

Aki Rintala

Digitaalisen videon pakkausmenetelmät

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

29.4.2015

Tekijä Otsikko	Aki Rintala Digitaalisen videon pakkausmenetelmät
Sivumäärä Aika	30 sivua + 2 liitettä 29.4.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Mediatekniikka
Suuntautumisvaihtoehto	Digitaalinen media
Ohjaaja	Yliopettaja Erkki Rämö
<p>Insinööriyössä tutkittiin videonpakkauksen eri vaiheissa käytettäviä menetelmiä, jotka mahdollistavat videotiedoston käytännöllisen ja kustannustehokkaan siirtämisen ja jakelun videotuotannossa. Insinööriyönä tuotettiin neljän opiskelijan voimin esittelyvideoita rakennusalan kehityshankkeeseen. Videoiden tarkoituksena oli saada hankkeelle näkyvyyttä verkkomedian keinoin.</p> <p>Tavoitteena oli tehdä hankkeen kotisivuilla käytettäväksi visuaaliset ja laadullisesti näyttävät videot, joista koostettiin myös Blu-ray-tallenne. Videot kuvattiin digitaalisella järjestelmäkameralla, koska se tarjosi kuvausten hetkellä parhaan kuvanlaadun käytettäväksi lähdemateriaalina.</p> <p>Videonpakkauksessa käytettäviä standardeja on useita, ja uusia menetelmiä kehitellään jatkuvasti vastaamaan alati kehittyvää kamera- ja näyttötekniikkaa. Tallennuskapasiteetti tai siirtonopeus on yleensä rajallista, joten videonpakkauksen keinoin näitä rajoituksia yritetään kiertää. Pakkaus on kuitenkin suoritettava siten, ettei se heikennä videon laatua liikaa. Videotuotannossa on tärkeää löytää juuri oman projektin tarpeita vastaava kuvanpakkausmenetelmä, jotta mahdollisimman pieni tiedostokoko voidaan saavuttaa, uhraamatta kuitenkaan kuvanlaatua.</p> <p>Insinööriyön tuloksena syntyi kolme esittelyvideota, jotka vastasivat laadultaan, visuaalisuudeltaan ja käyttötarkoitukseltaan kehityshankkeen asettamia tavoitteita. Videot tarjosivat visuaalisemman esitystavan hankkeen tavoitteista ja tuloksista.</p>	
Avainsanat	videonpakkaus, videotuotanto, MPEG, bittinopeus

Author Title	Aki Rintala Encoding methods for digital video
Number of Pages Date	30 pages + 2 appendices 29 April 2015
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructor	Erkki Rämö, Principal Lecturer
<p>The objective of the final year project was to study the different stages and methods of video encoding, which enable a practical and cost-efficient distribution of material in video production. Part of this project was also to produce instructional videos for a development project in the field of construction. The goal of these videos was to gain more visibility for the project through internet media.</p> <p>The objective was also to produce visually great quality videos, which would later be compiled into a Blu-ray disc. All of the videos were shot by using a DSLR camera, which during the time of shooting recorder the best quality video to be used as a source material.</p> <p>There are many different kind of standards in the field of video encoding and new kind of methods are being developed to meet the requirements in advancements in video and camera technology. Storage and transfer speed are usually limited, so video encoding methods have been developed to combat these limitations. The goal is to implement encoding in such a way that it will not affect the quality of the video too drastically. It is important to find the correct encoder to meet the requirements of video projects so the smallest possible file size can be achieved without sacrificing too much in terms of quality.</p> <p>Paying attention to these requirements, three videos were made that met the quality and visual standards set by the project. The videos were successful in offering a more visual approach presenting the goals and results of the development project.</p>	
Keywords	video encoding, video production, MPEG, bitrate

Sisällys

Lyhenteet

1	Johdanto	1
2	Johdatus videonpakkaukseen	2
3	Videonpakkauksen toimintaperiaate	6
3.1	Paikallinen ja ajallinen redundanssi	6
3.2	Intra- ja interkoodatut toimenpiteet	8
3.3	Entropian koodaus	16
3.4	Dekoodaus	20
4	Laadunmittaus ja pilvipalvelut	20
4.1	Laadunmittaus	20
4.2	Pilvipalvelut	23
5	Videonpakkausstandardit	24
6	EEMontti-projektin kulku	25
6.1	Projektin videoiden määrittely	26
6.2	Kuvaus	26
6.3	Videonpakkaus ja laadunvalvonta	27
7	Yhteenveto	29
	Lähteet	31

Liitteet

Liite 1. Kuvassa 14 käytettävä esimerkkikuva suuremmassa koossa.

Liite 2. Kuvassa 15 käytettävä esimerkkikuva suuremmassa koossa.

Lyhenteet

SD	Standard definition, standardipiirto
HD	High Definition, teräväpiirto
UHD	Ultra High Definition, ultra-teräväpiirto
FPS	Frames Per Second, kehysten määrä sekunnissa
CCD	Charged Coupled Devices, järjestelmäkameran kennotyyppi
CMOS	Complementary Metal Oxide Semiconductor, järjestelmäkameran kennotyyppi
RGB	Red Green Blue -väriavaruus
Y'CrCb	Luminanssiin perustuva väriavaruus
DCT	Diskreetti kosinimuunnos
RLC	Run Length Coding, jononpituuskoodaus
VLC	Variable Length Coding, vaihtuvamittainen koodaus
INTRA	Yhdessä kehyksessä tapahtuva pakkaus
INTER	Peräkkäisten kehysten erodataan perustuva pakkaus
MPEG	Moving Picture Experts Group, videopakkausstandardi
AVC	Advanced Video Coding, videopakkausstandardi
HEVC	High Efficiency Video Coding, videopakkausstandardi
PSNR	Peak Signal to Noise Ratio, matemaattinen laadunmittausmenetelmä
SNR	Signal to Noise Ratio
MSE	Mean Square Error
DSCQS	Double Stimulus Continuous Quality Scale, aistinvarainen laadunmittausmenetelmä

1 Johdanto

Insinööriyön tarkoituksena on valmistaa kolmen muun Metropolia Ammattikorkeakoulun opiskelijan kanssa esittelyvideoita Green Net Finland ry:n koordinoimaa EEMontti-kehityshankkeeseen. Hankkeen tarkoituksena on tuoda pientalojen energiaremonttien hyödyt suuren yleisön tietoisuuteen verkkomedian avulla. Tavoitteena on luoda videoista näyttäviä sekä visuaalisesti että kuvanlaadullisesti. Valmiit videot on tarkoitus julkaista videopilvipalvelun avulla projektin kotisivuilla, ja myöhemmin fyysisesti Blu-ray-levynä.

Insinööriyöraportissa tarkastellaan myös, kuinka videokuva tallennetaan kameran avulla, minkälaisista eri komponenteista videokuva koostuu ja kuinka videota käsitellään pakkauksen aikana. Lisäksi keskitytään digitaalisen videomateriaalin pakkauksen kulkuun ja niihin menetelmiin, jotka mahdollistavat videomateriaalin siirtämisen ja säilyttämisen pienemmässä tiedostokoossa vaikuttamatta kuitenkaan heikentävästi videon visuaaliseen laatuun. Raportissa vertaillaan myös aistinvaraisesti ja matemaattisesti suoritettavien laaduntarkkailumenetelmien tehokkuutta videoprojektissa ja sovelletaan niitä EEMontti-esimerkkiprojektiin.

Videomateriaalia on nykyisin saatavilla suuri määrä erilaisten jakelukanavien kautta, mutta sitä ei katsella enää pelkästään television kautta. Internetin ansiosta videomateriaali on levinnyt kaikkialle, ja sitä voidaan toistaa monilla erilaisilla tavoilla, esimerkiksi tietokoneilla tai mobiililaitteilla. Internetissä sijaitsevat palvelut tarjoavat kuluttajille mahdollisuuden tallentaa ja jakaa omaa videomateriaaliaan muille käyttäjille ympäri maailman.

Videoilla tarkoitettun pilvipalvelun käyttö ei kuitenkaan ole pakollista, sillä videomateriaalia pystytään yhtä hyvin jakamaan oman kotisivun kautta. Tällaisessa tapauksessa kuluttaja tai yritys joutuu itse huolehtimaan datansiirtomaksuistaan, joten esitettävän videomateriaalin on oltava pienssä koossa kustannusten minimoimiseksi. Tällaisen videoinformaation käytännöllinen ja kustannustehokas jakelu tai tallentaminen ei olisi mahdollista ilman tehokasta videonpakkausta.

2 Johdatus videonpakkaukseen

Televisio- ja kameratekniikassa 2000-luvulla tapahtuneen kehityksen ansiosta videomedian kuvatarkkuus eli resoluutio on kasvanut huomattavasti. Samalla on siirrytty SD-aikakaudelta HD-aikakauteen. Tulevaisuudessa on jo nähtävissä seuraava kehityssaskel kuvatarkkuuden paranemisessa niin sanotun UHD-kuvaformaatin avulla. Taulukosta 1 nähdään UHD:n lisäksi PAL-alueen yleisimmät kuvaformatit ja se, kuinka resoluution kasvun myötä myös kuvan pikselimäärä kasvaa moninkertaisesti.

Taulukko 1. Kuvaformatit, niiden resoluutio ja pikselimäärä [5, s. 19; 15].

