



SUOMI – FINLAND
(FI)

PATENTTI- JA REKISTERIHALLITUS
PATENT- OCH REGISTERSTYRELSEN



FI000120125B

(12) PATENTTIJULKAISU
PATENTSKRIFT

(10) FI 120125 B

(45) Patenti myönnetty - Patent beviljats

30.06.2009

(51) Kv.lk. - Int.kl.

H04N 7/32 (2006.01)

(21) Patentihakemus - Patentansökning

20001847

(22) Tekemispäivä - Ingivningsdag

21.08.2000

(24) Alkupäivä - Löpdag

21.08.2000

(41) Tullut julkiseksi - Blivit offentlig

22.02.2002

(73) Haltija - Innehavare

1 • Nokia Corporation, Helsinki, Keilalahdentie 4, 02150 Espoo, SUOMI - FINLAND, (FI)

(72) Keksijä - Uppfinnare

1 • Hannuksela, Miska, Kukkaniitynkatu 4 B, 33710 Tampere, SUOMI - FINLAND, (FI)

2 • Caglar, Kerem, Lähteenkatu 5 H 116, 33500 Tampere, SUOMI - FINLAND, (FI)

(74) Asiamies - Ombud

Nokia IPR-osasto, PL 206, 00045 Nokia Group

(54) Keksinnön nimitys - Uppfinningens benämning

Kuvankoodaus

Bildkodning

(56) Viitejulkaisut - Anförda publikationer

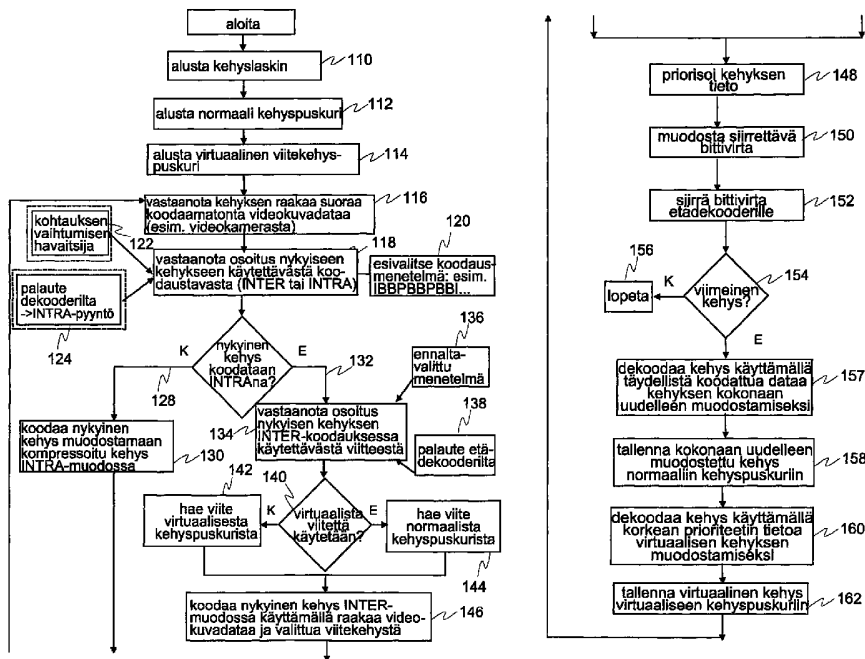
EP 844792 A, US 5528284 A, WO 01/06795 A, WO 01/89227 A,

Boyce, J.M.: Packet Loss Resilient Transmission of MPEG Video over Internet, Signal Processing: Image Communication 15 (1999) 7-25, p. 7-24, Elsevier, Alankomaat

(57) Tiivistelmä - Sammandrag

Menetelmä videokuva-signaalin koodaamiseksi käsittäen seuraavat vaiheet: koodataan ensimmäinen kokonainen kehyks muodostamalla bittivirta, joka sisältää tietoa kehyksen myöhempää täydellistä uudelleenmuodostamista (150) varten, joka tieto on priorisoitu (148) korkeamman ja matalamman prioriteetin tietoon; määritetään (160) vähintään yksi virtuaalikehyks ensimmäisen sellaisen kokonaisen kehyksen version perusteella, joka versio on muodostettu käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa, ja koodataan (146) toinen kokonainen kehyks muodostamalla bittivirta, joka sisältää tietoa sen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tietoon, mikä mahdollistaa toisen kokonaisen kehyksen täydellisen uudelleenmuodostamisen virtuaalikehyksen perusteella eikä ensimmäisen kokonaisen kehyksen perusteella. Vastaava dekodausmenetelmä on myös selostettu.

Ett förfarande för kodning av en videosignal, vilket förfarande omfattar följande steg: att koda en första komplett ram genom att skapa en bitström som innehåller information för senare, full rekonstruktion (150) av ramen, varvid informationen är prioriterad (148) till information med hög respektive låg prioritet; att definiera (160) åtminstone en virtuell ram på basis av en första komplett ramversion som har skapats med användande av högprioritetsinformationen om den första kompletta ramen i frånvaro av åtminstone en del av lågprioritetsinformationen om den första kompletta ramen; och att koda (146) en andra komplett ram genom att skapa en bitström som innehåller information för senare, full rekonstruktion av den, varvid informationen är prioriterad till information med hög respektive låg prioritet, vilket möjliggör full rekonstruktion av den andra kompletta ramen på basis av den virtuella ramen och inte på basis av den första kompletta ramen. Även motsvarande avkodningsförfarande beskrivs.



KUVANKOODAUS

Esillä oleva keksintö liittyy tiedonsiirtoon ja liittyy erityisesti, mutta siihen rajoittumatta, kuvasarjoja, kuten videokuvaa, edustavan tiedon siirtämiseen.

- 5 Keksintö soveltuu erityisen hyvin virheille ja tiedon häviölle alttiiden linkkien välityksellä tapahtuvaan tiedonsiirtoon, kuten matkaviestinjärjestelmän ilmarajapinnan välittämä tiedonsiirto.

- 10 Viime vuosien aikana internetin välityksellä saatavilla olevan multimediasisällön määrä on kasvanut huomattavasti. Koska matkaviestimien tiedonvälitysnopeudet ovat nousemassa riittävän suuriksi multimediasisällön hakemista varten, on tarpeen mahdollistaa kyseinen hakeminen internetistä. Esimerkki nopeasta tiedonvälitysjärjestelmästä on suunnitellun GSM 2+ -vaiheen pakettikytkentäinen datapalvelu GPRS (General Packet Radio Service).

15

Termillä multimedia tarkoitetaan tässä sekä ääntä että kuvaa, pelkkää ääntä tai pelkkää kuvaa. Ääni voi olla puhetta ja musiikkia.

- 20 Multimediasisällön lähettäminen internetissä on pakettikytkentäistä. Verkkoliikenne internetin välityksellä perustuu kuljetusprotokollaan (transport protocol) nimeltä internetprotokolla (IP). IP:n avulla datapaketteja kuljetetaan paikasta toiseen. Se helpottaa pakettien reitittämistä väliyhdykäytävien kautta, toisin sanoen se sallii tiedon lähettämisen koneille, jotka eivät ole suoraan kytkettyjä samaan fyysiseen verkkoon. IP-kerroksen kuljettama datayksikkö on
- 25 nimeltään IP-datagrammi. IP:n tarjoama välityspalvelu on yhteydetön, toisin sanoen IP-datagrammit reititetään internetin kautta toisistaan riippumatta. Koska yhdyskäytävien sisällä ei ole tiettyä yhteyttä varten pysyvästi varattuja resursseja, yhdyskäytävät voivat toisinaan joutua hylkäämään datagrammeja puskuritilan tai muun resurssin puutteen vuoksi. Näin ollen IP:n tarjoama
- 30 välityspalvelu on pikemminkin paras mahdollinen saatavilla oleva (best effort service) kuin taattu palvelu.

Internetissä olevaa multimediaa suoratoistetaan tavallisesti käyttämällä UDP:tä (User Datagram Protocol), TCP:tä (Transmission Control Protocol) tai HTTP:tä

(Hypertext Transfer Protocol). UDP ei tarkista, onko datagrammit vastaanotettu, ei lähetä puuttuvia datagrammeja uudelleen eikä takaa, että datagrammit vastaanotetaan samassa järjestyksessä kuin ne on lähetetty. UDP on yhteydetön. TCP tarkistaa, että datagrammit on vastaanotettu ja lähettää puuttuvat datagrammit uudelleen. Se myös takaa, että datagrammit vastaanotetaan samassa järjestyksessä kuin ne on lähetetty. TCP on yhteydellinen.

Riittävän laadukkaan multimediasisällön toimittamisen varmistamiseksi sisältö voidaan toimittaa luotettavan verkkoyhteyden, kuten TCP:n välityksellä; näin voidaan varmistaa vastaanotettu datan virheettömyys ja oikea järjestys. Kadonneet tai vahingoittuneet protokolladatayksiköt (protocol data units) lähetetään uudelleen.

Toisinaan kadonneen tiedon uudelleenlähettämistä ei käsittele kuljetusprotokolla vaan joku korkeamman tason protokolla. Tällainen protokolla voi valita multimediasisällöstä tärkeimmät puuttuvat osat ja pyytää niiden uudelleenlähettämistä. Tärkeimpiä osia voidaan käyttää esimerkiksi virran muiden osien ennustamiseen.

20

Multimediasisältö sisältää tavallisesti videokuvaa. Jotta videokuvan lähettäminen olisi tehokasta, videokuva tavallisesti kompressoituu. Sen vuoksi kompressointitehokkuus on tärkeä parametri videokuvansiirtojärjestelmissä. Toinen tärkeä parametri on siirtovirheiden toleranssi. Kumman tahansa parametrin tehostaminen vaikuttaa haitallisesti toiseen, sen vuoksi videokuvansiirtojärjestelmässä parametrien välillä pitää olla sopiva tasapaino.

Kuvio 1 esittää erästä videokuvansiirtojärjestelmää. Järjestelmä käsittää lähdekooderin (source coder), joka kompressoituu kompressoimattoman videokuvasignaalin haluttuun bittinopeuteen ja tuottaa siten koodatun ja kompressoituu videokuvasignaalin, sekä lähdekooderin (source decoder), joka dekodaa koodatun ja kompressoituu videokuvasignaalin ja muodostaa uudelleen kompressoimattoman videokuvasignaalin. Lähdekooderi käsittää aaltomuotokooderin (waveform coder) ja entropiakooderin (entropy coder).

Aaltomuotokooderi kompressoii videokuvasignaalin häviöllisesti ja entropiakooderi muuntaa häviöttömästi aaltomuotokooderin lähdon binäärijaksoksi. Binäärijakso siirretään lähdekooderilta kuljetuskooderille (transport coder), joka kapseloi kompressoitua videokuvan sopivan
5 kuljetusprotokollan mukaisesti ja lähettää sen vastaanottimelle, joka käsittää kuljetusdekooderin (transport decoder) ja lähdedekooderin. Kuljetuskooderi lähettää tiedon kuljetusdekooderille siirtokanavan yli. Kuljetuskooderi voi käsitellä kompressoitua videokuvaa myös muilla tavoin. Se voi esimerkiksi lomittaa ja moduloida tietoa. Vastaanotettuaan tiedon kuljetusdekooderi välittää sen
10 lähdedekooderille. Lähdedekooderi käsittää aaltomuotodekooderin (waveform decoder) ja entropiadekooderin (entropy decoder). Kuljetusdekooderi ja lähdedekooderi suorittavat käänteiset toiminnot tuodakseen uudelleenmuodostetun videosignaalin näyttöön. Vastaanotin voi myös antaa lähettimelle palautetta. Vastaanotin voi esimerkiksi osoittaa onnistuneesti
15 vastaanotettujen tietoyksikköjen siirtonopeuden.

Videokuvasarja koostuu sarjasta stillkuvia. Videokuvasarja kompressoidaan vähentämällä sen redundanteja ja havaitsemisen kannalta merkityksettömiä osia. Videokuvajakson redundanssi voidaan luokitella spatiaaliseksi, ajalliseksi ja
20 spektraaliseksi. Spatiaalinen redundanssi viittaa korrelaatioon naapuripikselien välillä saman videokuvan sisällä. Ajallinen redundanssi viittaa siihen, että edellisessä videokuvassa esiintyvät kohteet esiintyvät todennäköisesti myös tämän hetkessä videokuvassa. Spektraalinen redundanssi viittaa korrelaatioon videokuvan eri värikomponenttien välillä.

25

Ajallista redundanssia voidaan vähentää luomalla liikekompensointitietoa, joka kuvaa suhteellista liikettä esillä olevan ja edellisen kuvan välillä (tästä käytetään nimitystä vertailu- tai ankkurikuva). Käytännössä esillä oleva videokuva muodostetaan ennusteena edelliselle, ja menetelmä, jolla ennuste saadaan
30 aikaan, tunnetaan yleisesti nimellä liikekompensoitu ennustaminen (motion compensated prediction) tai liikekompensointi (motion compensation). Sen lisäksi, että kokonainen videokuva voidaan ennustaa edellisen perusteella, yksittäisen videokuvan osat tai alueet voidaan ennustaa kyseisen videokuvan muiden osien tai alueiden perusteella.

Yleensä pelkästään vähentämällä videokuvajakson redundanssia ei saavuteta riittävää kompressoititasoa. Sen vuoksi videokooderit pyrkivät myös vähentämään niiden videokuvasarjan osien laatua, jotka ovat subjektiivisesti vähemmän tärkeitä. Lisäksi koodatun bittivirran redundanssia vähennetään kompressoointiparametrien ja -kertoimien tehokkaan häviöttömän koodauksen avulla. Merkittävin menetelmä on vaihtuvamittaisten koodien (variable length codes) käyttäminen.

10 Kuvankompressointimenetelmät yleensä erittelevät kuvat sen perusteella, hyödyntävätkö ne ajallista redundanssin vähentämistä vai eivät (toisin sanoen, onko ne ennustettu vai ei). Kompressoituja kuvia, jotka eivät hyödynnä ajallisia redundanssin vähentämismenetelmiä, kutsutaan yleensä INTRA- tai I-kehyksiksi. INTRA-kehyksiä käytetään usein estämään pakettihäviön vaikutusten leviämistä spatiaalisesti ja ajallisesti. INTRA-kehysten ansiosta uudet vastaanottimet voivat broadcast-tilanteissa aloittaa virran dekoodaamisen, toisin sanoen ne tarjoavat "liityntäpisteitä" (access points). Kuvankoodausjärjestelmät yleensä mahdollistavat INTRA-kehysten lisäämisen jaksottaisesti n sekunnin välein. On myös edullista käyttää INTRA-kehyksiä kohdissa, joissa otos vaihtuu ja joissa kuvan sisältö muuttuu niin merkittävästi, että ajallinen ennustaminen edellisen kuvan perusteella ei todennäköisesti onnistu tai ole toivottavaa kompressointitehokkuuden kannalta.

20 Kompressoituja kuvia, jotka hyödyntävät ajallisia redundanssin vähentämismenetelmiä, kutsutaan yleensä INTER- tai P-kehyksiksi. Liikekompensointia käyttävät INTER-kehukset ovat harvoin tarpeeksi täsmällisiä salliakseen riittävän tarkan kuvan uudelleenmuodostamisen, joten jokaiseen INTER-kehukseen liittyy myös spatiaalisesti kompressoitu ennustusvirhekuva. Tämä edustaa esillä olevan kehyksen ja sen ennusteen välistä eroa.

30 Kuvajoukko (group of pictures, GOP) on INTRA-kehys ja sarja siitä ajallisesti ennustettuja kuvia.

Yksi koodatun bittivirran hyödyllinen ominaisuus on skaalattavuus (scalability). Seuraavassa kuvataan bittinopeuksista skaalattavuutta (bit-rate scalability), joka viittaa kompressoitujen kuvasarjojen kykyyn tulla dekodatuksi eri datanopeuksilla. Tällainen kompressoitu sarja voidaan suoratoistaa eri kaistanleveyksisten kanavien yli ja se voidaan dekodata ja toistaa reaaliajassa eri vastaanotinpäätteillä.

Skaalattava multimedia järjestetään tyypillisesti hierarkkisiin tietokerroksiin. Pohjakerros (base layer) sisältää itsenäisen esitysmuodon multimedialeikkeestä kuten videokuvajaksosta, ja parannuskerrokset (enhancement layers) sisältävät jalostustietoa pohjakerroksen lisäksi. Multimedialeikkeen laatu paranee asteittain kun parannuskerroksia lisätään pohjakerrokseen.

Skaalattavuus on toivottava ominaisuus heterogeenisissä ja virheille alttiissa ympäristöissä kuten internetissä ja matkaviestinverkkojen langattomilla kanavilla. Tämä ominaisuus on toivottava erilaisten rajoitusten kuten bittinopeutta, näytön resoluutiota, verkon läpäisykykyä ja dekooderin monimutkaisuutta koskevien rajoitusten torjumiseksi.

Monipiste- ja broadcast-multimediasovelluksissa verkon läpäisykykyä koskevia rajoituksia ei ehkä voida ennakoida koodaushetkellä. Siksi on edullista koodata multimediasisältö muodostamaan skaalattava bittivirta. Kuviossa 3 esitetään esimerkki IP-monijakelussa (multi-casting) käytettävästä skaalattavasta bittivirrasta. Kukin reititin (R1-R3) voi poistaa bittivirtaa ominaisuuksiensa mukaisesti. Tässä esimerkissä palvelimella on multimedialeike, joka voidaan skaalata ainakin kolmelle bittinopeudelle, 120 kbit/s, 60 kbit/s ja 28 kbit/s. Monijakelulähetyksessä (multi-casting transmission), missä sama bittivirta toimitetaan usealle asiakaslaitteelle samanaikaisesti niin, että luodaan verkkoon mahdollisimman vähän kopioita bittivirrasta, on verkon kaistanleveyden kannalta edullista lähettää yksittäinen bittinopeusskaalattava (bit-rate scalable) bittivirta.

Jos kuvasarja alasladataan ja toistetaan eri laitteissa, joilla on erilaiset käsittelytehot, bittinopeusskaalattavuutta voidaan käyttää heikomman käsittelytehon omaavissa laitteissa, joissa voidaan dekodata vain osa

bittivirrasta ja siten saada heikompilaatuinen esitys videokuvasarjasta. Laitteissa, joissa on suurempi käsittelyteho, voidaan kuvasarja dekodata ja toistaa täysilaatuisena. Lisäksi bittinopeusskaalattavuus tarkoittaa sitä, että videokuvasarjan heikompilaatuisen esityksen dekoodaamiseen tarvittava käsittelyteho on pienempi kuin täysilaatuisen sarjan dekoodaamisen tarvittava teho. Tämä on yksi laskennallisen skaalattavuuden (computational scalability) muoto.

Jos videokuvasarja on ennalta tallennettu suoratoistopalvelimeen ja palvelimen on tilapäisesti vähennettävä bittivirran lähettämiseen käyttämänsä bittinopeutta esimerkiksi välttääkseen ruuhkan muodostumista verkossa, on edullista, jos palvelin voi vähentää bittivirran bittinopeutta ja samalla kuitenkin lähettää käyttökelpoista bittivirtaa. Tähän päästään tavallisesti käyttämällä bittinopeusskaalattavaa koodausta.

Skaalattavuutta voidaan käyttää sellaisen kuljetusjärjestelmän virheensietokyvyn parantamiseen, jossa kerrostettu koodaus on yhdistetty kuljetuksen priorisointiin (transport prioritisation). Termiä kuljetuksen priorisointi käytetään kuvaamaan mekanismeja, joilla tarjotaan erilaisia palvelun laatuja (qualities of service) kuljetuksessa. Näihin kuuluu epätasainen virheensuojaus (unequal error protection), joka tarjoaa erilaisia kanavavirhe-/häviönopeuksia, ja erilaisten prioriteettien määrittäminen erilaisia viive- ja häviövaatimuksia varten. Esimerkiksi skaalattavasti koodatun bittivirran pohjakerros voidaan toimittaa korkean virheensuojaustason omaavan siirtokanavan välityksellä, kun taas parannuskerrokset voidaan lähettää virheille alttiimpien kanavien välityksellä.

Yksi skaalattavan multimediakoodauksen ongelma on sen usein ei-skaalattavaa koodausta heikempi kompressointitehokkuus. Korkealaatuinen skaalattava videokuvasarja yleensä vaatii enemmän kaistanleveyttä kuin vastaavanlaatuinen ei-skaalattava yksikerroksinen videokuvasarja. Kuitenkin tähän sääntöön on olemassa poikkeuksia, esimerkiksi ajallisesti skaalattavat B-kehukset videokuvan kompressoinnissa saattavat parantaa koodauksen tehokkuutta varsinkin suurilla kehysnopeuksilla. Seuraavassa käsitellään B-kehysä.

- Useita videokuvan koodausstandardeja on ehdotettu. Eräs näistä on International Telecommunications Union (ITU) videokoodausta koskeva suositus H.263, jossa määritellään bittivirran syntaksi ja bittivirran dekodaus. Tällä hetkellä H.263:sta on olemassa kaksi versiota. Versio 1 käsittää ydinalgoritmin ja neljä valinnaista koodaustapaa. H.263:n versio 2 on version 1 jatke, joka tarjoaa 12 neuvoteltavissa oleva koodaustapaa. H.263:n parhaillaan kehitteillä olevan version 3 on tarkoitus sisältää kaksi uutta koodaustapaa ja joukon täydentäviä parannustiedon koodipisteitä.
- 10 H.263:n mukaan kuvat koodataan luminanssikomponenttina ja kahden värin eroa (krominanssi) osoittavana komponenttina (Y , C_B ja C_R). Krominanssikomponenteista otetaan näytteitä puolella resoluutiolla luminanssikomponenttiin verrattuna molempien koordinaattiakseleiden osalta.
- 15 Jokainen koodattu kuva sekä vastaava koodattu bittivirta järjestetään hierarkkiseen rakenteeseen, jonka neljä kerrosta ovat ylhäältä alaspäin kuvakerros (picture layer), kuvasegmenttikerros (picture segment layer), makrolohkokerros (macroblock (MB) layer) ja lohkerros (block layer). Kuvasegmenttikerros voi olla joko lohkokryhmakerros (group of blocks layer) tai viipalekerros (slice layer).
- 20 Kuvakerrostieto sisältää parametrejä, jotka vaikuttavat koko kuva-alueeseen ja kuvatiedon dekodaukseen. Koodattu tieto järjestetään niin sanotuksi kuvaotsikoksi (picture header).
- 25 Oletuksena on se, että jokainen kuva on jaettu lohkokryhmiin. Lohkokryhmä (GOB) käsittää tavallisesti 16 peräkkäistä pikseliriviä. Jokaisen GOB:n tiedot koostuvat valinnaisesta GOB-otsikosta, jota seuraavat MB:n tiedot.
- 30 Jos käytetään valinnaista viipalerakenteista muotoa, jokainen kuva on jaettu viipaleisiin eikä GOB:ihin. Jokaisen viipaleen tiedot koostuvat valinnaisesta viipaleotsikosta, jota seuraavat MB:n tiedot.

Viipale määrittää alueen koodatun kuvan sisällä. Yleensä alue on joukko MB:itä normaalissa pyyhkäisyjärjestyksessä. Eri viipalerajojen välillä ei ole ennusteriippuvuuksia saman koodatun kuvan sisällä. Kuitenkin ajallinen ennustaminen voi ylittää viipaleiden rajat, ellei käytössä ole H.263 liitteen R (riippumaton segmentin dekodaus) mukainen ratkaisu. Viipaleet voidaan dekodata muusta kuvatiedosta erillään (lukuun ottamatta kuvaotsikkoa). Näin ollen viipaleet parantavat virheensietoa pakettihäviöllisissä verkoissa.

Kuva, GOB ja viipaleotsikko alkavat synkronointikoodilla. Mikään muu koodisana tai voimassaoleva koodisanayhdistelmä ei voi muodostaa samaa bittikuviota kuin synkronointikoodit. Näin ollen synkronointikoodeja voidaan käyttää bittivirtavirheen havaitsemiseen ja uudelleensynkronointiin bittivirheiden jälkeen. Mitä enemmän synkronointikoodeja lisätään bittivirtaan, sitä virhekestävämmäksi koodaus tulee.

Jokainen GOB tai viipale on jaettu MB:ihin. MB käsittää luminanssitiedon 16 x 16 pikseliä ja spatiaalisesti vastaavat krominanssitiedon 8 x 8 pikseliä. Toisin sanoen MB koostuu neljästä 8 x 8 luminanssilohkosta ja kahdesta spatiaalisesti vastaavasta 8 x 8 krominanssilohkosta.

Lohko käsittää luminanssi- tai krominanssitiedon 8 x 8 pikseliä. Lohkokerrostieto koostuu tasaisesti kvantisoiduista (uniformly quantised) diskreetin kosinimuunnoksen kertoimista, joita selataan siksak järjestyksessä, käsitellään jononpituuskooderilla (run-length encoder) ja koodataan vaihtuvamittaisilla koodeilla ITU-T:n suosituksessa H.263 esitetyllä tavalla.

Monissa videonkompressoitimenetelmissä käytetään ajallisesti kaksisuuntaisesti ennustettavia kehyksiä, joita kutsutaan yleisesti B-kuviksi tai B-kehyyksiksi. B-kehyykset sijoitetaan ankkurikehyksparien väliin ja ne ennustetaan joko yhdestä ankkurikehyksestä tai molemmista kuviossa 2 esitetyn tavan mukaisesti. B-kehyyksiä ei käytetä ankkurikehyyksinä, toisin sanoen niistä ei koskaan ennusteta muita kehyksiä. B-kehyyksiä käytetään havaitun kuvanlaadun vahvistamiseen kuvannäyttönopeutta lisäämällä. Ne voidaan jättää pois ilman, että se vaikuttaa seuraavien kehysten dekodointiin, jolloin videokuvausarjaa

voidaan dekodata eri nopeuksilla lähetysverkon kaistanleveyttä koskevien rajoitusten tai eri dekooderien ominaisuuksien mukaan. B-kehykset tarjoavat siis myös ajallista skaalattavuutta. Vaikka B-kehykset voivatkin parantaa kompressointitehoa P-kehyksiin verrattuna, niiden käyttö edellyttää suurempaa laskentatehoa ja muistia ja aiheuttaa lisäviivästyksiä.

Spatiaalinen skaalattavuus liittyy läheisesti erääseen toiseen skaalattavuuden muotoon, nimittäin signaali-kohinasuhteen (SNR) skaalattavuuteen. Kuviossa 4 esitetään esimerkki SNR-skaalattavista kuvista. SNR-skaalattavuuteen kuuluu moninopeuksisten bittivirtojen luominen. Se sallii alkuperäisen ja uudelleenmuodostetun kuvan välisten koodausvirheiden tai -erojen saamisen takaisin. Tämä tapahtuu käyttämällä hienompaa kvantisoijaa koodaamaan erokuvaa (difference picture) parannuskerroksessa. Tämä lisätieto lisää SNR:ää uudelleenmuodostetussa kuvassa.

Spatiaalisen skaalattavuuden avulla voidaan luoda moniresoluutioisia bittivirtoja vaihtelevien näyttövaatimusten/-rajoitteiden täyttämiseksi. Spatiaalisesti skaalattava rakenne on esitetty kuviossa 5. Se on samankaltainen kuin SNR-skaalattavuudessa käytetty rakenne. Spatiaalisessa skaalattavuudessa käytetään spatiaalista parannuskerrosta saamaan takaisin koodaushäviö parannuskerroksen viitteenä käyttämän uudelleenmuodostetun kerroksen tiheimmin näytteistetyyn version, toisin sanoen viitekerroksen, ja alkuperäisen kuvan suurempiresoluutioisen version välillä. Jos viitekerroksella on esimerkiksi Quarter Common Intermediate Format (QCIF) -resoluutio, eli 176x144 pikseliä, ja parannuskerroksella on Common Intermediate Format (CIF) -resoluutio, joka on 352x288 pikseliä, viitekerroksen kuva on skaalattava sen mukaisesti siten, että parannuskerroksen kuva voidaan ennustaa siitä asianmukaisesti. H.263:n mukaan resoluutiota kasvatetaan kaksinkertaisesti yhden parannuskerroksen osalta joko ainoastaan pystysuunnassa, ainoastaan vaakasuunnassa, tai sekä pysty- että vaakasuunnassa. Parannuskerroksia voi olla useampia, ja jokainen niistä parantaa kuvan resoluutiota edelliseen kerrokseen verrattuna. Viitekerroksen kuvan tiheimmin näytteistämiseen käytetyt interpolaatiosuodattimet on tarkasti määritelty H.263:ssa. Viitekerrokselta parannuskerrokselle tapahtuvaa ylösnäytteistämistä prosessia lukuun ottamatta



spatiaalisesti skaalatun videokuvan käsittely ja syntaksi ovat identtiset SNR-skaalatun videokuvan kanssa. Spatiaalinen skaalattavuus tarjoaa paremman spatiaalisen resoluution SNR-skaalattavuuteen verrattuna.

