другого проходу. Модуль 740 стійкості до помилок обмінюється даними з модулем 731 упаковки потоку бітів і модулем 722 упорядкування серій макроблоків/MB. Модуль 740 стійкості до помилок приймає метадані від пре-процесора 228 і вибирає схему стійкості до помилок, наприклад, поєднання серій макроблоків і одиниць доступу з границями кадрів, ієрархія з прогнозом і адаптивне внутрішнє оновлення. Вибір схеми стійкості до помилок може бути заснований на інформації, що приймається в метаданих, або на інформації, що передається в модуль стійкості до помилок, з модуля 731 упаковки потоку бітів і модуля 722 упорядкування серій макроблоків/MB. Модуль 740 стійкості до помилок надає інформацію в модуль упорядкування серій макроблоків/макроблоків (MB) в частині 702 першого проходу, щоб реалізувати вибрані процеси стійкості до помилок. Передача відео в відмовостійких оточеннях може використовувати стійкі до помилок стратегії і алгоритми, які можуть приводити до представлення більш чітких і менш наповнених помилками даних глядачеві. Нижченаведений опис стійкості до помилок може застосовуватися до будь-яких окремих або комбінацій існуючого або

майбутнього варіанта застосування, транспортно-го і фізичного рівня і інших технологій. Ефективні алгоритми стійкості до помилок інтегрують розуміння властивостей сприйнятливості до помилок і можливостей захисту від помилок для OSI-рівнів в поєднанні з необхідними властивостями системи зв'язку, такої як низька затримка і висока пропускна здатність. Обробка стійкості до помилок може бути заснована на інформації вмісту мультимедійних даних, наприклад, класифікації вмісту мультимедійних даних. Одна з основних переваг - це відновлюваність після помилок загасання і багатопроменевого каналу. Підходи стійкості до помилок, описані нижче, належать, зокрема, до процесів, які можуть міститися в кодері 228 (наприклад, зокрема, в модулі 740 стійкості до помилок і модулі 722 упорядкування серій макроблоків/MB), і можуть бути розширені, загалом, до обміну даними у відмовостійких оточеннях.

Стійкість до помилок

Для заснованої на прогнозі гібридної системи стиснення внутрішньо кодовані кадри незалежно кодуються без якого-небудь часового прогнозу. Зовні кодовані кадри можуть бути прогнозовані у часі з попередніх кадрів (Р-кадрів) і майбутніх кадрів (В-кадрів). Оптимальний предиктор може бути ідентифікований за допомогою процесу пошуку в опорному кадрі (одному або більше) і вимірювання спотворень, наприклад, SAD, використовується для того, щоб визначити найкращий збіг. Кодована з прогнозом зона поточного кадру може бути блоком змінної довжини і форми (16×16, 32×32, 8×4 і т. д.) або групою пікселів, ідентифікованих як об'єкт за допомогою, наприклад, сегментації.

Часовий прогноз в типовому варіанті розтягується на множину кадрів (наприклад, десять або десятки кадрів) і завершується, коли кадр кодований як І-кадр, причому GOP в типовому варіанті задається за допомогою частоти І-кадрів. Для максимальної ефективності кодування GOP - це сцена, наприклад, границі GOP суміщені з границями сцени, і кадри зміни сцен кодуються як І-кадри. При невеликому русі послідовності містять відносно статичний фон, і рух, загалом, обмежений об'єктом на передньому плані. Приклади вмісту таких послідовностей з невеликим рухом включають в себе програми новин і прогнозу погоди, де більше 30% найбільш перегляданого вмісту має такий характер. У послідовностях з невеликим рухом більшість зон є зовнішньо кодованими, і спрогнозовані кадри посилаються зворотно на І-кадр за допомогою проміжних спрогнозованих кадрів.

Посилаючись на Фіг. 22, внутрішньо кодований блок 2205 в І-кадрі є предиктором для зовнішньо кодованого блока 2210 в кодованому кадрі (або AU) P1. У цьому прикладі зона даних блоків є стаціонарною частиною фону. За допомогою послідовного часового прогнозу чутливість внутрішньо кодованого блока 2205 до помилок зростає, оскільки це "хороший" предиктор, що також має на увазі, що його "важливість" вище. Додатково, внутрішньо кодований блок 2205 внаслідок цього ланцюжка часового прогнозу, званого ланцюжком прогнозу, залишається довше на дисплеї (протягом сцени в прикладі на кресленні).

Ієрархія прогнозу задається як дерево блоків, створюване на основі цього рівня важливості, або вимірювання збережуваності з батьком вгорі (внутрішньо кодований блок 2205) і дочірніми елементами внизу. Зазначимо, що зовні кодований блок 2215 на P1 знаходиться на другому рівні ієрархії, і т. д. Листя - це блоки, які завершують ланцюжок прогнозу.

Ієрархія прогнозу може бути створена для відеопослідовностей незалежно від типу вмісту (наприклад, також музики і спорту, а не тільки новин) і застосовна до стиснення заснованого на прогнозі відео (і даних) загалом (це застосовується до всіх винаходів, описаних в цій заявці). Після того як ієрархія прогнозу встановлена, алгоритми стійкості до помилок, такі як адаптивне внутрішнє оновлення, описане нижче, можуть бути застосовані більш ефективно. Вимірювання важливості може бути засноване на відновлюваності даного блока після помилок, наприклад, за допомогою операцій маскування і застосування адаптивного внутрішнього оновлення для того, щоб підвищувати стійкість кодованого потоку бітів до помилок. Оцінка вимірювання важливості може бути заснована на кількості разів, коли блок використовується як прогнозний параметр, також згадуваний як показник збережуваності. Показник збережуваності також використовується для того, щоб підвищувати ефективність кодування за рахунок припинення поширення помилок прогнозу. Показник збережуваності також збільшує призначення бітів для блоків з більш високою важливістю.

Адаптивне внутрішнє оновлення

Адаптивне внутрішнє оновлення використовується в методиці стійкості до помилок, яка може бути заснована на інформації вмісту мультимедійних даних. У процесі внутрішнього оновлення деякі MB кодуються внутрішньо, навіть якщо стандартна оптимізація R-D повинна диктувати те, що вони повинні бути зовнішньо кодованими MB. AIR використовує внутрішнє оновлення із зважуванням руху для того, щоб ввести внутрішньо кодовані MB в Р- або В-кадри. Ці внутрішньо кодовані MB, що містяться в базовому рівні, можуть бути кодовані за допомогою QP<sub>b</sub> або QP<sub>e</sub>. Якщо QP<sub>e</sub> використовується для базового рівня, не потрібно виконувати уточнення на рівні поліпшення. Якщо QP<sub>b</sub> використовується для базового рівня, то уточнення може потребуватися, інакше на рівні поліпшення погіршення якості буде помітним. Оскільки зовнішнє кодування більш ефективне, ніж внутрішнє кодування, відносно ефективності кодування, ці уточнення на рівні поліпшення внутрішньо кодуються. Таким чином, коефіцієнти базового рівня не використовуються для рівня поліпшення, і якість підвищується на рівні поліпшення без введення нових операцій.

У деяких аспектах адаптивне внутрішнє оновлення може бути засноване на інформації вмісту мультимедійних даних (наприклад, класифікації вмісту) замість або крім основи зваженого руху. Наприклад, якщо класифікація вмісту відносно висока (наприклад, сцени, що мають високу просторову і часову складність), адаптивне внутрішнє оновлення може вводити відносно більше число внутрішньо кодованих MB в Р- або В-кадри. Аль-



тернативно, якщо класифікація вмісту відносно низька (вказуючи менш динамічну сцену з низькою просторовою і/або часовою складністю), то адаптивне внутрішнє оновлення може вводити менше число внутрішньо кодованих MB в P- і B-кадри. Ці показники і способи підвищення стійкості до помилок можуть бути застосовані не тільки в контексті безпроводного мультимедійного зв'язку, але також до стиснення даних і мультимедійної обробки, загалом, (наприклад, при рендерингу графіки).

#### Кадр перемикання каналів

Кадр перемикання каналів (CSF), що задається в даному документі, є широким терміном, що описує кадр довільного доступу, який вставляється у відповідне місце ширококомовного потоку для швидкого виявлення каналу і тим самим швидкої зміни каналів між потоками при ширококомовній мультимедійній передачі. Кадри перемикання каналів також підвищують стійкість до помилок, оскільки вони надають надмірні дані, які можуть бути використані в тому випадку, якщо первинний кадр переданий з помилкою. I-кадри або прогресивний I-кадр, такий як прогресивний кадр оновлення декодера, в H.264 в типовому варіанті виступає як точка довільного доступу. Проте, часті I-кадри (або короткі GOP, коротше тривалості сцен) приводять до значного зниження ефективності стиснення. Оскільки внутрішньо кодовані блоки можуть бути використані для стійкості до помилок, довільний доступ і стійкість до помилок можуть бути ефективно комбіновані за допомогою ефективності прогнозу, щоб підвищувати ефективність кодування при збільшенні стійкості до помилок.

Удосконалення перемикання з довільним доступом і стійкості до помилок може бути досягнуте у взаємодії і може бути засноване на інформації вмісту, такий як класифікація вмісту. Для послідовностей з невеликим рухом ланцюжки прогнозування є довгими, і значна частина інформації, необхідної для того, щоб відновлювати суперкадр або сцену, міститься в I-кадрі, який був на початку сцени. Помилки каналів часто є пульсуючими, і коли загасання виникає, і FEC і каналне кодування завершується збоєм, є серйозна залишкова помилка того, що маскування збоїв. Це особливо серйозно для послідовностей з невеликим рухом (і, отже, низькою швидкістю передачі бітів), оскільки об'єм кодованих даних недостатньо великий для того, щоб надавати хороше часове рознесення в потоці бітів відео, і оскільки це послідовності з високою мірою стиснення, що робить кожний біт придатним для відновлення. Послідовності з великим рухом більш стійкі до помилок внаслідок характеру вмісту - більше нової інформації в кожному кадрі збільшує число внутрішньо кодованих блоків, які є незалежно декодованими і більш стійкими до помилок за природою. Адаптивне внутрішнє оновлення на основі ієрархії прогнозу надає високу продуктивність для послідовностей з великим рухом, і підвищення продуктивності незначне для послідовностей з невеликим рухом. Отже, кадр перемикання каналів, що містить велику частину I-кадру, є хорошим джерелом рознесення для послідовностей з невеликим рухом. Коли помилка виникає в суперкадрі, декодування в послідовному кадрі починається з CSF, який відновлює втрачену

інформацію завдяки прогнозу, і досягається стійкість до помилок.

У випадку послідовностей з великим рухом, таких як послідовності, що мають відносно високу класифікацію вмісту (наприклад, 6-8), CSF може складатися з блоків, які містяться в SF, тобто тих, які є хорошими предикторами. Всі інші зони CSF не повинні кодуватися, оскільки вони є блоками, які мають короткі ланцюжки прогнозу, що має на увазі те, що вони завершуються внутрішніми блоками. Отже, CSF як і раніше служить для того, щоб виконувати відновлення втраченої інформації завдяки прогнозу, коли виникає помилка. CSF для послідовностей з невеликим рухом співпадають з розміром I-кадрів, і вони можуть кодуватися при меншій швидкості передачі бітів за рахунок більш сильного квантування, тоді як CSF для послідовностей з великим рухом набагато менше відповідних I-кадрів.

Стійкість до помилок на основі ієрархії прогнозу може добре працювати з масштабованістю і дозволяє досягати високоефективного багаторівневого кодування. Масштабованість, для того, щоб підтримувати ієрархічну модуляцію в технологіях фізичного рівня, може вимагати секціонування даних потоку бітів відео з конкретними коефіцієнтами смуги пропускання. Це не завжди можуть бути ідеальні коефіцієнти для оптимальної масштабованості (наприклад, з невеликою додатковою службовою інформацією). У деяких аспектах використовується дворівнева масштабованість з коефіцієнтом смуги пропускання 1:1. Секціонування потоку бітів відео на два рівні однакового розміру може бути не таким ефективним для послідовностей з невеликим рухом. Для послідовностей з невеликим рухом базовий рівень, що містить всю інформацію заголовків і метаданих, більше рівня поліпшення. Проте, оскільки CSF для послідовностей з невеликим рухом більше, вони точно вписуються в смугу пропускання, що залишилася, рівня поліпшення.

Послідовності з великим рухом мають достатньо залишкової інформації для того, щоб секціонування даних 1:1 могло бути виконане з найменшою службовою інформацією. Додатково, кадр перемикання каналів для таких послідовностей набагато менше для послідовностей з великим рухом. Отже, стійкість до помилок на основі ієрархії прогнозу також може добре працювати з масштабованістю для послідовностей з великим рухом. Розширення концепцій, описаних вище, до кліпів з помірним рухом можливе на основі описів цих алгоритмів, і концепції, що пропонуються, застосовуються до кодування відео загалом.

#### Мультимплексор

У деяких аспектах кодера мультимплексор може бути використаний для кодування декількох мультимедійних потоків, сформованих за допомогою кодера, і використаний для того, щоб готувати кодовані біти для ширококомовної передачі. Наприклад, в ілюстративному аспекті кодера 228, показаному на Фіг. 2, синхронізуючий рівень 240 містить мультимплексор. Мультимплексор може бути реалізований так, щоб надавати керування призначенням швидкості передачі бітів. Оцінена складність може надаватися в мультимплексор, який

потім може призначати доступну смугу пропускання для набору мультиплексованих відеоканалів згідно зі складністю кодування, передбачуваною для цих відеоканалів, що в такому випадку дає можливість якості конкретного каналу залишатися відносно постійною, навіть якщо смуга пропускання для набору мультиплексованих відеопотоків відносно постійна. Це дає можливість каналу в рамках набору каналів мати змінну швидкість передачі бітів і відносно постійну візуальну якість замість відносно постійної швидкості передачі бітів і змінної візуальної якості.

Фіг. 18 - це блок-схема, яка ілюструє систему кодування декількох мультимедійних потоків або каналів 1802. Мультимедійні потоки 1802 кодуються за допомогою відповідних кодерів 1804, які підтримують обмін даними з мультиплексором (MUX) 1806, який, в свою чергу, підтримує обмін даними з середовищем 1808 передачі. Наприклад, мультимедійні потоки 1802 можуть відповідати різним каналам вмісту, таким як канали новин, спортивні канали, канали з фільмами і т. п. Кодери 1804 кодують мультимедійні потоки 1802 в формат кодування, вказаний для системи. Хоч описуються в контексті кодування відеопотоків, принципи і переваги розкритих методик, загалом, застосовні до мультимедійних потоків, що включають в себе, наприклад, аудіопотоки. Кодовані мультимедійні потоки надаються в мультиплексор 1806, який комбінує різні кодовані мультимедійні потоки і відправляє комбінований потік в середовище 1808 передачі для передачі.

Середовище 1808 передачі може відповідати множині середовищ, таких як, але не тільки, цифрові супутникові канали зв'язку, наприклад, DirecTV®, цифрові кабельні, провідні і безпроводні Інтернет-канали зв'язку, оптичні мережі, стільникові телефонні мережі і т. п. Середовище 1808 передачі може включати в себе, наприклад, модуляцію до радіочастоти (RF). У типовому варіанті внаслідок спектральних обмежень і т. п., середовище передачі має обмежену смугу пропускання, і дані з мультиплексора 1806 в середовище передачі підтримуються з відносно постійною швидкістю передачі бітів (CBR).

У традиційних системах застосування постійної швидкості передачі бітів (CBR) при виведенні з мультиплексора 1806 може вимагати того, щоб кодоване мультимедіа або відеопотоки, які надходять в мультиплексор 1806, також були CBR. Як описано в рівні техніки, застосування CBR при кодуванні відеовмісту може приводити до змінної візуальної якості, що в типовому варіанті небажано.