Formaatti	Resoluutio	Pikselimäärä
SD	720 × 576	414 720
HD 720p	1280 × 720	921 600
HD 1080p	1920 × 1080	2 073 600
4K UHD	3840 × 2160	8 294 400
8K UHD	8192 × 4320	33 177 600

Video koostuu käytännössä yksittäisistä kuvista, jotka tarpeeksi nopeassa rytmissä peräjälkeen esitettynä luovat katsojalle illusion liikkeestä. Nopeus, jolla kuvia esitetään, ilmaistaan käsitteellä fps (frames per second). PAL-alueella videostandardiksi on muodostunut 25 kuvaa sekunnissa eli 25 fps. Ihmissilmä pystyy erottamaan noin 10–12 yksittäistä kuvaa sekunnissa. Näin ollen sekunnissa esitettävän kuvamäärän on ylitettävä tämä lukema, jotta illuusio liikkeestä säilyisi. Kuvien erottelukyky on kuitenkin yksilöllistä, eivätkä samat arvot päde kaikkiin ihmisiin. Tämä on kuitenkin yksi monista syistä, miksi videokuvan standardiksi on muodostunut juuri 25 fps. [3, s. 24–26.]

Yksittäinen 1920 x 1080 -HD-kuva koostuu noin kahdesta miljoonasta pikselistä, mikä tarkoittaa, että 25 fps -nopeudella sekunnin aikana on esitettävä $25 * 2\,073\,600$ pikseliä. Oletetaan vielä, että jokaisen yksittäisen pikselin sisältämän informaation esittämiseen tarvitaan 32 bittiä. Näin ollen sekunnin videokuvan esittämiseksi tarvitaan $25 * 2\,073\,600 * 32$ bittiä. Pakkaamattomana HD-materiaalin esittämiseen suoratoiston avulla vaadittaisiin noin 1500 Mb/s:n tasainen siirtonopeus ja 10 minuutin mittaisen HD-videon tallentamiseen vaadittaisiin noin 90 gigatavua muistia.

Tällaisen datamäärän siirtäminen sellaisenaan on hyvin epäkäytännöllistä ja kallista. Sen takia datamäärän pienentämiseksi on kehitetty kuvanpakkausmenetelmiä, joiden tarkoituksena on pakata dataa pienempään tilaan sen siirtämisen ja käsiteltävyyden helpottamiseksi. On kuitenkin tärkeää, että pakkaus suoritetaan siten, ettei katsoja huomaa eroa pakatun ja pakkaamattoman videomateriaalin välillä. [2, s. 1–2.]

Kuvassa 1 esitetään yksinkertaistetusti videodatan siirto julkaisukanavassa. Ennen videon siirtämistä lähdemateriaali pakataan pienempään tilaan kooderin avulla. Tämän jälkeen koodattu videodata lähetetään haluttuun julkaisukanavaan, kuten Blu-ray tai suoratoisto. Lähetetty data palautetaan lopuksi alkuperäiseen muotoonsa dekooderin avulla, minkä jälkeen videomateriaali on katseltavissa. Näiden kahden toimenpiteen yhdistelmää kutsutaan nimellä kodekki (codec). Kodekin sisältämien kooderin ja dekooderin välissä tapahtuvaa datan pakkausta kutsutaan pakkaussuhteeksi. [2, s. 2.]



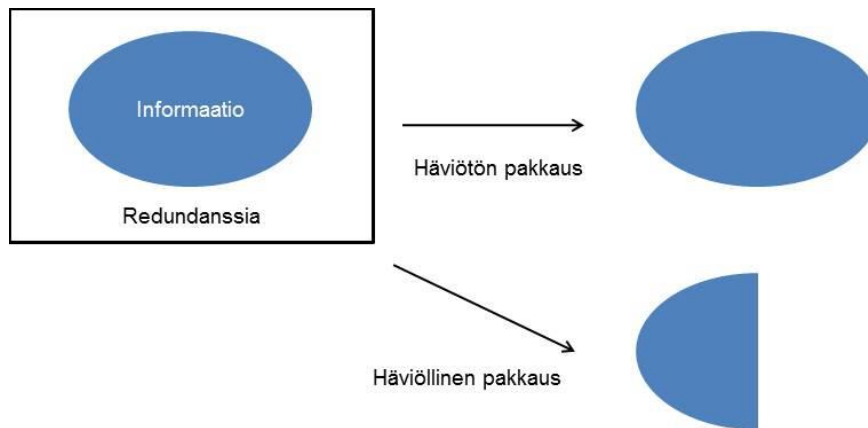
Kuva 1. Datan kulku lähdemateriaalista lopputuotteeksi [2, s. 2].

Häviötön ja häviöllinen pakkaus

Pakkaustekniikoita on monenlaisia, mutta ne kaikki voidaan kuitenkin jakaa kahteen pääluokkaan: häviöttömään ja häviölliseen pakkaukseen. Häviöttömässä pakkauksessa dataa ei menetetä pakkauksen aikana ja dekooderin jälkeen lopputuote on täysin samanlainen alkuperäisen lähdemateriaalin kanssa jokaista bittiä myöten. Ongelmana häviöttömässä pakkauksessa on huono pakkaussuhde: se on parhaimmillaan noin kahden suhde yhteen, riippuen pakattavasta lähdemateriaalista. Häviötöntä pakkausta käytetään pääasiassa tietokoneiden sovelluksissa, joissa bittijonon rakennetta ei voi muuttaa ilman vakavia seurauksia sovellusten kannalta. [2, s. 3–4.]

Häviöllisessä pakkauksessa lähdemateriaali ja lopputuote eivät täysin vastaa toisiaan. Tarkempi tarkastelu paljastaisi suuria eroja pakatun ja pakkaamattoman materiaalin välillä. Häviöllinen pakkaus ei tämän takia sovellu tietokoneissa käytettäviin sovelluksiin, mutta on suosittua videonpakkauksessa, koska se mahdollistaa paljon suuremman pakkaussuhteen. Suurempi pakkaussuhde mahdollistetaan käyttämällä hyödyksi ihmissilmän heikkouksia huomata yksityiskohtia ja poistamalla datasta informaatiota, jota silmä ei pysty havaitsemaan. [2, s. 3–4.]

Informaatiota, jota tarvitaan videomateriaalin esittämiseen, kutsutaan usein entropiaksi. Kuvasta 2 esitetään, kuinka entropian lisäksi kuvassa esiintyy redundanttia informaatiota. Häviöttömässä pakkauksessa pyrkimyksenä on poistaa redundantti eli informaation esittämisen kannalta turha data.



Kuva 2. Häviöttömän ja häviöllisen pakkauksen toiminta informaation ja redundanssin kanssa [2, s. 4].

Häviöllisessä pakkauksessa redundanssin poistamisen lisäksi videomateriaalin informaatiosta poistetaan dataa. Datan poistaminen tässä tapauksessa ei välttämättä tarkoita sen tuhoamista, vaan muuttamista erilaiseen muotoon siirtämisen ajaksi. [2, s. 3–4.]

Videokuvan tallentaminen

Todellisessa maailmassa visuaalinen kuva ympäristöstä koostuu yleensä useista eri objekteista, muodoista, väreistä, kirkkautenvaihteluista ja tekstuureista. Tällainen ih-

missilmin nähty visuaalinen kuva on myös ajallisesti ja paikallisesti jatkuva, eli niin sanottua visuaalista dataa ilmenee myös näkökenttämme ulkopuolella. [5, s. 7.]

Luonnollisen visuaalisen kuvan tallentamiseksi se täytyy näytteistää. Yleensä kohtaus näytteistetään paikallisesti suorakaiteen muotoon yhdelle videokuvan tasolle eli kehykselle ja ajallisesti peräkkäisille kehyksille. Digitaalinen video on näytteistetyn kuvasarjan esitysmuoto digitaalisessa muodossa, jossa jokaista näytettä vastaa yksi tai useampi numero. Tämä numeroarvo voi kuvastaa muun muassa näytteen kirkkautta, luminanssia tai väriä.

Näytteistetyn kuvan tallentamiseksi kamera keskittää linssiin osuvan säteilyn sensorille, kuten CCD- (Charged Coupled Devices) tai CMOS- (Complementary Metal Oxide Semiconductor) -kennolle, joka tallentaa jokaisen elementin ja värin erillisenä komponenttina. Tuloksena CCD-kennolta saadaan vaihteleva analoginen signaali, joka kuvastaa tallennettua kuvaa ja sen objekteja signaalimuodossa. Tämä signaalin näytteistäminen tietyllä aikavälillä luo edellä mainitun näytteistetyn kehyksen, jolla on neliönmuotoisessa näytekohdassa tietty numeroarvo. Tällaista neliönmuotoista näytekohtaa kutsutaan yleensä pikseliksi. Analogisessa signaalissa käytettävien näytekohtien määrä on suoraan verrannollinen digitaalisen kuvan pikselimäärään. Pitkät näytevälit signaalissa luovat huonon kuvalaadun, kun taas lyhyet näytevälit parantavat kuvan laatua. [5, s. 8–10.]

Bittinopeus

Bittinopeudella tarkoitetaan bittimäärää, joka sekunnin aikana tarvitaan videokuvan esittämiseksi. Bittinopeutta kuvataan yleensä käsitteillä Mb/s (megabittiä sekunnissa) tai kb/s (kilobittiä sekunnissa). Bittinopeus myös kertoo, kuinka paljon tiedonsiirtonopeutta laitteiden välillä tarvitaan, jotta videon esittämisessä ei olisi taukoja.