- 5 Joko SNR-skaalattavuudessa tai spatiaalisessa skaalattavuudessa parannuskerroskuviin viitataan EI- tai EP-kuvina. Jos parannuskerroskuva ennustetaan ylöspäin viitekerroksen INTRA-kuvasta, silloin parannuskerroskuvaan viitataan Enhancement-I (EI)-kuvana. Joissain tapauksissa kun viitekerroskuvat on ennustettu heikosti, parannuskerroksessa
- 10 voi tapahtua kuvan staattisten osien ylikoodausta, mikä vaatii ylisuurta bittinopeutta. Tämän ongelman välttämiseksi ennustus eteenpäin on sallittu parannuskerroksessa. Edelliseltä parannuskerrokselta eteenpäin ennustettuun kuvaan tai ennustetusta kuvasta viitekerrokselta ylöspäin ennustettuun kuvaan viitataan nimellä Enhancement-P (EP) -kuva. Sekä ylöspäin että eteenpäin
- 15 ennustettavien kuvien keskiarvon laskeminen voi tarjota kaksisuuntaisen ennustusvaihtoehdon EP-kuville. EI- ja EP-kuvien ylöspäin ennustaminen viitekerroskuvasta viittaa siihen, että liikevektoreita ei tarvita. EP-kuvien eteenpäin ennustamisessa liikevektoreita sen sijaan tarvitaan.
- 20 H.263:n skaalattavuusmuodossa (liite O) määritetään syntaksi tukemaan ajallisen, SNR:n ja spatiaalisen skaalattavuuden ominaisuuksia.
- Yksi perinteisen SNR-skaalattavuuskoodauksen ongelma on virheenkulkeutuminen (drifting). Virheenkulkeutuminen viittaa lähetysvirheen vaikutukseen. Virheen aiheuttama visuaalinen artefakti kulkeutuu ajallisesti virheellisestä kuvasta. Liikekompensoinnin ansiosta visuaalisen artefaktin alue voi kasvaa kuvasta toiseen. Skaalattavan koodauksen tapauksessa visuaalinen artefakti kulkeutuu myös alemmilta parannuskerroksilta ylempiin kerroksiin. Virheenkulkeutusvaikutusta voidaan selventää viitaten kuvioon 7, jossa
- 25 esitetään perinteiset skaalattavassa koodauksessa käytetyt ennustussuhteet. Kun virhe tai pakettihäviö on tapahtunut parannuskerroksessa, se leviää kuvajoukon (GOP) päähän, koska kuvat ennustetaan sarjana toistensa perusteella. Koska parannuskerrokset perustuvat pohjakerrokseen, virhe pohjakerroksessa aiheuttaa lisäksi virheitä parannuskerroksissa, jotka myös
- 30

perustuvat toisiinsa. Näin ollen tästä seuraa vakava virheenkulkeutumisongelma seuraavien ennustuskehysten ylemmissä kerroksissa. Vaikka myöhemmin voi olla käytettävissä riittävästi kaistanleveyttä tiedon lähettämiseen virheen korjaamiseksi, dekooderi ei pysty poistamaan virhettä ennen kuin ennustusketjun

5 pysäyttää uuden GOP:n aloittava toinen INTRA-kuva.

Tämän ongelman ratkaisemiseksi on kehitetty skaalattavuuden muoto, joka tunnetaan nimellä Fine Granularity Scalability (FGS). FGS:ssä heikompilaatuinen pohjakerros koodataan käyttämällä hybridiä ennustussilmukkaa, ja (lisä-)

10 parannuskerros kuljettaa progressiivisesti koodattua jäännöstä rekonstruoidun pohjakerroksen ja alkuperäisen kehyksen välillä. FGS:ää on ehdotettu esimerkiksi MPEG-4 visuaalisessa standardoinnissa.

Esimerkki ennustussuhteista FGS-koodauksessa on esitetty kuviossa 6. FGS-skaalattavassa videokuvankoodausmenetelmässä pohjakerroksen videokuvaa

15 lähetetään tarkoin hallitulla kanavalla virheiden tai pakettihäviön minimoimiseksi niin, että pohjakerros on koodattu sopimaan pienimpään mahdolliseen kanavan kaistanleveyteen. Tämä minimi on pienin kaistanleveys, joka voi esiintyä käytön aikana. Kaikki ennustuskehysten parannuskerrokset koodataan viitekehysten

20 pohjakerroksen perusteella. Näin yhden kehyksen parannuskerroksen virheet eivät aiheuta virheenkulkeutumisongelmaa myöhempien ennustettavien kehysten parannuskerroksissa, ja koodausmenetelmä voi mukautua kanavaolosuhteisiin. Koska ennustaminen perustuu kuitenkin aina heikkotasoiseen pohjakerrokseen, FGS-koodauksen koodaustehokkuus ei ole

25 yhtä hyvä kuin perinteisten skaalattavuus menetelmien, esimerkiksi H.263 liitteessä O esitetyn SNR-skaalattavuusmenetelmän, vaan on joskus jopa paljon heikompi.

Sekä FGS-koodauksen että perinteisen kerrosskaalattavuuskoodauksen

30 (layered scalability coding) etujen yhdistämiseksi on ehdotettu kuviossa 8 esitettyä hybridiä koodausmenetelmää, joka on nimeltään Progressive FGS (PFGS). On syytä huomioida kaksi asiaa. Ensinnäkin PFGS:ssä käytetään mahdollisimman monta ennustusta samasta kerroksesta koodaustehokkuuden säilyttämiseksi. Toiseksi, ennustuspolku käyttää aina ennustusta viitekehysten

alemmasta kerroksesta mahdollistaakseen virheestä toipumisen ja kanavaan mukautumisen. Ensimmäisen kohdan tarkoitus on varmistaa, että tietyssä videokuvakerroksessa liikkeenennustus on mahdollisimman tarkkaa, jolloin säilytetään koodaustehokkuus. Toisella kohdalla varmistetaan, että virheenkulkeutumisiongelmaa ei synny kanavaruuhan, pakettihäviön tai pakettivirheen tapahtuessa. Tätä koodausrakennetta käytettäessä kadonneita/virheellisiä paketteja ei tarvitse lähettää uudelleen, koska parannuskerrokset voidaan vaiheittain ja automaattisesti muodostaa uudelleen muutaman kehyksen aikana.

10

Kuviossa 8 kehys 2 ennustetaan kehyksen 1 parillisista kerroksista (toisin sanoen pohjakerroksesta ja toisesta kerroksesta). Kehys 3 ennustetaan kehyksen 3 parittomista kerroksista (toisin sanoen ensimmäisestä ja kolmannesta kerroksesta). Kehys 4 puolestaan ennustetaan kehyksen 3 parillisista kerroksista. Pariton/parillinen ennustusmalli jatkuu. Termiä ryhmäsyvyys (group depth) käytetään kuvaamaan yhteiseen viitekerrokseen viittaavien kerrosten lukumäärää. Kuvio 8 havainnollistaa tapausta, jossa ryhmäsyvyys on 2. Ryhmäsyvyttä voidaan muuttaa. Jos syvyys on 1, tilanne on olennaisesti vastaava kuin perinteisessä skaalattavuusmenetelmässä. Jos syvyys vastaa kerrosten kokonaislukumäärää, menetelmästä tulee vastaava FGS:n kanssa. Näin ollen kuviossa 8 esitetty progressiivinen FGS-koodausmenetelmä tarjoaa kompromissin, joka tarjoaa molempien edellisten menetelmien edut, kuten koodaustehokkuuden ja virheestä toipumisen.

20

PFGS tarjoaa etuja silloin, kun sitä sovelletaan videokuvan lähettämiseen internetin tai langattomien kanavien välityksellä. Koodattu bittivirta voi mukautua saatavilla olevaan kanavan kaistanleveyteen ilman huomattavaa virheenkulkeutumista. Kuvio 9 esittää esimerkin PFGS-skaalattavuuden kyvystä mukautua kaistanleveyteen tilanteessa, jossa videokuvasarjaa edustavat kehykset, joissa on pohjakerros ja 3 parannuskerrosta. Paksu piste-katkoviiva jäljittää lähetetyt videokuvakerrokset. Kehyksen 2 kohdalla kaistanleveys on merkittävästi pienentynyt. Lähetin (palvelin) reagoi tähän pudottamalla ylempiä parannuskerroksia (kerrokset 2 ja 3) edustavat bitit. Kehyksen 2 jälkeen kaistanleveys kasvaa jonkin verran ja lähetin pystyy lähettämään kahta

30

parannuskerrosta edustavat lisäbitit. Kehyksen 4 lähettämiseen mennessä saatavilla oleva kaistanleveys on kasvanut entisestään ja siinä on taas riittävästi kapasiteettia pohjakerroksen ja kaikkien parannuskerrosten lähettämiseksi. Nämä toiminnot eivät edellytä videokuvan bittivirran uudelleenkodeusta tai uudelleenlähettämistä. Videokuvasarjan jokaisen kehyksen kaikki kerrokset on koodattu tehokkaasti ja upotettu yhteen bittivirtaan.

Edellä kuvatut tekniikan tason mukaiset skaalattavat koodausmenetelmät perustuvat koodatun bittivirran yksittäiseen tulkintaan. Toisin sanoen, dekoderi tulkitsee koodattua bittivirta vain kerran ja luo uudelleen muodostetut kuvat. Uudelleen muodostettuja kuvia käytetään viitekuvina liikekompensointia varten.

Yleisesti edellä kuvatuissa ajallisten viittausten käyttöön tarkoitetuissa menetelmissä ennustusviitteet ovat ajallisesti ja spatiaalisesti mahdollisimman lähellä koodattavaa kuvaa tai aluetta. Kuitenkin ennustava koodaus on altis lähetyksvirheille, koska virhe vaikuttaa kaikkiin virheellistä kuvaa seuraaviin kuviin ennustetussa kuvaketjussa. Sen vuoksi yksi tavallinen keino tehdä videokuvan lähettämisyjärjestelmä paremmin virheitä sietäväksi on lyhentää ennustusketjuja.

Spatiaalinen, SNR- ja FGS-skaalattavuusmenetelmät tarjoavat kaikki keinon pienentää kriittisiä ennustuspolkuja bittien lukumäärän suhteen. Kriittinen ennustuspolku on se osa bittivirtaa, joka täytyy dekodata hyväksyttävän esityksen saamiseksi videokuvasarjan sisällöstä. Bittinopeuskaalattavassa koodauksessa kriittinen ennustuspolku viittaa GOP:n pohjakerrokseen. On edullista turvata huolellisesti vain kriittinen ennustuspolku koko yksikerroksisen bittivirran sijasta.

Ennustuspolkujen lyhentämiseksi voidaan käyttää B-kehyksiä ajallisesti vastaavien INTER-kehyksien sijasta. Jos aika peräkkäisten ankkurikehysten välillä on kuitenkin suhteellisen pitkä, B-kehysten käyttö pienentää kompressoititehokkuutta. Tässä tilanteessa B-kehykset ennustetaan ajallisesti toisistaan kauempana olevista ankkurikehyksistä, jolloin niistä ennustetut B-kehykset ja viitekehykset ovat vähemmän samankaltaisia. Tästä seuraa heikompi ennustettu B-kehys ja näin ollen bittejä tarvitaan enemmän

koodaamaan siihen liittyvä ennustusvirhekehys. Lisäksi ankkurikehysten välisen ajallisen etäisyyden kasvaessa peräkkäiset ankkurikehykset ovat vähemmän samankaltaisia. Tästä seuraa jälleen heikompi ennustettu ankkurikehys, ja bittejä tarvitaan enemmän koodaamaan siihen liittyvä ennustusvirhekuva.

5

Ajallinen ennustus tapahtuu tavallisesti kuvion 10 mukaisesti.

Jos INTER-kehysten ennustusviittaus voidaan valita (kuten esimerkiksi H.263:n viitekuvanvalintamoodissa (Reference Picture Selection)), ennustuspolkuja voidaan lyhentää ennustamalla sen hetkinen kehys muusta kuin sitä luonnollisessa numerojärjestyksessä välittömästi edeltävästä kehyksestä. Tämä kuvataan kuviossa 11.

On ehdotettu, että videoredundanssikoodauksen (VRC) avulla voidaan videokuvan laatua hienovaraisesti alentaa pakettikytkentäisissä verkoissa tapahtuvien pakettihäviöiden seurauksena. VRC:n periaate on jakaa videokuva-sarja kahdeksi tai useammaksi säikeeksi (thread) siten, että kaikki kuvat määrätään yhdelle säikeistä kiertovuorotteluna. Jokainen säie koodataan itsenäisesti. Säännöllisin väliajoin kaikki säikeet konvergoituvat niin sanottuun tahdistuskehukseen (Sync frame). Tästä tahdistuskehuksesta aloitetaan uusi säiesarja. Yhden säikeen kehysnopeus on näin ollen pienempi kuin kokonaiskehysnopeus: puolet kahden säikeen kohdalla, kolmannes kolmen säikeen kohdalla ja niin edelleen. Tämä johtaa huomattavaan koodauskustannuksen samassa sarjassa olevien peräkkäisten kuvien välisten yleisesti suurempien erojen takia sekä pitempien liikevektorien takia, joita tyypillisesti tarvitaan edustamaan säikeeseen kuuluvien kuvien välillä tapahtuvia liikkeeseen liittyviä muutoksia. Kuvio 12 esittää kahta säiettä ja kolmea kehystä säiettä kohti käyttävää VRC:tä.

Jos yksi säikeistä vahingoittuu esimerkiksi pakettihäviön vuoksi, jäljelle jääneet säikeet säilyvät todennäköisesti vahingoittumattomina ja niitä voidaan käyttää seuraavan tahdistuskehysten ennustamiseen. On mahdollista jatkaa vahingoittuneen säikeen dekodauksella, joka johtaa lievään kuvanlaadun heikkenemiseen, tai lopettaa sen dekodaus, joka johtaa kehysnopeuden

pienenemiseen. Jos säikeet ovat kuitenkin kohtuullisen lyhyitä, molemmat laadun heikkenemismuodot kestävät vain hyvin vähän aikaa, toisin sanoen siihen asti kunnes päästään seuraavaan tahdistuskehyykseen. Kuvio 13 esittää VRC:n toimintaa silloin, kun yksi kahdesta säikeestä on vahingoittunut.

5

Tahdistuskehyykset ennustetaan aina vahingoittumattomista säikeistä. Tämä tarkoittaa sitä, että lähetettyjen INTRA-kuvien lukumäärä voidaan pitää pienenä, koska täydelliseen uudelleensynkronointiin ei yleisesti ottaen ole tarvetta. Tahdistuskehyysten oikean muodostamisen estää ainoastaan se, että kaikki
10 säikeet kahden tahdistuskehyyksen välillä ovat vahingoittuneet. Tässä tilanteessa häiritsevät artefaktit jatkuvat, kunnes seuraava INTRA-kuva dekodataan oikein, kuten olisi käynyt ilman VRC:n käyttöä.

Tällä hetkellä VRC:tä voidaan käyttää ITU-T H.263 –
15 videokuvankoodausstandardin (versio 2) mukaisesti, jos valinnainen viitekuvan valintatila (liite N) on päällä. Merkittäviä esteitä VRC:n sisällyttämiselle muihin videokuvan kompressoitimenetelmiin ei kuitenkaan ole.

P-kehyyden ennustamista taaksepäin on myös ehdotettu ennustusketjujen
20 lyhentämismenetelmäksi kuvion 14 esittämällä tavalla. Siinä esitetään videokuvasarjan muutama peräkkäinen kehys. Kuviossa näkyy INTRA-kehys (I1), joka on lisätty koodatun videokuvasarjaan INTRA-kehyspyynnön tai tasavälisen INTRA-kehyyksen virkistä-toiminnon (periodic refresh) seurauksena. INTRA-kehyspyyntö voidaan tehdä kohdassa, jossa otos vaihtuu. Muutaman
25 INTER-koodatun kehyyksen (P2, P3, P4 ja P5) jälkeen tehdään toinen INTRA-kehyspyyntö (tai säännöllisin väliajoin tapahtuva INTRA-kehyyksen virkistä-toiminto). Sen sijaan, että INTRA-kehys lisättäisiin välittömästi INTRA-kehyspyynnön (tai säännöllisin väliajoin tapahtuva INTRA-kehyyksen virkistä-toiminnon) jälkeen, se lisätään muutaman ajallisesti ennustetun kehyyksen
30 jälkeen. INTRA-kehyspyynnön ja INTRA-kehyyksen I1 väliset kehykset ennustetaan sarjassa taaksepäin INTER-muodossa yksi toisensa jälkeen niin, että ennustusketju saa alkunsa I1:stä. Taaksepäin ennustettuja INTER-kehyyksiä ei voida dekodata ennen kuin I1 on dekodattu. Näin ollen tarvitaan

alkupuskurointiivie, joka on suurempi kuin otoksen vaihtumisen ja sitä seuraavan INTRA-kehysten välinen aika, jotta voidaan estää tauko toistossa.

Tämän lähestymistavan etu on selvä, kun ajatellaan kuinka monta kehystä on
 5 lähetettävä onnistuneesti, jotta kehysten P5 dekodaus on mahdollista. Jos
 käytetään perinteistä, esimerkiksi kuviossa 15 esitetyn kaltaista kehysjärjestelyä,
 P5:n onnistunut dekodaus edellyttää, että I1, P2, P3, P4 ja P5 lähetetään ja
 dekodataan oikein. Kuviossa 14 esitetystä menetelmästä P5:n onnistunut
 10 dekodaus edellyttää, että I1, P4 ja P5 lähetetään ja dekodataan oikein. Toisin
 sanoen tämä menetelmä tarjoaa suuremman varmuuden siitä, että P5
 dekodataan oikein verrattuna menetelmään, jossa käytetään perinteistä
 kehysjärjestelyä ja kehysten ennustamista.

Vaikka viitekuvan valinnalla (reference Picture Selection) voidaankin estää
 15 videokuvasarjan virheen ajallista etenemistä, se pienentää
 kompressoititehokkuutta.

Perinteinen SNR- ja spatiaalinen skaalattavuuskoodaus sekä FGS-koodaus
 vähentävät kompressoititehokkuutta. Lisäksi ne edellyttävät, että lähetin
 20 päättää kuinka videokuvadata kerrostetaan koodauksen aikana.

Kuvio 16 esittää videoviestintäjärjestelmää 10, joka toimii ITU-T H.26L –
 standardin mukaisesti perustuen testimalliin (TML) TML-3 muunneltuna
 nykyisten TML-4:ää koskevien ohjeiden mukaisesti. Järjestelmässä 10 on
 25 lähetinpuoli 12 ja vastaanotinpuoli 14. On ymmärrettävä, että järjestelmä on
 varustettu kaksisuuntaista lähetystä ja vastaanottamista varten, joten lähetin- ja
 vastaanotin- puolet 12 ja 14 voivat suorittaa sekä lähetys- että
 vastaanottotoimintoja ja ovat keskenään vaihdettavissa. Järjestelmä 10 käsittää
 30 videonkoodauskerroksen (video coding layer, VCL) ja verkkotietoisien (network
 aware) verkonsovituskerroksen (NAL) (toisin sanoen NAL voi sovittaa tiedon
 järjestelyn verkkoon sopivaksi). VCL sisältää sekä aaltomuotokooderin että
 entropiakooderin ja dekodaustoiminnallisuuden. NAL paketoit koodattua
 videokuvadataa palveludatayksiköiksi (service data units, paketeiksi), jotka
 toimitetaan kuljetuskooderille ennen lähettämistä kanavan yli. Lisäksi NAL

purkaa kanavan yli tapahtuneen lähetyksen jälkeen koodatun videokuvadatan kuljetuskooderilta vastaanotetuista palveludatayksiköistä. NAL pystyy osittamaan videokuvan bittivirran koodatuiksi lohkokuvioiksi ja ennustusvirhekertoimiksi erillään muusta tärkeämmästä datasta, kuten kuvatyyppi ja
5 liikekompensointitieto.

VCL:n päätehtävä on koodata videodataa tehokkaalla tavalla. Kuitenkin, kuten edellä on selostettu, virheet vaikuttavat haitallisesti tehokkaasti koodattuun dataan, joten siihen on sisällytetty jonkin verran tietoisuutta mahdollisista
10 virheistä. VCL pystyy keskeyttämään ennustavan koodausketjun ja ryhtymään toimiin virheiden tapahtumisen ja leviämisen kompensoimiseksi. Tämä voi tapahtua useilla tavoilla:

keskeytetään ajallinen ennustusketju lisäämällä INTRA-kehukset ja INTRA-MB:t, keskeytetään virheen leviäminen ottamalla käyttöön viipalekäsitel, toisin sanoen
15 rajoittamalla liikevektorin ennustus viipaleiden sisälle, lisäämällä vaihtuvamittainen koodi, joka voidaan dekodata itsenäisesti esimerkiksi ilman adaptiivista aritmeettista koodausta kehuksesta toiseen, ja käytetään nopeaa nopeudenallokointia (rate allocation) vaihtuviin bittinopeuskanaviin mukautumiseksi.

20

Lisäksi VCL tunnistaa prioriteettiluokat tukeakseen palvelulaatumekanismeja (quality of service, QoS) verkoissa.

Koodausmenetelmät sisältävät tietoa, joka kuvaa koodattuja videokuvakehyksiä tai kuvia. Tieto määritetään syntaksielementteinä. Syntaksielementti on
25 koodisana tai joukko koodisanoja, joiden toiminnallisuus koodausmenetelmässä on samankaltainen. Syntaksielementit jaetaan prioriteettiluokkiin (priority classes). Syntaksielementin prioriteettiluokka määritellään sen koodaus- ja dekodausriippuvuuksien mukaan suhteessa muihin luokkiin.
30 Dekoodausriippuvuudet ovat seurausta ajallisesta ja spatiaalisesta ennustamisesta sekä vaihtuvamittaisesta koodista. Yleiset säännöt prioriteettiluokkien määrittämiseksi ovat seuraavat:

1. Jos syntaksielementti A voidaan dekodata oikein tuntematta syntaksielementtiä B, ja syntaksielementti B:tä ei voida dekodata oikein

tuntematta syntaksielementtiä A, silloin syntaksielementin A prioriteetti on korkeampi kuin syntaksielementin B.

2. Jos syntaksielementit A ja B voidaan dekodata itsenäisesti, syntaksielementin prioriteettiluokan määrää sen vaikutus kuvan laatuun.

5

Virheelliset tai puuttuvat syntaksielementit vaikuttavat ainoastaan niiden syntaksielementtien dekodointiin, jotka ovat sen hetkessä haarassa muualla kuin riippuvuuspuun juuressa. Sen vuoksi puun juurta lähempänä olevien syntaksielementtien vaikutus dekodatun kuvan laatuun on suurempi kuin alemmissä prioriteettiluokissa olevien vaikutus.

10

Prioriteettiluokat määritetään kehys kehykseltä. Viipalepohjaista kuvankoodausmuotoa käytettäessä syntaksielementtien sijoittamista prioriteettiluokkiin mukautetaan jonkin verran.

15

Eri syntaksielementtien väliset riippuvuudet nykyisessä H.26L-testimallissa on esitetty kuviossa 17. Nykyisessä testimallissa on seuraavat 10 prioriteettiluokkaa, joissa luokka 1 on ylin ja luokka 10 alin.

Luokka 1: PSYNC, PTYPE: Sisältää PSYNC-, PTYPE-syntaksielementit

- 20 Luokka 2: MB_TYPE, REF_FRAME: Sisältää kaikki kehyksen MB-tyyppiset ja viitekehysyntaksielementit. Tämä luokka ei sisällä elementtejä INTRA-kuvissa/kehyksissä.

Luokka 3: IPM: Sisältää INTRA-prediction-Mode-syntaksielementit

- 25 Luokka 4: MVD, MACC: Sisältää liikevektori- ja liiketarkkuussyntaksielementit (TML-2). Tämä luokka ei sisällä elementtejä INTRA-kuvissa/kehyksissä.

Luokka 5: CBP-Intra: Sisältää kaikki yhdessä kehyksessä INTRA-MB:ihin sijoitetut CBP-syntaksielementit.

- 30 Luokka 6: LUM_DC-Intra, CHR_DC-Intra: Sisältää kaikki DC-luminanssikertoimet ja kaikki DC-krominanssikertoimet kaikkien INTRA-MB:iden lohkoille.

Luokka 7: LUM_AC-Intra, CHR_AC-Intra: Sisältää kaikki AC-luminanssikertoimet ja kaikki AC-krominanssikertoimet kaikkien INTRA-MB:iden lohkoille.

Luokka 8: CBP-Inter, sisältää kaikki kehyksessä INTER-MB:ihin sijoitetut CBP-syntaksielementit .

Luokka 9: LUM_DC-Inter, CHR_DC-Inter: Sisältää jokaisen lohkon ensimmäisen luminanssikertoimen ja INTER-MB:iden kaikkien lohkojen DC-
5 krominanssikertoimet.

Luokka 10: LUM_AC-Inter, CHR_AC-Inter: Sisältää INTER-MB:iden kaikkien lohkojen jäljellä olevat luminanssikertoimet ja krominanssikertoimet.

NAL:n päätehtävä on lähettää aiemmin kuvattujen prioriteettiluokkien sisältämää
10 dataa optimaalisella tavalla alla oleviin verkkoihin sovitettuna. Sen vuoksi jokaiselle alapuoliselle verkolle tai verkkotyypille on määritelty yksilöllinen datan kapselointimenetelmä. NAL suorittaa seuraavat tehtävät:

1. Se yhdistää aiemmin kuvattujen syntaksielementtiluokkien sisältämän datan videopaketteihin.
- 15 2. Se siirtää näin muodostuneet datapaketit alapuoliseen verkkoon sovitetulla tavalla.

Se voi myös tarjota virheensuojausmekanismin.

Kompressoitun videokuvan koodaamiseen käytettävien syntaksielementtien
20 priorisoiminen eri prioriteettiluokkiin yksinkertaistaa sovittamista alapuoliseen verkkoon. Priorisointimekanismeja tukevat verkot hyötyvät erityisesti syntaksielementtien priorisoinnista. Esimerkkejä tästä ovat:

- priorisointimenetelmien käyttö IP:ssä (kuten RVSP),
- QoS-mekanismien käyttö UMTS:ssä,
- 25 H.223:n liitteen C tai D käyttö, ja
- alapuolisten verkkojen tarjoaman epätasaisen virheensuojauksen käyttö.

Eri data-/viestintäverkkojen ominaisuudet ovat merkittävästi erilaiset. Esimerkiksi eri pakettikytkentäiset verkot käyttävät protokollia, jossa on määritelty
30 vähimmäis- ja enimmäispakettipituudet. Jotkut protokollat varmistavat, että datapaketit toimitetaan oikeassa järjestyksessä. Sen vuoksi useampaa kuin yhden luokan datan yhdistämistä yksittäiseksi datapaketiksi tai tiettyä prioriteettiluokkaa edustavan datan jakamista useiden datapakettien kesken sovelletaan tarpeen mukaan.

Yleisesti ottaen VCL tarkistaa verkkoa ja lähetysprotokollaa hyödyntäen, että tietty luokka ja kaikki korkeammat prioriteettiluokat tietyn kehyksen sisällä tunnistetaan ja on vastaanotettu oikein, toisin sanoen ilman bittivirheitä ja oikean mittaisina.