У проілюстрованій системі два або більше кодери 1804 передають передбачувану складність кодування вхідних даних. Один або більше кодерів 1804 можуть приймати адаптивне керування швидкістю передачі бітів від мультиплексора 1806 у відповідь. Це дозволяє кодеру 1804, який, як очікується, кодує відносно складне відео, приймати більш високу швидкість передачі бітів або більш високу смугу пропускання (більше бітів на кадр) для цих кадрів відео способом квазізмінної швидкості передачі бітів. Це дає можливість мультимедійному потоку 1802 бути кодованим при узгодже-

ній візуальній якості. Додаткова смуга пропускання, яка використовується за допомогою конкретного кодера 1804, що кодує відносно складне відео, надходить від бітів, які в іншому випадку були б використані для кодування інших відеопотоків 1804, якби кодери були реалізовані так, щоб працювати при постійних швидкостях передачі бітів. Це підтримує виведення мультиплексора 1806 на постійній швидкості передачі бітів (CBR).

Хоч окремих мультимедійний потік 1802 може бути відносно "пульсуючим", тобто варіювати використовувану смугу пропускання, кумулятивна сума з декількох мультимедійних потоків може бути менш пульсуючою. Швидкість передачі бітів від каналів, які кодують менш складне відео, може бути наново виділена, наприклад, за допомогою мультиплексора 1806, каналам, які кодують відносно складне відео, і це дозволяє підвищувати візуальну якість комбінованих відеопотоків загалом.

Кодери 1804 надають в мультиплексор 1806 індикатор складності набору відеокадрів, які повинні бути кодовані і мультиплексовані разом. Вивід мультиплексора 1806 повинен надавати виведення, яке не вище швидкості передачі бітів, заданої для середовища 1808 передачі. Індикатори складності можуть бути засновані на класифікації вмісту, як описано вище, щоб надавати вибраний рівень якості. Мультиплексор 1006 аналізує індикатори складності і надає в різні кодери 1004 призначене число бітів смуги пропускання, і кодери 1804 використовують цю інформацію для того, щоб кодувати відеокадри в наборі. Це дає можливість набору відеокадрів індивідуально мати змінну швидкість передачі бітів і при цьому досягати постійної швидкості передачі бітів як групи.

Класифікація вмісту також може бути використана для забезпечення заснованого на якості стиснення мультимедіа, загалом, для будь-якого узагальненого модуля стиснення. Класифікація вмісту і способи і пристрої, описані в даному документі, можуть бути використані в заснованій на якості і/або заснованій на вмісті мультимедійній обробці будь-яких мультимедійних даних. Один приклад - це її використання при стисненні мультимедіа, загалом, для будь-якого узагальненого модуля стиснення. Інший приклад полягає в розпаковуванні або декодуванні в будь-якому модулі розпаковування або декодері або постпроцесорі, наприклад, операції інтерполяції, повторної дискретизації, поліпшення, відновлення і представлення.

Посилаючись тепер на Фіг. 19, типова система відеозв'язку включає в себе систему відеостиснення, що складається з відеокодера і відеодекодера, сполучених за допомогою мережі зв'язку. Безпроводні мережі є одним класом відмовостійких мереж, де канал зв'язку надає логарифмічно-нормальне загасання або корекцію і багатопромежне загасання в мобільних сценаріях крім втрат в каналах. Щоб протистояти канальним помилкам і надавати надійний зв'язок для даних прикладного рівня, RF-модулятор включає в себе пряму корекцію помилок, що включає в себе модуль переможування, і канальне кодування, таке як згорткове або турбокодування.

Відеокодування знижує надмірність у вихідному відео і збільшує обсяг інформації, яка переноситься в кожному біті кодованих відеоданих. Це підвищує вплив на якість, коли навіть невелика частина кодованого відео втрачена. Просторове і часове кодування, властиве системам стиснення відео, посилює втрати і примушує помилки розповсюджуватися, приводячи до видимих перешкод у відновленому відео. Алгоритми стійкості до помилок у відеокодері і алгоритми відновлення після помилок у відеокодері підвищують відмовостійкість системи стиснення відео.

У типовому варіанті система стиснення відео знає про базову мережу. Проте, у відмовостійких мережах інтеграція або поєднання алгоритмів захисту від помилок в прикладному рівні з FEC і канальним кодуванням в канальному/фізичному рівні дуже бажане і надає найбільшу ефективність для підвищення продуктивності по помилках системи.

Фіг. 14 ілюструє один приклад потоку даних спотворень залежно від швидкості передачі, який виникає в кодері 228, щоб кодувати кадри. Процес починається 1400 з початку 1402 і переходить до етапу 1404 прийняття рішення, де він приймає введення 1410 модуля виявлення зміни сцен з препроцесора 226 (наприклад, за допомогою метаданих), і стійке до помилок введення 1406 одержується. Якщо інформація вказує, що вибраний кадр є I-кадром, процес внутрішньо кодує кадр. Якщо інформація вказує те, що вибраний кадр - це P- або B-кадр, процес використовує внутрішнє кодування і (внутрішнє) кодування з оцінкою руху для того, щоб кодувати кадр.

Після того як стверджувальний режим виникає для режимів етапу 1404, процес 1400 переходить до етапу 1414 підготовки, на якому швидкість R задається рівною значенню  $R=R_{qual}$ , необхідній цільовій якості на основі кривих R-D. Ця настройка приймається з етапу 1416 даних, що містить криві R-D. Процес 1400 далі переходить до етапу 1418, на якому виконується призначення бітів керування швидкістю передачі ( $Q_{pi}$ ) на основі інформації дій зображень/відео (наприклад, класифікації вмісту) з процесу класифікації вмісту на етапі 1420.

Етап 1418 призначення бітів керування швидкістю передачі використовується, в свою чергу, для оцінки руху на етапі 1422. Оцінка 1422 руху також може приймати введення метаданих з препроцесора 1412, згладжування векторів руху (MPEG-2+передісторія) з етапу 1424 і декількох опорних кадрів (причинні+непричинні макроблоки MB) з етапу 1426. Процес 1400 далі переходить до етапу 1428, на якому обчислення внутрішньо кодованих режимів визначається для призначення бітів керування швидкістю передачі ( $Q_{pi}$ ). Потім процес 1400 переходить до етапу 1430, на якому визначаються параметри режиму і квантування. Визначення режиму на етапі 1430 виконується на основі оцінки введення етапу 1422, введення стійкості 1406 до помилок і масштабованості R-D, яка визначається на етапі 1432. Після того як режим визначений, послідовність операцій переходить до етапу 1432. Потрібно зазначити, що послідовність операцій з етапу 1430 до етапу 1432 здійснюється, коли дані передаються з першого проходу в частину другого проходу кодера.

На етапі 1432 перетворення і квантування виконується за допомогою другого проходу кодера 228. Процес перетворення/квантування коректується або точно настроюється так, як вказано за допомогою етапу 1444. На цей процес перетворення/квантування може впливати модуль точної настройки керування швидкістю передачі (Фіг. 7). Процес 1400 далі переходить до етапу 1434 для зигзагоподібного сортування і кодування по ентропії, щоб надати кодований базовий рівень. Зигзагоподібне сортування готує дані в ефективному форматі для кодування. Кодування по ентропії - це методика стиснення, яка використовує послідовності бітових кодів для того, щоб представляти набори можливих символів. Результат етапу 1432 перетворення/квантування рівня поліпшення також відправляється в суматор 1436, який віднімає базовий рівень і відправляє результат в кодер 1438 ZZ/по ентропії для рівня поліпшення, як раніше описано з посиланням на Фіг. 31-36. Додатково помітимо, що рівень поліпшення відправляється зворотно (див. рядок 1440 оновлення істинної швидкості) для оновлення класифікації 1420 вмісту істинної швидкості і операції визначення довгострокової і короткострокової передісторії швидкостей передачі бітів, що використовується за допомогою керування швидкістю передачі.

Фіг. 17A - це блок-схема послідовності операцій способу, яка ілюструє обробку мультимедійних даних, які одержані, прийняті або іншим чином доступні. Процес 1700 починається на етапі 1702, де він класифікує вміст мультимедійних даних. У одному ілюстративному аспекті класифікація вмісту може бути виконана за допомогою засобу класифікації, наприклад, модуля 712 класифікації вмісту на Фіг. 7. Процес 1700 переходить до етапу 1704, де він кодує мультимедійні дані в першій групі даних і другій групі даних на основі класифікації вмісту. Це кодування виконується так, що перша група даних містить коефіцієнт, а друга група даних містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних. Це може бути виконане за допомогою засобу кодування, описаного в даному документі, наприклад, кодера 228 на Фіг. 7.

Фіг. 17B - це блок-схема системи 1710 кодування мультимедіа, яка може виконувати процес, проілюстрований на Фіг. 17A. У деяких аспектах системою 1710 кодування мультимедіа може бути транскодер, наприклад, транскодер 200. У інших аспектах система 1710 кодування може містити частину транскодера. Система 1710 кодування мультимедіа включає в себе засіб класифікації вмісту мультимедійних даних, модуль класифікації вмісту мультимедійних даних 1712. Засобом класифікації вмісту може бути, наприклад, модуль класифікації в препроцесорі (наприклад, препроцесорі 226) або кодері (наприклад, кодері 228). Система 1710 кодування також включає в себе засіб кодування мультимедійних даних, модуль кодування мультимедійних даних 1714, який може бути сконфігурований так, щоб кодувати мультимедійні дані в першій групі даних і у другій групі даних на основі класифікації вмісту, причому це кодування виконується таким чином, що перша група даних містить коефіцієнт, а друга група да-

них містить перше диференціальне уточнення, асоціативно зв'язане з коефіцієнтом першої групи даних. Інші компоненти транскодера, такі як описані в даному документі, також можуть бути включені в систему 1710 кодування.

Фіг. 23, 24, 27 і 28 - це блок-схеми послідовності операцій, які ілюструють приклади кодування мультимедійних даних, які здійснюють аспекти, описані в даному документі. Фіг. 23 - це блок-схема послідовності операцій, яка ілюструє процес 2300 кодування мультимедійних даних на основі інформації вмісту. На етапі 2305 процес 2300 приймає кодовані мультимедійні дані, а на етапі 2310 процес 2300 декодує мультимедійні дані. На етапі 2315 процес 2300 визначає інформацію вмісту, асоціативно зв'язану з декодованими мультимедійними даними. На етапі 2320 процес 2300 кодує мультимедійні дані на основі інформації вмісту.

Фіг. 24 - це блок-схема послідовності операцій способу, яка ілюструє процес 2400 способу кодування мультимедійних даних, так щоб суміщати границі даних на основі рівня інформації вмісту. На етапі 2405 процес 2400 одержує інформацію вмісту, асоціативно зв'язану з мультимедійними даними, що може бути виконано, наприклад, за допомогою препроцесора 226 або модуля 712 класифікації вмісту, показаного на Фіг. 7. На етапі 2410 процес 2400 кодує мультимедійні дані, так щоб суміщати границі даних на основі інформації вмісту. Наприклад, границі серій макроблоків і границі одиниць доступу поєднуються з границями кадрів на основі класифікації вмісту мультимедійних даних, що кодуються. Кодовані дані потім доступні для подальшої обробки і/або передачі в мобільний пристрій, і процес 2400 завершується.

Фіг. 27 - це блок-схема послідовності операцій способу, яка ілюструє процес 2700 кодування даних за допомогою адаптивного внутрішнього оновлення на основі інформації вмісту. Коли процес 2700 починається, мультимедійні дані були одержані. На етапі 2705 процес 2700 одержує інформацію вмісту мультимедійних даних. Одержання інформації вмісту може бути виконане, наприклад, за допомогою препроцесора 226 або модуля 712 класифікації вмісту, як описано вище. Процес 2700 переходить до етапу 2710, на якому він кодує мультимедійні дані за допомогою схеми стійкості до помилок на основі адаптивного внутрішнього оновлення, причому схема стійкості до помилок на основі адаптивного внутрішнього оновлення заснована на інформації вмісту. Функціональність етапу 2710 може бути здійснена за допомогою кодера 228. Кодовані дані стають доступними для подальшої обробки і передачі, і процес 2700 потім завершується.

Фіг. 28 - це блок-схема послідовності операцій способу, яка ілюструє процес кодування мультимедійних даних за допомогою надмірних I-кадрів на основі інформації мультимедійного вмісту. Коли процес 2800 починається, мультимедійні дані доступні для обробки. На етапі 2805 процес 2800 одержує інформацію вмісту мультимедійних даних. Як описано вище, це може бути виконане, наприклад, за допомогою препроцесора 226 і/або кодера 228. На етапі 2810 процес 2800 кодує мультимедійні

дані, так щоб вставити один або більше додаткових I-кадрів в кодовані дані на основі інформації вмісту. Це може бути зроблене за допомогою кодера 228, як описано вище в зв'язку зі схемою стійкості до помилок, вставки I-кадрів в базовий рівень або рівень поліпшення залежно від використовуваної схеми стійкості до помилок. Після етапу 2810 кодовані дані доступні для подальшої обробки і/або передачі в мобільний пристрій.

Потрібно зазначити, що способи, описані в даному документі, можуть бути реалізовані для множини апаратних засобів, процесорів і систем зв'язку, відомих фахівцям в даній галузі техніки. Наприклад, загальна вимога до клієнта працювати так, як описано в даному документі, полягає в тому, що клієнт має дисплей для того, щоб відображати вміст і інформацію, процесор, щоб керувати роботою клієнта, і запам'ятовуючий пристрій для збереження даних і програм, пов'язаних з роботою клієнта. У одному аспекті клієнтом є стильниковий телефон. У іншому аспекті клієнтом є кишеньковий комп'ютер, що має засоби зв'язку. У ще одному іншому аспекті клієнтом є персональний комп'ютер, що має засоби зв'язку. Крім цього, такі апаратні засоби, як приймальний GPS-пристрій, можуть міститися в клієнті для того, щоб реалізовувати різні аспекти.

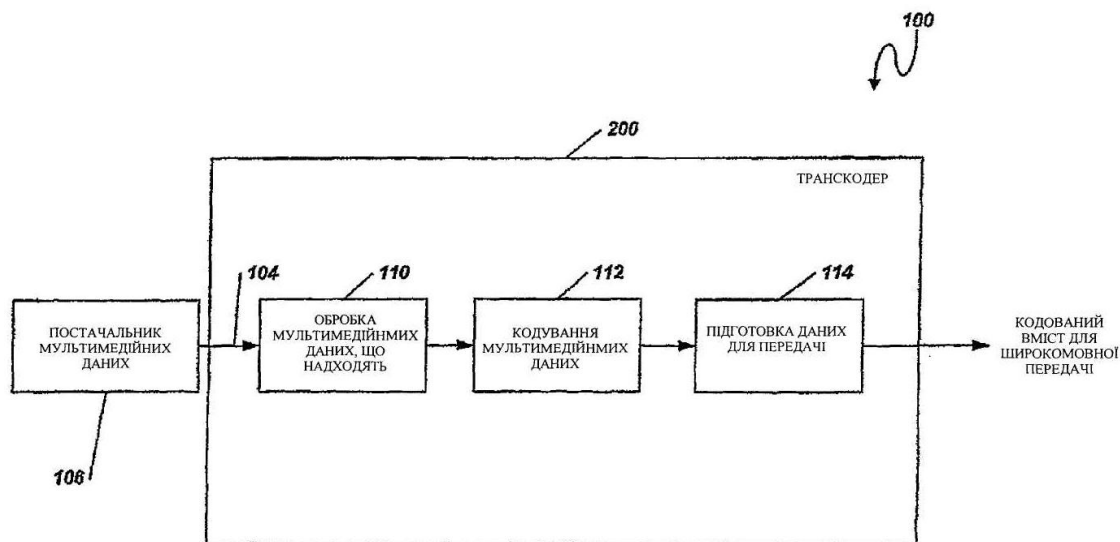
Різні ілюстративні логічні вузли, логічні блоки, модулі і схеми, описані в зв'язку з розкритими в даному документі варіантами здійснення, можуть бути реалізовані або виконані за допомогою процесора загального призначення, процесора цифрових сигналів (DSP), спеціалізованої інтегральної схеми (ASIC), програмованої користувачем матричної БІС (FPGA) або іншого програмованого логічного пристрою, дискретного логічного елемента або транзисторної логіки, дискретних компонентів апаратних засобів або будь-якої їх комбінації, призначеної для того, щоб виконувати описані в даному документі функції. Процесором загального призначення може бути мікропроцесор, але в альтернативному варіанті, процесором може бути будь-який традиційний процесор, контролер, мікроконтролер або кінцевий автомат. Процесор також може бути реалізований як поєднання обчислювальних пристроїв, наприклад, поєднання DSP і мікропроцесора, множина мікропроцесорів, один або більше мікропроцесорів разом з ядром DSP або будь-яка інша подібна конфігурація.