Videota pakattaessa käyttäjä valitsee, minkälaista bittinopeutta pakattu video käyttää toimenpiteiden jälkeen. Lähdemateriaalista riippuen bittinopeus ei voi pysyä koko ajan samana, vaan sille määritetään maksimi- ja minimiarvot, joiden väliin video pyritään pakkaamaan. DVD-levyn tapauksessa suurin käytettävissä oleva bittinopeus on 9,8 Mb/s, mikä johtuu laitteiston kapasiteetista käsitellä ja siirtää dataa. Nykyisten Blu-ray-levyjen tapauksessa suurin mahdollinen siirtonopeus on 40 Mb/s. [10.]

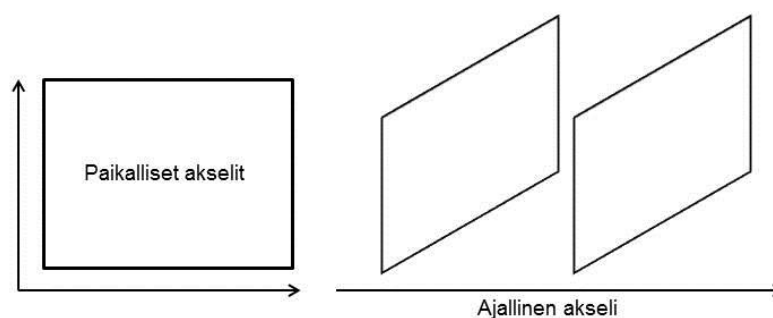
Pakkauksessa käytettävän bittinopeuden valinnassa on otettava huomioon pakattavan lähdemateriaalin laatu ja alkuperäinen bittinopeus. Pakkaamattoman 1080p-videokuvan alkuperäinen bittinopeus on 1500 Mb/s.

Tällaisen kuvan pakkaaminen 18–30 Mb/s -bittinopeuteen tietysti heikentää kuvanlaatua, mutta kuitenkin hyväksyttävissä rajoissa. Jos samanlaista 18–30 Mb/s -pakkausta pyrittäisiin käyttämään 4K- tai 8K-UHD-formaatteihin, olisi sillä paljon suurempi heikentävä vaikutus kuvanlaatuun. Tämä takia pakatun kuvamateriaalin bittinopeus on osittain sidottu lähdemateriaaliin, jotta hyväksyttävä kuvanlaatu säilyisi.

3 Videonpakkauksen toimintaperiaate

3.1 Paikallinen ja ajallinen redundanssi

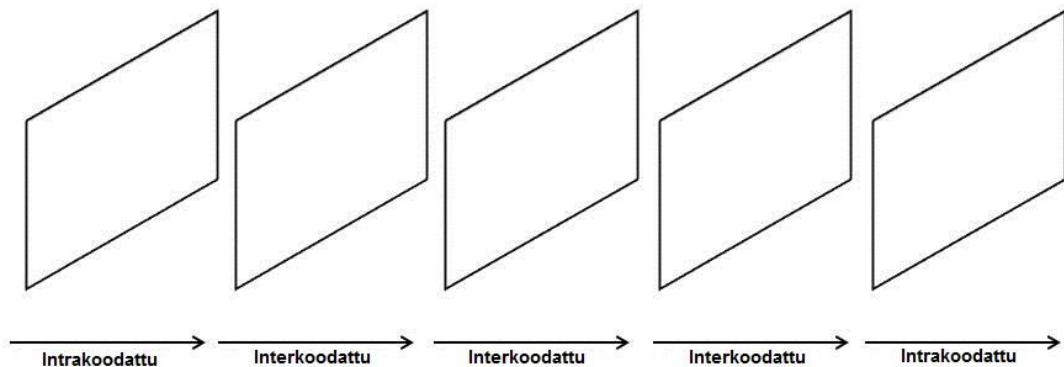
Videonpakkaus perustuu videosignaalin olevan redundantin informaation poistamiseen. Redundantilla informaatiolla tarkoitetaan dataa, jota ei pystytä aistinvaraisesti havaitsemaan tai data esiintyy useaan otteeseen videosignaalin. Redundanttia informaatiota voi ilmetä paikallisesti yhdessä videon kehyksessä tai ajallisesti useissa peräkkäisissä kehyksissä. Paikallisesti ilmenevän redundanssin poistamiseen tarkoitettuja toimenpiteitä kutsutaan intrakoodatuiksi toimenpiteiksi, kun taas ajallisesti tapahtuvan redundanssin poistamiseksi tarkoitettuja toimenpiteitä kutsutaan interkoodatuiksi toimenpiteiksi. [2, s. 9–10.] Kuvassa 3 esitetään paikallinen ja ajallinen akseli havainnollistamisen parantamiseksi.



Kuva 3. Paikallinen ja ajallinen akseli kuvasignaalin [5, s. 8].

Koska intrakoodatut toimenpiteet tapahtuvat vain yhdessä videokuvan kehyksessä, ne ovat hyvin yhtäläisiä still-kuvan pakkauksen kanssa. Tällaisia toimenpiteitä ovat muun muassa väriavaruuden muuntaminen, diskreetti kosinimuunnos (DCT), kvantisointi ja entropian koodaus. Yhdessä kehyksessä redundanssia esiintyy kolmessa ulottuvuudessa, spatiaalisilla akseleilla pysty- ja vaakasuunnassa sekä pikselien näytearvoissa. [2, s. 9–10.]

Ajallisesti tapahtuvissa eli interkoodatuissa toimenpiteissä redundanssin poistamiseksi hyödynnetään videosignaalin tilastollista ennakoitavuutta. Tällä tarkoitetaan videokuvan peräkkäisissä kehyksissä ilmenevää samankaltaisuutta, jota voidaan hyvällä todennäköisyydellä laskennallisesti ennustaa. Sen sijaan, että jokaisen kehyksen informaatio lähetettäisiin yksittäin, peräkkäisiä kehyksiä verrataan toisiinsa ja vain näiden kehysten välinen erodata lähetetään. Samaa informaatiota ei siis lähetetä useaan otteeseen, vaan käytetään hyväksi jo olemassa olevaa informaatiota. [2, s. 9–11.] Kuvassa 4 esitetään, kuinka peräkkäisiä kehyksiä käsitellään informaatiota lähetettäessä.



Kuva 4. Intrakoodattu kehys, jossa kaikki informaatio lähetetään. Interkoodattu kehys, jossa peräkkäisiä kehyksiä verrataan toisiinsa ja vain eroavaisuudet lähetetään. [2, s. 10.]

Tällainen informaation lähetys on kuitenkin hyvin altis virheille, sillä yhdessä kehyksessä tapahtuva koodausvirhe heijastuu myös kaikkiin myöhempisiin kehyksiin. Sen takia tietyin väliajoin lähetetään intrakoodattu kuva, jonka sisältöä ei ole ennustettu aikaisemmista kehyksistä. Intrakoodattuja kehyksiä merkitään usein kirjaimella "I" (Intra) ja interkoodattuja kirjaimella "P" (Predicted).

Videon esittämiseen voidaan käyttää kolmea erilaista kehystyyppiä. I-kehysten esittämiseen tarvitaan näistä kolmesta eniten bittejä, sillä niihin sovelletaan vain intra-

koodattuja toimepiteitä. I-kehyksiä on kuitenkin pakko käyttää tietyin väliajoin, koska ne toimivat referenssinä muille koodatuille kehyksille. [2, s. 10–11.]

P-kehukset käyttävät apunaan aikaisemmin esitettyjen kehysten dataa ja perustuvat näiden peräkkäisten kehysten erodatan käyttöön. Tarkoituksena erodatan käytössä on lähettää vain sellaista informaatiota, mikä on muuttunut edellisestä kehyksestä.

Kolmas kehystyyppi ”B” (Bi-directional) käyttää apunaan sitä edeltävää ja sen jälkeen saapuvaa kehystä. Tämän erodatan takia B-kehysten esittämiseen tarvittava bittimäärä on hyvin pieni. Koodekista riippuen B-kehyksiä voidaan käyttää joko yksi tai kaksi peräkkäin, ennen kuin on käytettävä taas I- tai P-kehystä. P- tai B-kehyksiä ei kuitenkaan voida esittää ilman referenssinä toimivaa I-kehystä, koska niiden sisältämä informaatio perustuu täysin referenssin informaatioon.

Tilastollista ennakoitavuutta voidaan siis käyttää vain tiettyyn määrään peräkkäisiä kehyksiä kerrallaan. Kuvassa 3 esitettiin, kuinka aluksi lähetetään intrakoodattu kehys, jota seuraa tietty määrä interkoodattuja kehyksiä. Pakkauksen kannalta olisi kuitenkin hyödyllistä käyttää mahdollisimman paljon erodatakehyksiä, jolloin siirrettävän datan määrä saataisiin pysymään mahdollisimman pienenä ja bittinopeus tasaisempina. Intra- ja interkoodattujen kehysten välinen bittinopeus vaihtelee huomattavasti, sillä interkoodattujen kehysten erodatamäärä on paljon pienempi kuin intrakoodattujen. Tämän seurauksena videosignaalia on puskuroitava valmiiksi muistiin tasaisen esittämisenopeuden saavuttamiseksi. [2, s. 10–13, 157.]

3.2 Intra- ja interkoodatut toimenpiteet

Väriredundanssi

Videokuvan väriavaruus koostuu yleensä R-, G- ja B- (Red, Green, Blue) värikomponenteista. Tämä tarkoittaa sitä, että yhden väriyhdistelmän esittämiseksi tarvitaan kolme erillistä numeroarvoa, yksi numeroarvo jokaista värikomponenttia kohden, jotka lisäämällä toisiinsa luovat yhdessä todellisen värin. Jokainen RGB-väriavaruuden komponenteista tallennetaan erikseen kamerassa CCD- tai CMOS-kennon avulla, eli yhdestä kehyksestä on periaatteessa kome erillistä tallennetta värikomponenteille. Jokainen näistä komponenteista on yhtä arvokas todellisen värin esittämiseksi, joten yhden-

kin värikomponentin resoluution muuttaminen vaikuttaisi todelliseen esitettävään väriin. [5, s. 12–16.]