Koodattua videobittivirtaa kapseloidaan useilla tavoilla alapuolisen verkon ja käytössä olevan sovelluksen mukaan. Seuraavaksi esitellään erilaisia kapselointimenetelmiä.

10

H.324 (piirikytkentäinen kuvapuhelin - circuit switched videophone)

H.324:n kuljetuskooderi H.223:n palveludataysikkö on enimmäiskooltaan 254 tavua. Tämä ei tavallisesti riitä kuljettamaan koko kuvaa, minkä vuoksi VCL todennäköisesti jakaa kuvan useisiin osiin siten, että jokainen osa sopii yhteen palveludatayksikköön. Tällä hetkellä koodisanat on ryhmitelty osiin niiden tyyppin perusteella (toisin sanoen kuvan samantyyppiset koodisanat samaan osaan). Osien koodisana- (ja tavu-) järjestys järjestetään pienenevässä tärkeysjärjestyksessä. Jos bittivirhe vaikuttaa kuvadataa kuljettavaan H.223 palveludatayksikköön, dekooderi voi menettää dekodauksen synkronoinnin (parametrien vaihtuvamittaisen koodauksen vuoksi), ja loppua palveludatayksiköissä olevasta datasta ei voida dekodata. Dekooderi pystyy kuitenkin todennäköisesti luomaan heikomman esityksen kuvan sisällöstä, koska tärkein data esiintyy palveludatayksikön alussa.

25 IP kuvapuhelin (IP videophone)

IP-paketin enimmäiskoko on noin 1500 tavua (historiallisista syistä). On edullista käyttää mahdollisimman suuria IP-paketteja kahdesta syystä:

1. IP-verkkoelementit, kuten reitittimet, voivat tukkeutua liiallisen IP-liikenteen vuoksi ja tämä aiheuttaa sisäisten puskureiden ylivuotoja. Puskurit ovat tavallisesti pakettiorientoituneita (packet orientated), toisin sanoen ne voivat sisältää tietyn määrän paketteja. Niinpä verkkoon muodostuvan ruuhkan välttämiseksi on suotavaa käyttää harvoin muodostettuja suuria paketteja kuin usein muodostettuja pieniä paketteja.

30

2. Kukin IP-paketti sisältää otsikkotietoa. Tavallinen reaaliaikaisessa kuvaviestinnässä käytetty protokollayhdistelmä, eli RTP/UDP/IP, sisältää 40-tavuisen otsikko-osan pakettia kohden. IP-verkkoa kytkeydyttäessä käytetään usein piirikytkentäistä pienen kaistanleveyden valintayhteys. Paketoinnin aiheuttama lisäinformaatio muodostuu merkittäväksi alhainen bittinopeuden linkeillä, jos pieniä paketteja on useita.

- INTER-koodattu videokuva voi käsittää riittävän vähän bittejä mahtuakseen yhteen IP-pakettiin (kuvan koosta ja monimutkaisuudesta riippuen). On olemassa useita tapoja tarjota epätasaista virheensuojausta IP-verkoissa. Niitä ovat esimerkiksi paketin monistus, virheen korjaaminen eteenpäin (forward error correction, FEC) –paketit, eriytetyt palvelut (differentiated services, eli paketin priorisointi verkossa), ja integroidut palvelut (intergrated services, RSVP-protokolla). Tavallisesti nämä menetelmät edellyttävät, että tärkeydeltään samanarvoista dataa on kapseloitu yhteen pakettiin.

IP videokuvan suoratoisto (IP video streaming)

- Koska videokuvan suoratoisto ei ole keskustelusovellus (conversational application), siihen ei liity tiukkoja vaatimuksia päästä-päähän viiveen osalta. Näin ollen paketointimenetelmä voi hyödyntää tietoa useista kuvista. Dataa voidaan luokitella esimerkiksi samantapaisella tavalla kuin IP-kuvapuhelimen kohdalla, tosin tärkeä data useasta kuvasta paketoidaan samaan pakettiin.

- Vaihtoehtoisesti jokainen kuva tai viipale voidaan paketoita omaan pakettiin. Datan osittamista sovelletaan siten, että tärkein data esiintyy paketin alussa. Virheen korjaaminen eteenpäin (FEC) –paketit lasketaan sarjasta jo lähetettyjä paketteja. FEC-algoritmi valitaan niin, että se suojaa vain tietyn määrän paketin alussa esiintyvistä tavuista. Jos tavallinen datapaketti katoaa, kadonneen datapaketin alku voidaan korjata vastaanottavassa päässä käyttämällä FEC-pakettia (kuten on ehdotettu asiakirjassa A. H. Li, J. D. Villasenor, "A Generic Uneven Level Protection (ULP) Proposal for Annex I of H.323", ITU-T, SG16, Question 15, document Q15-J-61, 16th May, 2000).

- Keksinnön ensimmäisen aspektin mukaisesti tarjotaan menetelmä videokuvasignaalin koodaamiseksi, jossa koodataan ensimmäinen kokonainen kehys muodostamalla bittivirta, joka sisältää tietoa kehyyksen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tietoon, tunnettu siitä että:
- 5 määritetään vähintään yksi virtuaalikehys ensimmäisen sellaisen kokonaisen kehyyksen version perusteella, joka versio on muodostettu käyttäen ensimmäisen kokonaisen kehyyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa; ja
- 10 koodataan toinen kokonainen kehys muodostamalla bittivirta, joka sisältää tietoa kehyyksen myöhempää täydellistä uudelleenmuodostamista varten, siten, että toinen kokonainen kehys voidaan uudelleenmuodostaa täysin virtuaalikehyyksen perusteella eikä ensimmäisen kokonaisen kehyyksen perusteella.
- 15 Eräässä suoritusmuodossa keksintö on skaalattava koodausmenetelmä. Tässä tapauksessa virtuaalikehykset voidaan tulkita skaalattavan bittivirran pohjakerrokseksi.
- Edullisesti menetelmä käsittää vaiheet, joissa:
- 20 priorisoidaan toisen kokonaisen kehyyksen tieto korkeamman ja matalamman prioriteetin tiedoksi; ja
- koodataan kolmas kokonainen kehys määrittämällä virtuaalikehys kolmannen kokonaisen kehyyksen version perusteella, joka versio on muodostettu käyttämällä kolmannen kokonaisen kehyyksen korkeamman prioriteetin tietoa
- 25 kolmannen kokonaisen kehyyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa, joka kolmas virtuaalikehys ennustetaan toisen kokonaisen kehyyksen sellaisesta versiosta, joka on muodostettu käyttämällä toisen kokonaisen kehyyksen korkeamman prioriteetin tietoa toisen kokonaisen kehyyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa. Näin
- 30 ollen virtuaalikehys voidaan määrittää ennustettuna toisesta virtuaalikehyyksestä ja voidaan tarkentaa käyttämällä sen omaa korkeamman prioriteetin tietoa. Tämän mukaisesti voidaan tarjota myös ennustusketjuja.

Edullisesti tieto kokonaisen kehyksen myöhempää uudelleenmuodostamista varten on priorisoitu korkeampaan ja matalampaan prioriteettiin sen mukaan, kuinka merkittävää tieto on kokonaisen kehyksen täysin uudelleenmuodostetun version tuottamisessa.

5

Kokonaiset kehykset voivat olla skaalattavan kehyksen pohjakerroksia. Ne ovat kokonaisia siinä mielessä, että voidaan muodostaa näytettäväksi kelpaava kuva. Tämä ei välttämättä päde virtuaalikehyksiin.

- 10 Edullisesti ennustusketjussa ennustetaan joukko virtuaalikehyksiä. Alkuperäinen kehys ketjussa voi olla normaali INTRA-kehys tai se voi olla virtuaalinen INTRA-kehys.

- 15 Virtuaalikehys muodostetaan käyttämällä korkeamman prioriteetin tietoa ja jättämällä tarkoituksella käyttämättä matalamman prioriteetin tietoa. Edullisesti virtuaalikehystä ei näytetä näytöllä. Vaihtoehtoisesti, jos se näytetään, sitä käytetään vaihtoehtona kokonaiselle kehykselle. Näin voi käydä, jos kokonaista kehystä ei ole saatavilla lähetysvirheen takia.

- 20 Keksintö mahdollistaa koodaustehokkuuden parantamisen ajallisen ennustusketjun lyhentämisen yhteydessä.

- 25 Edullisesti tieto käsittää koodisanoja. Virtuaalikehykset voidaan muodostaa tai määrittää yksinomaan korkeamman prioriteetin tiedosta, mutta ne voidaan muodostaa tai määrittää myös jostain matalamman prioriteetin tiedosta.

- 30 Virtuaalikehys voidaan ennustaa aiemmasta virtuaalikehyksestä tai kokonaisesta kehyksestä. Vaihtoehtoisesti tai lisäksi virtuaalikehys voidaan ennustaa seuraavasta virtuaalikehyksestä tai kokonaisesta kehyksestä käyttämällä virtuaalikehysten ennustusta taaksepäin. INTER-kehysten ennustusta taaksepäin on selostettu edellä viitaten kuvioon 14. On ymmärrettävä, että tätä periaatetta voidaan soveltaa virtuaalikehyksiin.

Edullisesti virtuaalikehys voidaan dekodata käyttämällä sekä sen korkeamman että matalamman prioriteetin tietoa (eikä sen tietoa ole välttämättä edes jaettu korkeampaan ja matalampaan prioriteettiin) ja se voidaan ennustaa toisen virtuaalikehysten perusteella. Tällä tavoin virtuaalikehys voidaan määrittää tai muodostaa vaikka sen tietoa ei olisi priorisoitu korkeampaan ja matalampaan prioriteettiin.

Edullisesti bittivirran dekodamisessa virtuaalikehystä varten käytetään eri algoritmia kuin bittivirran dekodamisessa kokonaista kehystä varten. Virtuaalikehysten dekodamista varten voi olla useita algoritmeja. Tietyn algoritmin valinta voidaan osoittaa bittivirrassa.

Edullisesti matalamman prioriteetin tiedon puuttuessa se voidaan korvata oletusarvoilla, jotta virtuaalikehysten dekodaus voidaan suorittaa. Oletusarvojen valinta voi vaihdella, ja oikea valinta voidaan osoittaa bittivirrassa.

Keksinnön toisen aspektin mukaisesti tarjotaan menetelmä videokuva-signaalin dekodamiseksi, jossa dekodataan ensimmäinen kokonainen kehys bittivirrasta, joka sisältää tietoa kehysten myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tietoon, tunnettu siitä että:

määritetään vähintään yksi virtuaalikehys ensimmäisen sellaisen kokonaisen kehysten version perusteella, joka versio on muodostettu käyttäen ensimmäisen kokonaisen kehysten korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehysten vähintään jonkun matalamman prioriteetin tiedon puuttuessa; ja ennustetaan toinen kokonainen kehys virtuaalikehysten perusteella eikä ensimmäisen kokonaisen kehysten perusteella.

Edullisesti menetelmä käsittää vaiheen, jossa dekodataan kolmas kokonainen kehys määrittämällä virtuaalikehys kolmannen kokonaisen kehysten sellaisen version perusteella, joka on muodostettu käyttämällä kolmannen kokonaisen kehysten korkeamman prioriteetin tietoa kolmannen kokonaisen kehysten vähintään jonkun matalamman prioriteetin tiedon puuttuessa, joka kolmas virtuaalikehys ennustetaan toisen kokonaisen kehysten sellaisesta

virtuaaliversiosta, joka on muodostettua käyttämällä toisen kokonaisen kehyksen korkeamman prioriteetin tietoa toisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa. Näin ollen virtuaalikehys voidaan määrittää ennustettuna toisesta virtuaalikehyksestä ja tarkentaa käyttämällä sen
 5 omaa korkeamman prioriteetin tietoa. Tämän mukaisesti voidaan tarjota myös ennustusketjuja. Kokonainen kehys voidaan dekodata virtuaalikehyksestä. Kokonainen kehys voidaan dekodata virtuaalikehysten ennustusketjusta.

Keksinnön kolmannen aspektin mukaisesti tarjotaan videokuvakooderi
 10 videokuvasignaalin koodaamiseksi käsittäen kokonaisen kehyksen kooderin ensimmäisen kokonaisen kehyksen bittivirran muodostamiseksi sisältäen tietoa ensimmäisen kokonaisen kehyksen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tiedoksi, tunnettu siitä että videokuvakooderi käsittää:
 15 virtuaalikehyksen kooderin, joka määrittää vähintään yhden virtuaalikehyksen ensimmäisen kokonaisen kehyksen versiona, joka on muodostettu käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa; ja
 20 kehyksen ennustimen toisen kokonaisen kehyksen ennustamiseen virtuaalikehyksen perusteella eikä ensimmäisen kokonaisen kehyksen perusteella.

Edullisesti kooderi lähettää signaalin dekooderille osoittaakseen, mikä osa
 25 kehyksen bittivirrasta on riittävä hyväksyttävän kuvan tuottamiseen täysilaatuisen kuvan korvaamiseksi lähetysvirheen tai häviön sattuessa. Signaointi voi sisältyä bittivirtaan tai se voidaan lähettää erillään bittivirrasta. Kehyksen sijaan signaointia voidaan soveltaa kuvan osaan, esimerkiksi viipaleeseen, lohkoon, makrolohkoon, tai lohkokoryhmään. Signaloinnilla voidaan osoittaa, mikä usean
 30 kuvan joukosta on riittävä hyväksyttävän kuvan tuottamiseksi korvaamaan täysilaatuisen kuvan.

Edullisesti kooderi voi lähettää signaalin dekooderille osoittaakseen miten muodostetaan virtuaalinen varaviitekuva, jota käytetään, jos varsinainen viitekuva katoaa tai on liian vahingoittunut.

- 5 Keksinnön neljännen aspektin mukaisesti tarjotaan dekooderi videokuvasignaalin dekoodaamiseksi käsittäen kokonaisen kehyksdekooderin ensimmäisen kokonaisen kehyksen dekoodaamiseksi bittivirrasta, joka sisältää tietoa kehyksen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tietoon, tunnettu siitä
- 10 että dekooderi käsittää:
virtuaalikehyksen dekooderin vähintään yhden virtuaalikehyksen muodostamiseksi ensimmäisen kokonaisen kehyksen bittivirrasta käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon
- 15 puuttuessa; ja
kehyksen ennustimen toisen kokonaisen kehyksen ennustamiseen virtuaalikehyksen eikä ensimmäisen kokonaisen kehyksen perusteella.

- 20 Keksintö pätee tilanteisiin, joissa matalamman prioriteetin tietoa ei käytetä virtuaalikehysten muodostamisessa, koska matalamman prioriteetin tieto on kadonnut esimerkiksi lähetyksen aikana. Vaihtoehtoisesti dekooderi voi päättää olla käyttämättä hallussaan olevaa matalamman prioriteetin tietoa. Näin voi tapahtua, jos keksinnössä käytetään viitekuvanvalintaa (Reference Picture Selection) ja dekooderi päättää ennustaa viitekehyksestä, joka ei ole seuraava
- 25 ennustusketjussa luonnollisesti esiintyvä kehys. Tällä tavoin ennustusketjuja voidaan jakaa osiin. Näin voidaan ratkaista virheenkulkeutumisongelmat.

- 30 Viitekuvanvalinnan kohdalla kooderi ja dekooderi voidaan varustaa kokonaisten kehysten säilyttämiseen tarkoitetuilla monikehyspuskureilla ja virtuaalikehysten säilyttämiseen tarkoitettulla monikehyspuskurilla.

Edullisesti toisen kehyksen ennustamiseen käytetyn viitekehysten voi valita esimerkiksi kooderi, dekooderi tai molemmat. Viitekehysten valinta voidaan suorittaa erikseen jokaiselle kehykselle, kuvasegmentille, makrolohkolle, lohkolle,

tai mille tahansa kuvan alaelementille. Viitekehys voi olla mikä tahansa kokonainen tai virtuaalikehys, joka on saatavilla tai joka voidaan muodostaa kooderissa ja dekooderissa.

- 5 Tällä tavoin jokainen kokonainen kehys ei ole rajattu yhteen virtuaalikehykseen, vaan se voidaan liittää useisiin eri virtuaalikehyksiin, joista jokaisella on erilainen tapa luokitella kokonaisen kehyksen bittivirtaa. Nämä erilaiset bittivirran luokittelutavat voivat olla eri viitekuvia (virtuaalisia tai kokonaisia) liikekompensointia varten ja/tai erilaisia tapoja dekodata bittivirran korkeamman
- 10 prioriteetin osaa.

- Edullisesti dekooderi antaa kooderille palautetta (feedback). Palaute voi olla osoituksen muodossa, joka koskee yhden tai useamman määritetyn kuvan osoitettuja koodisanoja. Osoitus voi osoittaa, että koodisanat on vastaanotettu, ei
- 15 ole vastaanotettu, tai on vastaanotettu vahingoittuneina. Tämän seurauksena kooderi voi lähettää koodisanat uudelleen. Osoitus voi määrittää tietyn kuvan tietyn alueen koodisanat tai useiden kuvien tietyn alueen koodisanat.

- Keksinnön viidennen aspektin mukaisesti tarjotaan videokuvaviestintäjärjestelmä
- 20 videosignaalin koodaamiseksi ja dekodaaamiseksi, joka järjestelmä käsittää kooderin ja dekooderin, jossa kooderi käsittää kokonaisen kehyksen kooderin ensimmäisen kokonaisen kehyksen bittivirran muodostamiseksi sisältäen tietoa ensimmäisen kokonaisen kehyksen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja
- 25 matalamman prioriteetin tiedoksi, tunnettu siitä että kooderi käsittää: virtuaalikehyksen kooderin, joka määrittää vähintään yhden virtuaalikehyksen ensimmäisen kokonaisen kehyksen versiona, joka on muodostettu käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon
- 30 puuttuessa; ja kehyksen ennustimen toisen kokonaisen kehyksen ennustamiseen virtuaalikehyksen perusteella eikä ensimmäisen kokonaisen kehyksen perusteella; ja jossa dekooderi käsittää:

kokonaisen kehysdekooderin ensimmäisen kokonaisen kehyksen dekodoraamiseksi bittivirrasta, joka sisältää tietoa kehyksen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tietoon;

- 5 virtuaalikehyksen dekooderin vähintään yhden virtuaalikehyksen muodostamiseksi ensimmäisen kokonaisen kehyksen bittivirrasta käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa; ja
- 10 kehyksen ennustimen toisen kokonaisen kehyksen ennustamiseen virtuaalikehyksen eikä ensimmäisen kokonaisen kehyksen perusteella.

Keksinnön kuudennen aspektin mukaisesti tarjotaan videokuvaviestintäpääte, joka käsittää videokuvakooderin, joka käsittää kokonaisen kehyksen kooderin

15 ensimmäisen kokonaisen kehyksen bittivirran muodostamiseksi sisältäen tietoa ensimmäisen kokonaisen kehyksen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tiedoksi, tunnettu siitä että videokuvakooderi käsittää:

- 20 virtuaalikehyksen kooderin, joka määrittää vähintään yhden virtuaalikehyksen ensimmäisen kokonaisen kehyksen versiona, joka on muodostettu käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa; ja
- 25 kehyksen ennustimen toisen kokonaisen kehyksen ennustamiseen virtuaalikehyksen perusteella eikä ensimmäisen kokonaisen kehyksen perusteella.

- Keksinnön seitsemännen aspektin mukaisesti tarjotaan videokuvaviestintäpääte, joka käsittää videokuvadekooderin, joka käsittää kokonaisen kehysdekooderin
- 30 ensimmäisen kokonaisen kehyksen dekodoraamiseksi bittivirrasta, joka sisältää tietoa kehyksen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tietoon, tunnettu siitä että videokuvadekooderi käsittää:

virtuaalikehyksen dekooderin vähintään yhden virtuaalikehyksen muodostamiseksi ensimmäisen kokonaisen kehyksen bittivirrasta käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa; ja

5 kehyksen ennustimen toisen kokonaisen kehyksen ennustamiseen virtuaalikehyksen eikä ensimmäisen kokonaisen kehyksen perusteella.

Keksinnön kahdeksannen aspektin mukaisesti tarjotaan tietokoneohjelmatuote tietokoneen toimimiseksi videokuvakooderina, joka ohjelmatuote käsittää:

10 tietokonesuoritettavan koodin ensimmäisen kokonaisen kehyksen koodaamiseksi muodostamalla bittivirta, joka sisältää tietoa kehyksen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tietoon;

15 tietokonesuoritettavan koodin vähintään yhden virtuaalikehyksen määrittämiseksi ensimmäisen sellaisen kokonaisen kehyksen version perusteella, joka versio on muodostettu käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa; ja

20 tietokonesuoritettavan koodin toisen kokonaisen kehyksen koodaamiseksi muodostamalla bittivirta, joka sisältää tietoa sen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tietoon, mikä mahdollistaa toisen kokonaisen kehyksen täydellisen uudelleenmuodostamisen virtuaalikehyksen perusteella eikä

25 ensimmäisen kokonaisen kehyksen perusteella.

Keksinnön yhdeksännen aspektin mukaisesti tarjotaan tietokoneohjelmatuote tietokoneen toimimiseksi videokuvadekooderina, joka ohjelmatuote käsittää:

30 tietokonesuoritettavan koodin ensimmäisen kokonaisen kehyksen dekoodaamiseksi bittivirrasta, joka sisältää tietoa kehyksen myöhempää täydellistä uudelleenmuodostamista varten, joka tieto on priorisoitu korkeamman ja matalamman prioriteetin tietoon;

tietokonesuoritettavan koodin vähintään yhden virtuaalikehyksen määrittämiseksi ensimmäisen sellaisen täydellisen kehyksen version perusteella, joka versio on

muodostettu käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa; ja

5 tietokonesuoritettavan koodin toisen kokonaisen kehyksen ennustamiseen virtuaalikehyksen perusteella eikä ensimmäisen kokonaisen kehyksen perusteella.

Edullisesti kahdeksannen ja yhdeksannen aspektin mukaiset tietokoneohjelmat ovat tallennettuina muistivälineissä. Kyseessä voi olla kannettava muistiväline tai

10 muistiväline laitteen sisällä. Laite voi olla kannettava, esimerkiksi sylimikro, PDA-laite tai matkapuhelin.

Viittausten "kehyksiin" keksinnön yhteydessä on tarkoitus sisältää myös kehysten osat, kuten kehyksen sisällä olevat viipaleet, lohkot ja MB:t.

15

Verrattuna PFGS:ään keksintö tarjoaa paremman kompressointitehokkuuden. Tämä johtuu siitä, että sen skaalattavuushierarkia on joustavampi. PFGS ja keksintö voivat olla samassa koodausmenetelmässä. Tässä tapauksessa keksintö toimii PFGS:n pohjakerroksen alla.

20 Keksintö mahdollistaa ennustamisen virtuaalikehyksistä, toisin sanoen ennustamisen bittivirran merkittävimmästä osasta. Tekniikan tasossa liikekompensoinnissa käytetään vain normaalikehyksiä, toisin sanoen koko bittivirtaa, liikekompensoinnin viitteinä. Jos korkeamman ja matalamman prioriteetin tiedon suojaus on epätasaista, keksintö tarjoaa parempaa kompressointitehokkuutta. Epätasainen virheensuojaus voi käsittää korkeamman ja matalamman prioriteetin tietoa sisältävien pakettien lähettämistä siten, että korkeamman prioriteetin tietoa sisältävien pakettien katoaminen on vähemmän todennäköistä. Jos videokuvan bittivirta sisältää INTRA- ja INTER-kehyksiä,

25 matalamman prioriteetin pakettien käyttöä rajoitetaan pitkissä ennustusketjuissa pakettihäviöihin varautumiseksi. Tästä esimerkkinä viitekuvan valintaa (reference picture selection) voidaan käyttää tuottamaan ajallisesti skaalattavaa bittivirtaa seuraavasti: I0 P2 P4 P6 P8 P10(I0) P12 P14 P16 P18 P20(P10) , missä kehystyyppin jälkeinen arvo vastaa aikaleimaa (ajallista viitettä) ja sulkujen sisällä

30

oleva kehys on liikekompensoinnin viitekehys. Jos yhtään viitekehystä ei ole lueteltu, edellistä kehystä käytetään viitteenä. Kehykset I0, P10, P20, voidaan kapseloida korkeamman prioriteetin paketteihin ja loput kehyksistä voidaan kapseloida matalamman prioriteetin paketteihin. Keksintö tarjoaa edullisen tavan

5 hyödyntää korkeamman prioriteetin pakettien kantamaa tietoa. Tämä voi käsittää matalamman prioriteetin kehysten korkeamman prioriteetin osien kapseloimisen korkeamman prioriteetin paketteihin, esimerkiksi lähettämällä P2:n, P4:n, P6:n, ja P8:n liiketiedot korkeamman prioriteetin paketissa. Keksintö tarjoaa menetelmän, jolla kooderi voi käyttää matalamman prioriteetin kehyksistä vain tätä

10 korkeamman prioriteetin osaa ennustusviitteenä. Kompressoitietu saadaan, koska korkeamman prioriteetin paketeissa kannetaan enemmän dataa, ja koska uusi koodausmenetelmä sallii kaiken korkeamman prioriteetin pakettien kantaman datan hyödyntämisen (vaikka data ei ehkä sisälläkään kaikkea tarpeellista tietoa kehyksen täydelliseksi rekonstruoimiseksi). Keksintö tarjoaa

15 (virtuaaliset) viitekehukset, jotka muistuttavat ennustettavia kehyksiä enemmän kuin tekniikan tasossa. Näin ollen keksinnössä tarvitaan vähemmän bittejä ennustusvirhetiedon koodaamiseen.

Keksintöä selostetaan seuraavassa esimerkinomaisesti viittaamalla oheisiin

20 piirustuksiin, joissa:

kuvio 1 esittää erästä videokuvansiirtojärjestelmää,
 kuvio 2 esittää B-kuvien ennustamista ankkurikuvapareista,
 kuvio 3 esittää IP-monijakelujärjestelmää (multi-casting suýstem),
 kuvio 4 esittää SNR-skaalattavia kuvia,

25 kuvio 5 esittää spatiaalisesti skaalattavia kuvia,
 kuvio 6 esittää ennustussuhteita FGS-koodauksessa,
 kuvio 7 esittää skaalattavassa koodauksessa käytettäviä perinteisiä ennustussuhteita,
 kuvio 8 esittää ennustussuhteita PFGS-koodauksessa,

30 kuvio 9 esittää kanavansovitusta (channel adaptation) PFGS:ssä,
 kuvio 10 esittää perinteistä ajallista ennustamista,
 kuvio 11 havainnollistaa ennustuspolkujen lyhentämistä viitekuva valintaa (reference picture selection) käyttämällä,
 kuvio 12 havainnollistaa videoredundanssikoodausta,

- kuvio 13 esittää videoredundanssikoodausta käsittelemässä vahingoittuneita säikeitä,
 kuvio 14 havainnollistaa INTRA-kehysten uudelleenkohdistamista ja INTER-kehysten ennustamista taaksepäin,
 5 kuvio 15 esittää INTRA-kehystä seuraavia perinteisiä kehysten ennustamissuhteita,
 kuvio 16 esittää videokuvansiirtojärjestelmää,
 kuvio 17 esittää syntaksielementtien riippuvuuksia H.26L:n TML-4-testimallissa,
 kuvio 18 esittää koodausmenettelyä,
 10 kuvio 19 esittää dekodeausmenettelyä,
 kuvio 20 esittää kuvion 19 esittämän dekodeausmenettelyn muunnelmaa,
 kuvio 21 havainnollistaa keksinnön mukaista videokuvankoodausmenetelmää,
 kuvio 22 havainnollistaa keksinnön mukaista toista videokuvankoodausmenetelmää,
 15 kuvio 23 esittää keksinnön mukaista videokuvansiirtojärjestelmää,
 kuvio 24 esittää ZPE-kuvia hyödyntävää videokuvansiirtojärjestelmää.

Kuvioita 1-17 on jo kuvattu edellä.