Розкриті способи і пристрої надають перекодування відеоданих, кодованих в одному форматі, у відеодані, кодовані в іншому форматі, причому кодування засноване на вмісті відеоданих і кодування є стійким до помилок. Способи або алгоритми, описані в зв'язку з розкритими в даному документі прикладами, можуть бути реалізовані безпосередньо в апаратних засобах, в програмному модулі, що приводиться у виконання процесором, мікропрограмному забезпеченні або в поєднанні двох або більше з цих елементів. Програмний модуль може постійно розміщуватися в пам'яті типу RAM, флеш-пам'яті, пам'яті типу ROM, пам'яті типу EEPROM, регістрах, на жорсткому диску, змінному диску, компакт-диску або будь-якій іншій формі носія зберігання даних, відомій в даній галузі техніки. Зразковий носій збері-

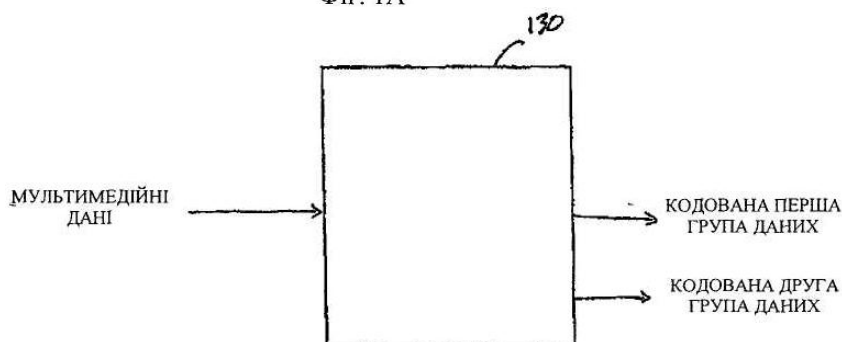
гання даних підключений до процесора таким чином, щоб процесор міг зчитувати інформацію і записувати інформацію на носій зберігання даних. У альтернативному варіанті носій зберігання може бути вбудований в процесор. Процесор і носій зберігання даних можуть постійно розміщуватися в ASIC. ASIC може постійно розміщуватися в користувацькому терміналі. У альтернативному варіанті процесор і носій зберігання даних можуть постійно розміщуватися як дискретні компоненти в користувацькому терміналі.

Приклади, описані вище, є просто ілюстративними, і фахівці в даній галузі техніки тепер можуть виконувати різні застосування і відхилення від вищеописаних прикладів без відхилення від розкритих в даному документі понять винаходу. Різні модифікації в цих прикладах можуть бути очевидними для фахівців в даній галузі техніки, а описані в даному документі загальні принципи можуть бути застосовані до інших прикладів, напри-

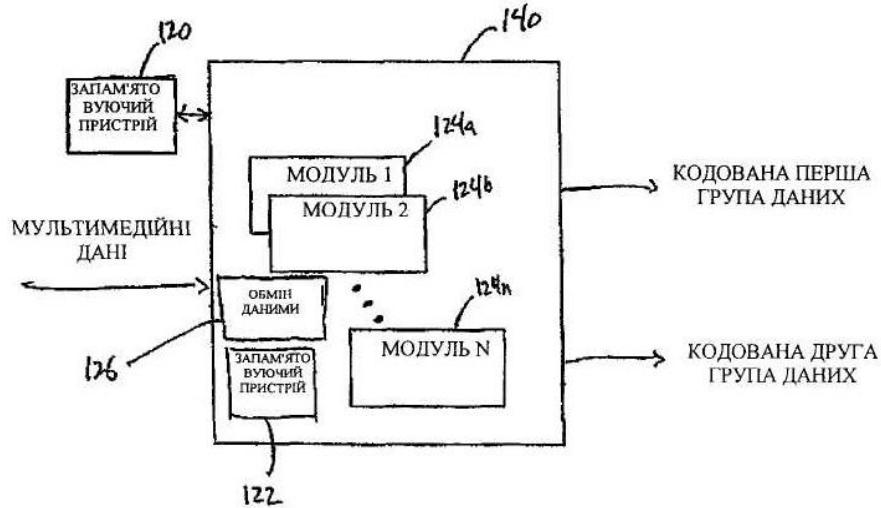
клад, до служби обміну миттєвими повідомленнями або будь-яких загальних додатків обміну безпровідними даними, без відхилення від духу і галузі застосування нових аспектів, що описуються в цьому документі. Таким чином, галузь застосування винаходу не призначена, щоб бути обмеженою показаними в даному документі прикладами, а повинна задовольняти самій широкій галузі застосування, узгодженій з принципами, і новими ознаками, розкритими в даному документі. Слово "зразковий" використовується в даному документі виключно для того, щоб означати "служить як приклад, окремий випадок або ілюстрація". Будь-який приклад, описаний в даному документі як "зразковий", не обов'язково повинен бути витлумачений як переважний або вигідний в порівнянні з іншими прикладами. Відповідно, нові ознаки, що описуються в даному документі, повинні задаватися виключно за допомогою галузі застосування нижченаведеної формули винаходу.