Ihmissilmä on kuitenkin herkempi valon intensiteetille eli luminanssille kuin valon värille. Tämän takia voidaan väriredundanssia poistaa siirtymällä Y'CrCb-väriavaruuteen, joka koostuu yhdestä luminanssikomponentista (Y') sekä värikomponenteista (Cr) ja (Cb). Luminanssikomponentti Y' voidaan laskea R-, G- ja B-komponenttien painotetusta keskiarvosta kaavan 1 mukaisesti.

$$Y' = k_r R + k_g G + k_b B \quad (1)$$

Loppu väri-informaatio voidaan esittää värierokomponentteina, jossa jokainen värikomponentti Cr, Cb ja Cg on erotus R-, G-, B-värikomponenttien ja luminanssin Y' välillä.

$$C_r = R - Y$$

$$C_g = G - Y$$

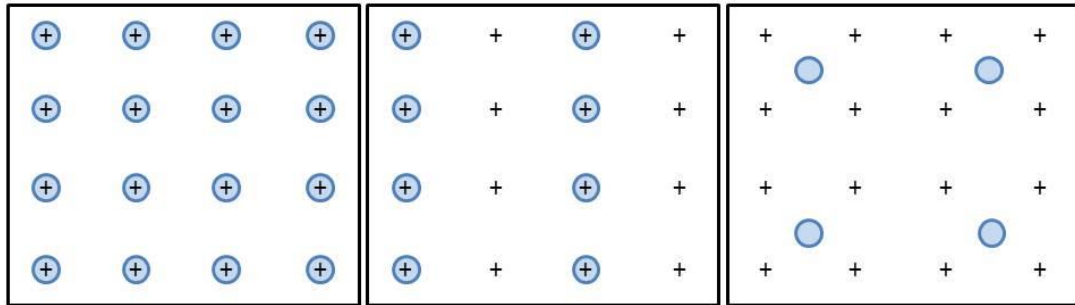
$$C_b = B - Y$$

Nimensä mukaisesti Y'CrCb-väriavaruudessa vain luminanssikomponentti Y' ja värikomponentit Cr ja Cb lähetetään, koska loppuinformaatio saadaan lähetetystä tiedosta laskennallisesti.

RGB-väriavaruuteen nähden todellinen hyöty saadaan siitä, että Y'CrCb-väriavaruuden värikomponentteja voidaan näyttää pienemmällä resoluutiolla kuin luminanssikomponenttia. Näin ollen väri-informaatio pystytään esittämään pienemmällä datamäärällä, ilman että sillä on huomattavaa vaikutusta kuvanlaatuun.

Y'CrCb-väriavaruuteen siirytään usein kuvanpakkausta ja siirtoa varten, mutta kuvan esittämiseksi se palautetaan takaisin RGB-väriavaruuteen dekooderin avulla. Koska siirrettävien komponenttien arvot ovat vakioita, ne voidaan laskennallisesti palauttaa alkuperäisiin muotoihinsa ilman, että informaatiota katoaa. [5, s. 12–16.]

Kuvassa 5 on esitetty Y'CrCb-väriavaruuden yleisimmät näytteistystavat, joita käytetään muun muassa MPEG4-videonpakkauksessa.



Kuva 5. Y'CrCb näytteistystavat 4:4:4, 4:2:2 ja 4:2:0 [5, s. 15].

Numeroarvo näytteistystavan nimessä kertoo vaaka-akselilla tapahtuvan näytteenotto-taajuuden. 4:4:4-näytteistystavassa jokaisella värikomponentilla on sama resoluutio, joten ne esiintyvät kaikissa pikseleissä samanarvoisesti. 4:2:2-näytteistystavassa väri-komponenteilla on sama resoluutio pystyakselilla, mutta vain puolet luminanssikomponentin resoluutiosta vaaka-akselilla. 4:2:0-näytteistystavassa värikomponenteilla on puolet luminanssikomponentin resoluutiosta sekä pysty- että vaaka-akselilla. Näistä kolmesta näytteistystavasta 4:2:0 on eniten käytetty, sillä se tarjoaa kaikista suurimman pakkauksen datamäärässä. [5, s. 12–16.]

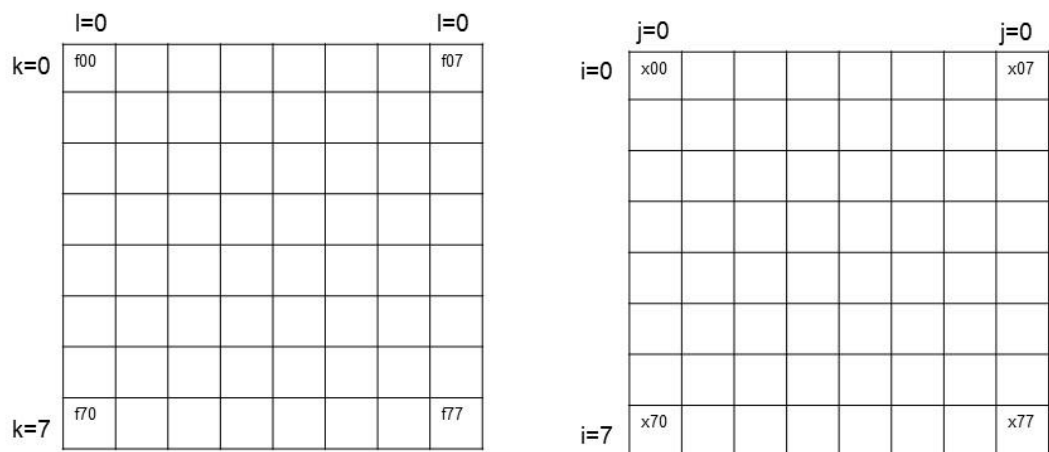
Diskreetti kosinimuunnos (DCT)

Diskreetin kosinimuunnoksen tarkoituksena ei ole niinkään pakata kuvadataa, vaan saattaa se muotoon, jossa redundanssia on helpompi havaita, käsitellä ja poistaa. Yksittäinen kuva jaetaan yleensä 8 x 8 pikselin kokoiisiin kuvalohkoihin, joihin DCT-muunnos kohdistetaan. Tällaista 8 x 8 -kuvalohkoa kutsutaan myös englanninkielisellä nimellä ”macroblock”.

DCT-muunnos muuttaa kuvan paikka-asteikolta taajuusasteikkoon, minkä jälkeen pikseleissä esiintyviä kosinikomponentteja voidaan käsitellä matemaattisesti matriisina. Pieneen alueeseen keskittyminen poistaa suuren puskurimuistin tarpeen, ja laskutoimitus pysyy kevyempänä. Alue, jolle DCT-muunnos suoritetaan, ei ole täysin sidottu 8 x 8 -kuvalohkon muotoon. Erikokoisia kuvalohkoja voidaan käyttää videokuvan resoluution mukaan. [2, s. 61–89; 1.]

DCT-muunnoksella saavutetut kertoimet asettuvat kuvalohkon alueella niin, että matalimmat taajuudet asettuvat neliön vasempaan yläkulmaan ja korkeimmat taajuudet oikeaan alakulmaan. Vasemmassa yläkulmassa sijaitseva arvo on kaikista tärkein, sillä se on koko kuvalohkon kirkkauden keskiarvo. Tätä arvoa kutsutaan DC-kertoimeksi. Kuvalohkon muita horisontaalisia ja vertiaalisia arvoja kutsutaan AC-kertoimiksi. AC-kertoimien lukuarvot muuttuvat sen mukaan, missä ne sijaisevat kuvalohkon alueella, ja taajuuden kasvaessa lukuarvo yleensä pienenee.

Korkeat taajuudet sisältävät paljon redundanssia yleensä yksityiskohtien muodossa. Nämä yksityiskohdat ovat kuitenkin niin hienovaraisia, ettei ihmissilmä pysty erottamaan niitä liikkuvasta videokuvasta. Näin ollen ne voidaan poistaa kuvalohkosta kvantisoinnin avulla. Muunnoksen jälkeen on kuitenkin tärkeää, ettei kuvalohkon reunoja pystytä havaitsemaan ja kuva näyttää edelleen saumattomalta. [2, s. 61–89; 1.] Diskreetin kosinimuunnoksen matemaattinen kaava on esitetty kuvassa 6.



$$f(k, l) = \frac{1}{4} C_k C_l + \sum_{i=0}^7 \sum_{j=0}^7 x(i, j) \cos \frac{\pi(2i+1)k}{16} \cos \frac{\pi(2j+1)l}{16} \quad (2)$$

- u, v, x, y on $0, 1, 2, \dots, N-1$
- $C(0), C(0)$ $1/\sqrt{2}$
- $C(u), C(v)$ 1 , kun $u, v = 1, 2, \dots, N-1$
- x, y alkuperäisen lohkon koordinaatit
- u, v muunnetun lohkon koordinaatit
- $f(x, y)$ alkuperäisen lohkon pisteen (x, y) arvo
- $f(u, v)$ muunnetun lohkon kertoimen (u, v) arvo

Kuva 6. Diskreetin kosinimuunnoksen matemaattinen kaava [2, s. 88].