- 20 Keksintö voidaan toteuttaa kuvion 16 mukaisessa videokuvansiirtojärjestelmässä.

- Keksinnön kuvaamiseksi sitä selostetaan ensin algoritminä sen periaatteiden havainnollistamiseksi, ja sitten sitä selostetaan yksityiskohtaisemmin viitaten
 25 kuvioihin 18, 19 ja 20. Algoritmillä on ensimmäinen vaihe, joka tapahtuu koodaamisen aikana ja toinen vaihe, joka tapahtuu dekodeamisen aikana.

- Ensimmäisessä vaiheessa (paketointi) muodostaan muunneltu bittivirta, joka sisältää vain signaloidut parametrit. Se voidaan saada suoraan NAL:ltä tai se
 30 voidaan luoda käyttäen algoritmiä. Algoritmi jäsentää ensin bittivirran, joka vastaa viimeistä kuvaa, jota pitäisi käyttää viitteen luomiseen. Sitten se luo bittivirrasta toisen esitysmuodon, joka sisältää vain signaloidut parametriluokat ja jossa käytetään oletusarvoja ei-signaloiduille parametreille (nolla, ei olemassa, jne). Samaan aikaan se päättää, mitä kehyksiä käytetään jäsenetyn kehysten

ennustusviitteinä. Jos viitekehys ei ole virtuaalikehys ja jos se on signaloidussa kehysjoukossa, algoritmi jäsentää rekursiivisesti viitekehysten bittivirran.

5 Toisessa vaiheessa algoritmi dekodaa juuri muodostetun bittivirran aloittaen siitä bittivirran osasta, joka vastaa ennustamiseen käytettyä ensimmäistä signaloitua kuvaa. Jos bittivirta viittaa viitekehykseen, jota ei ole virtuaalisessa monikehysviitepuskurissa, se saadaan normaalista monikehysviitepuskurista. Muuten algoritmi käyttää uudelleenmuodostettua kehystä virtuaalisesta monikehysviitepuskurista. Jokainen uudelleenmuodostettu kehys sijoitetaan 10 virtuaaliseen monikehysviitepuskuriin. Algoritmin ulostulo sijaitsee virtuaalisessa monikehysviitepuskurissa.

Algoritmi edellyttää, että on olemassa samankaltainen, mutta erillinen monikehysviitepuskuri, jota käytetään normaalia dekodausta varten. Jokaisen 15 puskurin paikan pitäisi pystyä varastoimaan uudelleenmuodostettu kehys ja kehysten bittivirta. H.263:n liiteluonnoksessa U (H.26L:ään tehtyine asianmukaisine muutoksineen) kuvattua monikuvapuskuria voidaan käyttää.

Seuraavaksi keksintöä selostetaan yksityiskohtaisemmin sarjana 20 menettelyvaiheita viitaten kuvioon 18, joka havainnollistaa kooderin suorittamaa koodausmenettelyä, ja kuvioon 19, joka havainnollistaa kooderia vastaavan dekooderin dekodausmenettelyä.

Seuraavaksi viitataan kuvion 18 esittämään menettelyyn. Alustusvaiheessa 25 kooderi alustaa kehyslaskimen (vaihe 110), alustaa normaalin viitekehyspuskurin (vaihe 112) ja alustaa virtuaalisen viitekehyspuskurin (vaihe 114). Sen jälkeen kooderi vastaanottaa raakaa koodaamatonta videokuvadataa lähteestä (vaihe 16), esimerkiksi videokamerasta. Videokuvadata voi myös olla peräisin suorasta syötöstä. Kooderi vastaanottaa osoituksen sen hetkisen kehysten koodauksessa 30 käytettävästä koodausmuodosta (vaihe 118), toisin sanoen, pitääkö siitä tulla INTRA-kehys vai INTER-kehys. Osoitus voi tulla valitusta koodausmenettelystä (lohko 120). Osoitus voi valinnaisesti tulla otoksen vaihtumisen tunnistajalta (scene cut detector, lohko 122), jos sellainen on, tai palautteena dekooderilta

(lohko 124). Sitten kooderi päättää, koodaako se sen hetkisen kehyyksen INTRA-kehyyksenä (vaihe 126).

5 Jos päätös on "KYLLÄ" (päättös 128), sen hetkinen kehys koodataan muodostamaan kompressoitu kehys INTRA-kehysmuodossa (vaihe 130).

10 Jos päätös on "EI" (päättös 132), kooderi vastaanottaa osoituksen viitteenä käytettävästä kehyyksestä sen hetkistä kehystä koodatessa INTER-kehysmuotoon (vaihe 134). Tämä voidaan määrätä ennalta valitun koodausmenettelyn seurauksena (lohko 136). Keksinnön eräässä toisessa suoritusmuodossa tätä vaihetta voidaan ohjata dekooderilta saadun palautteen avulla (lohko 138). Tätä kuvataan myöhemmin. Tunnistettu viitekehys voi olla normaalikehys tai virtuaalikehys, joten kooderi päättää, käytetäänkö virtuaalista viitekehystä (vaihe 140).

15

Jos käytetään virtuaaliviitekehystä, se haetaan virtuaalisesta viitekehyspuskurista (vaihe 142). Jos virtuaaliviitekehystä ei käytetä, normaali viitekehys haetaan normaalista kehyspuskurista (vaihe 144). Tämä edellyttää normaalin ja virtuaalisen viitekehyyksen olemassaoloa omissa puskuureissaan. Jos
20 kooderi lähettää ensimmäisen kehyyksen alustuksen jälkeen, kyseessä on yleensä INTRA-kehys, joten viitekehystä ei käytetä. Sen jälkeen sen hetkinen kehys koodataan INTER-kehysmuotoon raakaa videokuvadataa ja valittua viitekehystä käyttäen (vaihe 146).

25 Sen jälkeen sovelletaan seuraavia vaiheita riippumatta siitä, koodataanko sen hetkinen kehys INTRA- vai INTER-kehysmuotoon. Koodattu kehysdata priorisoidaan (vaihe 148) sen mukaisesti, onko käytetty INTER- vai INTRA-kehyskoodausta. Priorisoinnissa data jaetaan matalamman ja korkeamman prioriteetin dataan sen perusteella, kuinka oleellista data on koodattavan kuvan uudelleen muodostamisen kannalta. Jakamisen jälkeen muodostetaan bittivirta
30 lähetystä varten. Bittivirran muodostamisessa käytetään soveltuvaa paketointimenetelmää, esimerkiksi jotain edellä mainituista. Sen jälkeen bittivirta lähetetään dekooderille (vaihe 152). Jos sen hetkinen kehys on viimeinen kehys, tehdään päätös (vaihe 154) lopettaa menettely (lohko 156) tässä vaiheessa.

Jos oletetaan, että sen hetkinen kehys ei ole viimeinen kehys, sen hetkistä kehystä edustava bittivirta dekodataan asianmukaisen viitekehysten perusteella käyttämällä sekä korkeamman että matalamman prioriteetin dataa kehyksen muodostamiseksi kokonaan uudelleen (vaihe 157). Kokonaan uudelleen muodostettu kehys tallennetaan normaaliin viitekehyspuskuriin. Seuraavaksi sen hetkistä kehystä edustava bittivirta dekodataan asianmukaisen viitekehysten perusteella käyttämällä ainoastaan korkeamman prioriteetin dataa virtuaalikehyksen uudelleen muodostamiseksi (vaihe 160). Uudelleen muodostettu virtuaalikehys tallennetaan virtuaaliseen viitekehyspuskuriin (vaihe 162). Sarja menettelyvaihteita alkaa jälleen vaiheesta 116, ja seuraava kehys koodataan ja muodostetaan bittivirraksi.

Edellä esitettyjen vaiheiden järjestys voi olla erilainen keksinnön vaihtoehtoisissa suoritusmuodoissa. Esimerkiksi alustusvaiheet voivat tapahtua missä tahansa soveltuvassa järjestyksessä, samoin kokonaan uudelleen muodostetun normaalin viitekehysten ja uudelleen muodostetun virtuaaliviitekehysten dekodausvaiheet.

Vaikka edellä kuvataan kehyksen ennustamista yhdestä viitekehyksestä, keksinnön toisessa suoritusmuodossa voidaan yhden kuvan ennustamiseen käyttää useampaa kuin yhtä viitekehystä. Viitekehysten valinta voidaan suorittaa jokaiselle koodattavan/dekodattavan kehyksen kuvasegmentille, makrolohkolle, lohkolle, tai mille tahansa kuvan alaelementille erikseen. Viitekehys voi olla mikä tahansa kokonainen tai virtuaalikehys, joka on saatavilla tai joka voidaan luoda kooderissa ja dekodeerissa. Joissain tapauksissa, kuten B-kehysten kohdalla, samaan kuva-alueeseen assosioituu kaksi tai useampaa viitekehystä ja koodattavan alueen ennustamiseen käytetään jonkinlaista interpolaatiomenetelmää. Jokainen kokonainen kehys voi assosioitua useisiin erilaisiin virtuaalikehyksiin, jotka edustavat:

erilaisia tapoja luokitella bittivirta kokonaiselle kehykselle, erilaisia viitekuvia (virtuaalisia tai kokonaisia) liikekompensointia varten ja/tai erilaisia tapoja dekodata bittivirran korkeamman prioriteetin osa.

Sen vuoksi tällaisessa suoritusmuodossa viittaukset normaaliin ja virtuaaliseen viitekehyspuskuriin olisivat todella viittauksia normaaliin ja virtuaaliseen viitekehyspuskuriin.

- 5 Seuraavaksi viitataan kuvion 19 esittämään menettelyyn. Alustusvaiheessa dekooderi alustaa virtuaalisen viitekehyspuskurin (vaihe 210), normaalin viitekehyspuskurin (vaihe 211) ja kehyslaskimen (vaihe 212). Sen jälkeen dekooderi ottaa vastaan kompressoituun sen hetkiseen kehykseen liittyvän bittivirran (vaihe 214). Sitten dekooderi päättää koodataanko sen hetkinen kehys
- 10 INTRA- vai INTER-kehysmuotoon (vaihe 216). Tämä voidaan päätellä esimerkiksi kuvan otsikossa saadusta tiedosta.

- Jos sen hetkinen kehys on INTRA-kehysmuodossa, se dekodataan käyttämällä kokonaista bittivirtaa INTRA-kehysmuodostamiseksi kokonaan uudelleen
- 15 (vaihe 218).

- Jos sen hetkinen kehys on viimeinen kehys, tehdään päätös (vaihe 220) lopettaa menettely (vaihe 222). Jos oletetaan, että sen hetkinen kehys ei ole viimeinen kehys, silloin sen hetkistä kehystä edustava bittivirta dekodataan käyttämällä
- 20 korkeamman prioriteetin dataa virtuaaliviitekehysmuodostamiseksi (vaihe 224).

- Jos sen hetkinen kehys on INTER-kehysmuodossa, sen ennustamisessa (kooderilla) käytetty viitekehys tunnistetaan (vaihe 226). Viitekehys voidaan
- 25 tunnistaa bittivirrassa lähetetyn tiedon perusteella.

- Tunnistettu viitekehys voi olla normaalikehys tai virtuaalikehys, joten dekooderi päättää, käytetäänkö virtuaalista viitekehystä (vaihe 228).

- 30 Jos käytetään virtuaalista viitekehystä, se haetaan virtuaalisesta viitekehyspuskurista (vaihe 230). Muussa tapauksessa normaali viitekehys haetaan normaalista viitekehyspuskurista (vaihe 232). On syytä huomata, että korkeamman prioriteetin tiedon dekoodaaminen virtuaalikehysmuodostamiseksi ei noudata samaa dekoodausmenettelyä kuin kehysmuodostamiseksi.

kokonaisen esitystavan dekoddaaminen. Esimerkiksi matalamman prioriteetin tiedon puuttuessa se voidaan korvata joillain oletusarvoilla, jotta virtuaalikehyksen dekoddaus voidaan suorittaa.

- 5 Tämä edellyttää normaalin ja virtuaalisen viitekehyksen olemassaoloa omilla puskureissaan. Jos dekooderi vastaanottaa ensimmäisen kehyksen alustuksen jälkeen, kyseessä on yleensä INTRA-kehys, joten viitekehystä ei käytetä.

- 10 Sen jälkeen sen hetkinen kehys dekoddataan INTER-kehysmuodossa käyttämällä kokonaista vastaanotettua bittivirtaa ja tunnistettua viitekehystä ennustusviitteenä (vaihe 234).

- 15 Jos sen hetkinen kehys on viimeinen kehys, tehdään päätös (vaihe 236) lopettaa menettely (vaihe 222). Jos oletetaan, että sen hetkinen kehys ei ole viimeinen kehys, silloin sen hetkistä kehystä edustava bittivirta dekoddataan käyttämällä korkeamman prioriteetin dataa virtuaalisen viitekehyksen muodostamiseksi (vaihe 238). Sitten tämä virtuaalinen viitekehys tallennetaan virtuaaliseen viitekehyspuskuriin (vaihe 240).

- 20 Kuten edellä on mainittu, keksinnön eräässä suoritusmuodossa normaalin tai virtuaalisen viitekehyksen valitseminen kooderissa suoritetaan dekooderilta saadun palautteen perusteella.

- 25 Kuvio 20 esittää lisävaiheita, joilla muokataan kuvion 19 menettelyä palautteen antamiseksi. Kuvion 20 esittämät lisävaiheet lisätään kuvion 19 vaiheiden 214 ja 216 väliin. Koska kuvio 19 on kokonaan kuvattu edellä, seuraavaksi selostetaan vain lisävaiheet. Tämä on vain yksi toteutus useiden mahdollisuuksien joukossa. Esimerkiksi INTRA-kuvan pyytämisen sijaan dekooderi voisi osoittaa, että kaikki kehyksen data on kadonnut, ja kooderin pitäisi reagoida tähän osoitukseen siten, että se ei viittaisi kadonneeseen kehykseen liikekompensoinnissa. Toisin sanoen
30 INTRA-kuvapäivitystä ei välttämättä tarvittaisi.

Kun kompressoidun sen hetkisen kehyksen bittivirta on vastaanotettu (vaihe 214), dekooderi tarkistaa (vaihe 310) onko bittivirta vastaanotettu oikein. Tähän

kuuluu yleinen virheentarkistus, jonka jälkeen tehdään tarkempia tarkistuksia virheen vakavuuden mukaan. Jos bittivirta on vastaanotettu oikein, dekooderi päättää koodataanko sen hetkinen kehys INTRA- vai INTER-kehysmuotoon (vaihe 216).

5

Jos bittivirtaa ei ole vastaanotettu oikein, dekooderi päättää pystyykö se dekoodaamaan kuvan otsikon (vaihe 312). Jos se ei pysty, se toimittaa INTRA-kehysten päivityspyynnön kooderin sisältävälle lähettävälle päätteelle (vaihe 314) ja menettely palaa vaiheeseen 214.

10

Jos dekooderi pystyy dekoodaamaan kehysten otsikon, se päättää pystyykö se dekoodaamaan korkeamman prioriteetin dataa (vaihe 316). Jos se ei pysty, seuraa vaihe 314.

15

Jos dekooderi pystyy dekoodaamaan korkeamman prioriteetin dataa, se päättää pystyykö se dekoodaamaan matalamman prioriteetin dataa (vaihe 318). Jos se ei pysty, se ohjeistaa kooderin sisältävän lähettävän päätteen koodaamaan seuraavan kehysten ennustettuna ainoastaan sen hetkisen kehysten korkeamman prioriteetin datasta eikä matalamman prioriteetin datasta (vaihe

20

320). Sen vuoksi keksinnössä on uudentyyppinen osoitus, joka koskee yhden tai useamman määritetyn kuvan osoitettuja koodisanoja. Osoitus voi osoittaa koodisanoja, jotka on vastaanotettu ja joita ei ole vastaanotettu. Tämä tarjoaa palautteen, johon viitattiin edellä liittyen lohkon 138. Tällä tavalla seuraava kehys koodataan virtuaalista viitekehystä käyttäen sen hetkisen kehysten

25

perusteella. Jos dekooderi voi dekodata matalamman prioriteetin dataa, menettely palaa vaiheeseen 214. Tämä pätee, jos viive on niin pieni, että kooderi voi saada palautetiedon ennen seuraavan kehysten koodaamista. Jos näin ei ole, on edullisempaa lähettää osoitus siitä, että matalamman prioriteetin osa tietystä kehyksestä katosi. Sitten kooderi reagoi tähän osoitukseen siten, että se

30

ei käytä matalamman prioriteetin tietoa kehyksessä, jonka se aikoo seuraavaksi koodata. Toisin sanoen, kooderi luo virtuaalikehysten, jonka ennustusketju ei sisällä kadonnutta matalamman prioriteetin osaa.

Kuvion 18 sekä kuvioiden 19 ja 20 menettelyt voidaan toteuttaa soveltuvan algoritmin tai soveltuvien algoritmien muodossa.

5 Bittivirran dekoodaamisessa virtuaalikehyksiä varten voidaan käyttää eri algoritmiä kuin bittivirran dekoodaamisessa kokonaisia tai normaaleja kehyksiä varten. Keksinnön eräässä suoritusmuodossa on joukko tällaisia algoritmeja, ja oikean algoritmin valinta voidaan signaloida bittivirrassa. Matalamman prioriteetin tiedon puuttuessa se voidaan korvata joillakin oletusarvoilla, jotta virtuaalikehyksen dekoodaus voidaan suorittaa. Oletusarvojen valinta voi 10 vaihdella, ja oikea valinta voidaan signaloida bittivirrassa esimerkiksi käyttämällä osoitusta, johon viitattiin edellisessä kappaleessa.

Kuvioiden 18, 19 ja 20 menettelyt käyttävät kehys-kehykseltä-dekoodausmenetelmää. Keksinnön muissa suoritusmuodoissa kyseessä voi olla 15 lohko-lohkolta- tai viipale-viipaleelta-menetelmä. Yleisesti ottaen keksintöä voidaan soveltaa mihin tahansa kuvasegmenttiin, ei pelkästään viipaleisiin ja lohkoihin.

Vaikka INTER-kehysten koodausta onkin selostettu, B-kehysten koodausta ei ole 20 selostettu yksinkertaisuuden vuoksi. B-kehysten koodaus voisi kuitenkin sisältyä keksinnön johonkin suoritusmuotoon. Lisäksi kuvien toissijaiset esitystavat (toisin sanoen tahdistuskehukset) voidaan sisällyttää keksinnön suoritusmuotoon. Jos tahdistuskehysten ennustamiseen käytetään virtuaalikehystä, dekooderin ei tarvitse luoda sitä, jos ensisijainen esitystapa on vastaanotettu oikein. Tässä 25 tapauksessa ei ole välttämätöntä muodostaa virtuaalista viitekehystä tahdistuskehysten muita jäljennöksiä varten. Perinteisessä skaalattavuusmenetelmässä on aina välttämätöntä dekodata pohjakerros.

Eräässä suoritusmuodossa videokuvakehys kapseloidaan ainakin kahteen 30 pakettiin, joista toisen prioriteetti on korkeampi ja toisen matalampi. Jos käytetään H.26L:ää, matalamman prioriteetin paketti voi sisältää esimerkiksi koodattuja lohkokuvia ja ennustusvirhekertoimia.

Kuvioissa 18, 19 ja 20 viitataan (lohkoissa 160, 224, 238) kehyksen dekodointiin käyttämällä korkeamman prioriteetin tietoa virtuaalikehyksen muodostamiseksi. Tämä voidaan toteuttaa kahdessa alavaiheessa:

- 1) luodaan väliaikainen bittivirtaus kehyksestä, joka koostuu korkeamman prioriteetin tiedosta sekä matalamman prioriteetin tiedon oletusarvoista.
- 2) dekodataan väliaikainen esitys normaalisti (kuten tapauksessa, jossa kaikki tieto on saatavilla).

Tämä on vain yksi keksinnön suoritusmuodoista, koska oletusarvojen valintaa voidaan säätää ja koska virtuaalikehyksen dekodausalgoritmi ei ole välttämättä sama kuin normaalin kehyksen.

Kuviot 18 ja 19 liittyvät keksinnön erääseen tiettyyn suoritusmuotoon. Jokaisesta normaalista kehyksestä muodostamien virtuaalikehysten lukumäärälle ei ole tiettyä rajaa. Menettelyn edullisessa suoritusmuodossa on useita erillisiä virtuaalikehysketjuja, koska ennustusketjuihin kuuluvia virtuaalikehysketjuja luodaan tarvittaessa.

Keksinnön bittivirtaus syntaksi on hyvin lähellä yksikerroksista koodausta, jossa skaalattavuuden parannuskerroksia ei tarvita. Koska virtuaalikehysketjuja ei näytetä, kooderit voivat päättää kuinka virtuaalinen ennustusviite luodaan alkaessaan koodata siihen perustuvaa kehystä. Toisin sanoen kooderit voivat hyödyntää edellisten kehysten bittivirtoja joustavasti, ja kehykset voidaan lähettämisen jälkeen jakaa erilaisiin koodisanayhdistelmiin. Edellisten kehysten kerrostustieto, toisin sanoen tieto, joka osoittaa mitkä koodisanat kuuluvat korkeamman prioriteetin tietoon, lähetetään kun virtuaalinen ennustuskehys muodostetaan. Tekniikan tasossa kooderien on valittava kehyksen kerrokseen jako kehyksen koodaamisen yhteydessä, ja tieto lähetetään kehyksen bittivirtausissa.

Kuvio 21 esittää graafisessa muodossa INTRA-koodatun kehyksen I0 ja INTER-koodatut kehykset P1, P2 ja P3 sisältävän videokuvasarjan osan dekodointiin. Tämä kuvio osoittaa suhteessa kuvioihin 19 ja 20 selostetun menettelyn vaikutuksen. Kuten kuvioista 21 käy ilmi, siinä on yläriivi, keskiriivi ja alariivi. Yläriivi vastaa uudelleen muodostettuja ja näytettyjä kehyksiä, keskiriivi vastaa jokaisen kehyksen bittivirtaa, ja alariivi vastaa muodostettuja virtuaalisia

ennustusviitekehyksiä. Nuolet osoittavat tietyn uudelleen muodostetun virtuaalikehyksen muodostamiseksi käytettyjä syöttölähteitä. Esimerkiksi kehys I0 luodaan vastaavasta bittivirrasta I0 B-S ja kehys P1 muodostetaan uudelleen käyttämällä kehystä I0 liikekompensoinnin viitteenä ja vastaanotettua P1:n bittivirtaa. Samoin virtuaalikehys I0' luodaan kehystä I0 vastaavan bittivirran osasta, ja keinokehys P1' luodaan käyttämällä I0':aa liikekompensoinnin viitteenä ja käyttämällä osaa P1:n bittivirrasta. Keksinnön mukaisesti kehys P3 luodaan käyttämällä virtuaalikehystä P2' liikekompensoinnin viitteenä ja P3:n bittivirtaa. Kehystä P2 ei käytetä liikekompensoinnin viitteenä.

10

On tilanteita, joissa datan jakaminen korkeampaan ja matalampaan prioriteettiin ei ole välttämätöntä. Jos esimerkiksi kuvaan liittyvä data kokonaisuudessaan mahtuu yhteen kehykseen, voi olla edullisempaa olla jakamatta dataa. Tässä tapauksessa dataa kokonaisuudessaan voidaan käyttää ennustamiseen virtuaalikehyksestä. Seuraavaksi viitataan kuvioon 21. Tässä nimenomaisessa suoritusmuodossa kehys P1' muodostetaan ennustamalla virtuaalikehyksestä I0' ja dekodamalla kaikki P1:n bittivirtatieto. Uudelleen muodostettu virtuaalikehys P1' ei ole vastaava kehyksen P1 kanssa, koska ennustusviite kehykselle P1 on I0, kun taas ennustusviite kehykselle P1' on I0'. Näin ollen P1' on virtuaalikehys, vaikka tässä tapauksessa se ennustetaan kehyksestä (P1), jossa on tietoa, jota ei ole priorisoitu korkeampaan ja matalampaan prioriteettiin.

15

20

Seuraavaksi selostetaan keksinnön erästä suoritusmuotoa viitaten kuvioon 22 sovellettuna kuviossa 11 esitettyä vastaavaan tilanteeseen. Liike- ja otsikkodata on erotettu ennustusvirhedatasta. Liike- ja otsikkodata on kapseloitu siirtopakettiin nimeltä liikepaketti (motion packet), ja ennustusvirhedata on kapseloitu siirtopakettiin nimeltä ennustusvirhepaketti (prediction error packet). Näin tapahtuu usean peräkkäisen koodatun kuvan kohdalla. Liikepaketeilla on korkea prioriteetti ja ne lähetetään aina mahdollisuuksien ja tarpeen mukaan uudelleen, koska virheen peittäminen onnistuu paremmin, jos dekooderi vastaanottaa liiketiedon oikein. (Liikepaketit myös parantavat kompressointitehokkuutta.) Kooderi erottaa liike- ja otsikkodatan kehyksistä P1-P3 ja muodostaa tästä tiedosta liikepaketin (M1-3). Ennustusvirhetieto P-kehyksille P1-P3 lähetetään erillisessä ennustusvirhepaketissa (PE1, PE2, PE3). Sen

25

30

sijaan, että kooderi käyttäisi I1:tä liikekompensointiviitteenä, se luo keinokehykset P1', P2' ja P3' I1:n ja M1-3:n perusteella. Toisin sanoen kooderi dekodaa I1:tä ja liikeosaa ennustuskehysistä P1, P2 ja P3 siten, että P2' ennustetaan P1':stä ja P3' ennustetaan P2':sta. Kehys P3':a käytetään kehyksen P4
 5 liikekompensointiviitteenä. Keinokehysiin P1', P2' ja P3' viitataan nollaennustusvirhekehysinä (Zero-Prediction-Error ZPE frames), koska ne eivät sisällä ennustusvirhedataa.

On huomattava, että kehys ja sen virtuaalinen vastine dekodataan käyttämällä
 10 eri osia tarjolla olevasta bittivirrasta. Normaalit (toisin sanoen näytetyt) kehykset käyttävät kaiken tarjolla olevan bittivirran ja virtuaalikehykset käyttävät vain osaa bittivirrasta. Virtuaalikehysten käyttämä osa on se osa bittivirtaa, joka on merkittävin kehyksen dekoodaamisessa. Lisäksi on edullisempaa, että virtuaalikehysten käyttämä osa on voimakkaimmin suojattu lähetysvirheiltä, ja
 15 näin ollen kyseisen osan lähettäminen todennäköisimmin onnistuu. Tällä tavalla keksintö pystyy lyhentämään ennustavaa koodausketjua ja perustamaan ennustetun kehyksen virtuaaliseen liikekompensointiviitekehykseen, joka luodaan bittivirran merkittävimmästä osasta, eikä liikekompensointiviitteeseen, joka luodaan käyttämällä merkittävin osa ja vähemmän merkittävää osaa.

20

Kun kuvioden 18, 19 ja 20 menettelyjä sovelletaan H.26L:ään, kuvat koodataan sisältäen kuvaotsikot. Tämä on korkeimman prioriteetin tietoa, johon viitataan yllä mainitussa luokittelumenetelmässä, koska koko kuvaa ei voida dekodata ilman kuvaotsikkoa. Jokainen kuvaotsikko sisältää kuvatyyppikentän (Ptype). Tietty
 25 arvo sisällytetään osoittamaan käyttääkö kuva yhtä vai useampia virtuaalisia viitekuvia. Jos Ptype-kentän arvo osoittaa, että on käytettävä yhtä tai useampaa virtuaalista viitekuvaa, kuvaotsikko sisältää tietoa siitä, kuinka viitekuva luodaan. Esillä olevan keksinnön muissa suoritusmuodoissa tämä tieto voi sisältyä viipaleotsikoihin, MB-otsikoihin ja/tai lohko-otsikoihin sen mukaan, millaista
 30 paketointia on käytetty. Lisäksi jos tietyn kehyksen koodaamisen yhteydessä käytetään useita viitekuvia, yksi tai useampi viitekuva voi olla virtuaalinen. Seuraavia signalointimenetelmiä käytetään:

1. Osoitus siitä, mitä menneen bittivirran kehyksistä käytetään luomaan viitekuva, annetaan lähetetyssä bittivirrassa. Kaksi arvoa lähetetään:

ajallisesti viimeistä kuvaa vastaava ja sitä ajallisesti aikaisinta kuvaa vastaava, jota käytetään ennustamiseen. Kuvioiden 18 ja 19 menettelyä olisi sopivasti muokattava tämän osoituksen käyttämistä varten.