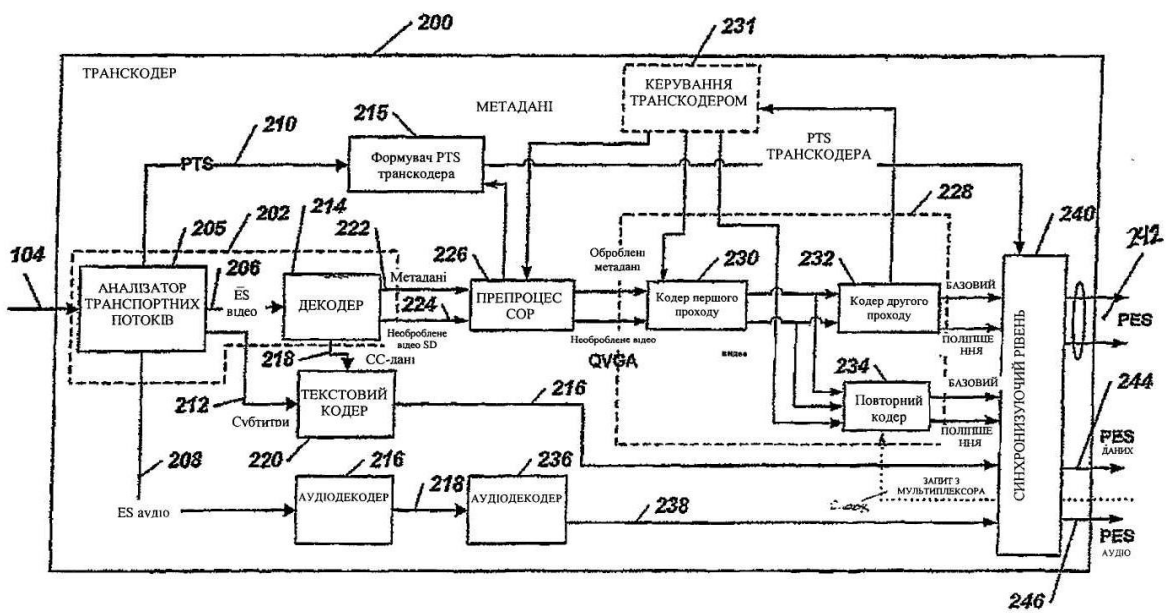
Фіг. 1А



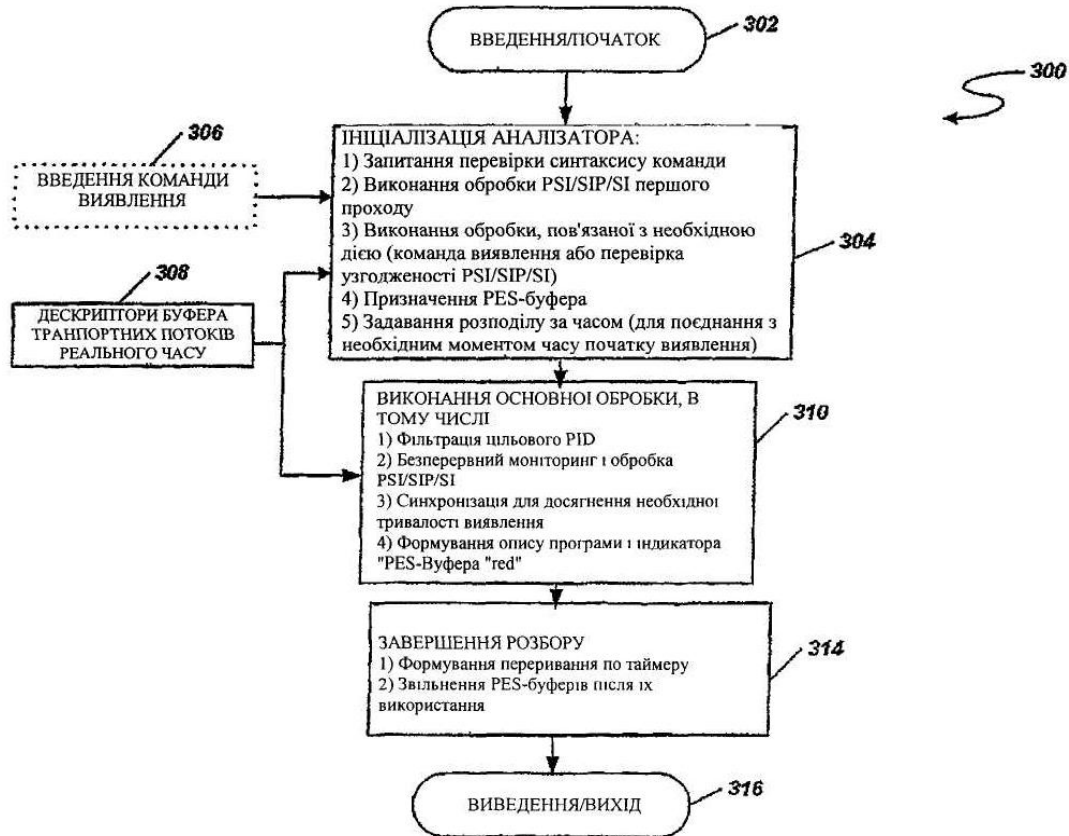
Фіг. 1В



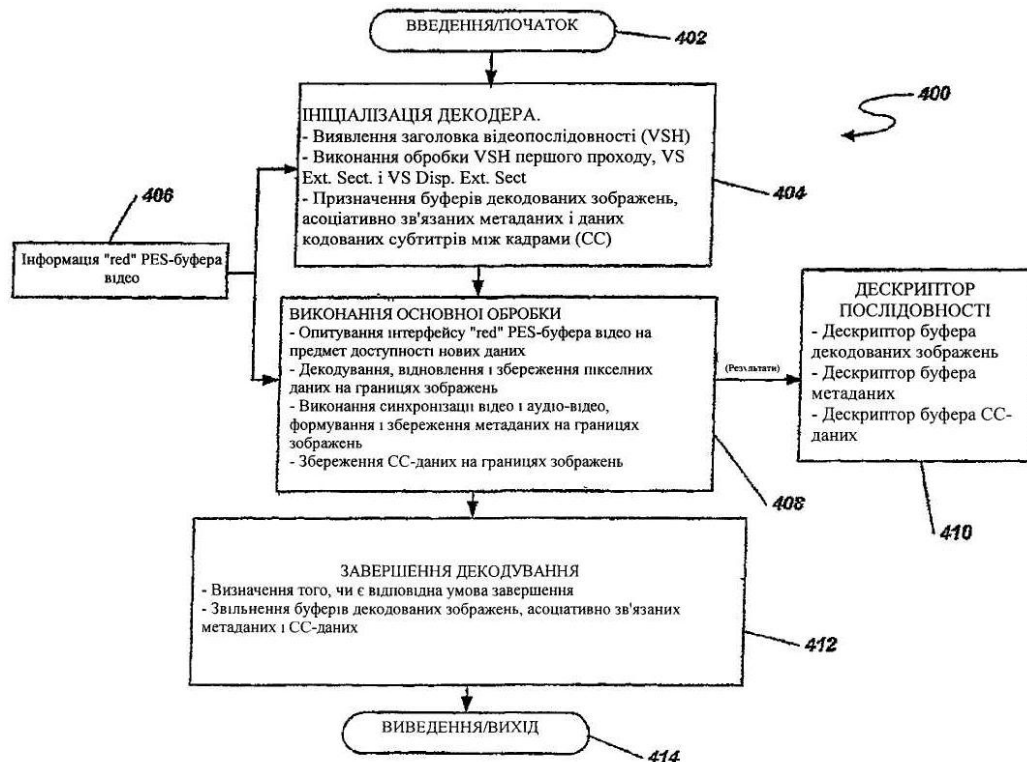
Фіг. 1С



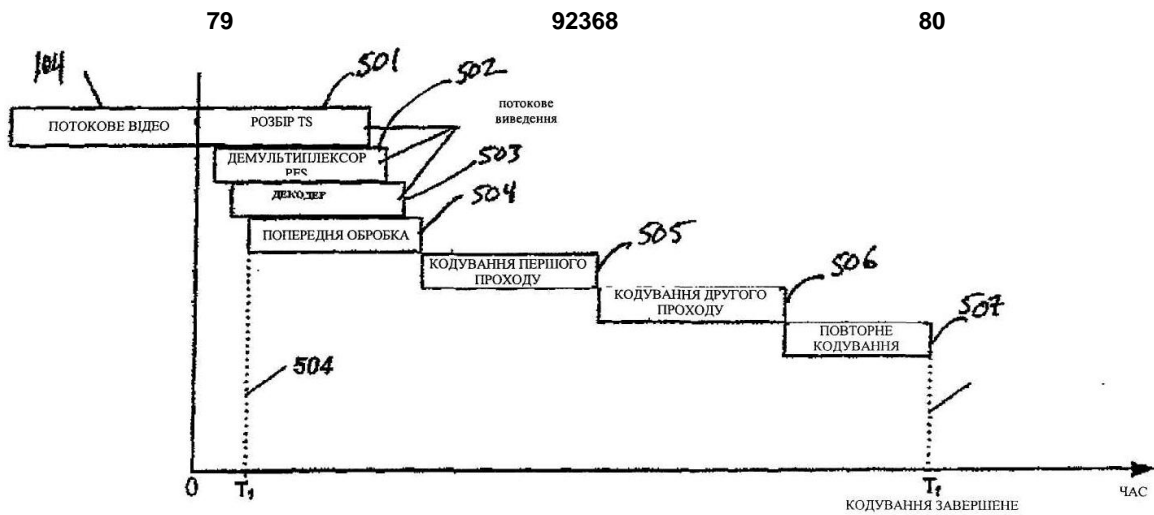
Фіг. 2



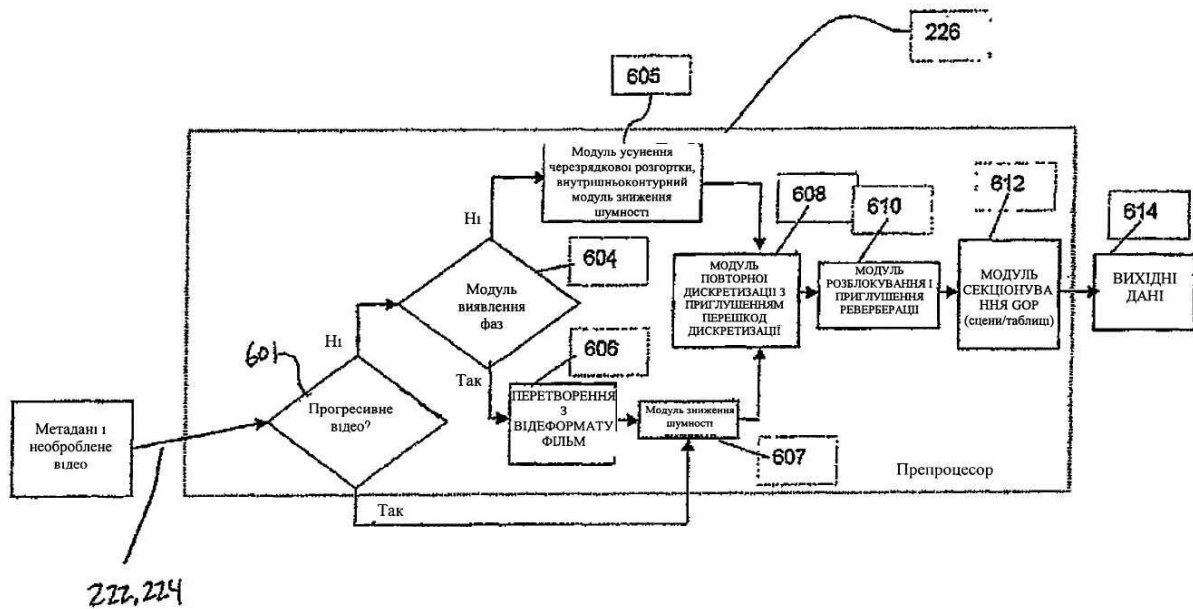
Фіг. 3



Фіг. 4

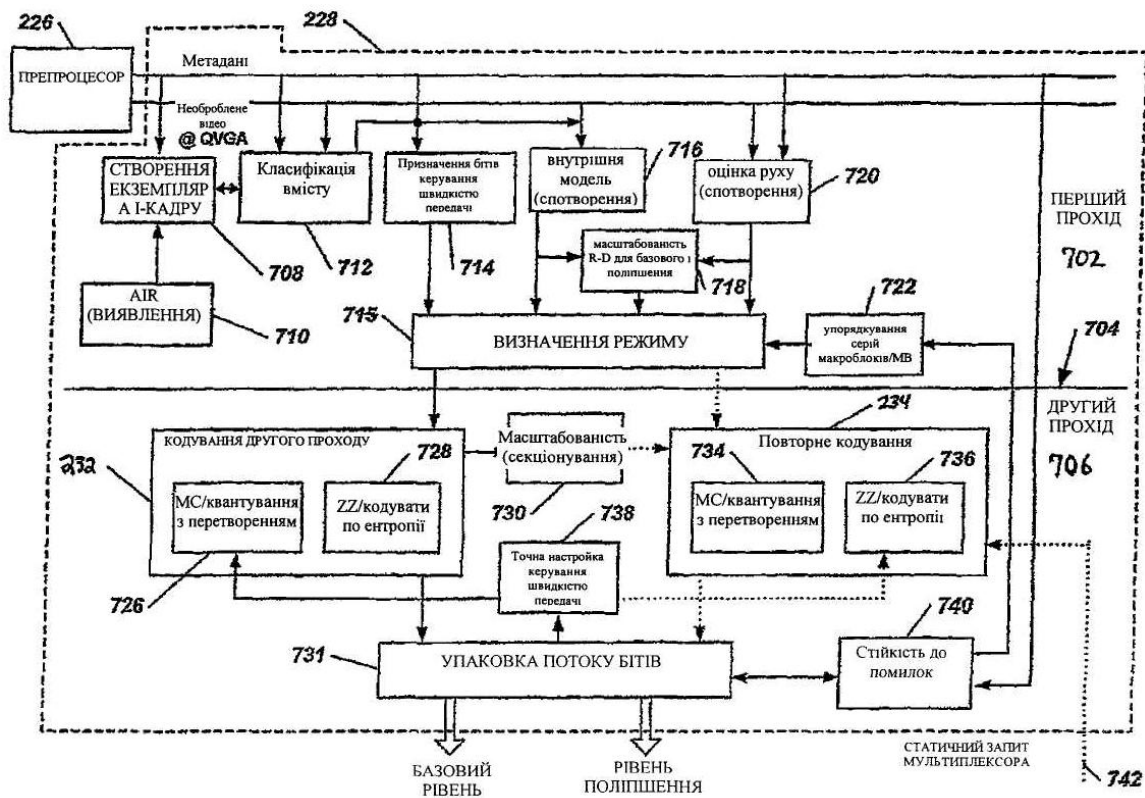


Фіг. 5

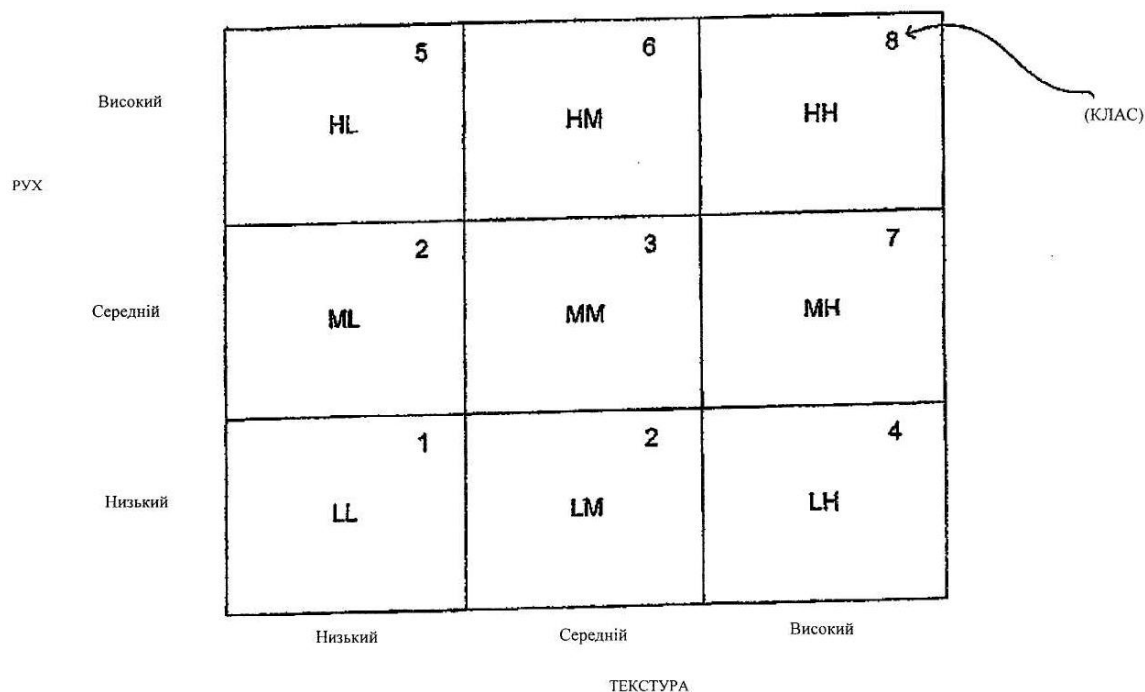


Фіг. 6

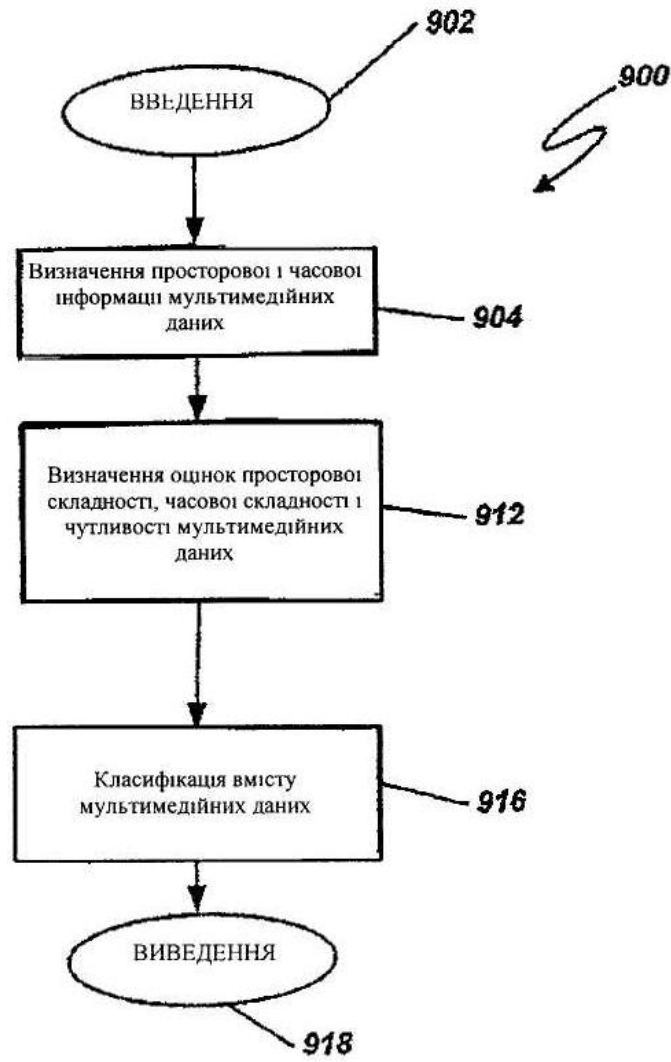




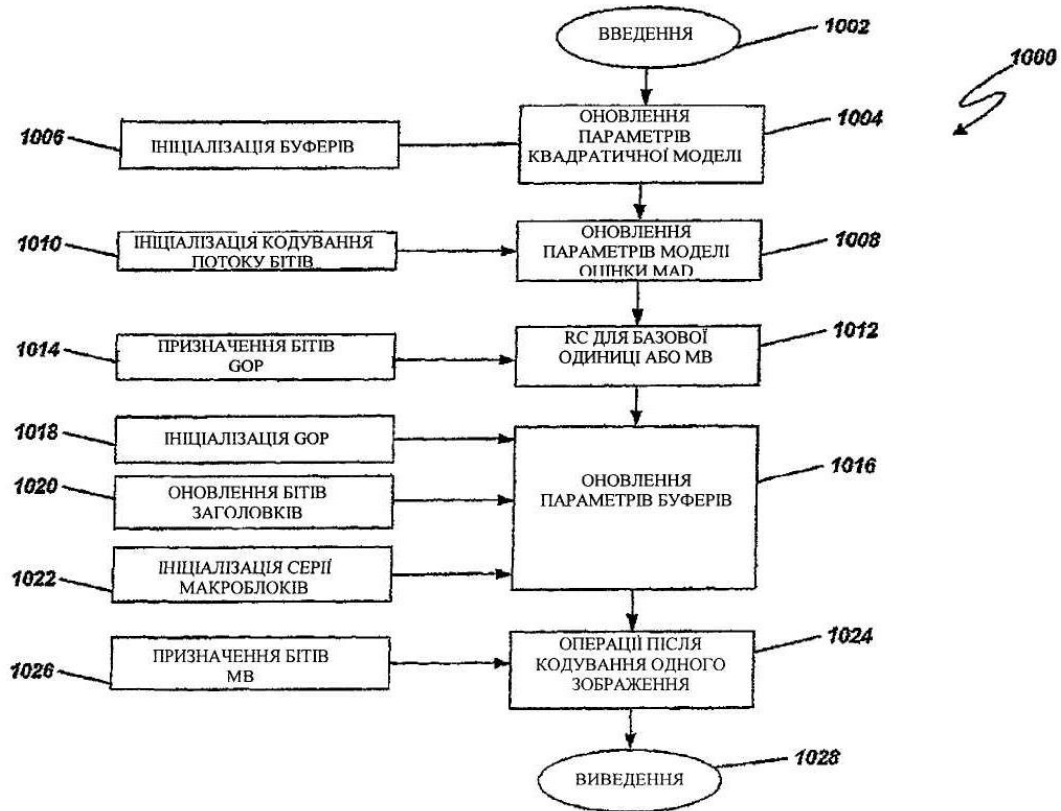
Фіг. 7