Kvantisointi

Kun kuvalohkon alueelta on saatu pikselien matemaattiset arvot esiin, se voidaan kvantisoida käyttäen kvantisointimatriisia. Toimenpiteen tarkoituksena on saattaa mahdollisimman moni korkeilla taajuuksilla sijaitseva AC-kerroin nolaksi jakamalla se tietyllä kvantisointiarvolla. Jakamisen jälkeen saatu tulos pyöristetään lähimpään kokonaislukuun. Vaikka DCT ja kvantisointi ovat periaatteessa häviöttömiä pakkausmenetelmiä, kvantisoinnin lopussa tapahtuva arvojen pyöristys hukkaa osan datasta eikä sitä voida tarkasti palauttaa edes käänteisellä kvantisoinnilla.

Kvantisoinnissa apuna käytettävät kvantisointimatriisit ovat yleensä sidottuja tiettyyn videopakkausstandardiin. MPEG4-standardin tapauksessa käytössä on kaksi erillistä kvantisointimatriisia. Ensimmäinen matriisi on käytössä intrakoodatuille kuvalohkoille ja toinen interkoodatuille kuvalohkoille. Käyttäjän on myös mahdollista määrittää omat kvantisointimatriisinsa, jolloin ne on myös sisällytettävä dekooderille lähetettävään dataan. [14.] Kuvassa 7 on esitetty MPEG4-standardin mukaiset intra- ja interkoodatuille kehyksille tarkoitetut kvantisointimatriisit.

8	17	18	19	21	23	25	27
17	18	19	21	23	25	27	28
20	21	22	23	24	26	28	30
21	22	23	24	26	28	30	32
22	23	24	26	28	30	32	35
23	24	26	28	30	32	35	38
25	26	268	30	32	35	38	41
27	28	30	32	35	38	41	45

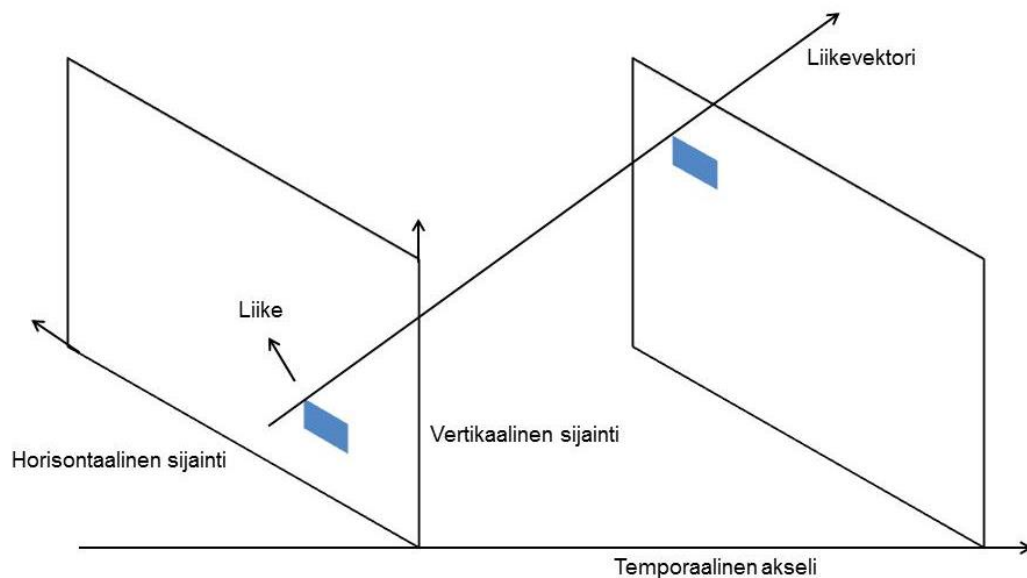
16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

Kuva 7. MPEG4-standardin mukaiset kvantisointimatriisit. Vasemmalla matriisi intrakoodatuille kuvalohkoille ja oikealla interkoodatuille kuvalohkoille. [14.]

Kvantisointimatriisin avulla voidaan suoraan vaikuttaa kuvan pakkauksen laatuun. Käyttämällä suurempia kvantisointiarvoja voidaan useampi AC-kertoimen arvo saattaa nolaksi. Kvantisointimatriisissa käytettävien numeroarvojen suuruus on siis suoraan verrannollinen kuvan laatuun. Kvantisointia voidaan myös keskittää tiettyihin kuvalohkon taajuusarvoihin, jolloin korkeilla taajuuksilla sijaitsevia kuvanlaadun kannalta vähemmän tärkeitä arvoja kvantisoidaan enemmän. [5, s. 87.]

pakkauksen interkoodattu prosessi, joka mittaa objektien liikettä eri kehysten välillä. Objektien liikkeiden mittaamisesta on huomattavaa hyötyä, kun kuvista etsitään redundanttista informaatiota.

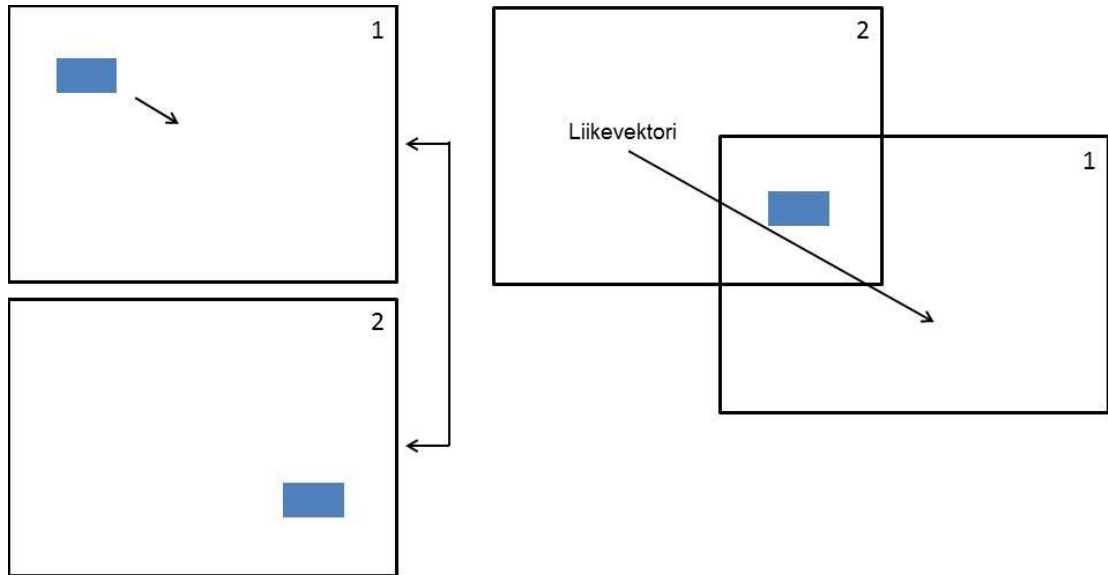
Staattisen objektin tapauksessa liikevektori on temporaalisen akselin suuntainen. Kuvassa liikkuvan objektin tapauksessa liikettä tapahtuu vektorin suuntaisesti, joka ei ole täysin temporaalisen akselin suuntainen. Objektin liikkeen havaitsemiseksi DCT-makroblokkeja verrataan toisiinsa peräkkäisten kehysten välillä kuvan 10 esittämällä tavalla. Kehysten välillä tapahtuva objektin liike tallennetaan paikallisesti liikevektoriin, jotta sitä voidaan käyttää uudestaan liikkeen etsimisessä seuraavien kehysten kanssa. Liikevektorit koostuvat kahdesta komponentista: suunnasta ja pituudesta, joita käytetään objektin paikantamiseen seuraavassa kehyksessä. [2, s. 11–15.]



Kuva 10. Objektin liikevektorin muodostuminen kahden peräkkäisen kehysten välillä. Liikevektori on temporaalisen akselin suuntainen, kun objekti on staattinen [2, s. 12].

Liikevektoria käytetään hyödyksi liikkuvan objektin redundanssin havaitsemiseksi. Kahden peräkkäisen kehysten vertaaminen toisiinsa ilman liikevektoria paljastaa paljon redundanssia staattisessa taustassa, mutta jos peräkkäisiä kehymiä siirretään ennen vertailun tekemistä liikevektorin suunnan mukaisesti, huomataan että, objektia esittävässä datassa on paljon redundanssia. Kuvassa 11 esitetään, kuinka redundanssin havaitseminen liikkuvasta objektista helpottuu liikevektorin avulla.

Liikkeen kompensoiminen on videonpakkauksen raskain ja aikaavievin osuus, sillä liikevektoreiden laskutoimitukset suoritetaan 8×8 -kuvalohkon sisällä sijaitseville yksittäisille pikseleille. Pakkaustoimenpiteet on kuitenkin suunniteltu niin, että raskain työ tehdään pakkauksen aikana, jotta purku voidaan suorittaa nopeasti käänteisenä dekooderin avulla. [2, s. 11–15; 5, s. 28–31.]



Kuva 11. Liikkeen kompensoinnissa peräkkäisiä kehyksiä siirretään liikevektorin avulla objektin redundanssin havaitsemiseksi [2, s.12].

Liikkeen kompensointi toimii kuvanpakkausprosessissa seuraavanlaisesti. Ensin intrakoodattu I-kehys lähetetään mutta tallennetaan samalla puskurimuistiin käytettäväksi referenssinä seuraavan kehyksen kanssa. I-kehystä verrataan seuraavana vuorossa olevaan kehykseen objektien liikevektoreiden laskemiseksi. Seuraavaksi I-kehystä siirretään liikevektorin mukaisesti objektin paikallisen liikkeen estämiseksi ja kehyksien välisen erodatan luomiseksi. Tulokseksi saatu erodata ja liikevektori lähetetään ja lisätään jo aikaisemmin lähetettyyn I-kehykseen. Tulokseksi saadaan erodataan ja liikkeenkompensointiin perustuva P-kehys. Kun haluttu määrä P-kehyksiä on lähetetty, lähetetään uusi I-kehys ja koko prosessi toistetaan.