2. Osoitus siitä, mitä koodausparametrejä käytetään viitekuvan luomiseen.
- 5 Bittivirrassa pitäisi olla osoitus ennustamiseen käytetystä suurimmasta prioriteettiluokasta. Jos bittivirrassa on esimerkiksi osoitus luokasta 4, ennustus muodostetaan luokkiin 1, 2, 3 ja 4 kuuluvista parametreistä. Tämä on yksinkertaistettu versio yleisemmästä menetelmästä, jossa käytetyt luokat pitäisi signaloida yksitellen.

10

- Kuvio 23 esittää keksinnön mukaista videokuvansiirtojärjestelmää 400. Järjestelmä käsittää viestivät videokuvapäätteet 402 ja 404. Tässä suoritusmuodossa esitetään päätteestä toiseen tapahtuva viestintä. Jossain toisessa suoritusmuodossa viestintä voi olla konfiguroitu tapahtumaan päätteeltä
- 15 palvelimelle tai palvelimelta päätteelle. Vaikka järjestelmä 400 on tarkoitettu mahdollistamaan kaksisuuntainen videokuvadatan lähettäminen bittivirran muodossa, se voi mahdollistaa vain yksisuuntaisen videokuvadatan lähettämisen. Yksinkertaisuuden vuoksi kuviossa 24 esitettyssä järjestelmässä 400 videokuvapäätte 402 on lähetävä (koodaava) videopäätte ja videopäätte 404
- 20 on vastaanottava (dekoodaava) videopäätte.

- Videopäätte 402 käsittää kooderin 410 ja lähetin-vastaanottimen 412. Kooderi 410 käsittää kokonaisen kehyksen kooderin 414, virtuaalikehyksen kooderin 416 ja kehyksen ennustimen 418. Lisäksi kooderi käsittää monikehyspuskurin 420
- 25 normaalien kehysten tallentamiseksi sekä monikehyspuskurin 422 virtuaalikehysten tallentamiseksi.

- Kokonaisen kehyksen kooderi 414 muodostaa bittivirran normaalista kehyksestä, joka sisältää tietoa sen myöhempää täydellistä uudelleenmuodostamista varten.
- 30 Se suorittaa kuvion 18 vaiheet 118–146 ja vaiheen 150. Kuitenkaan vaiheen 142 sisältävää haaraa ei seurata eikä tätä vaihetta käytetä. Tieto priorisoidaan korkeamman ja matalamman prioriteetin tietoon kuvion 18 vaiheen 148 mukaisesti.

Virtuaalikehyksen kooderi 416 määrittää virtuaalikehyksen normaalin kehyksen versiona, joka on muodostettu käyttäen normaalin kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa. Se suorittaa kuvion 18 vaiheet 160 ja 162.

Kehyksen ennustin 418 koodaa toisen normaalin kehyksen muodostamalla bittivirran, joka sisältää tietoa sen myöhempää täydellistä uudelleenmuodostamista varten. Se suorittaa kuvion 18 vaiheet 118–146 ja vaiheen 150. Kuitenkaan vaiheen 144 sisältävää haaraa ei seurata eikä tätä vaihetta käytetä. Näin ollen tämän toisen normaalin kehyksen koodaaminen suoritetaan vaiheiden 140 ja 142 mukaisesti niin, että se voidaan muodostaa kokonaan uudelleen vaiheen 146 mukaisesti virtuaalisen viitekehyksen perusteella eikä normaalin kehyksen perusteella.

Videopäätte 404 käsittää dekooderin 423 ja lähetin-vastaanottimen 424. Dekooderi 423 käsittää kokonaisen kehyksen dekooderin 425, virtuaalikehyksen dekooderin 426 ja kehysennustimen 428. Lisäksi dekooderi käsittää monikehyspuskurin 430 normaalien kehysten tallentamiseksi sekä monikehyspuskurin 432 virtuaalikehysten tallentamiseksi.

Kokonaisen kehyksen dekooderi 425 dekodaa normaalin kehyksen bittivirrasta, joka sisältää tietoa sen myöhempää täydellistä uudelleenmuodostamista varten. Se suorittaa kuvion 19 vaiheen 218 ja vaiheet 226–234. Bittivirran tieto oli aiemmin priorisoitu korkeamman ja matalamman prioriteetin tietoon.

Virtuaalikehyksen dekooderi 426 muodostaa virtuaalikehyksen ensimmäisen kokonaisen kehyksen bittivirrasta käyttäen ensimmäisen kokonaisen kehyksen korkeamman prioriteetin tietoa ensimmäisen kokonaisen kehyksen vähintään jonkun matalamman prioriteetin tiedon puuttuessa kuvion 19 vaiheiden 224, 238 ja 240 mukaisesti.

Kehysennustin 428 ennustaa toisen normaalin kehyksen kuvion 19 vaiheen 234 mukaisesti virtuaalikehyksen eikä normaalin kehyksen perusteella.

Videopääte 402 tuottaa koodatun videokuvabittivirran 434, jonka lähetin-vastaanotin 412 lähettää ja jonka lähetin-vastaanotin 424 ottaa vastaan sopivan lähetysmedian kautta. Keksinnön eräessä suoritusmuodossa lähetysmedia on langattoman viestintäjärjestelmän ilmarajapinta. Lähetin-vastaanotin 424 lähettää palautteen (feedback) 436 lähetin-vastaanottimelle 412. Tämän palautteen luonnetta on kuvattu edellä.

Seuraavaksi selostetaan ZPE-kehäksiä hyödyntävän videokuvansiirtojärjestelmän 500 toimintaa. Järjestelmä 500 esitetään kuviossa 24. Järjestelmässä 500 on lähetävä pääte 510 ja joukko vastaanottavia päätteitä 512 (joista vain yksi on esitetty), jotka viestivät siirtokanavan tai –verkon välityksellä. Lähetävä pääte 510 käsittää kooderin 514, paketoijan (packetiser) 516 ja lähettimen 518. Se käsittää myös TX-ZPE-dekooderin 520. Vastaanottavista päätteistä 512 jokainen käsittää vastaanottimen 522, paketoinnin purkajan (de-packetiser) 524 ja dekooderin 526. Niistä jokainen käsittää myös RX-ZPE-dekooderin 528. Kooderi 514 koodaa kompressoimatonta videokuvaa muodostaakseen kompressoituja videokuvia. Paketoija 516 kapseloi kompressoidut videokuvat siirtopaketteihin. Se voi järjestää kooderilta saadun tiedon uudelleen. Se myös tuottaa videokuvat, jotka eivät sisällä ennustusvirhetietoa liikekompensointia varten (nimeltään ZPE-bittivirta). TX-ZPE-dekooderi 520 on tavallinen videokuvadekooderi, jota käytetään ZPE-bittivirran dekodointiin. Lähetin 518 toimittaa paketit siirtokanavan tai –verkon välityksellä. Vastaanotin 522 vastaanottaa paketit siirtokanavalta tai –verkolta. Paketoinnin purkaja 524 purkaa siirtopaketit ja luo kompressoidut videokuvat. Jos joitakin paketteja katoaa lähetyksen aikana, paketoinnin purkaja 524 pyrkii peittämään häviöt kompressoiduissa videokuvissa. Lisäksi paketoinnin purkaja 524 tulostaa ZPE-bittivirtaa. Dekooderi 526 muodostaa kuvat uudelleen kompressoidusta videokuvabittivirrasta. RX-ZPE-dekooderi 528 on tavallinen videokuvadekooderi, jota käytetään ZPE-bittivirran dekodointiin.

Kooderi 514 toimii normaalisti lukuun ottamatta tapauksia, joissa paketoija 516 pyytää ZPE-kehystä käytettäväksi ennustusvirhetietoa. Silloin kooderi 514 muuttaa oletuksena olevan liikekompensointiviitekuvan TX-ZPE-dekooderin 520

		sä, L				(bps)	
Akiyo	8	50	17602	14	0.1%	158	0.9%
	10	53	12950	67	0.5%	262	2.0%
	13	55	9410	42	0.4%	222	2.4%
	15	59	7674	-2	0.0%	386	5.0%
	18	62	6083	24	0.4%	146	2.4%
	20	65	5306	7	0.1%	111	2.1%
Coastguard	8	16	107976	266	0.2%	1505	1.4%
	10	15	78458	182	0.2%	989	1.3%
	15	15	43854	154	0.4%	556	1.3%
	18	15	33021	187	0.6%	597	1.8%
	20	15	28370	248	0.9%	682	2.4%
Foreman	8	12	87741	173	0.2%	534	0.6%
	10	12	65309	346	0.5%	622	1.0%
	15	11	39711	95	0.2%	266	0.7%
	18	11	31718	179	0.6%	234	0.7%
	20	11	28562	-12	0.0%	-7	0.0%

Tulosten bittinopeuden kasvua osoittavista sarakkeista voidaan havaita, että nollaennustusvirhekehukset (Zero-Prediction-Error frames) parantavat kompressoititehokkuutta, kun käytetään viitekuvavalintaa (Reference Picture Selection).

5

Bittivirran erittäin tärkeä osa voidaan uudelleenmuodostaa ja käyttää bittivirran vähemmän tärkeän osan katoamisen tai vahingoittumisen kätkemiseen. Keksinnön eräs aspekti on valvoa virheen peittämistä yksikäsitteisellä tavalla.

10

Tämä tapahtuu siten, että kooderi signaloi dekooderille, mikä osa kehyksen bittivirrasta on riittävä hyväksyttävän kuvan tuottamiseen täysilaatuisen kuvan korvaamiseksi lähetysvirheen tai häviön sattuessa. Signointi voi sisältyä

bittivirtaan tai se voidaan lähettää erillään bittivirrasta. Dekooderi dekodaa erittäin tärkeän osan ja korvaa vähemmän tärkeän osan oletusarvoilla saadakseen näyttöön hyväksyttävän kuvan. Tätä periaatetta voidaan soveltaa kuvan osiin (viipaleisiin jne.) ja useisiin kuviin. Toisessa
 5 virheenpeittämismenetyelmässä kooderi voi signaloida dekooderille kuinka muodostetaan virtuaalinen keinotekoinen varaviitekuva, jota käytetään, jos varsinainen viitekuva katoaa tai on liian vahingoittunut, jotta sitä voisi käyttää.

On syytä huomata, että virtuaaliset ennustuskehykset eivät välttämättä edusta
 10 minkään sarjassa esiintyvän kompressoimattoman kuvan sisältöä. Tunnetuissa skaalattavuusmenetyelmissä ennustusviitekuvat ovat vastaavan alkuperäisen kuvan esitystapoja. Koska virtuaalisia ennustusviitekehyksiä ei ole tarkoitettu näytettäväksi, toisin kuin pohjakerros perinteisissä skaalattavuusmenetyelmissä, kooderin ei tarvitse välttämättä muodostaa hyväksyttävän laatuista virtuaalisia
 15 kuvia. Näin ollen keksinnön kompressoititehokkuus on lähellä yksikerroksista koodausmenetyelmään tehokkuutta.

On mahdollista poistaa tai dekodata sama tieto bittivirrasta erilaisilla tavoilla erilaisten virtuaalikehysten muodostamiseksi.

20

Esillä oleva keksintö parantaa koodaustehokkuutta viitekuvanvalintamenetyelmissä. Keksintö voidaan luokitella uudentyyppiseksi SNR-skaalattavuudeksi, toisin sanoen joustavammaksi kuin tekniikan tason mukaiset skaalattavuusmenetyelmät.

25

Keksinnön erityiset toteutukset ja suoritusmuodot on kuvattu edellä. Alan ammattimiehelle on selvää, että keksintö ei rajoitu yllä esitettyjen suoritusmuotojen yksityiskohtiin, vaan se voidaan vastaavia keinoja käyttäen toteuttaa muissa suoritusmuodoissa ilman, että keksinnön tunnusmerkeistä
 30 poiketaan. Keksinnön puitteita rajoittavat ainoastaan liitteenä olevat patenttivaatimukset.

1. Videokehysten koodaamismenetelmä koodattujen videokehysten tuottamiseksi (Kuvio 18), missä jokaista koodattua videokehystä edustaa bittivirta, ja koodatun videokehysten bittivirtailmentymä käsittää korkeamman ja matalamman prioriteetin tietoa (148), jolloin ainakin jonkin matalamman prioriteetin tiedon
- 5 dekodeeraus bittivirrasta riippuu ainakin jonkin korkeamman prioriteetin tiedon dekodeeraamisesta bittivirrasta, jossa menetelmässä:

10 koodataan ensimmäinen videokehys ensimmäisen koodatun videokehysten muodostamiseksi (130, 146), jota ensimmäistä koodattua videokehystä edustaa ensimmäinen bittivirta, joka sisältää ensimmäistä videokehystä edustavaa korkeamman ja matalamman prioriteetin tietoa,

muodostetaan ensimmäinen kokonainen viitekehys dekodeeraamalla korkeamman ja matalamman prioriteetin tieto ensimmäisestä bittivirrasta (157); ja

15 muodostetaan ensimmäinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekodeerattu ensimmäisestä bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa (160).

2. Patenttivaatimuksen 1 mukainen menetelmä, jossa lisäksi:

20 koodataan toinen videokehys toisen koodatun videokehysten muodostamiseksi (146), jota toista koodattua videokehystä edustaa toinen bittivirta;

muodostetaan toinen kokonainen viitekehys dekodeeraamalla korkeamman ja matalamman prioriteetin tieto toisesta bittivirrasta (157);

25 muodostetaan toinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekodeerattu toisesta bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa (160);

koodataan kolmas videokehys kolmannen koodatun videokehysten muodostamiseksi toisen virtuaalisen viitekehysten perusteella toisen kokonaisen viitekehysten sijaan (142, 146).

3. Patenttivaatimuksen 1 tai patenttivaatimuksen 2 mukainen menetelmä, joka käsittää lisäksi toisen videokehysten koodaamisen toisen koodatun videokehysten muodostamiseksi viitaten ensimmäiseen virtuaaliseen viitekehykseen (142, 146).

4. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, joka käsittää lisäksi seuraavan videokehysten koodauksen käyttämällä ajallista ennustamista suoraan edeltävästä virtuaalisesta viitekehyksestä suoraan edeltävän kokonaisen viitekehysten sijaan.

5. Patenttivaatimuksen 4 mukainen menetelmä, joka käsittää lisäksi vaiheen, jossa valitaan tietty viitekehys useiden vaihtoehtojen joukosta toisen kehysten ennustamista varten (140).

6. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, jossa jokainen videokehys assosioidaan joukkoon erilaisia virtuaalisia viitekehysiä, joista jokainen edustaa erilaista tapaa luokitella bittivirtaa videokehystä varten.

7. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, jossa ensimmäinen videokehys koodataan käyttämällä ajallista ennustamista suhteessa edeltävään kokonaiseen viitekehykseen.

8. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, jossa ensimmäinen videokehys koodataan käyttämällä ajallista ennustamista suhteessa edeltävään virtuaaliseen viitekehykseen.

9. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, joka käsittää lisäksi ensimmäisen kokonaisen viitekehysten muodostamisen käyttämällä korkeamman ja matalamman prioriteetin tietoa, joka on dekodattu

ensimmäisestä bittivirrasta ja viitaten edeltävään kokonaiseen viitekehykseen (144, 146).

10. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, joka käsittää lisäksi ensimmäisen virtuaalisen viitekehysten muodostamisen
5 käyttämällä korkeamman prioriteetin tietoa, joka on dekoodattu ensimmäisestä bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa ja viitaten edeltävään virtuaaliseen viitekehykseen.

11. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, joka käsittää lisäksi vaiheen, jossa muodostetaan koodattu videokehys yhden
10 laajempaan viitekehysjoukkoon kuuluvan viitekehysten perusteella.

12. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, joka käsittää lisäksi vaiheen, jossa tietyn viitekehysten valinta signaloidaan bittivirrassa.

13. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, joka
15 käsittää lisäksi sen, että ensimmäisessä bittivirrassa käytetään osoitinta, joka osoittaa korkeamman prioriteetin tiedon, jota käytetään ensimmäisen virtuaalisen viitekehysten muodostamisessa.

14. Minkä tahansa edeltävän patenttivaatimuksen mukainen menetelmä, joka käsittää lisäksi ensimmäisen virtuaalisen viitekehysten muodostamisen osittain
20 oletusarvojen perusteella.

15. Koodattujen videokehysten käsittelymenetelmä rekonstruoitujen videokehysten tuottamiseksi, jossa jokainen rekonstruoitu videokehys vastaa koodattua videokehystä (Kuvio 19), ja jokaista koodattua videokehystä edustaa bittivirta ja koodatun videokehysten bittivirtailmentymä käsittää korkeamman ja matalamman
25 prioriteetin tietoa (214), jolloin ainakin jonkin matalamman prioriteetin tiedon dekoodaaminen bittivirrasta riippuu ainakin jonkin korkeamman prioriteetin tiedon dekoodaamisesta bittivirrasta, jossa menetelmässä:

vastaanotetaan ensimmäinen koodattu videokehys, jossa ensimmäistä koodattua videokehystä edustaa ensimmäinen bittivirta, ensimmäisen bittivirran käsittäessä korkeamman ja matalamman prioriteetin tietoa (214);

5 rekonstruoidaan ensimmäinen videokehys dekodamalla korkeamman ja matalamman prioriteetin tieto ensimmäisestä bittivirrasta (218, 234);

tallennetaan ensimmäinen rekonstruoitu videokehys kokonaisena viitekehyyksenä (242); ja

10 muodostetaan ensimmäinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekodattu ensimmäisestä bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa (238).

16. Patenttivaatimuksen 15 mukainen menetelmä, jossa lisäksi:

vastaanotetaan toinen koodattu videokehys, jossa toista koodattua videokehystä edustaa toinen bittivirta, toisen bittivirran käsittäessä korkeamman ja matalamman prioriteetin tietoa (214);

15 rekonstruoidaan toinen videokehys dekodamalla korkeamman ja matalamman prioriteetin tieto toisesta bittivirrasta (218, 234);

tallennetaan toinen rekonstruoitu videokehys toisena kokonaisena viitekehyyksenä (242);

20 muodostetaan toinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekodattu toisesta bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa (238);

vastaanotetaan kolmas koodattu videokehys, jossa kolmatta koodattua videokehystä edustaa kolmas bittivirta, kolmannen bittivirran käsittäessä korkeamman ja matalamman prioriteetin tietoa (214); ja

rekonstruoidaan kolmas videokehys toisen virtuaalisen viitekehysten perusteella toisen kokonaisen viitekehysten sijaan (230, 234).

17. Patenttivaatimuksen 15 tai patenttivaatimuksen 16 mukainen menetelmä, joka käsittää lisäksi toisen videokehysten rekonstruoimisen toisen rekonstruoidun
5 videokehysten muodostamiseksi viitaten ensimmäiseen virtuaaliseen viitekehykseen (230, 234).

18. Minkä tahansa patenttivaatimuksen 15–17 mukainen menetelmä, joka käsittää lisäksi seuraavan videokehysten rekonstruoimisen käyttämällä ajallista ennustamista suoraan edeltävästä virtuaalisesta viitekehyksestä suoraan
10 edeltävän kokonaisen viitekehysten sijaan.

19. Minkä tahansa patenttivaatimuksen 15–18 mukainen menetelmä, joka käsittää lisäksi vaiheen, jossa valitaan tietty viitekehys useiden vaihtoehtojen joukosta toisen kehysten ennustamista varten.

20. Minkä tahansa patenttivaatimuksen 15–19 mukainen menetelmä, jossa
15 jokainen koodattu videokehys assosioidaan joukkoon erilaisia virtuaalisia viitekehysiksiä, joista jokainen edustaa erilaista tapaa tulkita bittivirtaa koodattua videokehystä varten.

21. Minkä tahansa patenttivaatimuksen 15–20 mukainen menetelmä, jossa ensimmäinen rekonstruoitu videokehys rekonstruoidaan käyttämällä ajallista
20 ennustamista suhteessa edeltävään kokonaiseen viitekehykseen.

22. Minkä tahansa patenttivaatimuksen 15–21 mukainen menetelmä, jossa ensimmäinen rekonstruoitu videokehys rekonstruoidaan käyttämällä ajallista ennustamista suhteessa edeltävään virtuaaliseen viitekehykseen.

23. Minkä tahansa patenttivaatimuksen 15–22 mukainen menetelmä, jossa lisäksi
25 ensimmäinen viitekehys rekonstruoidaan käyttämällä korkeamman ja matalamman prioriteetin tietoa, joka on dekodattu ensimmäisestä bittivirrasta ja viitaten edeltävään kokonaiseen viitekehykseen (232, 234).

24. Minkä tahansa patenttivaatimuksen 15–23 mukainen menetelmä, jossa muodostetaan lisäksi ensimmäinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekoodattu ensimmäisestä bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa ja viitaten edeltävään virtuaaliseen viitekehukseen.

25. Minkä tahansa patenttivaatimuksen 15–24 mukainen menetelmä, jossa lisäksi rekonstruoidaan koodattu videokehys yhden laajempaan viitekehysjoukkoon kuuluvan viitekehysten perusteella.

26. Minkä tahansa patenttivaatimuksen 15–25 mukainen menetelmä, joka käsittää lisäksi vaiheen, jossa bittivirrasta poistetaan viittaus tiettyyn viitekehukseen.

27. Minkä tahansa patenttivaatimuksen 15–26 mukainen menetelmä, jossa muodostetaan lisäksi ensimmäinen virtuaalinen viitekehys osittain oletusarvojen perusteella.

28. Videokehyksiä koodaava kooderi (410) koodattujen videokehysten tuottamiseksi, joita jokaista koodattua videokehystä edustaa bittivirta, ja koodatun videokehysten bittivirtailmentymä käsittää korkeamman ja matalamman prioriteetin tietoa, jolloin ainakin jonkin matalamman prioriteetin tiedon dekoodaaminen bittivirrasta riippuu ainakin jonkin korkeamman prioriteetin tiedon dekoodauksesta bittivirrasta,

20 kooderin ollessa järjestetty

koodaamaan ensimmäinen videokehys ensimmäisen koodatun videokehysten muodostamiseksi (414), jossa ensimmäistä koodattua videokehystä edustaa ensimmäinen bittivirta, joka sisältää ensimmäistä videokehystä edustavaa korkeamman ja matalamman prioriteetin tietoa;

25 muodostamaan ensimmäinen kokonainen viitekehys dekoodaamalla korkeamman ja matalamman prioriteetin tieto ensimmäisestä bittivirrasta (420); ja

muodostamaan ensimmäinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekoodattu ensimmäisestä bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa (422).

29. Patenttivaatimuksen 28 mukainen kooderi (410), jossa kooderi on järjestetty

5 koodaamaan toinen videokehys toisen koodatun videokehysten muodostamiseksi, jossa toista koodattua videokehystä edustaa toinen bittivirta (414);

muodostamaan toinen kokonainen viitekehys dekoodaamalla korkeamman ja matalamman prioriteetin tietoa toisesta bittivirrasta (420);

10 muodostamaan toinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekoodattu toisesta bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa (422);

15 koodaamaan kolmas videokehys kolmannen koodatun videokehysten muodostamiseksi toisen virtuaalisen viitekehysten perusteella toisen kokonaisen viitekehysten sijaan (414).

30. Patenttivaatimuksen 28 tai patenttivaatimuksen 29 mukainen kooderi (410), jossa kooderi on järjestetty koodaamaan toinen videokehys toisen koodatun videokehysten muodostamiseksi viitaten ensimmäiseen virtuaaliseen viitekehykseen.

20 31. Minkä tahansa patenttivaatimuksen 28–30 mukainen kooderi (410), jossa kooderi on järjestetty koodaamaan seuraava videokehys käyttämällä ajallista ennustamista suoraan edeltävästä virtuaalisesta viitekehyksestä suoraan edeltävän kokonaisen viitekehysten sijaan.

25 32. Patenttivaatimuksen 31 mukainen kooderi (410), jossa kooderi on järjestetty valitsemaan tietty viitekehys useiden vaihtoehtojen joukosta toisen kehyksen ennustamiseksi.

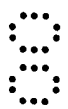
33. Minkä tahansa patenttivaatimuksen 28–32 mukainen kooderi (410), jossa kooderi on järjestetty assosioimaan kukin videokehys joukkoon erilaisia virtuaalisia viitekehyksiä, joista jokainen edustaa erilaista tapaa tulkita bittivirtaa koodattua videokehystä varten.

5 34. Minkä tahansa patenttivaatimuksen 28–33 mukainen kooderi (410), jossa kooderi on järjestetty koodaamaan ensimmäinen videokehys käyttämällä ajallista ennustamista suhteessa edeltävään kokonaiseen viitekehykseen.

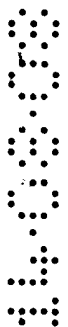
10 35. Minkä tahansa patenttivaatimuksen 28–34 mukainen kooderi (410), jossa kooderi on järjestetty koodaamaan ensimmäinen videokehys käyttämällä ajallista ennustamista suhteessa edeltävään virtuaaliseen viitekehykseen.

36. Minkä tahansa patenttivaatimuksen 28–35 mukainen kooderi (410), jossa kooderi on järjestetty muodostamaan ensimmäinen kokonainen viitekehys käyttämällä korkeamman ja matalamman prioriteetin tietoa, joka on dekodattu ensimmäisestä bittivirrasta ja viitaten edeltävään kokonaiseen viitekehykseen.

15 37. Minkä tahansa patenttivaatimuksen 20–36 mukainen kooderi (410), jossa kooderi on järjestetty muodostamaan ensimmäinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekodattu ensimmäisestä bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa ja viitaten edeltävään virtuaaliseen viitekehykseen.



20 38. Minkä tahansa patenttivaatimuksen 28–37 mukainen kooderi (410), jossa kooderi on järjestetty muodostamaan koodattu videokehys yhden laajempaan viitekehysjoukkoon kuuluvan viitekehyksen perusteella.



39. Minkä tahansa patenttivaatimuksen 28–38 mukainen kooderi (410), jossa kooderi on järjestetty signaloimaan tietyn viitekehyksen valinta bittivirrassa.

25 40. Minkä tahansa patenttivaatimuksen 28-39 mukainen kooderi (410), jossa kooderi on järjestetty käyttämään ensimmäisessä bittivirrassa osoitinta, joka

osoittaa ensimmäisen virtuaalisen viitekehysten muodostamisessa käytetyn korkeamman prioriteetin tiedon.

41. Minkä tahansa patenttivaatimuksen 28–40 mukainen kooderi (410), jossa kooderi on järjestetty muodostamaan ensimmäinen virtuaalinen viitekehys osittain oletusarvojen perusteella.

42. Koodattuja videokehymiä käsittelevä dekooderi (423), jolla tuotetaan rekonstruoituja videokehymiä, joista jokainen rekonstruoitu videokehys vastaa koodattua videokehystä, jokaista koodattua videokehystä edustaa bittivirta, koodatun videokehysten bittivirtailmentymän sisältäessä korkeamman ja matalamman prioriteetin tietoa, jolloin ainakin jonkin matalamman prioriteetin tiedon dekoodaaminen bittivirrasta riippuu ainakin jonkin korkeamman prioriteetin tiedon dekoodauksesta bittivirrasta,

dekooderin ollessa järjestetty

vastaanottamaan ensimmäinen koodattu videokehys, jossa ensimmäistä koodattua videokehystä edustaa ensimmäinen bittivirta, ensimmäisen bittivirran käsittäessä korkeamman ja matalamman prioriteetin tietoa (434);

rekonstruoimaan ensimmäinen videokehys dekoodaamalla korkeamman ja matalamman prioriteetin tieto ensimmäisestä bittivirrasta (425);

tallentamaan ensimmäinen rekonstruoitu videokehys kokonaisena viitekehysenä (430); ja

muodostamaan ensimmäinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekodattu ensimmäisestä bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa (426).