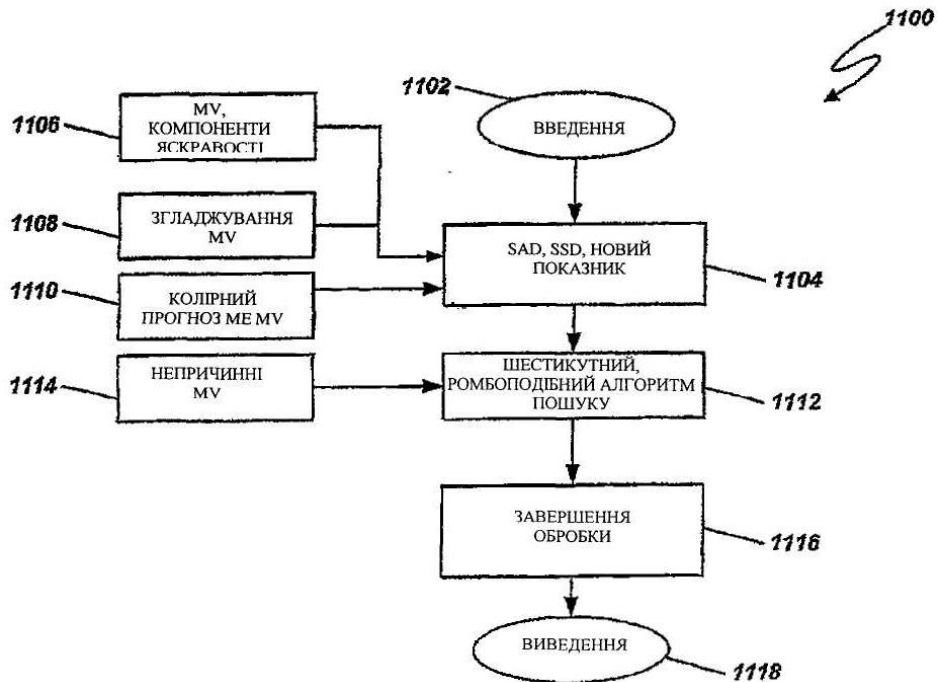
Фіг. 8



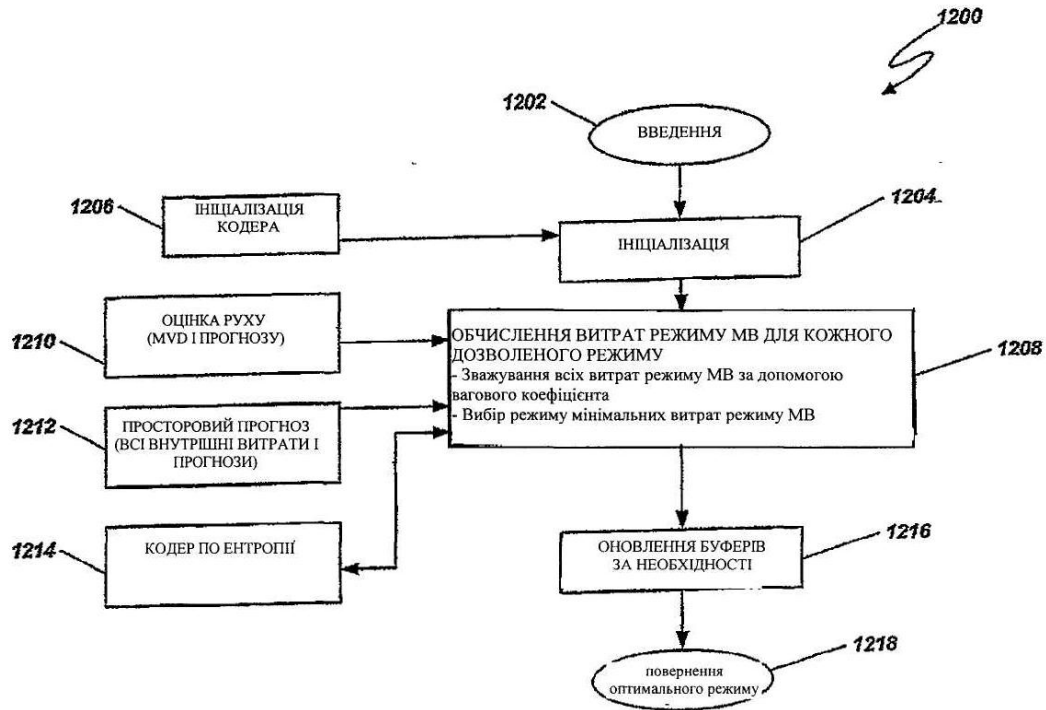
Фіг. 9



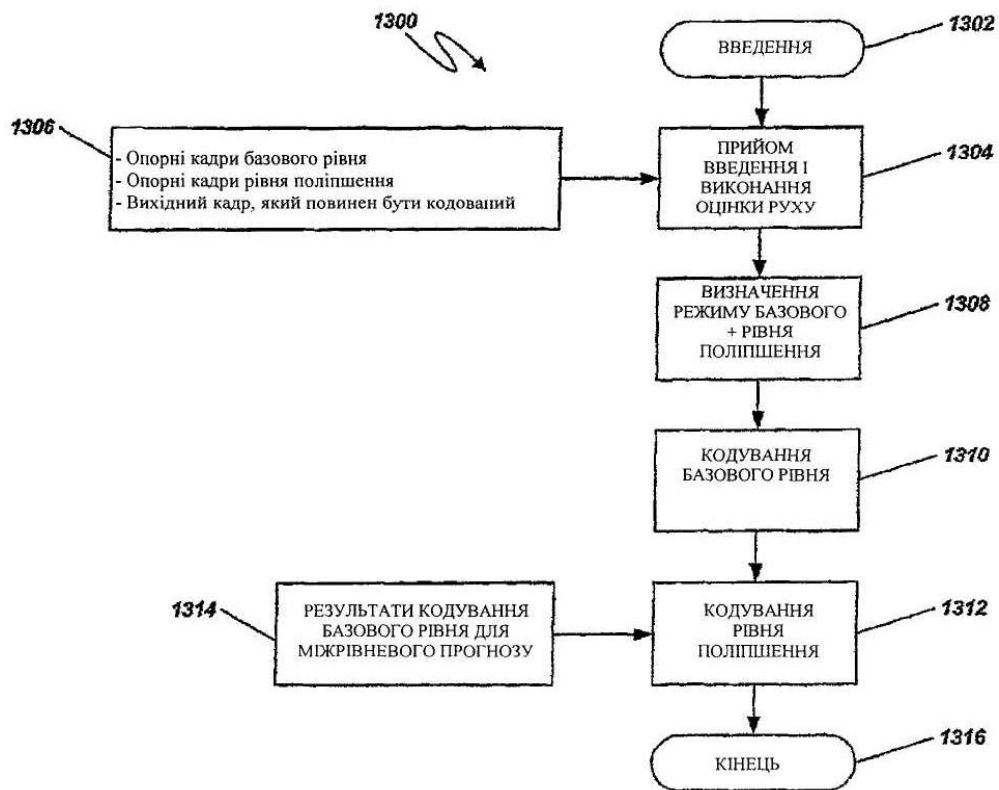
Фіг. 10



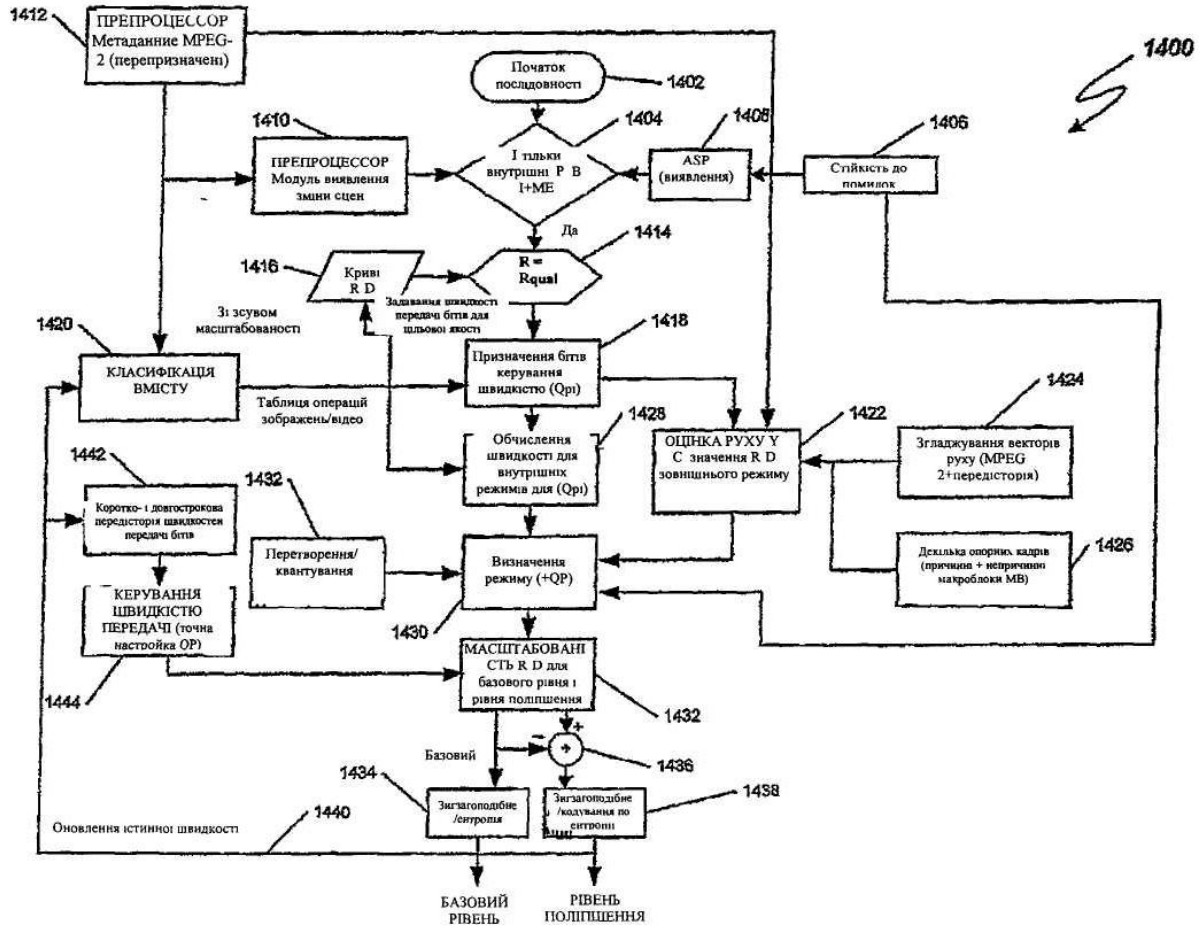
Фіг. 11



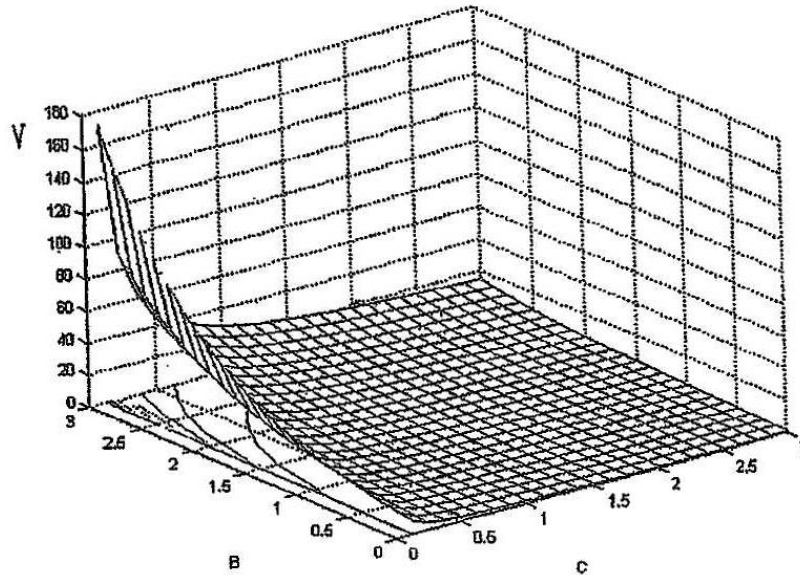
Фіг. 12



Фіг. 13



Фіг. 14



Фіг. 15

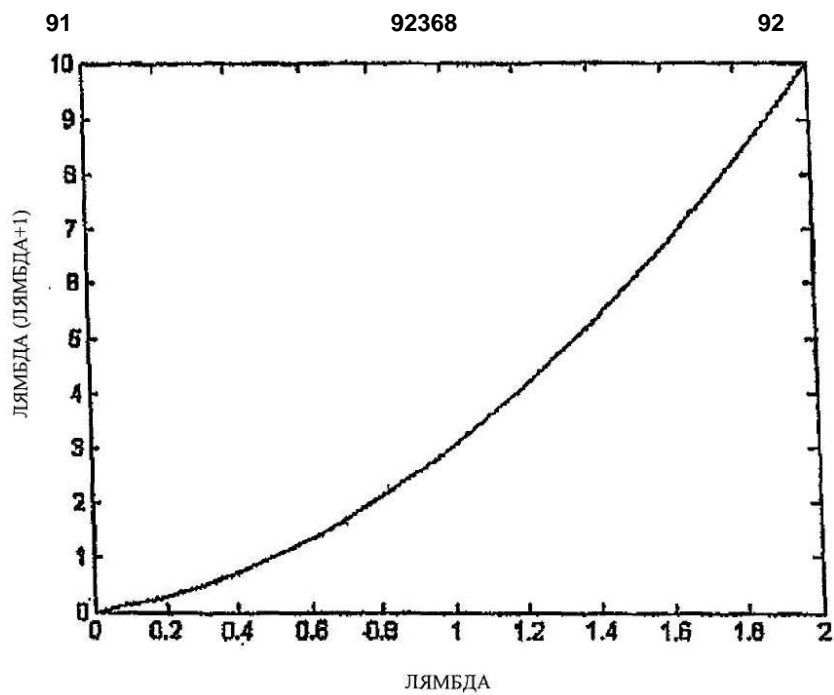


Fig. 16

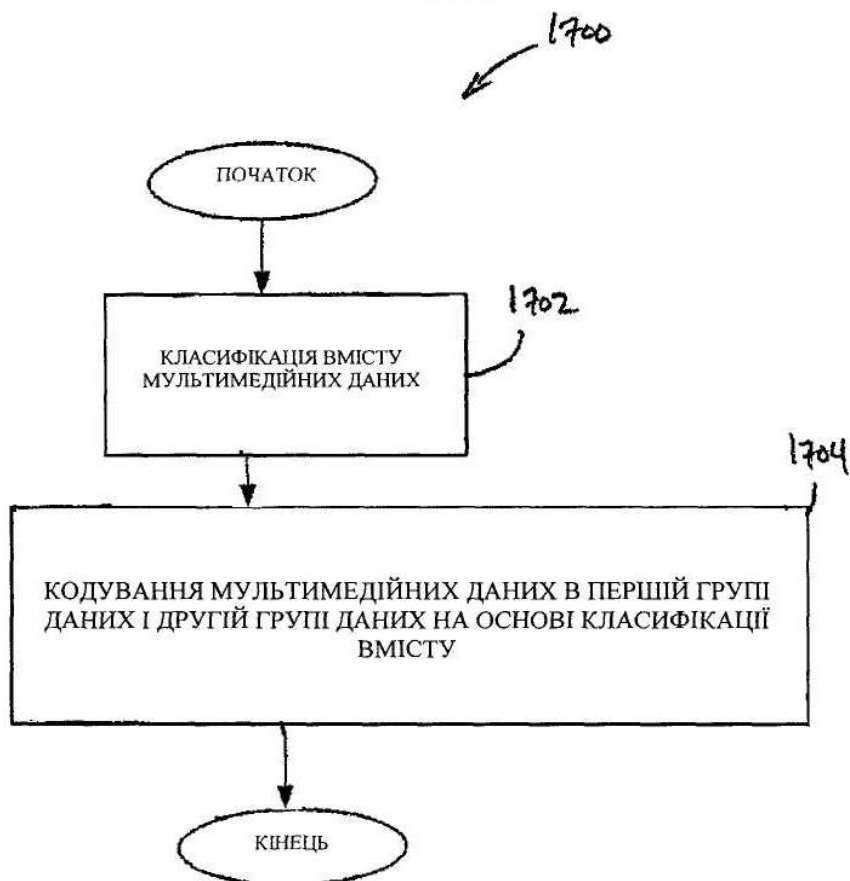
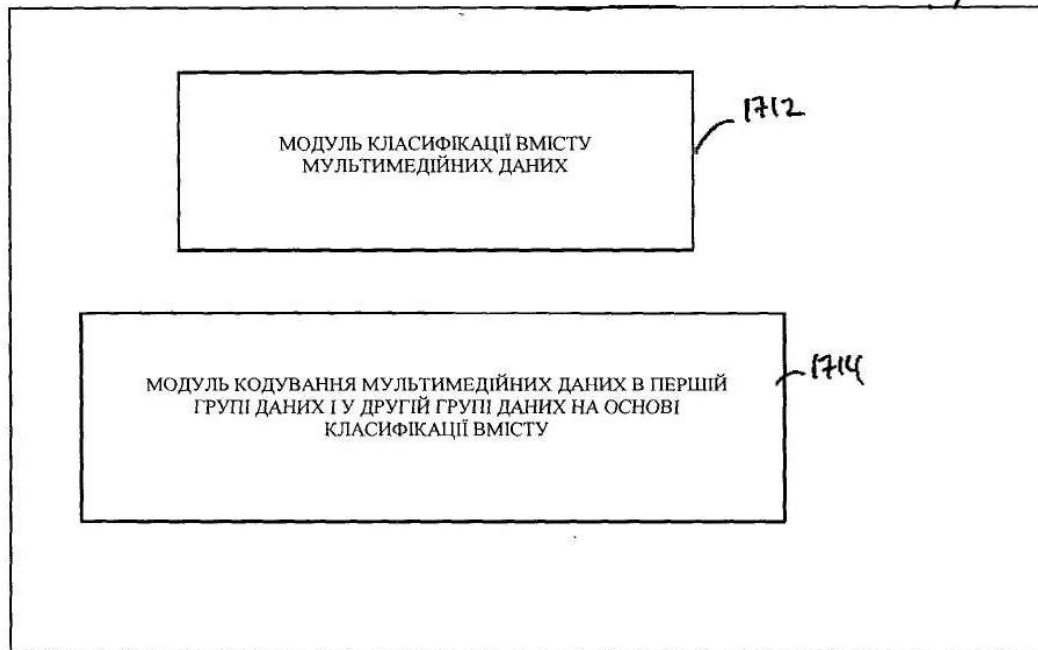
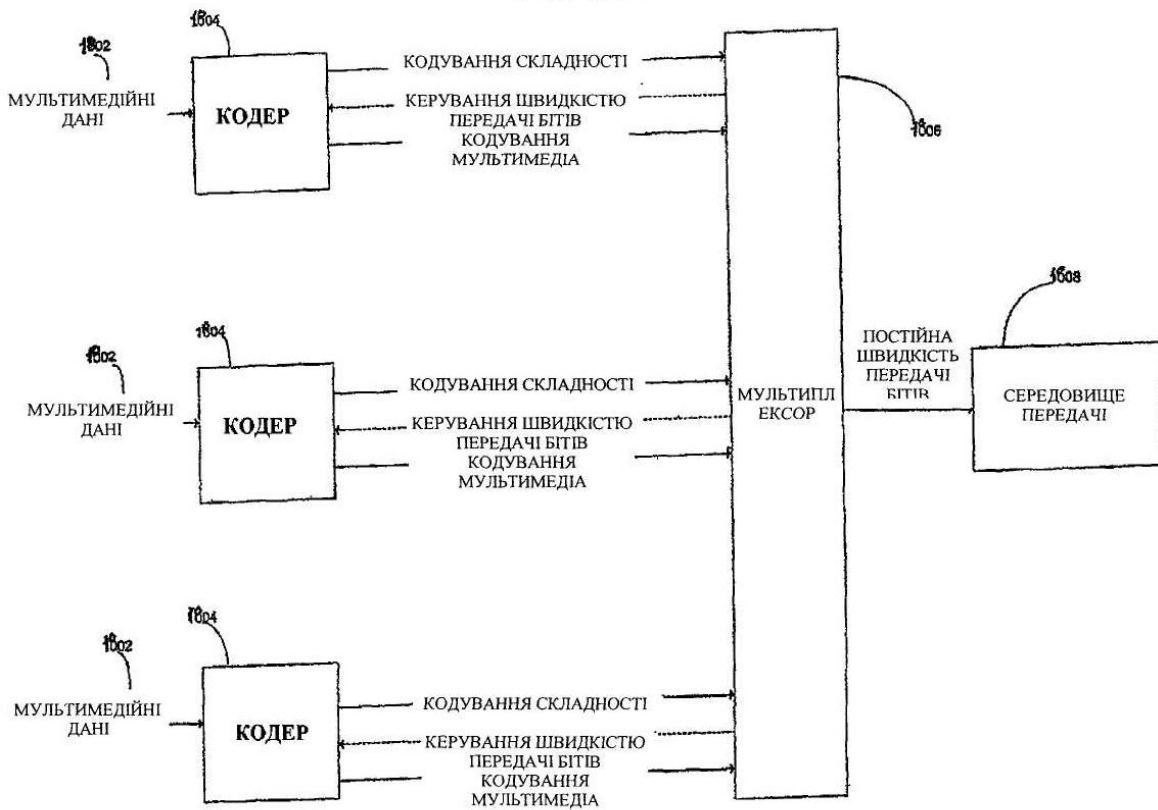