Erodataan perustuvia P-kehyksiä voidaan myös käsitellä yksittäisinä kuvina, minkä vuoksi niihin voidaan käyttää muitakin aikaisemmin mainittuja intrakoodattuja kuvanpakkaustekniikoita siirtodatan pienentämiseksi. [2, s. 11–15, 157.]

I- ja P-kehysten lisäksi on mahdollista käyttää B-kehystä. B-kehysten toiminta perustuu erodataan sitä edeltävän ja seuraavan kehysten kanssa. B-kehys ei voi toimia referenssikuvana erodatalle, ja sen on sijaittava I- ja P-kehysten välissä. B-kehysten siirtämiseen tarvittava bittimäärä on näistä kolmesta kaikkein pienin, mutta se on myös niistä vähiten käytetty.

MPEG2-standardissa liikkeenkompensointi on rajattu yhteen liikevektoriin kuvalohkoa kohden. Liikevektorin tarkkuus on myös rajattu puoleen pikseliin. Uudemmassa MPEG4-standardissa liikevektoreita voi olla enintään neljä kuvalohkoa kohden, ja näiden liikevektoreiden tarkkuus on rajattu neljäsosapikseliin. [5, s. 35.]

3.3 Entropian koodaus

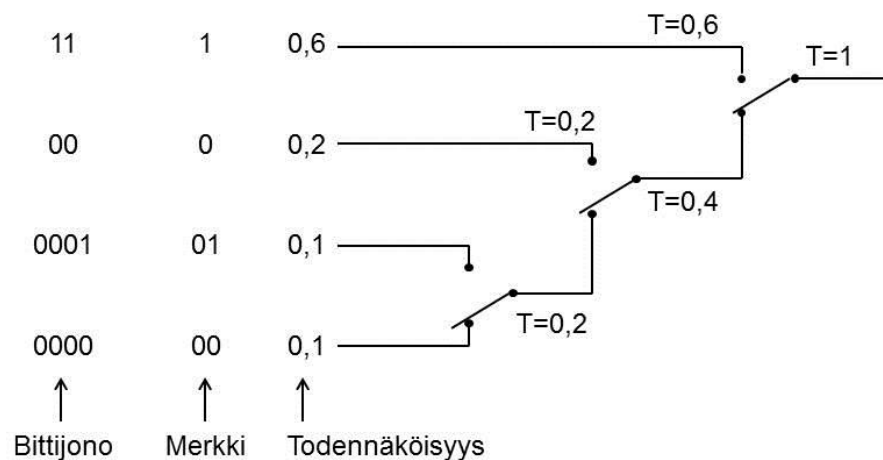
RLC

Jononpituuskoodauksen (RLC) tarkoituksena on lyhentää bittijonon pituutta lisäämällä jonoon kertoimia. Sen sijaan, että lähetettäisiin jonossa kymmenen nollaa peräkkäin, voidaan jonon eteen lisätä kerroin 10. Näin bittijono 0000000000 muuttuu muotoon [10]0. Bittijonoa saadaan lyhennettyä ja muokattua helpommin käsiteltävään muotoon. Jononpituuskoodauksesta saatua tulosta pyritään vielä pakkaamaan pienempään muotoon käyttämällä apuna vaihtuvamittasta koodausta. Jononpituuskoodaus ja vaihtuvamittainen koodaus ovat häviöttömiä pakkausmenetelmiä. Vaikka bittijonon ulkonäkö muuttuu, sen sisältö pysyy kuitenkin samana. [5, s. 60–62.]

VLC

Vaihtuvamittaisessa koodauksessa (VLC) yleisimmin bittijonossa esiintyvät merkit korvataan lyhyemmillä bittijonoilla ja harvemmin esiintyvät merkit pidemmillä bittijonoilla. Vaihtuvamittainen koodaus siis perustuu samoihin toimintaperiaatteisiin kuin vuonna 1838 kehitetty morsekoodaus. Esimerkiksi englannin kielessä yleisimmin esiintyvää kirjainta "e" vastaa yksi piste, kun taas harvoin esiintyvää kirjainta "z" vastaa pidempi ja monimutkaisempi merkkijono. Tällainen koodausmenetelmä perustuu kuitenkin vahvasti ennakkotietoon käytettävästä koodikielestä. Morse-esimerkin tapauksessa suomessa yleisin kirjain on "a", joten englanninkielinen morsetaulukko ei olisi suoraan verrannollinen suomen kielen morsetaulukkaan. [2, s. 4–7 ; 4.]

Sama periaate pätee myös bittimaailmassa, ja siksi pakattavaa bittijonoa on analysoida ennen koodaustaulukon luomista. Bittijonon analyysin perusteella suurella todennäköisyydellä esiintyvät arvot korvataan lyhyellä merkkijonolla. Todennäköisyyden laskeissa arvot saavat käyttöönsä pidempiä merkkijonoja. Tuloksena saatu vaihtuvamittainen bittijono ja koodaustaulukko toimivat yhdessä, joten taulukko on sisällytettävä siirrettävään dataan, jotta dekooderi pystyy purkamaan bittijonon oikein. [2, s. 4–7.] Kuvassa 6 esitetään David A. Huffmanin kehittämän vaihtuvamittaisen koodauksen toimintaperiaate.



herkkyyttä on pystytty parantamaan käyttämällä standardin mukaisia pakkaustaulukoi-
ta. MPEG4-pakkauksen yhteydessä käytetään 102 bittijonoyhdistelmää sisältävää
koodaustaulukkoa. Nämä yhdistelmät sisältävät bittijonon lopuksi niin sanotun merkki-
bitin "s", joka voi olla kertoimesta riippuen 1 tai 0. Jos kyseistä bittijonoa ei löydy merk-
kitaulukosta, sen tilalla käytetään erikoismerkkiä (ESCAPE) ja 13 bittiä koodaamatonta
bittijonoa. [5, s. 62–64.]

Aritmeettinen koodaus

Vaikka Huffmanin koodaukseen perustuvat vaihtuvamittaiset koodausmetodit pystyvät
pakkaamaan bittijonoa pienempään muotoon, ne ovat silti kaukana optimaalisesta pak-
kassuhteesta. Koodauksen teho nojaa vahvasti koodattavaan sisältöön, ja vaihtuvamit-
taisessa koodauksen teho alkaakin laskea, kun pakattavan bittisarjan esiintymistoden-
näköisyys ylittää 50 %.

Aritmeettinen koodaus tarjoaa vaihtoehdoisen lähestymistavan optimaalisen pakkaus-
tehon saavuttamiseksi. Ensin bittijonolle määrätään etäisyys numeroiden nolla ja yksi
välille. Tämän jälkeen bittijonossa esiintyvät bittisarjat saavat itselleen oman alialueen
bittijonon etäisyydeltä. Tämän alialueen koko perustuu bittisarjan esiintymistodennä-
köisyyteen bittijonon sisällä. Alialue määräytyy minimi- ja maksimiarvosta (L ja H), joi-
den välillä bittisarjan on sijaittava.

Bittisarja	Esiintymistodennäköisyys	Alialue
11	0,2	0,1–0,3
00	0,4	0,3–0,7
1111	0,1	0,9–1,0
0000	0,3	0,7–0,9

Aritmeettisen koodauksen kulku bittijonolle 00 11 00 1111

Toimenpide	Alue (L-H)	Bittisarja	Alialue (L-H)
1. Määritetään etäisyys.	0–1,0		
2. Etsitään bittisarjaa vastaava alialue.		00	0,3–0,7
3. Asetetaan uudeksi etäisyydeksi edellisen bittisarjan alialue.	0,3–0,7		
4. Etsitään uutta bittisarjaa vastaava alialue.		11	0,1–0,3
5. Asetetaan uudeksi etäisyydeksi alialue edellisen etäisyyden sisällä.	0,34–0,42		
6. Etsitään uutta bittisarjaa vastaava alialue.		00	0,3–0,7
7. Asetetaan uudeksi etäisyydeksi alialue edellisen etäisyyden sisällä.	0,364–0,396		
8. Etsitään uutta bittisarjaa vastaava alialue.		1111	0,9–1,0
9. Asetetaan uudeksi etäisyydeksi alialue edellisen etäisyyden sisällä.	0,3928–0,396		

Bittijonoa koodattaessa jokaisen uuden bittisarjan kohdalla alialue pienenee. Edellisessä esimerkissä lopullinen bittijonon L–H alue saavutetaan neljännen bittisarjan jälkeen. Koko bittijono voidaan nyt esittää millä tahansa murtoluvun arvolla välillä 0,3928–0,396. [5, s. 65–68.]

3.4 Dekoodaus

Edellä käytiin läpi toimenpiteitä, jotka mahdollistavat videokuvan pakkauksen pienempään tilaan siirtämisen ja käsiteltävyyden parantamiseksi. Nyt pakattu video on saapunut määränpäähensä ja pakkaus on purettava dekooderin avulla videokuvan näyttämiseksi. Koska pakkauksessa suoritettujen toimenpiteiden kulku on myös tallennettu pakattuun dataan, purku voidaan suorittaa käänteisesti alkuperäisen datan palauttamiseksi.