43. Patenttivaatimuksen 42 mukainen dekooderi (423), jossa dekooderi on järjestetty

vastaanottamaan toinen koodattu videokehys, jossa toista koodattua videokehystä edustaa toinen bittivirta, toisen bittivirran sisältäessä korkeamman ja matalamman prioriteetin tietoa (434);

5 rekonstruoimaan toinen videokehys dekodamalla korkeamman ja matalamman prioriteetin tieto toisesta bittivirrasta (425);

tallentamaan toinen rekonstruoitu videokehys toisena kokonaisena viitekehyyksenä (430);

10 muodostamaan toinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekodattu toisesta bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa (426);

vastaanottamaan kolmas koodattu videokehys, jossa kolmatta koodattua videokehystä edustaa kolmas bittivirta, kolmannen bittivirran käsittäessä korkeamman ja matalamman prioriteetin tietoa (434); ja

15 rekonstruoimaan kolmas videokehys toisen virtuaalisen viitekehyyksen perusteella toisen kokonaisen viitekehyyksen sijaan (425).

44. Patenttivaatimuksen 42 tai patenttivaatimuksen 43 mukainen dekooderi (423), jossa dekooderi on järjestetty rekonstruoimaan toinen videokehys toisen rekonstruoidun videokehyyksen muodostamiseksi viitaten ensimmäiseen virtuaaliseen viitekehyykseen.

20 45. Minkä tahansa patenttivaatimuksen 42–44 mukainen dekooderi (423), jossa dekooderi on järjestetty rekonstruoimaan seuraava videokehys käyttämällä ajallista ennustamista suoraan edeltävästä virtuaalisesta viitekehyyksestä suoraan edeltävän kokonaisen viitekehyyksen sijaan.

25 46. Patenttivaatimuksen 42–45 mukainen dekooderi (423), jossa dekooderi on järjestetty valitsemaan tietty viitekehys useiden vaihtoehtojen joukosta toisen kehyksen ennustamiseksi.

47. Minkä tahansa patenttivaatimuksen 42–45 mukainen dekooderi (423), jossa dekooderi on järjestetty assosioimaan kukin koodattu videokehys joukkoon erilaisia virtuaalisia viitekehyksiä, joista jokainen edustaa erilaista tapaa tulkita bittivirtaa koodattua videokehystä varten.

5 48. Minkä tahansa patenttivaatimuksen 42–47 mukainen dekooderi (423), jossa dekooderi on järjestetty rekonstruoimaan ensimmäinen rekonstruoitu videokehys käyttämällä ajallista ennustamista suhteessa edeltävään kokonaiseen viitekehykseen.

10 49. Minkä tahansa patenttivaatimuksen 42–48 mukainen dekooderi (423), jossa dekooderi on järjestetty rekonstruoimaan ensimmäinen rekonstruoitu videokehys käyttämällä ajallista ennustamista suhteessa edeltävään virtuaaliseen viitekehykseen.

15 50. Minkä tahansa patenttivaatimuksen 42–49 mukainen dekooderi (423), jossa dekooderi on järjestetty rekonstruoimaan ensimmäinen viitekehys käyttämällä korkeamman ja matalamman prioriteetin tietoa, joka on dekodattu ensimmäisestä bittivirrasta ja viitaten edeltävään kokonaiseen viitekehykseen.

20 51. Minkä tahansa patenttivaatimuksen 42–50 mukainen dekooderi (423), jossa dekooderi on järjestetty muodostamaan ensimmäinen virtuaalinen viitekehys käyttämällä korkeamman prioriteetin tietoa, joka on dekodattu ensimmäisestä bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa ja viitaten edeltävään virtuaaliseen viitekehykseen.

52. Minkä tahansa patenttivaatimuksen 42–51 mukainen dekooderi (423), jossa dekooderi on järjestetty rekonstruoimaan koodattu videokehys yhden laajempaan viitekehysjoukkoon kuuluvan viitekehymuksen perusteella.

25 53. Minkä tahansa patenttivaatimuksen 42–52 mukainen dekooderi (423), jossa dekooderi on järjestetty poistamaan bittivirrasta osoitin tiettyyn viitekehykseen.

54. Minkä tahansa patenttivaatimuksen 42–53 mukainen dekooderi (423), jossa dekooderi on järjestetty poistamaan bittivirrasta osoitin tiettyyn viitekehukseen.

55. Minkä tahansa patenttivaatimuksen 42–54 mukainen dekooderi (423), jossa dekooderi on järjestetty muodostamaan ensimmäinen virtuaalinen viitekehys osittain oletusarvojen perusteella.

56. Videokuvaviestintäpääte (402), joka käsittää ensimmäisen ja toisen videokehysten koodaavan videokooderin (410), joka videokooderi käsittää

10 kehyksen kooderin (414), joka on järjestetty koodaamaan ensimmäinen videokehys ensimmäisen koodatun videokehysten muodostamiseksi, jossa ensimmäistä videokehystä edustaa bittivirta; ensimmäisen koodatun videokehysten bittivirtailmentymä sisältää korkeamman ja matalamman prioriteetin tietoa (434), jossa ainakin jonkin matalamman prioriteetin tiedon dekodointi bittivirrasta riippuu vähintään jonkin korkeamman prioriteetin dekodointi bittivirrasta, ja kehyksen kooderi on järjestetty
15 lisäksi dekodointiin ensimmäinen koodattu videokehys käyttämällä ensimmäisen koodatun videokehysten korkeamman ja matalamman prioriteetin tietoa; ja

virtuaalinen kehyksenmuodostin (416), joka on järjestetty dekodointiin ensimmäinen koodattu videokehys käyttämällä ensimmäisen koodatun videokehysten korkeamman prioriteetin tietoa, ainakin jonkin ensimmäisen koodatun videokehysten matalamman prioriteetin tiedon puuttuessa, ensimmäisen virtuaalisen viitekehysten muodostamiseksi;

jossa kehyksen kooderi on järjestetty koodaamaan toinen videokehys ensimmäisen virtuaalisen viitekehysten perusteella.

25 57. Videokuvaviestintäpääte (404), joka käsittää videodekooderin (423), jolla dekodataan ensimmäinen ja toinen koodattu videokehys, jossa jokaista koodattua videokehystä edustaa bittivirta, ja koodatun videokehysten bittivirtailmentymä sisältää korkeamman ja matalamman prioriteetin tietoa (434),

jolloin ainakin jonkin matalamman prioriteetin tiedon dekoddaaminen bittivirrasta riippuu ainakin jonkin korkeamman prioriteetin tiedon dekodauksesta bittivirrasta, joka videodekooderi käsittää:

5 kokonaisen kehyksen dekooderin (425), joka on järjestetty dekoddaamaan ensimmäinen koodattu videokehys ensimmäisen koodatun videokehyksen korkeamman ja matalamman prioriteetin tiedon perusteella kokonaisen viitekehyksen muodostamiseksi;

10 virtuaalisen kehyksen dekoodaajan (426), joka on järjestetty dekoddaamaan ensimmäinen koodattu videokehys käyttämällä ensimmäisen koodatun videokehyksen korkeamman prioriteetin tietoa, ainakin jonkin ensimmäisen koodatun videokehyksen matalamman prioriteetin tiedon puuttuessa, ensimmäisen virtuaalisen viitekehyksen muodostamiseksi;

15 jossa kokonainen kehyksen dekooderi on lisäksi järjestetty dekoddaamaan toinen koodattu videokehys ensimmäisen virtuaalisen viitekehyksen perusteella.

58. Tietokoneohjelma tietokoneen videokuvakooderina (410) käyttämistä varten ensimmäisen ja toisen videokehyksen koodaamiseksi, joista jokaista koodattua videokehystä edustaa bittivirta, ja koodatun videokehyksen bittivirtailmentymä käsittää korkeamman ja matalamman prioriteetin tietoa (434), jolloin ainakin jonkin matalamman prioriteetin tiedon dekoddaaminen bittivirrasta riippuu ainakin jonkin korkeamman prioriteetin tiedon dekodauksesta bittivirrasta, joka tietokoneohjelma käsittää:

25 tietokoneohjelman koodin ensimmäisen videokehyksen koodaamista varten ensimmäisen koodatun videokehyksen muodostamiseksi, jossa ensimmäinen koodattu videokehys käsittää korkeamman ja matalamman prioriteetin tietoa, ensimmäisen kokonaisen viitekehyksen muodostamiseksi;

tietokoneohjelman koodin ensimmäisen koodatun videokehyksen dekodaaamiseksi käyttämällä ensimmäisen koodatun videokehyksen korkeamman ja matalamman prioriteetin tietoa;

5 tietokoneohjelman koodin ensimmäisen koodatun videokehyksen dekodaaamiseksi käyttämällä ensimmäisen koodatun videokehyksen korkeamman prioriteetin tietoa, ainakin jonkin ensimmäisen koodatun videokehyksen matalamman prioriteetin tiedon puuttuessa, ensimmäisen virtuaalisen viitekehyksen muodostamiseksi; ja

10 tietokoneohjelman koodin toisen videokehyksen koodaamiseksi ensimmäisen virtuaalisen viitekehyksen perusteella.

59. Tietokoneohjelma tietokoneen videokuvadekooderina (423) käyttämistä varten ensimmäisen ja toisen koodatun videokehyksen dekodaaamiseksi, joista jokaista koodattua videokehystä edustaa bittivirta, ja koodatun videokehyksen bittivirtailmentymä sisältää korkeamman ja matalamman prioriteetin tietoa (434),
15 jolloin ainakin jonkin matalamman prioriteetin tiedon dekodaaaminen bittivirrasta riippuu ainakin jonkin korkeamman prioriteetin tiedon dekodauksesta bittivirrasta, joka tietokoneohjelma käsittää:

20 tietokoneohjelman koodin ensimmäisen koodatun videokehyksen dekodaaamiseksi ainakin osittain ensimmäisen koodatun videokehyksen korkeamman ja matalamman prioriteetin tiedon perusteella;

25 tietokoneohjelman koodin ensimmäisen koodatun videokehyksen dekodaaamiseksi käyttämällä ensimmäisen koodatun videokehyksen korkeamman prioriteetin tietoa, ainakin jonkin ensimmäisen koodatun videokehyksen matalamman prioriteetin tiedon puuttuessa, ensimmäisen virtuaalisen viitekehyksen muodostamiseksi;

tietokoneohjelman koodin toisen koodatun videokehyksen dekodaaamiseksi ensimmäisen virtuaalisen viitekehyksen perusteella.

60. Koodatun videokehysten bittivirtailmentymä (434), jossa koodattu videokehys käsittää korkeamman ja matalamman prioriteetin tietoa, jossa ainakin jonkin matalamman prioriteetin tiedon dekodaus bittivirrasta riippuu ainakin jonkin korkeamman prioriteetin tiedon dekodauksesta bittivirrasta; bittivirta käsittää
- 5 dekooderin käyttämän osoittimen, jolla määritetään, mitä korkeamman prioriteetin tietoa kooderi on käyttänyt virtuaalisen viitekehysten muodostamiseksi, jossa mainittu virtuaalinen viitekehys on kooderin muodostama käyttäen korkeamman prioriteetin tietoa, joka on dekodattu bittivirrasta ainakin jonkin matalamman prioriteetin tiedon puuttuessa.



Patentkrav

1. Förfarande för kodning av videoramar för att producera kodade videoramar (Figur 18), vid vilket varje kodad videoram representeras av en bitström, och
 5 biströmsrepresentationen av en kodad videoram omfattar information (148) av högre och lägre prioritet, varvid dekodning av åtminstone en del av informationen av lägre prioritet från en bitströmm beror på dekodning av åtminstone en del av informationen av högre prioritet från bitströmmen, vid vilket förfarande

10 - kodas en första videoram för att skapa en första kodad videoram (130, 146), vilken första kodad videoram representeras av en första bitström med information av högre och lägre prioritet som representerar den första videoramen,

15 - skapas en första komplett referensram genom att dekodera informationen av högre och lägre prioritet från den första bitströmmen (157); och

- skapas en första virtuell referensram med användande av informationen av högre prioritet som dekodats från den första bitströmmen i frånvaro av
 20 åtminstone en del av informationen av lägre prioritet (160).

2. Förfarande enligt patentkrav 1, vid vilket dessutom

25 - kodas en andra videoram för att skapa en andra kodad videoram (146), vilken andra kodad videoram representeras av en andra bitström;

- skapas en andra komplett referensram genom att dekodera informationen av högre och lägre prioritet från den andra bitströmmen (157);

30 - skapas en andra virtuell referensram med användande av informationen av högre prioritet som dekodats från den andra bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet (160);

- kodas en tredje videoram för att skapa en tredje kodad videoram på basis av den andra virtuella referensramen i stället för den andra kompletta referensramen (142, 146).

5 3. Förfarande enligt patentkrav 1 eller patentkrav 2, vilket förfarande dessutom omfattar kodning av en andra videoram för att skapa en andra kodad videoram med hänvisning till den första virtuella referensramen (142, 146).

10 4. Förfarande enligt något av föregående patentkrav, vilket förfarande dessutom omfattar kodning av en påföljande videoram med användande tidsmässig prediktion från en direkt föregående virtuella referensram i stället för en direkt föregående komplett referensram.

15 5. Förfarande enligt patentkrav 4, vilket förfarande dessutom omfattar ett steg där väljs en viss referensram bland flera alternativ för prediktion av en annan ram (140).

20 6. Förfarande enligt något av föregående patentkrav, vid vilket förfarande associeras varje videoram med en mängd av olika virtuella referensramar som var och en representerar ett olika sätt att klassificera bitströmmen för videoramen.

25 7. Förfarande enligt något av föregående patentkrav, vid vilket förfarande kodas den första videoramen med användande av tidsmässig prediktion i förhållande till en föregående komplett referensram.

8. Förfarande enligt något av föregående patentkrav, vid vilket förfarande kodas den första videoramen med användande av tidsmässig prediktion i förhållande till en föregående virtuell referensram.

30 9. Förfarande enligt något av föregående patentkrav, vilket förfarande dessutom omfattar skapande av den första kompletta videoramen med användande av informationen av högre och lägre prioritet som dekodats från den första bitströmmen och med hänvisning till en föregående komplett videoram (144, 146).

10. Förfarande enligt något av föregående patentkrav, vilket förfarande dessutom omfattar skapande av den första virtuella referensramen med användande av informationen av högre prioritet som dekodats från den första bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet och med hänvisning till en föregående virtuell referensram.

11. Förfarande enligt något av föregående patentkrav, vilket förfarande dessutom omfattar ett steg där skapas en kodad videoram på basis av en referensram tillhörande en större mängd av referensramar.

12. Förfarande enligt något av föregående patentkrav, vilket förfarande dessutom omfattar ett steg där signaleras valet av en viss referensram i bitströmmen.

13. Förfarande enligt något av föregående patentkrav, vilket förfarande dessutom omfattar att i den första bitströmmen används en indikator som indicerar informationen av högre prioritet som används vid skapandet av den första virtuella referensramen.

14. Förfarande enligt något av föregående patentkrav, vilket förfarande dessutom omfattar skapande av den första virtuella referensramen delvis på basis av förinställda värden.

15. Förfarande för behandling av kodade videoramar för att producera rekonstruerade videoramar, vid vilket varje rekonstruerad videoram motsvarar en kodad videoram (Figur 19), och varje kodad videoram representeras av en bitström och bitströmsrepresentationen av en kodad videoram omfattar information (214) av högre och lägre prioritet, varvid dekodning av åtminstone en del av informationen av lägre prioritet från en bitström beror på dekodning av åtminstone en del av informationen av högre prioritet från bitströmmen, vid vilket förfarande

- mottas en första kodad videoram, varvid den första kodade videoramen representeras av en första bitström, varvid den första bitströmmen omfattar information (214) av högre och lägre prioritet;

5 - rekonstrueras den första videoramen genom att dekodera informationen av högre och lägre prioritet från den första bitströmmen (218, 234);

- lagras den första rekonstruerade videoramen som komplett referensram (242);
och

10

- skapas en första virtuell referensram med användande av informationen av högre prioritet som dekodats från den första bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet (238).

15 16. Förfarande enligt patentkrav 15, vid vilket förfarande dessutom

- mottas en andra kodad videoram, varvid den andra kodade videoramen representeras av en andra bitström, varvid den andra bitströmmen omfattar information (214) av högre och lägre prioritet;

20

- rekonstrueras den andra videoramen genom att dekodera informationen av högre och lägre prioritet från den andra bitströmmen (218, 234);

- lagras den andra rekonstruerade videoramen som andra komplett referensram (242);

25

- skapas en andra virtuell referensram med användande av informationen av högre prioritet som dekodats från den andra bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet (238);

30

- mottas en tredje kodad videoram, varvid den tredje kodade videoramen representeras av en tredje bitström, varvid den tredje bitströmmen omfattar information (214) av högre och lägre prioritet; och

- rekonstrueras den tredje videoramen på basis av den andra virtuella referensramen i stället för den andra kompletta referensramen (230, 234).

5 17. Förfarande enligt patentkrav 15 eller patentkrav 16, vilket förfarande dessutom omfattar rekonstruktion av en andra videoram för skapande av en andra rekonstruerad videoram med hänvisning till den första virtuella referensramen (230, 234).

10 18. Förfarande enligt något av patentkraven 15 - 17, vilket förfarande dessutom omfattar rekonstruktion av en påföljande videoram med användande av tidsmässig prediktion från en direkt föregående virtuell referensram i stället för en direkt föregående komplett referensram.

15 19. Förfarande enligt något av patentkraven 15 - 18, vilket förfarande dessutom omfattar ett steg där väljs en viss referensram bland flera alternativ för prediktion av en annan ram.

20 20. Förfarande enligt något av patentkraven 15 - 19, vid vilket förfarande associeras varje kodad videoram med en mängd av olika virtuella referensramar som var och en representerar ett olika sätt att tolka bitströmmen för den kodade videoramen.

25 21. Förfarande enligt något av patentkraven 15 - 20, vid vilket förfarande rekonstrueras den första rekonstruerade videoramen med användande av tidsmässig prediktion i förhållande till en föregående komplett referensram.

30 22. Förfarande enligt något av patentkraven 15 - 21, vid vilket förfarande rekonstrueras den första rekonstruerade videoramen med användande av tidsmässig prediktion i förhållande till en föregående virtuell referensram.

23. Förfarande enligt något av patentkraven 15 - 22, vid vilket förfarande dessutom rekonstrueras den första referensramen med användande av

informationen av högre och lägre prioritet som dekodats från den första bitströmmen och med hänvisning till en föregående komplett referensram.

24. Förfarande enligt något av patentkraven 15 - 23, vid vilket förfarande
5 dessutom skapas den första virtuella referensramen med användande av informationen av högre prioritet som dekodats från den första bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet och med hänvisning till en föregående virtuell referensram.

10 25. Förfarande enligt något av patentkraven 15 - 24, vid vilket förfarande dessutom rekonstrueras en kodad videoram på basis av en referensram tillhörande en större mängd av referensramar.

15 26. Förfarande enligt något av patentkraven 15 - 25, vilket förfarande dessutom omfattar ett steg där avlägsnas en indikation av en viss referensram från bitströmmen.

20 27. Förfarande enligt något av patentkraven 15 - 26, vid vilket förfarande dessutom skapas en första virtuell referensram delvis på basis av förinställda värden.

25 28. Kodare (410) som kodar videoramar för att producera kodade videoramar, varvid varje kodad videoram representeras av en bitström, och biströmsrepresentationen av en kodad videoram omfattar information av högre och lägre prioritet, varvid dekodning av åtminstone en del av informationen av lägre prioritet från bitströmmen beror på dekodning av åtminstone en del av informationen av högre prioritet från bitströmmen, varvid kodaren är anordnad att

30 - koda en första videoram för att skapa en första kodad videoram (414), varvid den första kodade videoramen representeras av en första bitström med information av högre och lägre prioritet som representerar den första videoramen;

- skapa en första komplett referensram genom att dekodera informationen av högre och lägre prioritet från den första bitströmmen (420); och
- att skapa en första virtuell referensram med användande av informationen av högre prioritet som dekodats från den första bitströmmen i frånvaro av
5 åtminstone en del av informationen av lägre prioritet (422).

29. Kodare (410) enligt patentkrav 28, varvid kodaren är anordnad att

- 10 - koda en andra videoram för att skapa en andra kodad videoram, varvid den andra kodade videoramen representeras av en andra bitström (414);
- skapa en andra komplett referensram genom att dekodera informationen av högre och lägre prioritet från den andra bitströmmen (420);
15
- skapa en andra virtuell referensram med användande av informationen av högre prioritet som dekodats från den andra bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet (422);
- 20 - koda en tredje videoram för att skapa en tredje kodad videoram på basis av den andra virtuella referensramen i stället för den andra kompletta referensramen (414).

30. Kodare (410) enligt patentkrav 28 eller patentkrav 29, varvid kodaren är
25 anordnad att koda en andra videoram för att skapa en andra kodad videoram med hänvisning till den första virtuella referensramen.

31. Kodare (410) enligt något av patentkraven 28 - 30, varvid kodaren är anordnad
30 att koda en påföljande videoram med användande av tidsmässig prediktion från en direkt föregående virtuell referensram i stället för en direkt föregående komplett referensram.

32. Kodare (410) enligt något av patentkrav 31, varvid kodaren är anordnad att välja en viss referensram bland flera alternativ för prediktion av en annan ram.

5 33. Kodare (410) enligt något av patentkraven 28 - 32, varvid kodaren är anordnad att associera varje videoram med en mängd av olika virtuella referensramar som var och en representerar ett olika sätt att tolka bitströmmen för den kodade videoramen.

10 34. Kodare (410) enligt något av patentkraven 28 - 33, varvid kodaren är anordnad att koda den första videoramen med användande av tidsmässig prediktion i förhållande till en föregående komplett referensram.

15 35. Kodare (410) enligt något av patentkraven 28 - 34, varvid kodaren är anordnad att koda den första videoramen med användande av tidsmässig prediktion i förhållande till en föregående virtuell referensram.

20 36. Kodare (410) enligt något av patentkraven 28 - 35, varvid kodaren är anordnad att skapa den första kompletta referensramen med användande av informationen av högre och lägre prioritet som dekodats från den första bitströmmen och med hänvisning till en föregående komplett referensram.

25 37. Kodare (410) enligt något av patentkraven 28 - 36, varvid kodaren är anordnad att skapa den första virtuella referensramen med användande av informationen av högre prioritet som dekodats från den första bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet och med hänvisning till en föregående virtuell referensram.

30 38. Kodare (410) enligt något av patentkraven 28 - 37, varvid kodaren är anordnad att skapa en kodad videoram på basis av en referensram tillhörande till en större mängd av referensramar.

39. Kodare (410) enligt något av patentkraven 28 - 38, varvid kodaren är anordnad att signalera valet av en viss referensram i bitströmmen.

40. Kodare (410) enligt något av patentkraven 28 - 39, varvid kodaren är anordnad att i den första bitströmmen använda en indikator som indicerar den vid skapandet av den första virtuella referensramen använda informationen av högre prioritet.

5 41. Kodare (410) enligt något av patentkraven 28 - 40, varvid kodaren är anordnad att skapa den första virtuella referensramen delvis på basis av förinställda värden.

42. Dekodare (423) som behandlar kodade videoramar för att producera rekonstruerade videoramar som var och en rekonstruerad videoram motsvarar en
10 kodad videoram, varvid varje kodad videoram representeras av en bitström, varvid biströmsrepresentationen av en kodad videoram omfattar information av högre och lägre prioritet, varvid dekodning av åtminstone en del av informationen av lägre prioritet från bitströmmen beror på dekodning av åtminstone en del av informationen av högre prioritet från bitströmmen, varvid dekodaren är anordnad
15 att

- mottaga en första kodad videoram, varvid den första kodade videoramen representeras av en första bitström, varvid den första bitströmmen omfattar information av högre och lägre prioritet (434);

20

- rekonstruera den första videoramen genom att dekoda informationen av högre och lägre prioritet från den första bitströmmen (425);

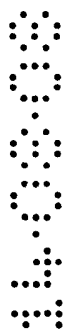
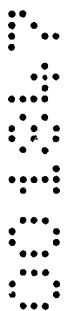
- lagra den första rekonstruerade videoramen som komplett referensram (430);
och

25

- att skapa en första virtuell referensram med användande av informationen av högre prioritet som dekodats från den första bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet (426).

30

43. Dekodare (423) enligt patentkrav 42, varvid dekodaren är anordnad att



- mottaga en andra kodad videoram, varvid den andra kodade videoramen representeras av en andra bitström, varvid den andra bitströmmen omfattar information av högre och lägre prioritet (434);
- 5
- rekonstruera den andra videoramen genom att dekodera informationen av högre och lägre prioritet från den andra bitströmmen (425);
 - lagra den andra rekonstruerade videoramen som komplett referensram (430);
- 10
- skapa en andra virtuell referensram med användande av informationen av högre prioritet som dekodats från den andra bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet (426);
 - mottaga en tredje kodad videoram, varvid den tredje kodade videoramen representeras av en tredje bitström, varvid den tredje bitströmmen omfattar information av högre och lägre prioritet (434); och att
- 15
- rekonstruera den tredje videoramen på basis av den andra virtuella referensramen i stället för den andra kompletta referensramen (425).
- 20
44. Dekodare (423) enligt patentkrav 42 eller patentkrav 43, varvid dekodaren är anordnad att rekonstruera en andra videoram för att skapa en andra rekonstruerad videoram med hänvisning till den första virtuella referensramen.
- 25
45. Dekodare (423) enligt något av patentkraven 42 - 44, varvid dekodaren är anordnad att rekonstruera en påföljande videoram med användande av tidsmässig prediktion från en direkt föregående virtuell referensram i stället för en direkt föregående komplett referensram.
- 30
46. Dekodare (423) enligt något av patentkraven 42 - 45, varvid dekodaren är anordnad att välja en viss referensram bland flera alternativ för prediktion av en annan videoram.

47. Dekodare (423) enligt något av patentkraven 42 - 46, varvid dekodaren är anordnad att associera varje kodad videoram med en mängd av olika virtuella referensramar som var och en representerar ett olika sätt att tolka bitströmmen för den kodade videoramen.

5

48. Dekodare (423) enligt något av patentkraven 42 - 47, varvid dekodaren är anordnad att rekonstruera den första rekonstruerade videoramen med användande av tidsmässig prediktion i förhållande till en föregående komplett referensram.

10

49. Dekodare (423) enligt något av patentkraven 42 - 48, varvid dekodaren är anordnad att rekonstruera den första rekonstruerade videoramen med användande av tidsmässig prediktion i förhållande till en föregående virtuell referensram.

15

50. Dekodare (423) enligt något av patentkraven 42 - 49, varvid dekodaren är anordnad att rekonstruera den första referensramen med användande av informationen av högre och lägre prioritet som dekodats från den första bitströmmen och med hänvisning till en föregående komplett referensram.

20

51. Dekodare (423) enligt något av patentkraven 42 - 50, varvid dekodaren är anordnad att rekonstruera den första virtuella referensramen med användande av informationen av högre prioritet som dekodats från den första bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet och med hänvisning till en föregående virtuell referensram.

25

52. Dekodare (423) enligt något av patentkraven 42 - 51, varvid dekodaren är anordnad att rekonstruera en kodad videoram på basis av en referensram tillhörande till en större mängd av referensramar.

30

53. Dekodare (423) enligt något av patentkraven 42 - 52, varvid dekodaren är anordnad att avlägsna en indikation av en viss referensram från bitströmmen.

54. Dekodare (423) enligt något av patentkraven 42 - 53, varvid dekodaren är anordnad att avlägsna en indikation av en viss referensram från bitströmmen.

55. Dekodare (423) enligt något av patentkraven 42 - 54, varvid dekodaren är anordnad att skapa den första virtuella referensramen delvis på basis av förinställda värden.