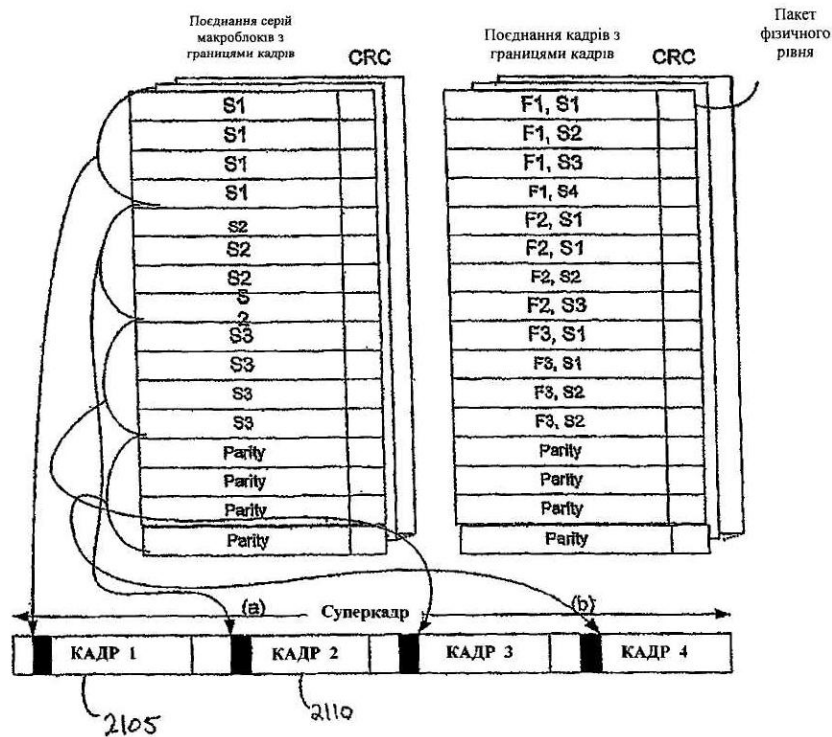
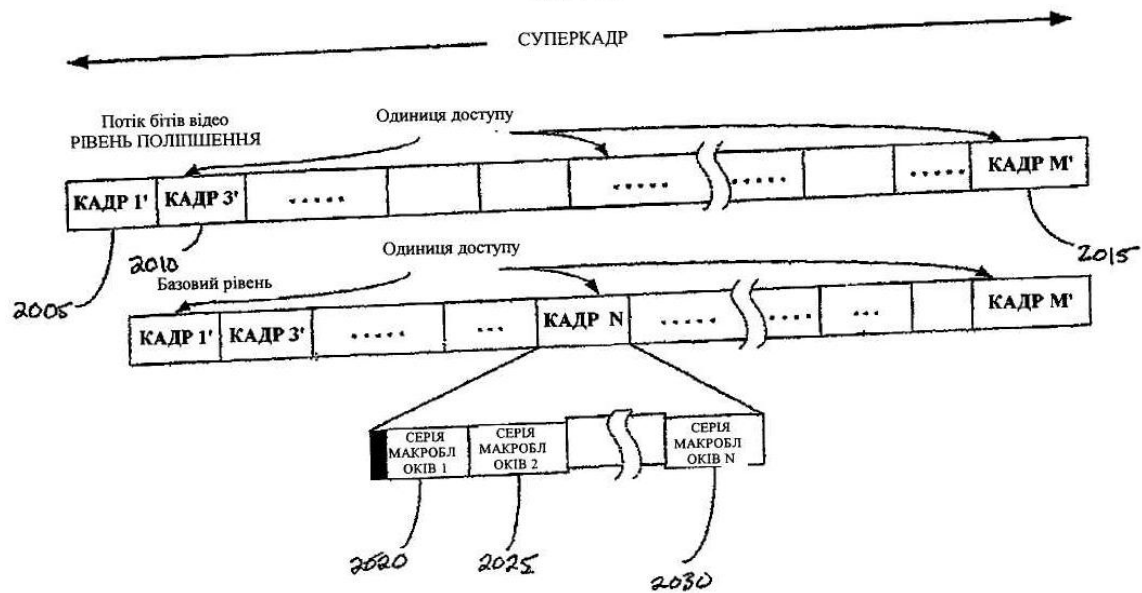
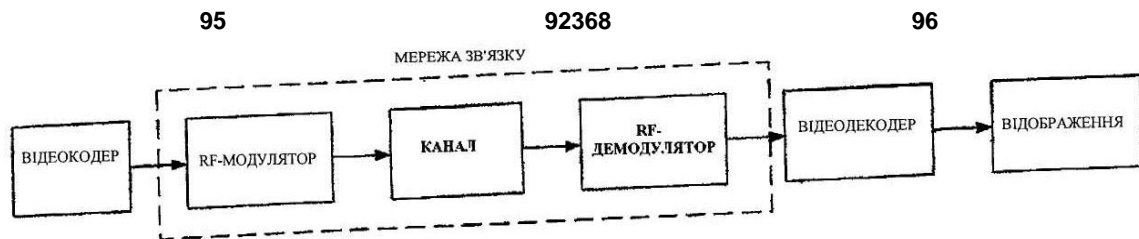
Fig. 17A