Käänteisessä kvantisoinnissa kuvalohkon alkuperäiset DC- ja AC-arvot palautetaan käyttämällä samoja kvantisointimatriiseja kuin pakkauksessa, mutta käänteisesti. Jos käyttäjä määritteli pakkauksessa omat kvantisointimatriisinsa, ne on kuljetettava videodatan mukana dekooderille purkamisen mahdollistamiseksi. Standardina käytettävät matriisit ovat dekooderilla valmiina, joten niitä ei tarvitse lisätä siirrettävään videodataan pakkausvaiheessa.

Käänteisen kvantisoinnin jälkeen DCT-kuvalohko palautetaan alkuperäiseen muotoonsa taajuusasteikolta paikka-asteikolle käyttäen käänteistä DCT-kaavaa 3. [5, s. 104–111.]

$$f(u, v) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C_u C_v F(u, v) \cos \frac{\pi(2x+1)u}{2N} \cos \frac{\pi(2y+1)v}{2N} \quad (3)$$

Kuva 13. Käänteinen DCT-kaava palauttaa kuvalohkon sen alkuperäiseen muotoon [5, s. 43].

Viimeinen vaihe dekooderilla on liikkeenkompensoinnin purku. Tässä vaiheessa P-kehysten erodata lisätään aikaisempaan referenssikuvaan liikevektoreiden avulla.

4 Laadunmittaus ja pilvipalvelut

4.1 Laadunmittaus

Standardinmukainen kuvanlaatu on lähtökohtaisesti sidottu kuvan resoluutioon. Tämä tarkoittaa, että HD-standardin täyttävässä kuvassa täytyy alkujaan olla 1920 x 1080

pikseliä yhtä kuvaa kohden. Pakkaamaton HD-kuva toimii näin ollen referenssinä, kun pakatun videon kuvanlaatua aletaan mitata. Hyvänä sääntönä voidaan pitää kuvan esittämiseen tarvittavaa bittimäärää. Mitä enemmän kuvassa on bittejä, sitä paremmalta se näyttää.

Pakatun videokuvan laatua voidaan mitata joko matemaattisesti tai aistinvaraisesti. Matemaattisessa mittaustavassa pakattua videokuvaa verrataan pakkaamattomaan referenssivideoon algoritmin avulla. Yksi eniten käytetty matemaattinen laadunmittausmenetelmä on PSNR (Peak Signal to Noise Ratio).

PSNR toimii logaritmisella asteikolla, ja sillä verrataan referenssivideon ja pakatun videon kehysten välistä MSE (Mean Squarer Error) arvoa. PSNR-arvo saadaan kaavan 4 mukaisella laskutoimituksella.

$$PSNR_{db} = 10 \log_{10} \frac{(2n - 1)^2}{MSE} \quad (4)$$

- MSE = signaalin näytekohdan suurimman arvon neliöjuuri
- n = näytekohdassa käytettävien bittien määrä

PSNR-arvon laskeminen on nopea toimenpide, ja siksi se onkin suosittu matemaattinen mittausten menetelmä. Tyypillinen PSNR-arvo häviöllisesti pakatulle videotiedostolle on 30–50 dB, jossa suurempi luku tarkoittaa parempaa laatua. [16; 5, s. 20–14.]

Matemaattiseen mittaukseen perustuvat operaatiot eivät kuitenkaan anna absoluuttista totuutta kuvan todellisesta laadusta, sillä se ei voi jäljitellä ihmisten havainnointikykyä. Tämän takia videomateriaalin laatua myös mitataan aistinvaraisesti käyttäen hyväksi esimerkiksi DSCQS (Double Stimulus Continuous Quality Scale) -menetelmää.

DSCQS-menetelmässä videokuvan laaduntarkkailijalle esitetään lyhyitä videosekvenssejä. Laaduntarkkailijan on arvioitava videosekvenssien laatua tietämättä, kumpi niistä on pakkaamaton referenssivideo ja kumpi pakattu video. Satunnaisessa järjestyksessä esitettävien videosekvenssien laatua arvioidaan asteikolla huonosta erinomaiseen.

Aistinvaraisesti kuvanlaadun mittaaminen on kuitenkin melko hankalaa, koska ihmisten aistit ja havainnointikyky ovat hyvin yksilöllisiä. Lisäksi aistinvaraisessa mittauksessa on myös tekijöitä, jotka vaikuttavat tuloksen luotettavuuteen. On osoitettu, että tarkkailuympäristön valaistus, mukavuus ja rauhallisuus vaikuttavat tulokseen positiivisesti, kun vastakohtaisesti epämukava ja häiritsevä ympäristö vaikuttaa tulokseen negatiivisesti. Ympäristöllä on jopa niin suuri vaikutus, että joissain tapauksissa on huomattu sen vaikuttavan tulokseen enemmän kuin itse kuvan tarkkuus. [5, s. 20–24.]

Myös laaduntarkkailijan mielentila vaikuttaa kuvanlaadun arvioinnin tulokseen. Tarkkailija voi joko aktiivisesti seurata kuvaa ja keskittyä tarkasti tiettyihin yksityiskohtiin tai tarkastella kuvaa passiivisesti keskittymällä koko kuvaan kerrallaan. Esimerkkinä voidaan mainita videosekvenssi, jossa esiintyy ihmiskasvoja. Aistimme on ohjelmoitu niin, että kiinnitämme aina enemmän huomiota ihmisten kasvoihin ja sen piirteisiin kuin ympäristöön. Kärjistetyksi sanottuna videosekvenssin taustalla voisi käytännössä tapahtua mitä vain, emmekä me huomaisi sitä. Tällaisen kohtauksen mittaaminen matemaattisesti saattaisi antaa huononkin tuloksen kuvanlaadun arvioinnissa, vaikka aistinvaraisesti se voi näyttää hyvältä.

Edellä mainittujen laadunmittauksien suorittamiseksi tarvitaan kuitenkin videon pakkaamaton lähdemateriaali referenssiksi. Joissain tapauksissa on mahdollista, että lähdemateriaalin saaminen tai käyttäminen on mahdotonta tai ei pystytä todistamaan, että saatavilla oleva materiaali on häiriötöntä. Laadullisia mittauksia voidaan suorittaa ilman referenssimateriaalia tietyillä mittausten menetelmillä ja algoritmeilla, mutta parhaassakin tapauksessa nämä mittaukset antavat tulokseksi hyvin epätarkan arvion kuvan laadusta.

Pakkaussuhde ja laatu eivät kuitenkaan kulje käsi kädessä. Videonpakkaus onkin tasapainottelua laadun ja videotiedoston koon välillä. Ei ole mahdollista saada parasta molemmista. Tärkeää onkin käyttää sopivaa pakkaussuhdetta kyseiseen videomateriaaliin ja löytää kultainen keskitie laadun ja koon välillä. [5, s. 20–24.]

4.2 Pilvipalvelut

Youtuben, Vimeon ja Netflixin kaltaiset palvelut eivät pystyisi kustannustehokkaasti toimimaan ilman videonpakkausta. Kuten aikaisemmin todettiin, on pakkaamattoman videon siirtäminen, käsitteleminen ja tallentaminen epäkäytännöllistä ja kallista.

Suoratoistaminen voidaan suorittaa kahdella tapaa. Progressiivinen suoratoisto vaati koko videomateriaalin lataamisen käyttäjän koneelle, ennen kuin se voidaan esittää. Suuria videotiedostoja toistettaessa progressiivinen suoratoistaminen on hyvin epäkäytännöllistä ja sen tehokkuus on hyvin paljon riippuvainen käyttäjän laitteen tiedonsiirtonopeudesta. [8.]

Tosiaikainen suoratoistaminen mahdollistaa videomateriaalin tarkastelun, ennen kuin se on kokonaan ladattu. Tämä mahdollistetaan puskurimuistin avulla, jonne videodataa ladataan lisää samalla, kun videota toistetaan. Tosiaikaisesti suoratoistetussa videossa käyttäjä pystyy myös hyppäämään videon eri ajankohtaan kesken videon latauksen. Tämä voi kuitenkin aiheuttaa hetkellisen viiveen videon toistamisessa, kunnes puskurimuistiin ladataan taas tarpeeksi dataa. [8.]

Kun käyttäjä lataa oman videonsa palveluntarjoajan videopalvelimelle, se pakataan palvelimella uudestaan vielä pienempään muotoon säilytystä varten. Säilyttääkseen mahdollisimman hyvän kuvanlaadun käyttäjän kannattaa pakata videota kevyesti tai käyttää kokonaan häviötöntä pakkausmentelmää ennen videon latausta. Youtuben tapauksessa videon uudelleenpakkaamiseen ei käytetä yhtä tiettyä kooderia, vaan koodereita on käytössä monia ja niistä valitaan paras käytettäväksi ladattavan videon spesifikaatioiden mukaan. Yleisimmin käytetty koodekki on H.264, mutta Youtube on yrittämässä siirtymistä avoimeen lähdekoodiin perustuviin koodekkeihin, kuten VP9. [8; 12.]

Verkon välityksellä tapahtuvan tiedonsiirron nopeus saattaa vaihdella runsaasti kesken videon suoratoiston. Tämän takia videon bittinopeutta pystytään tosiaikaisesti muuttamaan kesken toiston. Bittinopeuden muuttamisen avulla videon toistoa pystytään jatkamaan, vaikka tiedonsiirtonopeus laskisi hetkellisesti. [6; 8.]

5 Videopakkausstandardit

Pakattu videomateriaali koostuu käytännössä neljästä eri osa-alueesta. Videoinformaatiosta, ääni-informaatiosta, metatiedoista ja paketoinnista (container). Video ja ääni ovat yksiselitteisiä, ja ne sisältävät pakatun videomateriaalin video- ja ääniraidat. Metatiedoissa on informaatiota videosta, kuten: käytetty kooderi, kehysnopeus (fps), tekstitys tai navigaatio videon sisällä. Paketointi sitoo video-, ääni- ja metatiedostot yhteen ja muodostaa digitaalisen videotiedostopakettin. Selvimmin paketointi on nähtävillä videotiedoston nimen päätteellä, esimerkiksi: "video.mp4". [15.]