56. Videokommunikationsterminal (402) omfattande en videokodare (410) som kodar en första och en andra videoram, vilken videokodare omfattar

10

- en ramkodare (414) som är anordnad att koda den första videoramen för skapande av en första kodad videoram, varvid den första videoramen representeras av en bitström; varvid bitströmsrepresentationen av den första kodade videoramen omfattar information av högre och lägre prioritet (434), varvid dekodning av åtminstone en del av informationen av lägre prioritet från bitströmmen beror på dekodning av åtminstone en del av informationen av högre prioritet från bitströmmen, och ramkodaren är dessutom är anordnad att dekoda den första kodade videoramen med användande av den första kodade videoramens information av högre och lägre prioritet; och

20

- en virtuell ramskapare (416) som för skapande av en första virtuell referensram är anordnad att dekoda den första kodade videoramen med användande av den första kodade videoramens information av högre prioritet i frånvaro av åtminstone en del av den första kodade videoramens information av lägre prioritet;

25

- varvid ramkodaren är anordnad att koda den andra videoramen på basis av den första virtuella referensramen.

30

57. Videokommunikationsterminal (404) omfattande en videodekodare (423) som dekodar en första och en andra videoram, varvid varje kodad videoram representeras av en bitström, och bitströmsrepresentationen av den första kodade videoramen omfattar information av högre och lägre prioritet (434), varvid

dekodning av åtminstone en del av informationen av lägre prioritet från bitströmmen beror på dekodning av åtminstone en del av informationen av högre prioritet från bitströmmen, vilken videodekodare omfattar

- 5 - en dekodare (425) för kompletta ramar som för skapande av en komplett referensram är anordnad att dekodra en första kodad videoram på basis av den första kodade videoramens information av högre och lägre prioritet;
- 10 - en dekodare (426) för virtuella ramar som för skapande av en första virtuell referensram är anordnad att dekodra den första kodade videoramen med användande av den första kodade videoramens information av högre prioritet i frånvaro av åtminstone en del av den första kodade videoramens information av lägre prioritet;
- 15 - varvid dekodaren för kompletta ramar dessutom är anordnad att dekodra den andra videoramen på basis av den första virtuella referensramen.

58. Datorprogram för användning av en dator som videokodare (410) för kodning av en första och en andra videoram, varvid varje kodad videoram representeras av en bitström, och bitströmsrepresentationen av en kodad videoram omfattar information av högre och lägre prioritet (434), varvid dekodning av åtminstone en del av informationen av lägre prioritet från bitströmmen beror på dekodning av åtminstone en del av informationen av högre prioritet från bitströmmen, vilket datorprogram omfattar

20

25 - en datorprogramskod för kodning av en första videoram för skapande av en första kodad videoram, varvid den första kodade videoramen omfattar information av högre och lägre prioritet, för skapande av en första komplett referensram;

30 - en datorprogramskod för dekodning av den första kodade videoramen med användande av den första kodade videoramens information av högre och lägre prioritet;

- en datorprogramskod för dekodning av den första kodade videoramen med användande av den första kodade videoramens information av högre prioritet i frånvaro av åtminstone en del av den första kodade videoramens information av lägre prioritet för skapande av en första virtuell referensram; och

- en datorprogramskod för kodning av den andra videoramen på basis av den första virtuella referensramen.

10 59. Datorprogram för användning av en dator som videodekodare (423) för dekodning av en första och en andra kodad videoram, varvid varje kodad videoram representeras av en bitström, och bitströmsrepresentationen av en kodad videoram omfattar information av högre och lägre prioritet (434), varvid dekodning av åtminstone en del av informationen av lägre prioritet från
15 bitströmmen beror på dekodning av åtminstone en del av informationen av högre prioritet från bitströmmen, vilket datorprogramsprodukt omfattar

- en datorprogramskod för dekodning av den första kodade videoramen åtminstone delvis på basis av den första kodade videoramens information av högre och lägre prioritet;

- en datorprogramskod för dekodning av den första kodade videoramen med användande av den första kodade videoramens information av högre prioritet i frånvaro av åtminstone en del av den första kodade videoramens information av lägre prioritet för skapande av en första virtuell referensram;

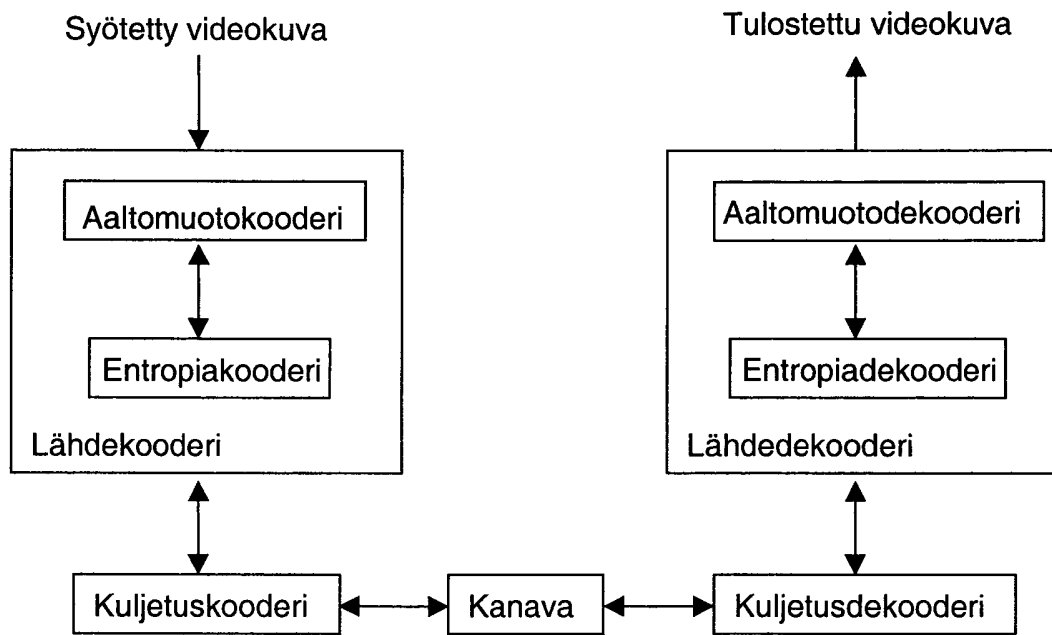
- en datorprogramskod för dekodning av den andra kodade videoramen på basis av den första virtuella referensramen.

30 60. Bitströmsrepresentation (434) av en kodad videoram, varvid den kodade videoramen omfattar information av högre och lägre information, varvid dekodning av åtminstone en del av informationen av lägre prioritet från bitströmmen beror på dekodning av åtminstone en del av informationen av högre prioritet från

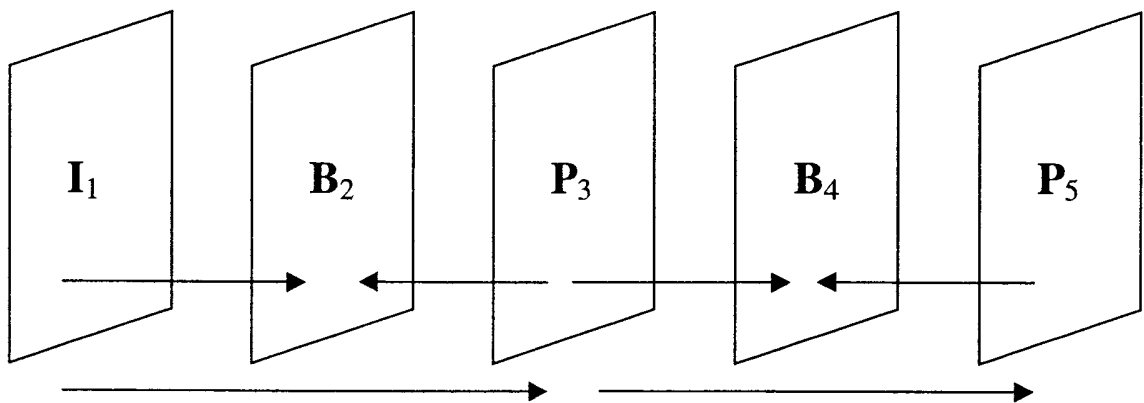


- bitströmmen; bitströmmen omfattar en indikator för att användas av en dekodare för att bestämma vilken information av högre prioritet en kodare har använt för skapande av en virtuell referensram, varvid kodaren har skapat nämnda virtuella referensram med användande av informationen av högre prioritet som dekodats från bitströmmen i frånvaro av åtminstone en del av informationen av lägre prioritet.
- 5