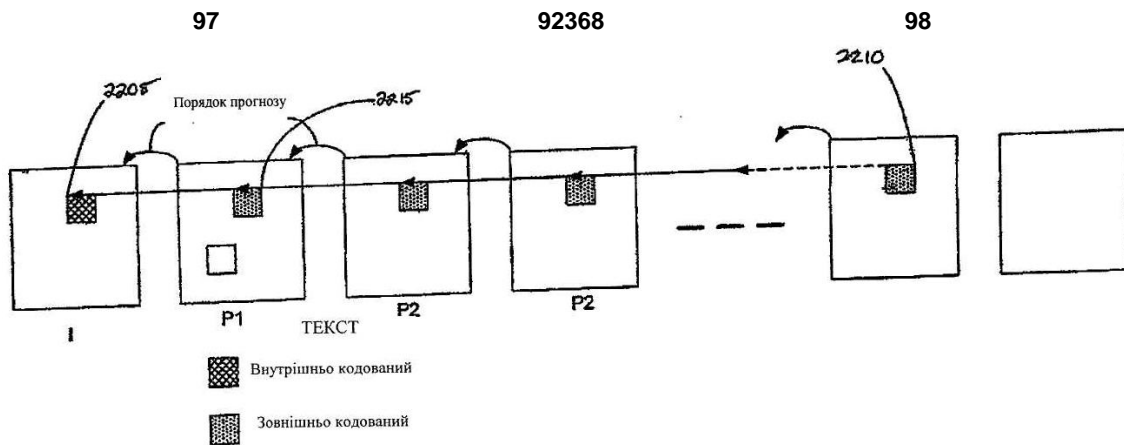
Фіг. 17В



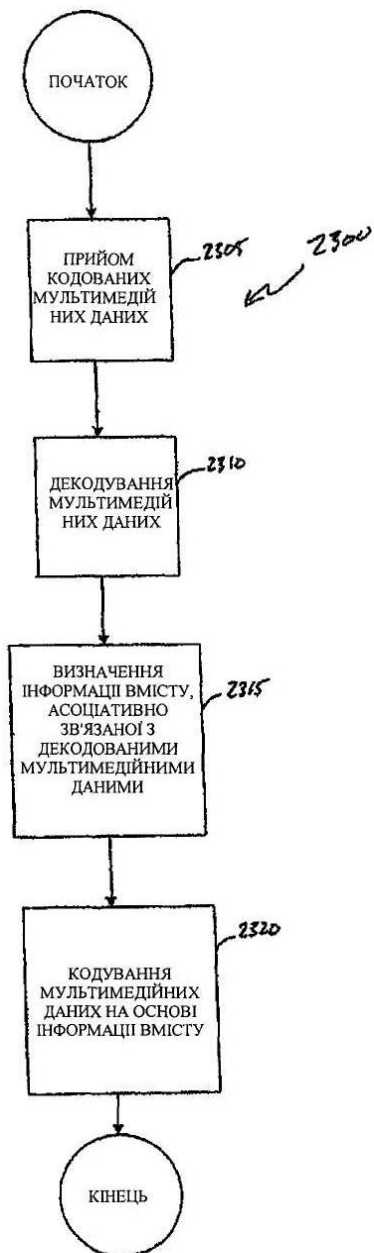
Фіг. 18







Фіг. 22



Фіг. 23



Фіг. 24

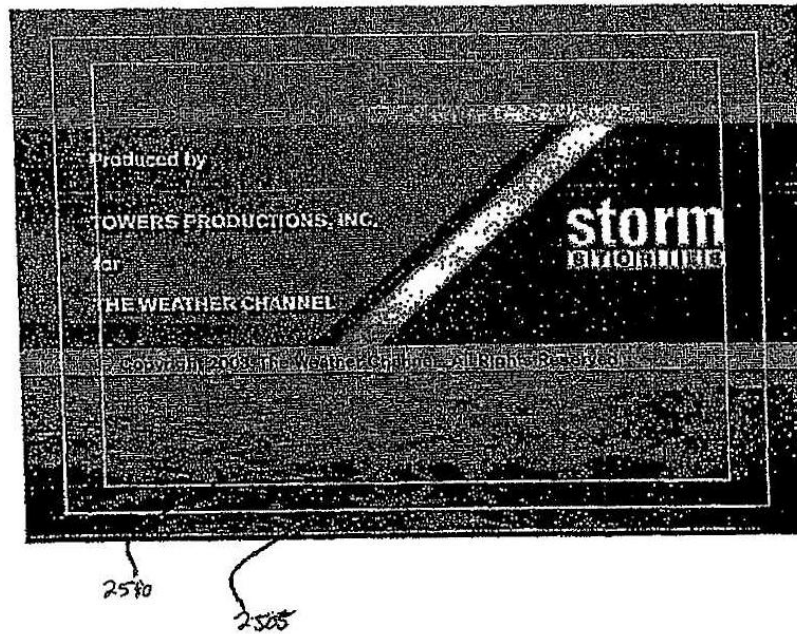


Fig. 25

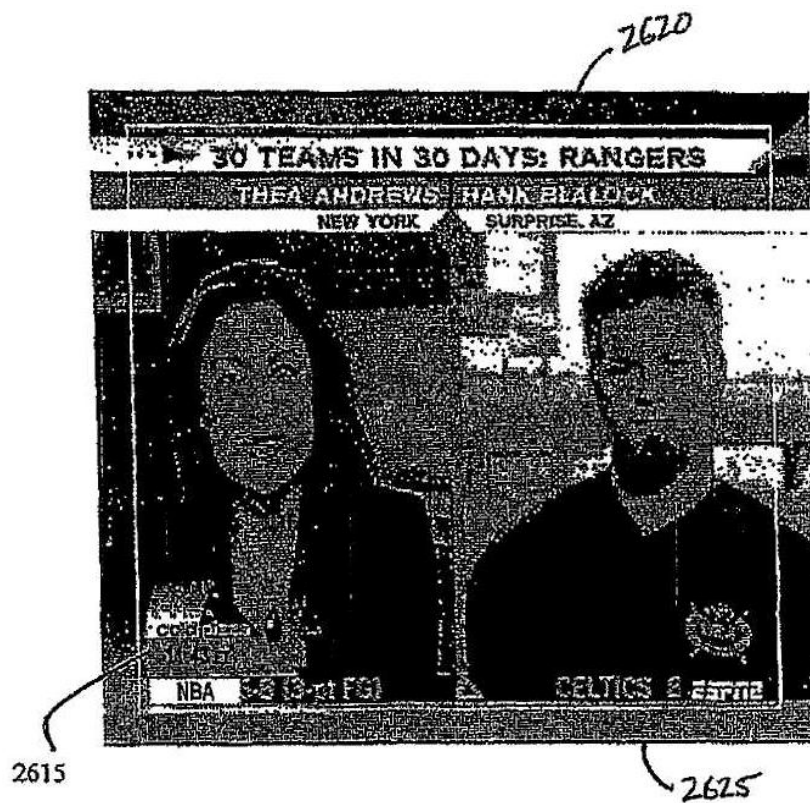
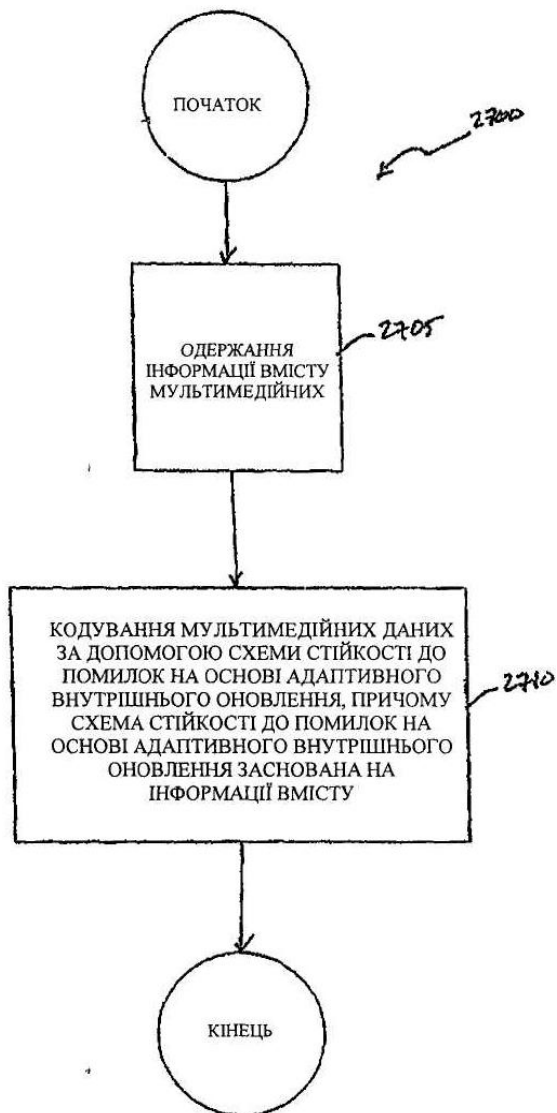
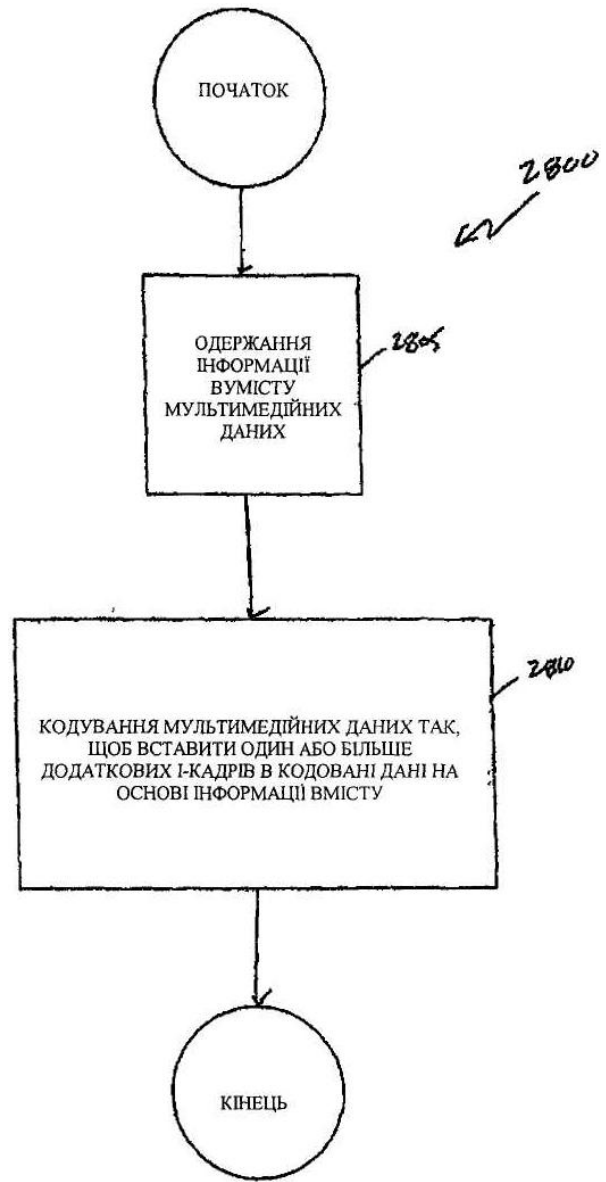