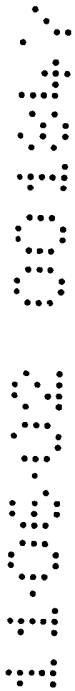


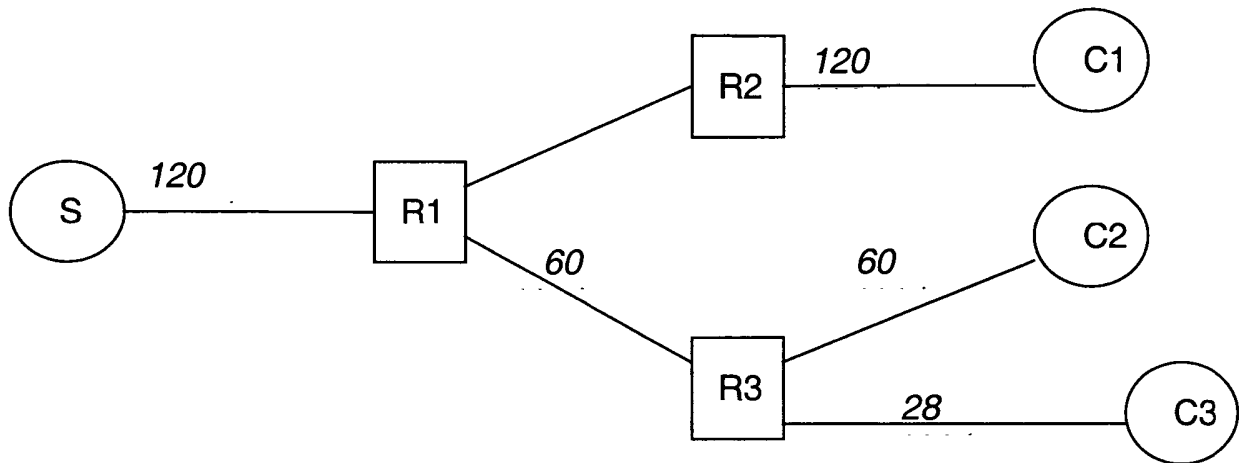


Kuvio 1

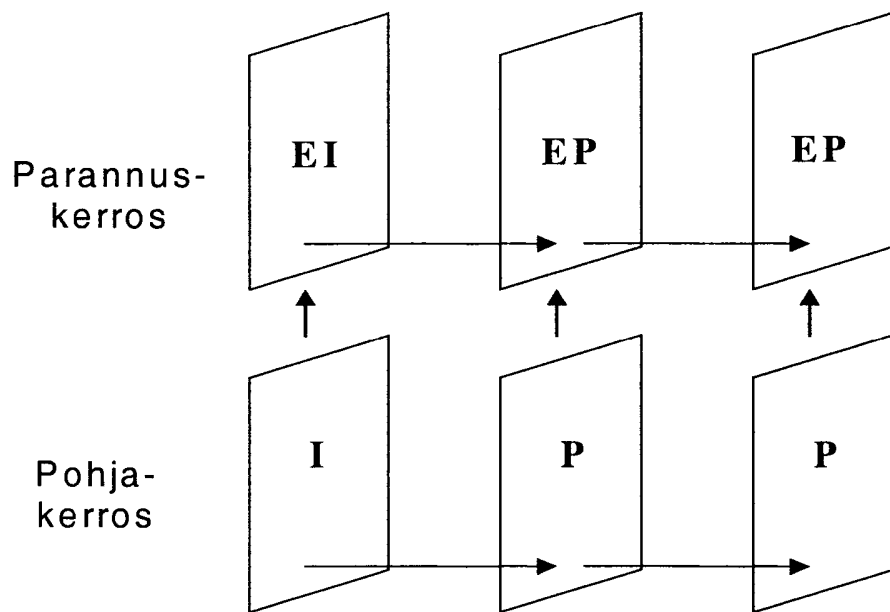


Kuvio 2



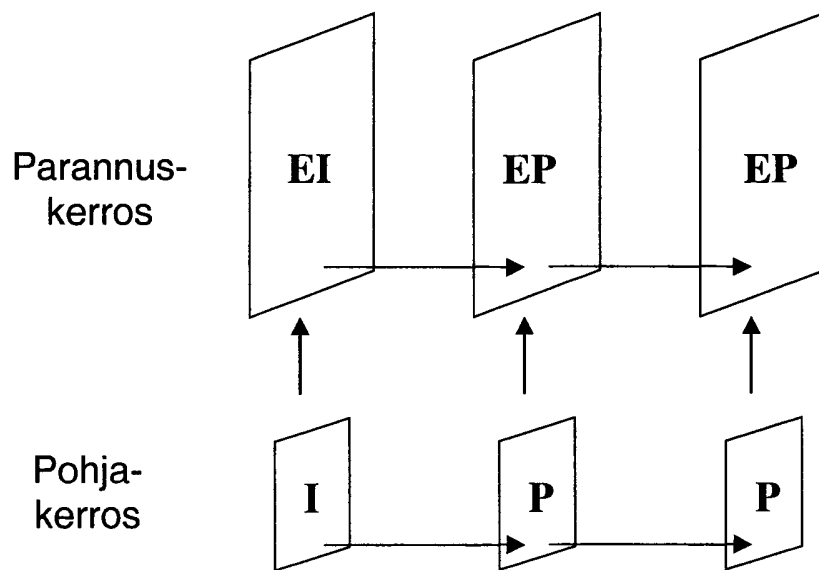


Kuvio 3



Kuvio 4

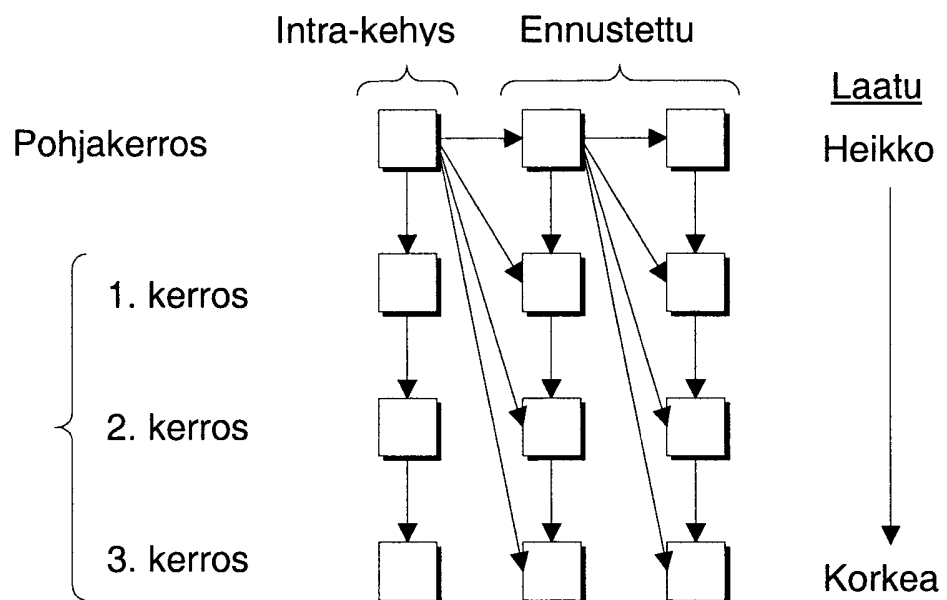




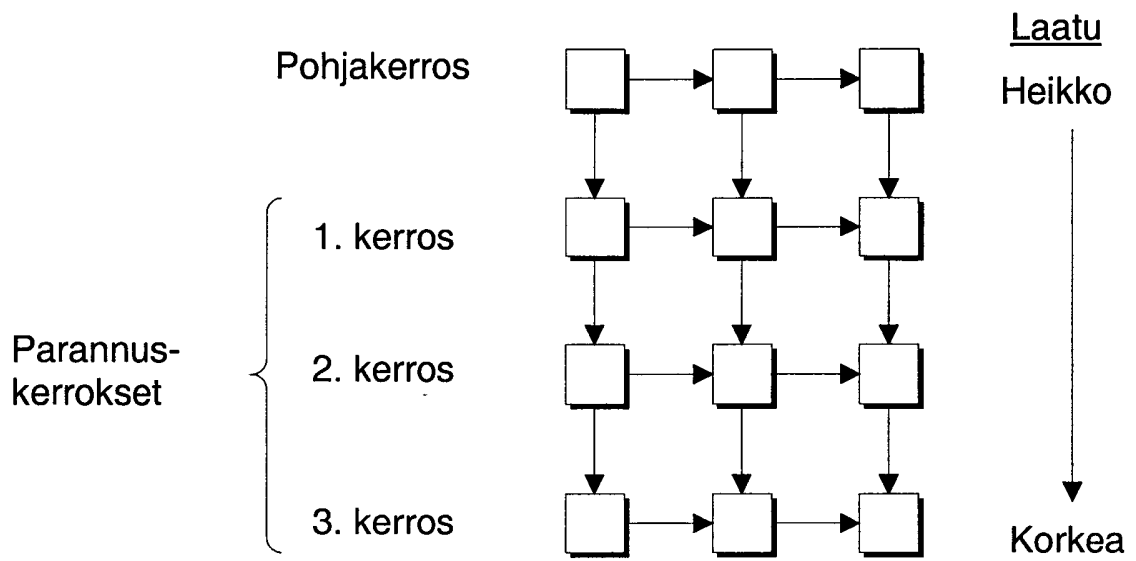
Kuvio 5



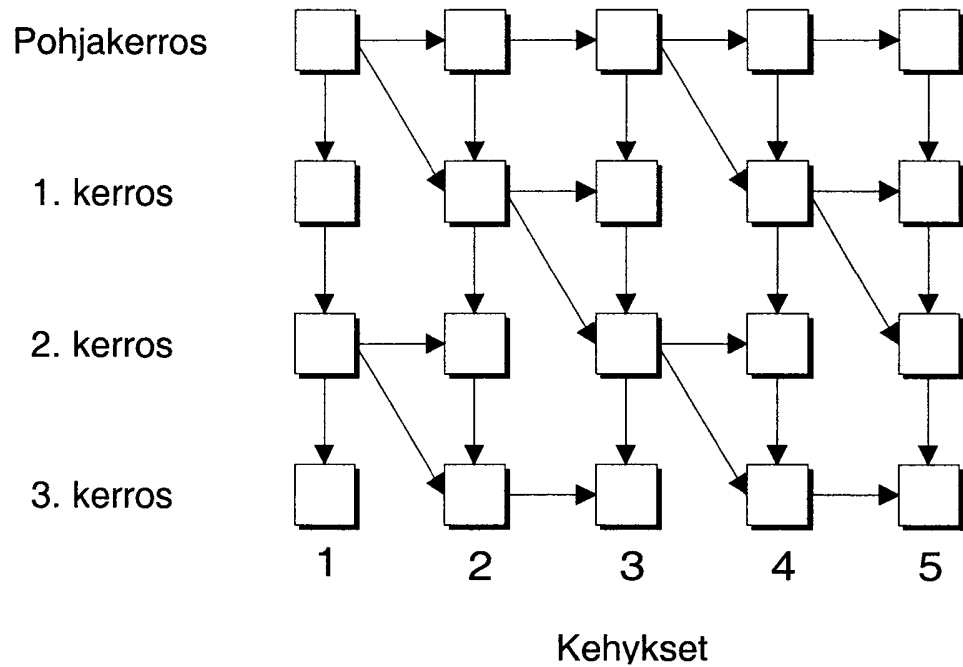
Parannuskerrokset



Kuvio 6

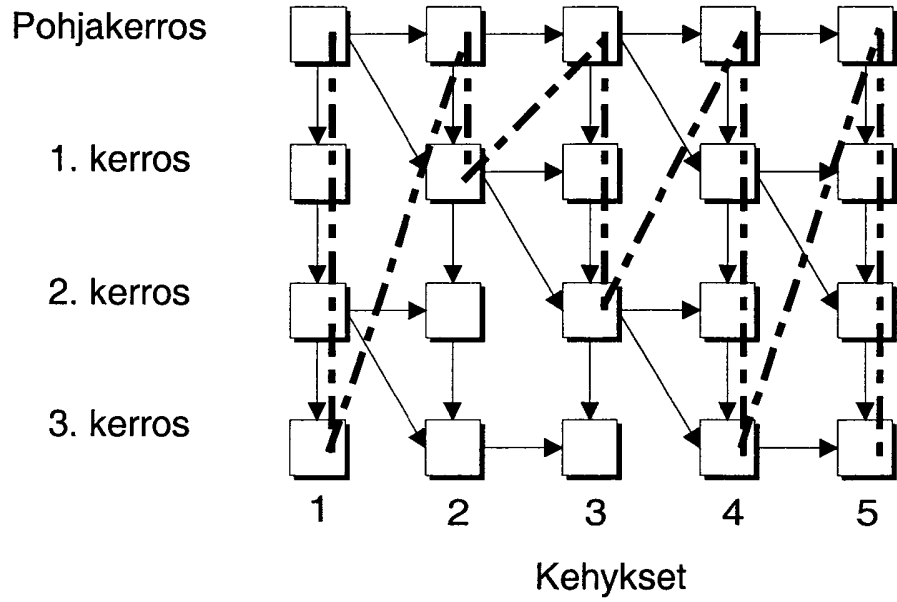


Kuvio 7

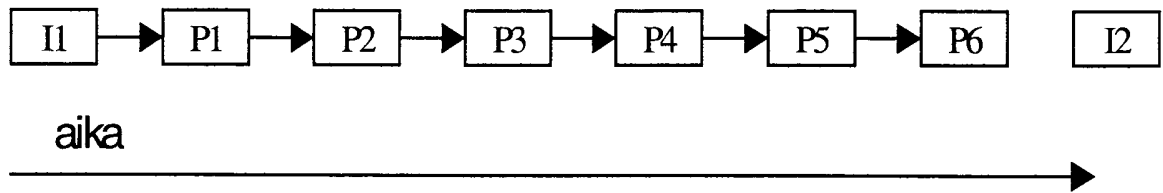


Kuvio 8



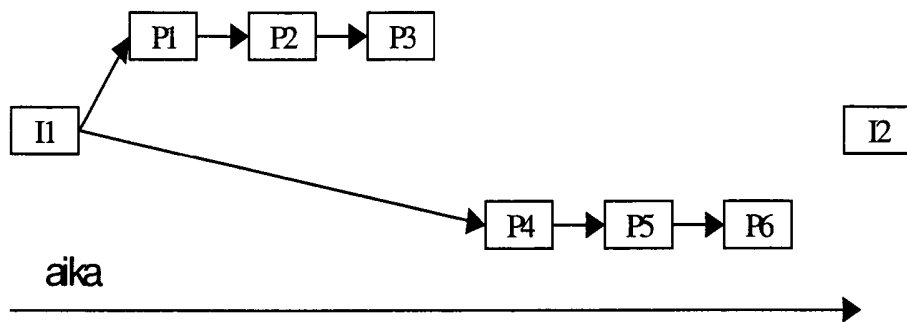


Kuvio 9

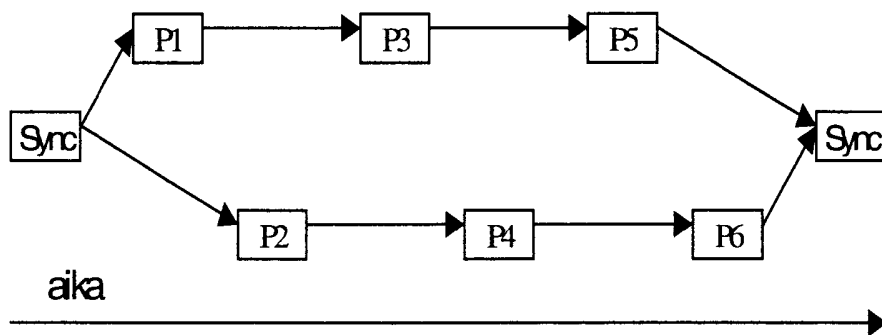


Kuvio 10

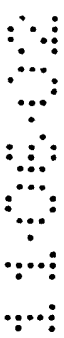


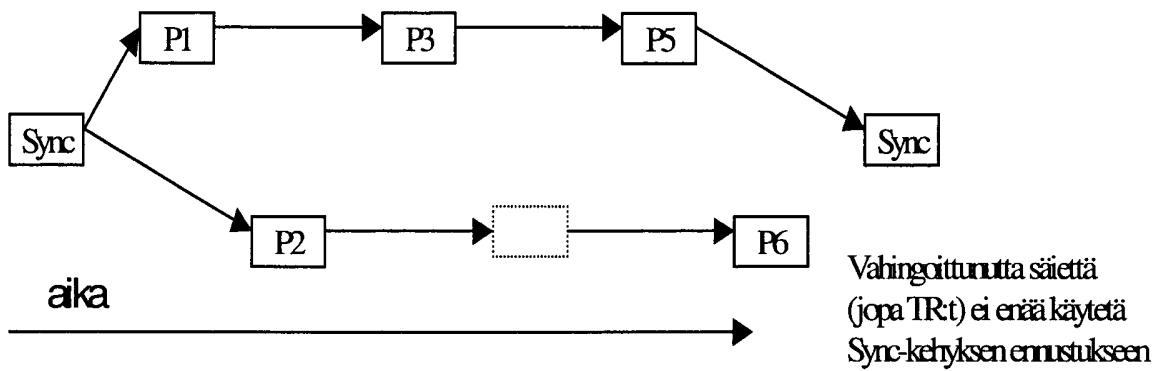


Kuvio 11

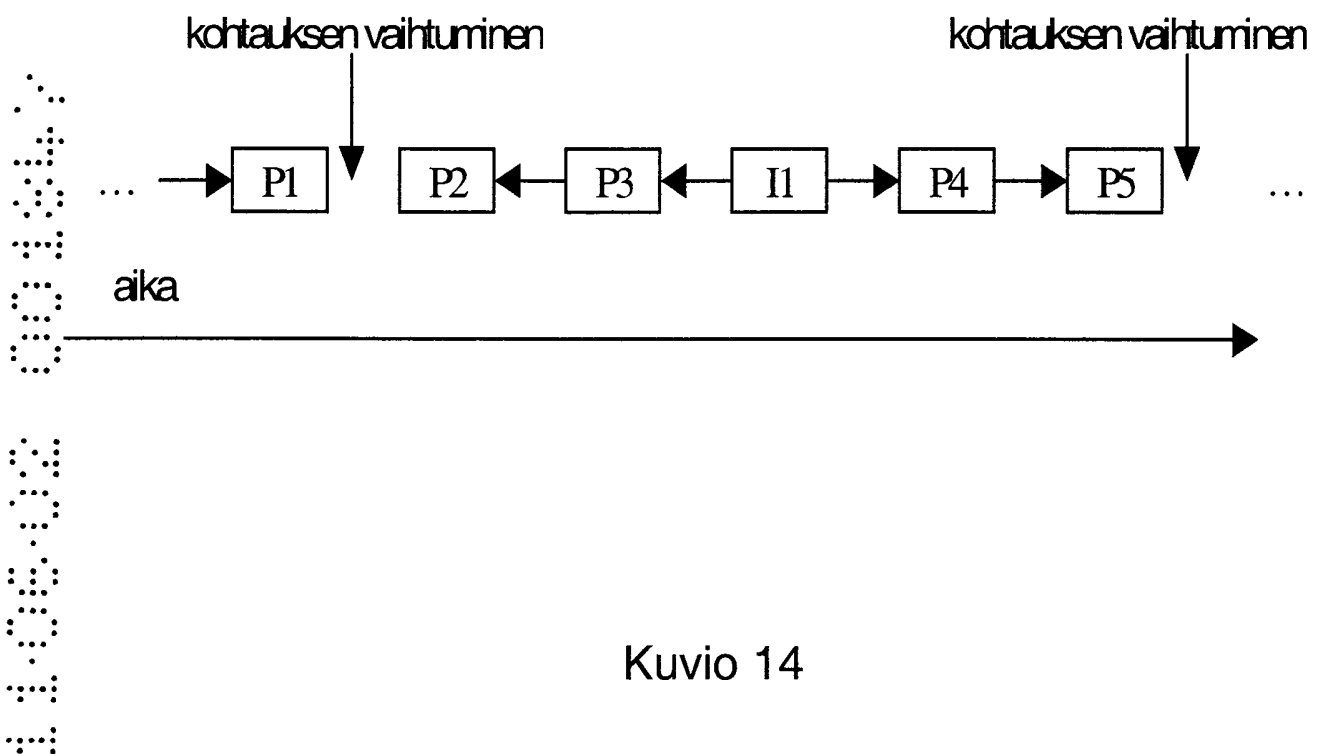


Kuvio 12

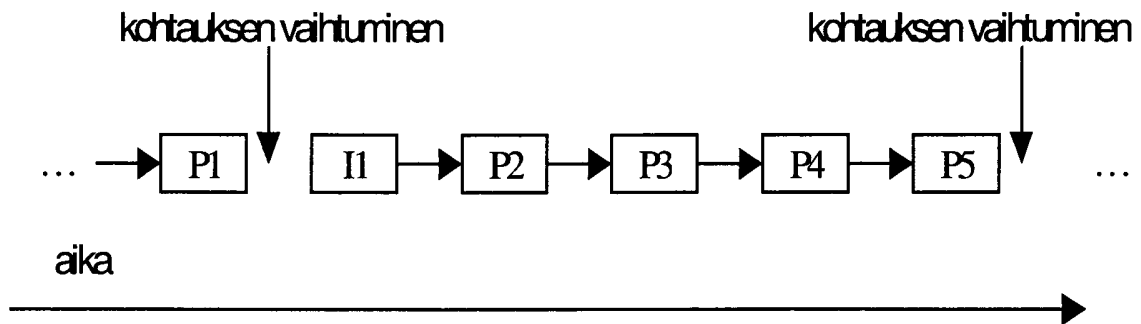




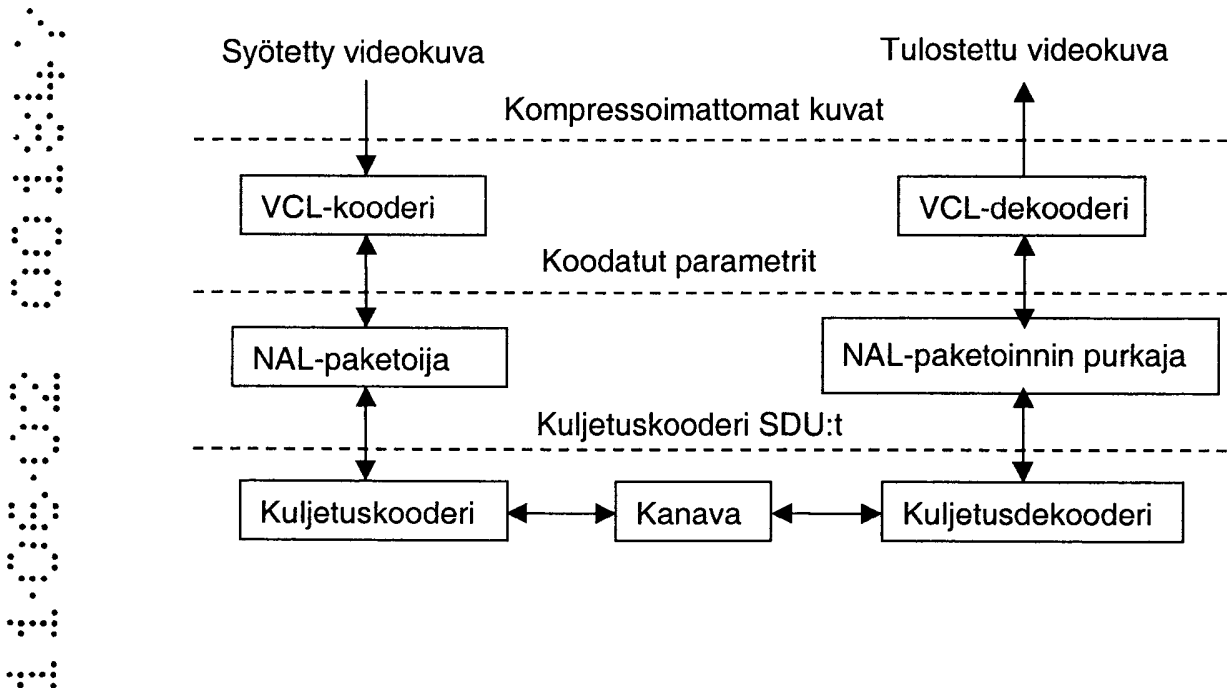
Kuvio 13



Kuvio 14



Kuvio 15



Kuvio 16

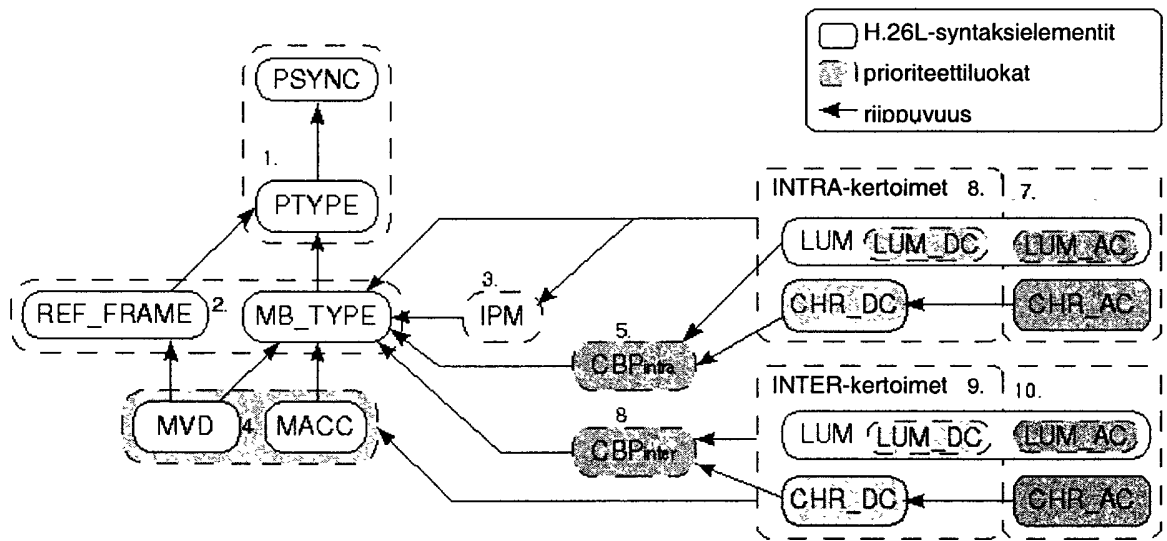
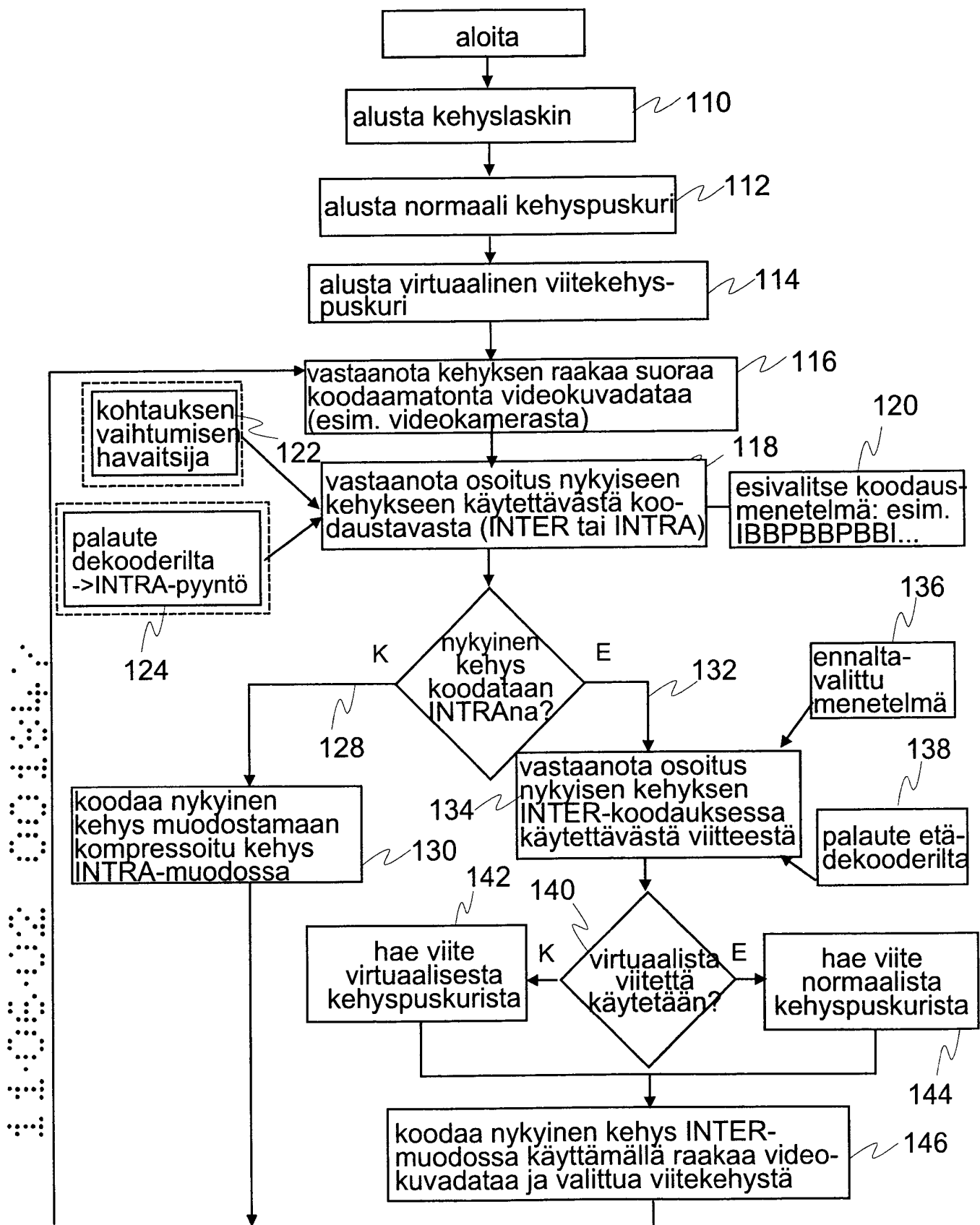
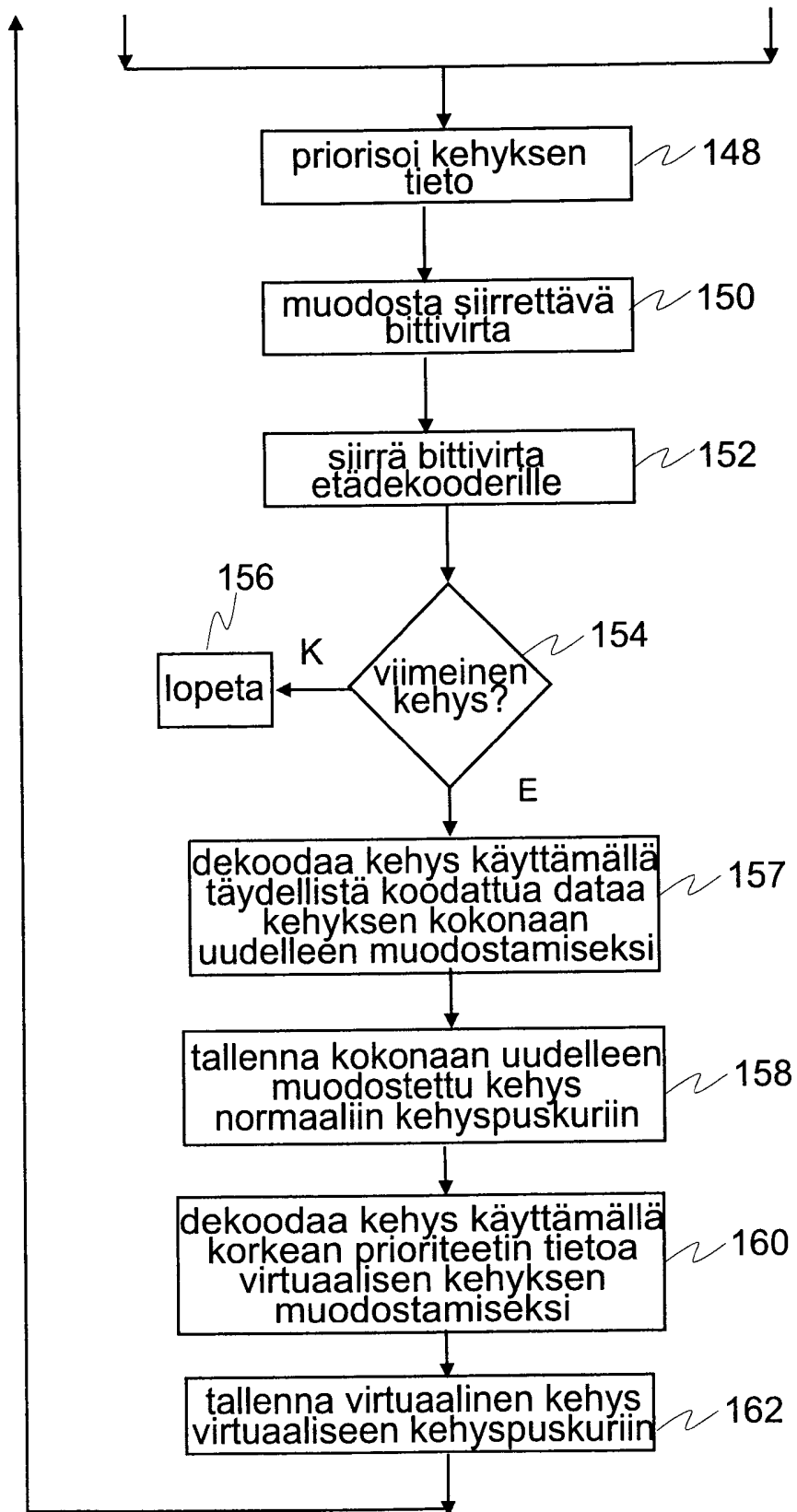


Fig. 17

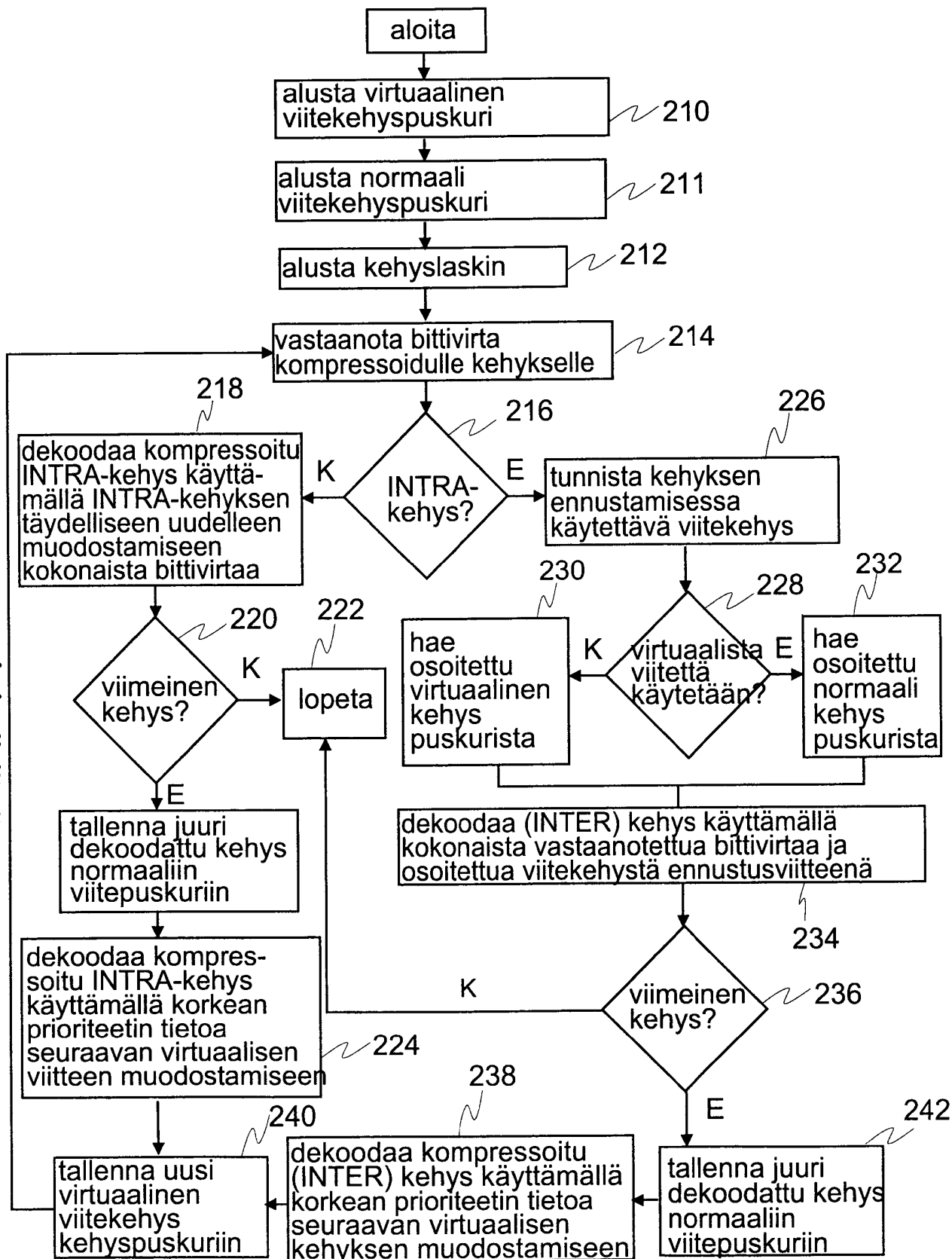
5
6
7
8
9
10



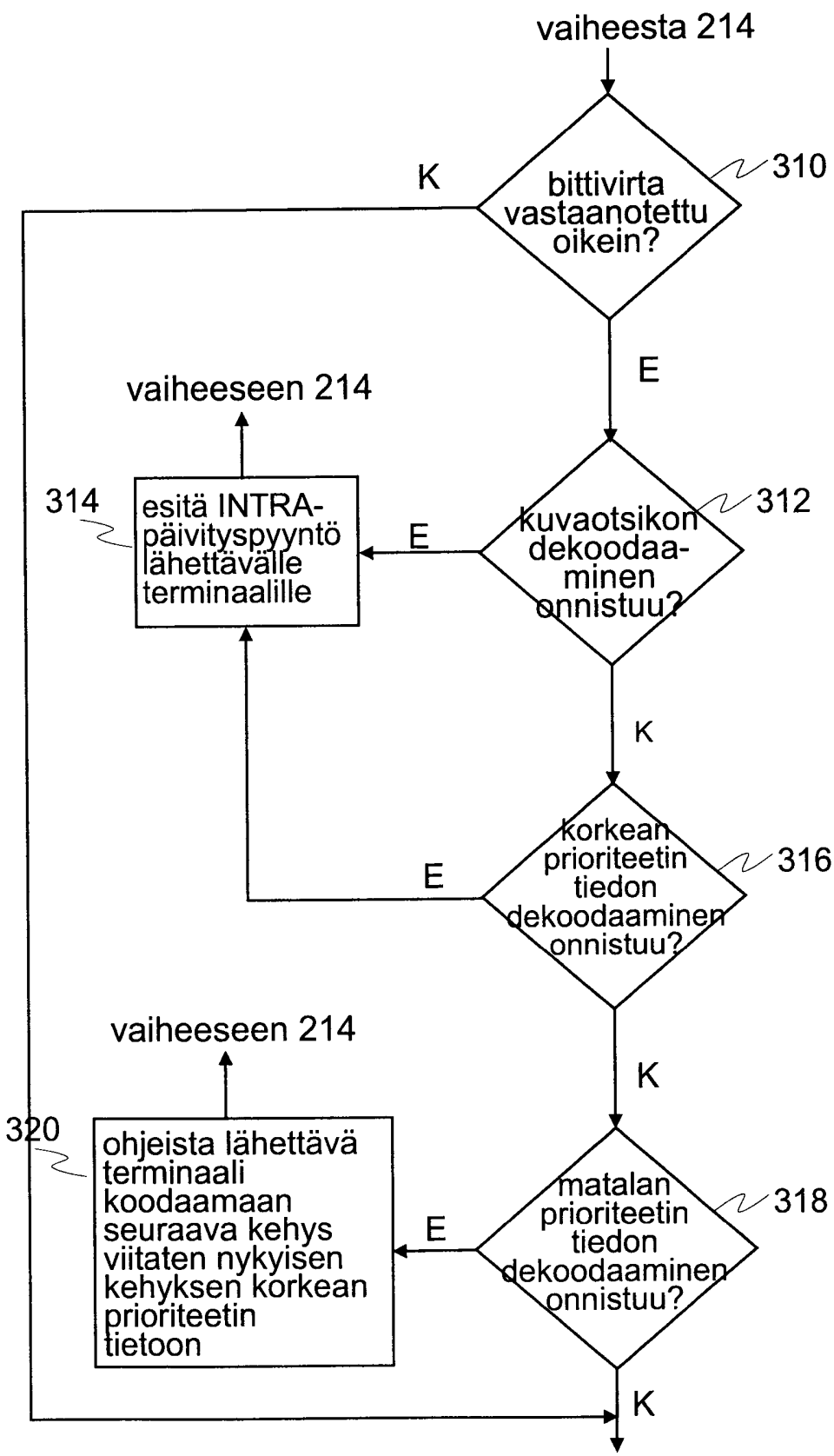
Kuvio 18a



Kuvio 18b

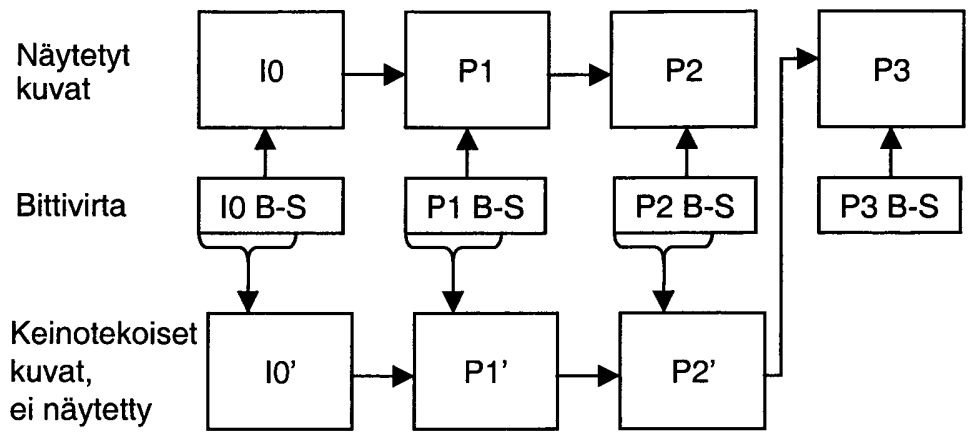


Kuvio 19

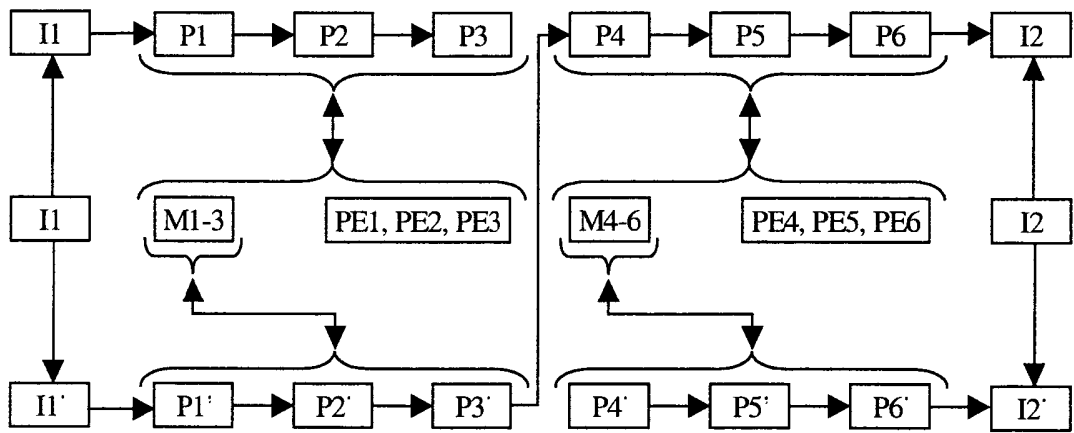


Kuvio 20 vaiheeseen 216



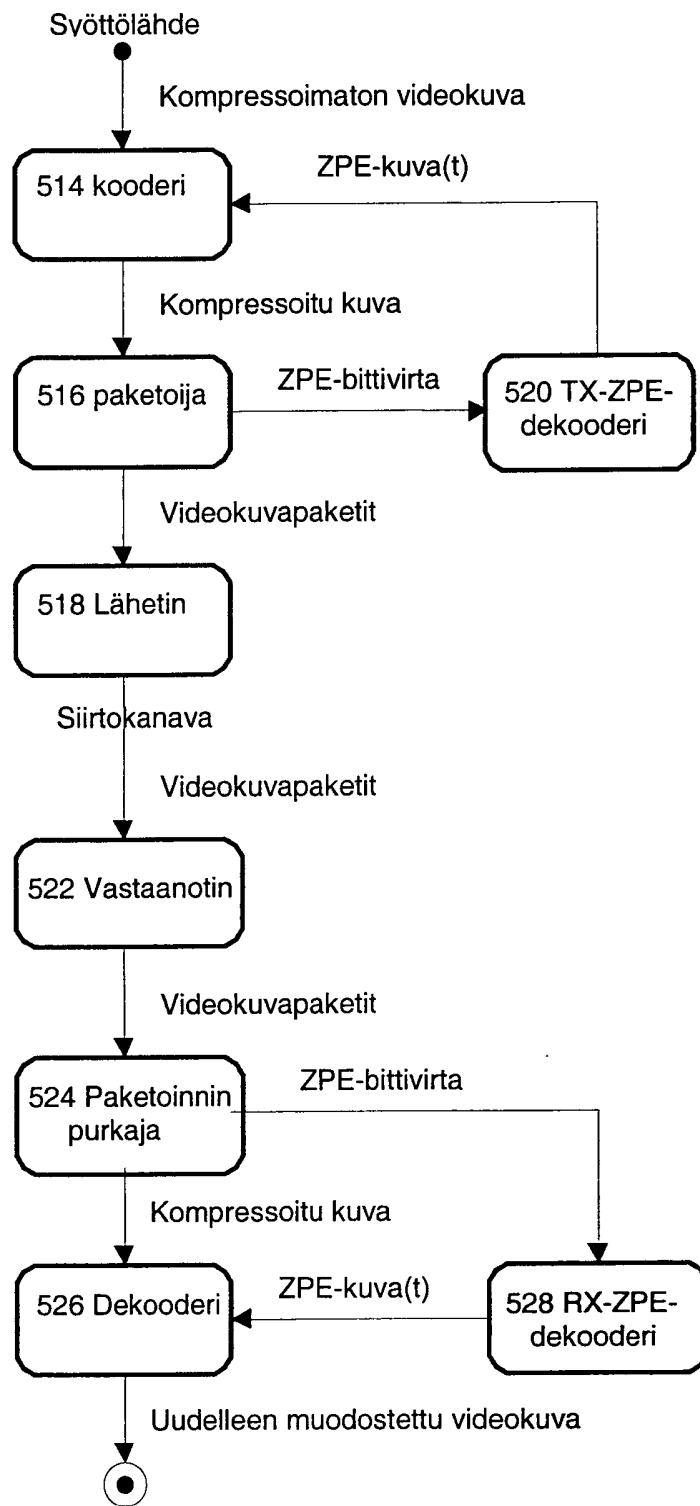


Kuvio 21



Kuvio 22





Kuvio 24