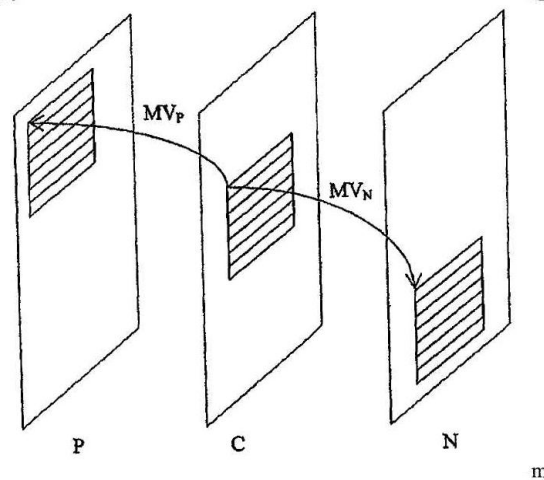
Fig. 26



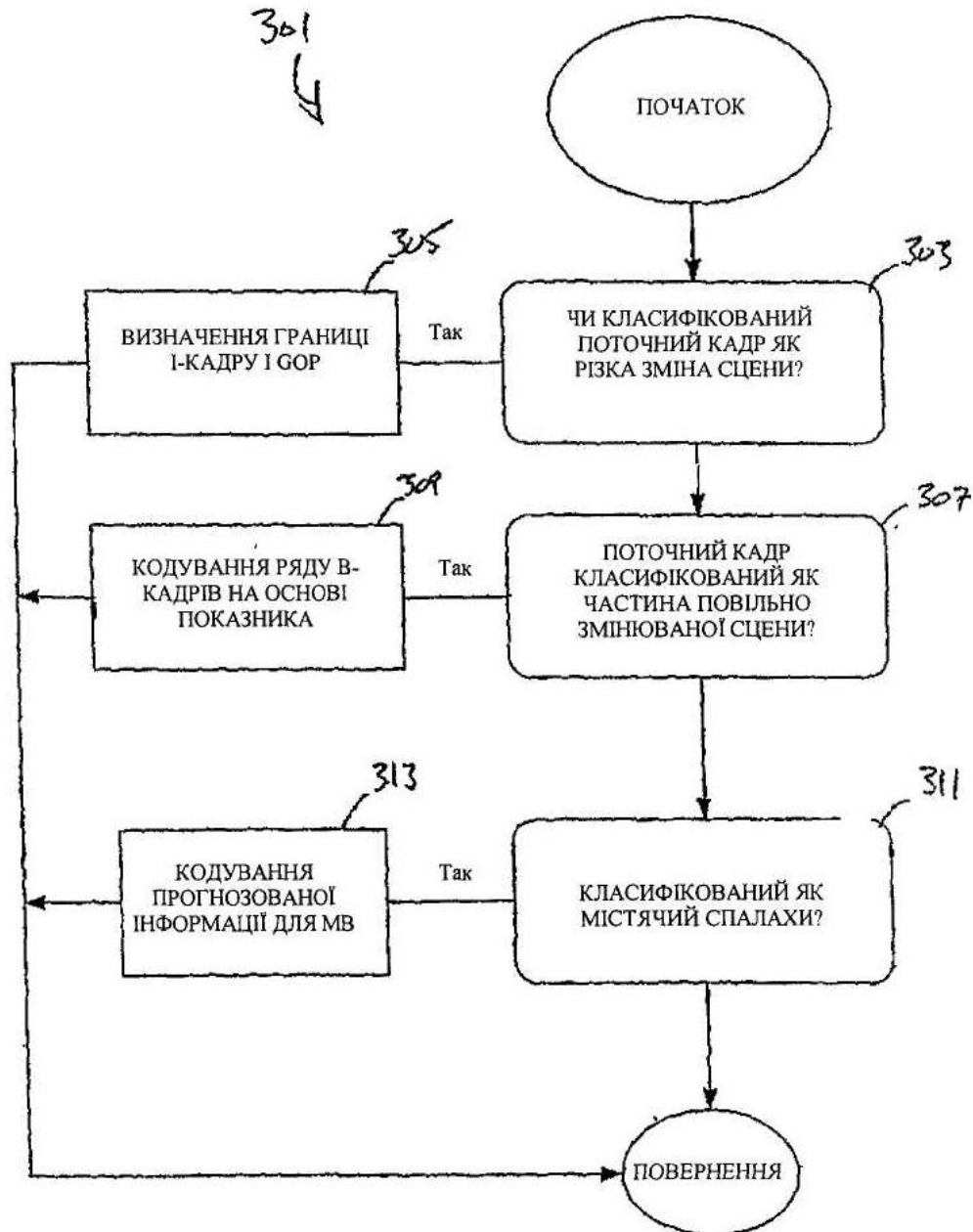
Фіг. 27



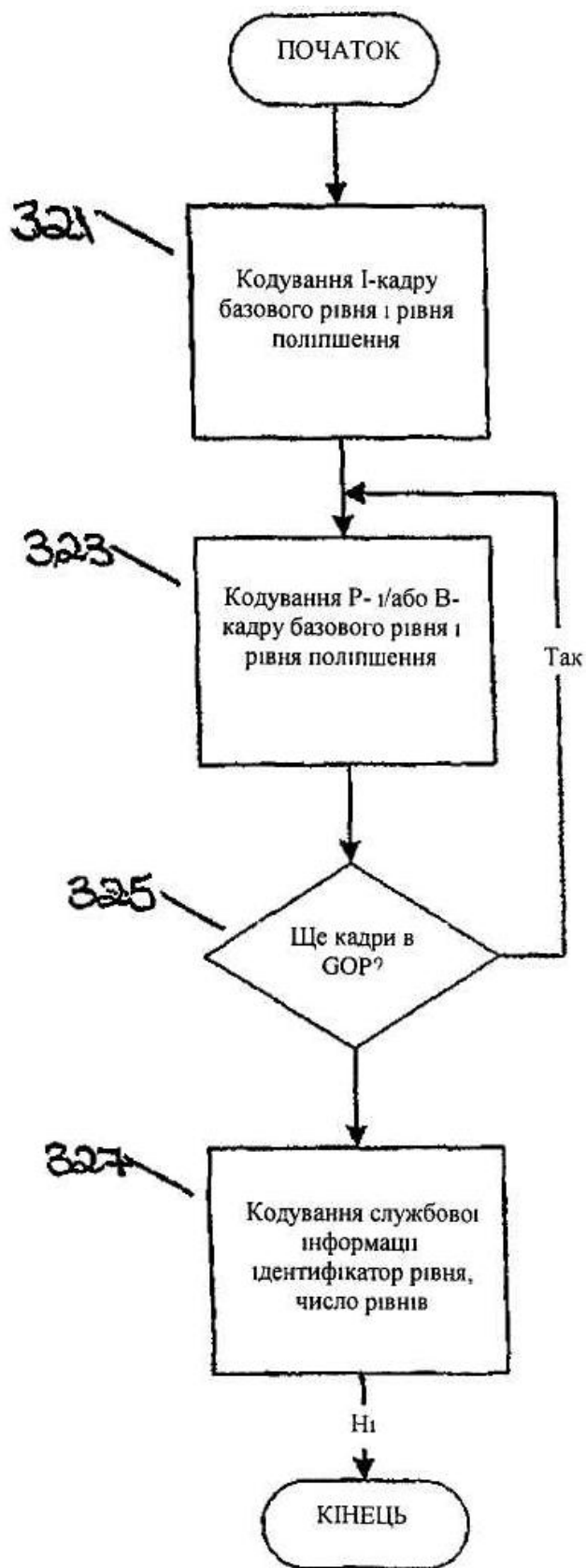
Фіг. 28



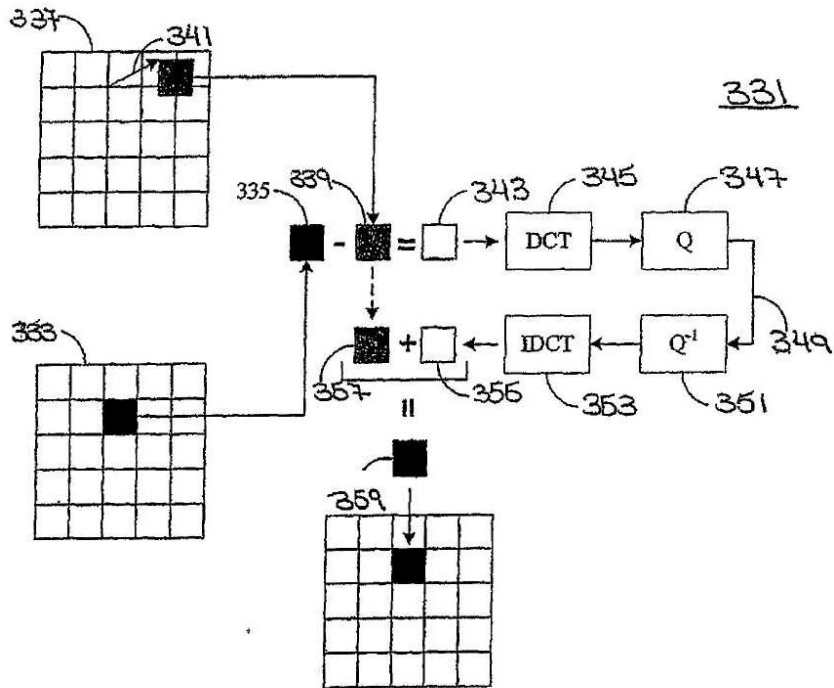
Фіг. 29



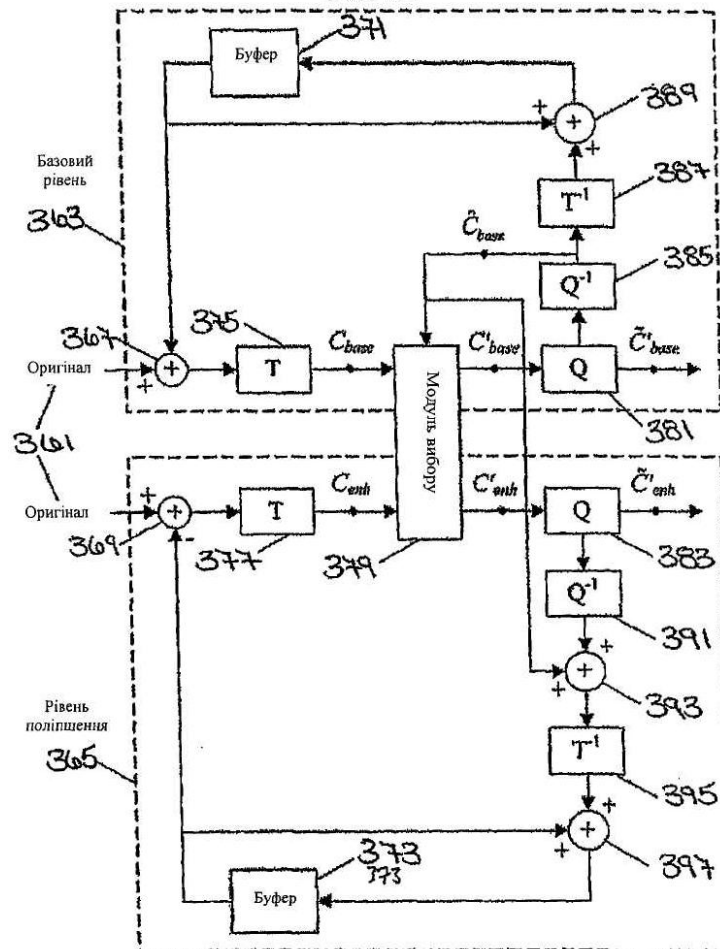
Фіг. 30



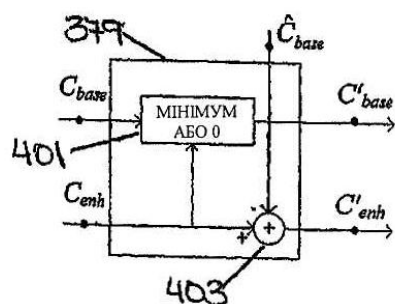
Фіг. 31



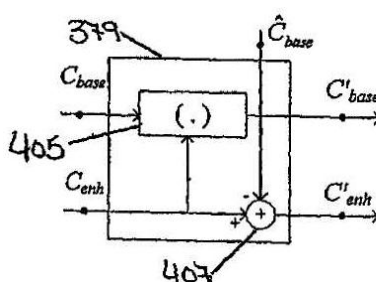
Фіг. 32



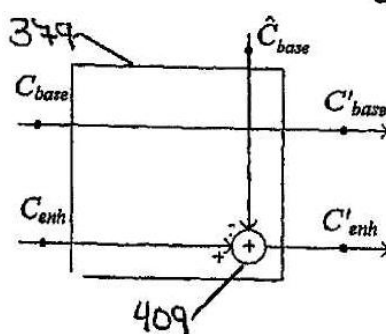
Фіг. 33



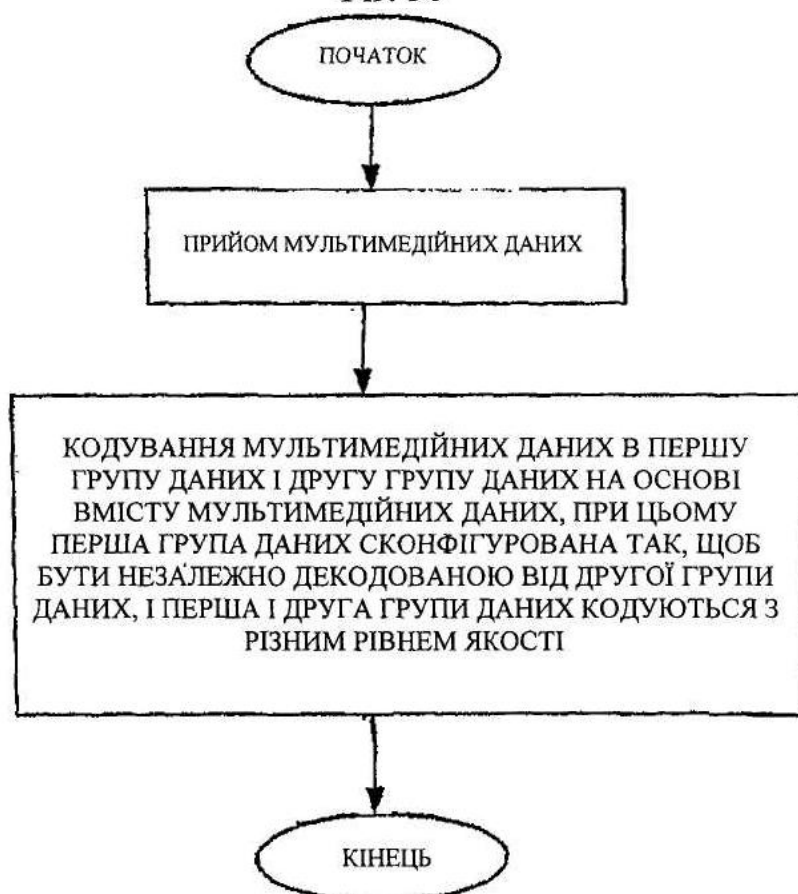
Фіг. 34



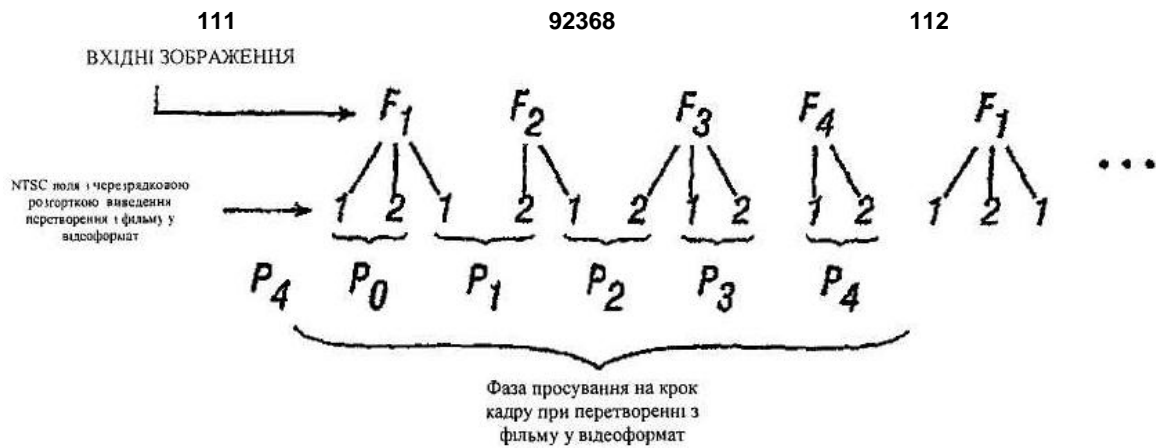
Фіг. 35



Фіг. 36

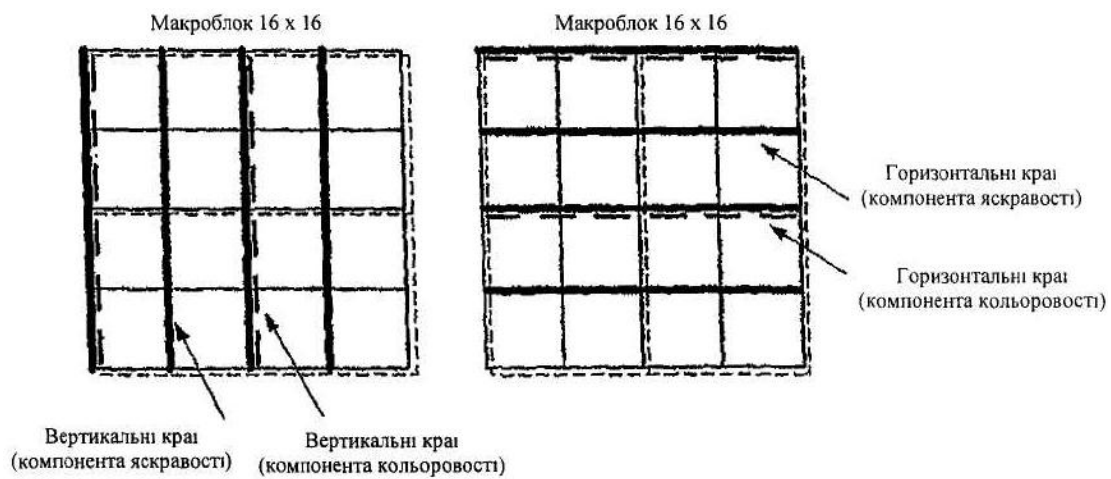


Фіг. 37



Можливі системні рішення

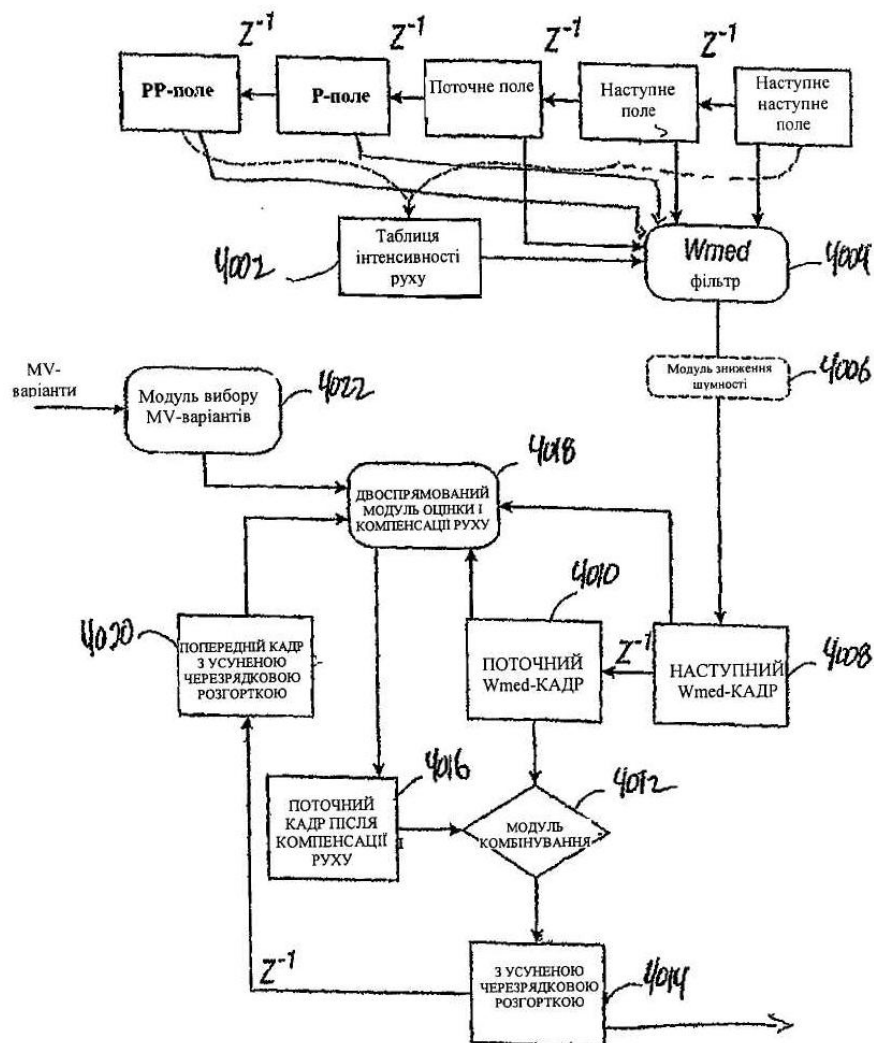
Фіг. 38



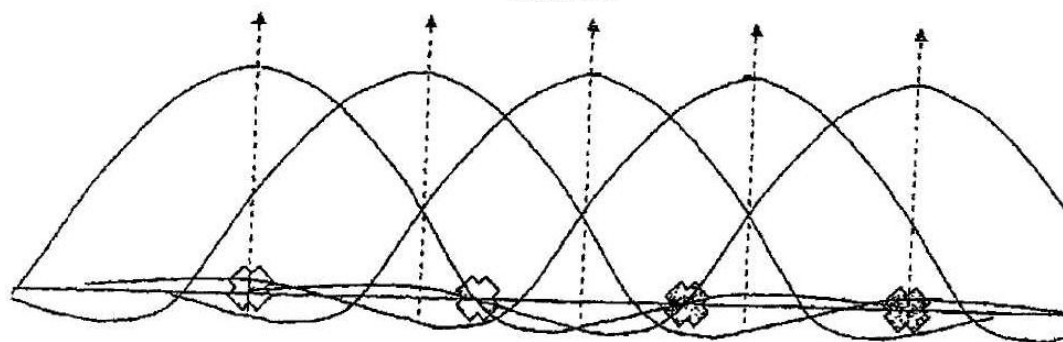
Границі в макроблоці, який повинен бути фільтрований (границі  
компоненти яскравості показані суцільними лініями, а границі  
компоненти кольоровості показані пунктирними лініями)

Фіг. 39



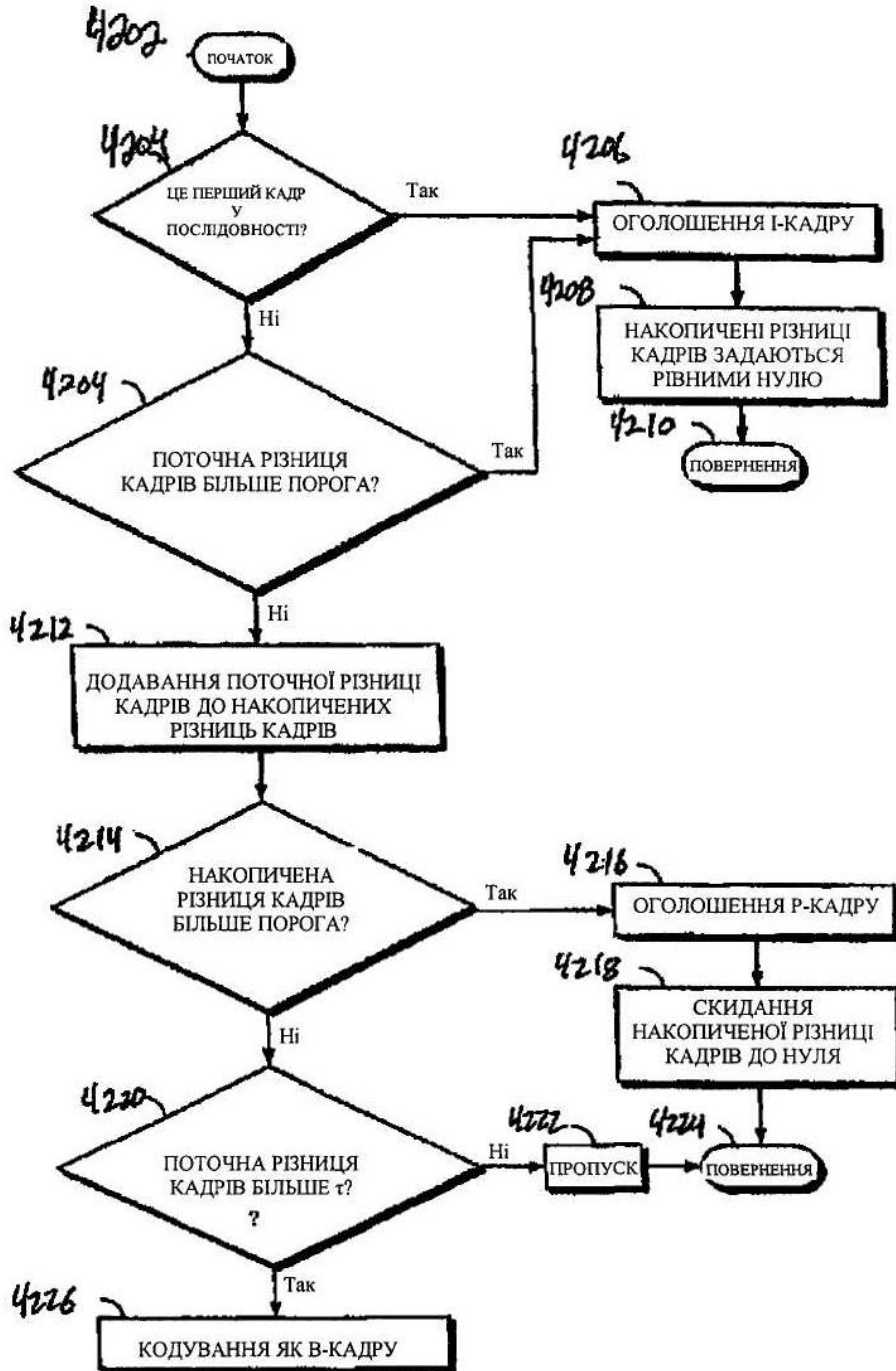


Фіг. 40



МОДУЛЬ ОДНОВИМІРНОЇ МОНОФАЗНОЇ ДИСКРЕТИЗАЦІЇ

Фіг. 41



Фіг. 42

(Блок-схема послідовності операцій  
способу адаптивного призначення  
типів кадрів кадрам)

В описі до патенту на винахід графічні зображення та текст подаються в редакції заявника

Комп'ютерна верстка О. Гапоненко

Підписне

Тираж 28 прим.

Міністерство освіти і науки України

Державний департамент інтелектуальної власності, вул. Урицького, 45, м. Київ, МСП, 03680, Україна

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