Kooderi ja paketointi tuovat hieman epäselvyyttä standardoinnin puutteesta kärsivään digitaalisen videon maailmaan, sillä videotiedostossa käytetty paketointi ei välttämättä kerro meille mitään tiedoston sisällöstä. Monet paketointityypit tukevat useita eri koodereita ja saattavat aiheuttaa ongelmia kun videotiedostoa yritetään toistaa. Käytössä oleva videosoitin saattaa tukea videossa käytettävää paketointia, mutta ei sen sisältävää kooderia. [7.]

H.264

H.264 tai MPEG4 on MPEG (Moving Picture Experts Group) -standardiin perustuva teräväpiirtokuvalla tarkoitettu kooderikoodi, joka kuuluu AVC- (Advanced Video Coding) paketoinnin ryhmään. H.264 kooderikoodi tuli standardiksi Blu-ray-levylle ja on edelleen yleisimmin käytetty kooderikoodi videopakkausissa. H.264 pystyy käyttämään pakkausissa hyvin pientä tai suurta bittinopeutta, joten se soveltuu sekä teräväpiirtomateriaalin ja verkkomateriaalin pakkaamiseen. [7.]

H.265

H.265 on UHD-kuvamateriaalille kehitetty kooderi HEVC- (High Efficiency Video Coding) paketoinnin ryhmässä, ja se pystyy samalla bittinopeudella parhaimmillaan tarjoamaan 200 % paremman pakkaussuhteen kuin H.264 ja tukee UHD-kuvaformaatteja parhaimmillaan 8192 x 4320 -resoluutioon asti. Tämä mahdollistetaan muun muassa suurempien kuvablokkien (64 x 64) ja tehokkaamman aritmeettisen koodauksen avulla. [11.]

VP9

VP9 on Googlen ja WebM:n yhteistyössä kehitettävä vapaan lähdekoodin kooderi, joka on suunniteltu kilpailemaan H.265-kooderin kanssa. VP9 ei kilpaile HEVC-standardien kanssa pääasiassa kuvanlaadussa, vaan juuri tekijäoikeusmaksuissa. Vaikka VP9 ei kuvanlaadullisesti pääsekään aivan standardoitujen pakkaustekniikoiden tasolle, se on silti DSCQS-kuvanlaatumittauksissa saanut hyvää palautetta kuvanlaadusta. Vapaan lähdekoodin kooderina se ei ole sidottu HEVC-standardin mukaisiin 0,20 \$/kpl maksuihin, mikä tekee VP9-kooderista varteenotettavan kilpailijan kustannuksia ajatellen. Tämä on yksi syy, miksi Youtube pyrkii siirtymään VP9-kooderin käyttöön. [12; 13.]

6 EEMontti-projektin kulku

Insinööriyön osana tehtiin esittelyvideot EEMontti-projektiin. EEMontti-projekti on vuonna 2012 tehty kehityshanke, jonka tarkoituksena oli etsiä innovatiivisia ratkaisuja pientalojen energiatehokkuuden parantamiseksi. Hanke oli yhteiskunnallisesti merkittävä, sillä lähes viidennes Suomen energian loppukäytöstä kuluu asuintalojen lämmittämiseen. Hankkeen koordinaattorina toimi Green Net Finland ry. [9.]

EEMontti-hankkeen kohteiksi valittiin neljä pientaloasuntoa pääkaupunkiseudulta. Näihin neljään pientaloastunoon järjestettiin tarjouskilpailu syksyllä 2011, ja tarkoituksena oli löytää paras kokonaisratkaisu energiankulutuksen puolittamiseen jokaisessa kohdekiinteistössä. Tarjouskilpailu oli avoin kaikille energiaremontin tarjoajille. Asiantuntijaraati valitsi kilpailuun saapuneista tarjouksista parhaan jokaista asuntoa kohden ja suosittelee sitä asunnon omistajalle. Hankkeen ulkopuolisten ongelmien vuoksi vain kaksi neljästä mukana olleista kiinteistöistä päätyi lopulta mukaan energiaremonttiin. [9.]

Hankkeelle oli tarkoitus saada medianäkyvyyttä julkaisemalla jokaisesta kohteesta ja remontista esittelyvideo. Nämä videot olisivat lopulta nähtävillä projektin verkkosivuilla sekä mahdollisesti Blu-ray-julkaisuna.

6.1 Projektin videoiden määrittely

Projektin alkaessa ei videoiden tuottamiseen ollut juuri kiinnitetty huomiota. Näin ollen neljästä opiskelijasta koostuva projektiryhmämme sai vastuulleen videoiden käsikirjoituksen ja toteutuksen suunnittelun. Alustava suunnitelma videoiden toteuttamisesta valmistui nopeasti kohdekiinteistöissä vierailemisen jälkeen.

Koska videoiden pääpaino oli verkkojulkaisussa, päätettiin videot pitää lyhyinä ja informatiivisina. Lopulliseksi pituudeksi päätettiin noin 2–5 minuuttia. Internet-käyttäytyminen on yleensä hyvin lyhytjänteistä, eikä pitkiin videoihin jakseta kiinnittää huomiota. Videoiden julkaisemisessa päätettiin käyttää hyväksi Vimeo-pilvipalvelua, jonka kautta valmiit videot upotettaisiin hankkeen kotisivuille.

Videossa esiintyvä informaatio päätettiin esittää kuvan ja tekstin avulla. Kertojaäänen käyttäminen videossa olisi ollut haastavaa projektin aikataulun takia. Lisäksi ajattelimme informaation jäävän paremmin mieleen ilman häiritsevää puhujaa. Ainoa puhe, jota videoilla kuullaan, on omistajien kommentteja remontin kulusta ja sen tuloksista.

Kuvattavista kiinteistöistä päätettiin ottaa referenssikuvaa käytössä olevista lämmitysjärjestelmistä ja ratkaisuista ennen remonttia. Samalla kuvaisimme haastattelun muodossa asuntojen omistajien mielteitä tulevasta remontista. Remontin aikana kävisimme kuvaamassa remontin kulkua ja eri työvaiheita. Valmiin remontin jälkeen palaisimme jälleen kuvauspaikalle ottamaan videokuvaa lopputuloksista sekä uudet haastattelut omistajilta.

Koska vain kaksi kiinteistöä päätyi mukaan varsinaiseen projektiin, päätettiin lisäksi tehdä myös esittelyvideo energiaremontin tärkeydestä ja tavoitteista. Yhteensä videoita tehtiin siis kolme, ja ne ovat nähtävillä osoitteessa <<http://www.eemontti.fi>>.

6.2 Kuvaus

Kuvaukset tehtiin digitaalisella järjestelmäkameralla Canon 5D mark 2. Järjestelmäkameralla pääsimme tavallista videokameraa parempaan kuvanlaatuun, koska 5D mahdollisti 1080p-HD-videokuvan tallentamisen. Päädyimme 1080p-kuvanlaatuun mahdol-

lisen Blu-ray-julkaisun takia. Kuvaamalla parhaalla mahdollisella kuvanlaadulla voimme soveltaa videokuvaa moniin eri julkaisukanaviin ja säilyttää silti laadukkaan ulkonäön.

Järjestelmäkamerassa on myös mahdollisuus käyttää erilaisia objektiiveja haluttujen kuvauksellisten tehosteiden saavuttamiseksi. Tämä ei ollut mahdollista senhetkisillä käytössämme olleilla tavallisilla videokameroilla. Tavallisen kuvamateriaalin lisäksi käytettiin myös tehostekuvia, esimerkiksi 3D-grafiikkaa ja time-lapse-kuvia. Tehostemateriaalin luomisessa oli huomioitava lähdemateriaalin resoluutio ja tuotettava tehostemateriaali käyttäen samaa resoluutiota videokuvan kanssa.

Käytimme apuna myös steadicam-järjestelmää, jonka tarkoituksena on mukailla kuvaa-
jan liikkeitä ja tehdä kameraliikkeistä tasaisempia. Tätä järjestelmää käytettiin apuna varsinkin kamera-ajoissa, jotka usein olivat osana siirtymisessä ulkotiloista sisätiloihin. Steadicam-järjestelmän käyttö aiheutti hieman ongelmia, sillä järjestelmäkameraa ei varsinaisesti ollut suunniteltu käytettäväksi juuri käytössämme ollen steadicamin kanssa. Järjestelmäkameramme oli liian kevyt, joten steadicam ei aina toiminut haluamallamme tavalla.

6.3 Videonpakkaus ja laadunvalvonta

Videot kuvattiin 1080p-HD-resoluutiolla, mikä tarkoittaa, että pakkaamattomana kolme 2–5 minuuttia pitkää videota veisivät tilaa noin 54–135 gigatavua. Pakkaamattoman videon käyttö oli tietysti poissa laskuista, sillä Blu-ray-levyn tallennuskapasiteetti on 25–100 gigatavua riippuen levyn kerrosten määrästä. [10.]

Videonpakkauksessa päätettiin käyttää H.264 pakkausmenetelmää, koska videon julkaisun aikaan se tarjosi parhaan pakkaussuhteen säilyttäen silti hyvän kuvanlaadun. Projektin aikana videon laatua arvioitiin aistinvaraisesti neljän hengen toimesta. Lopullisesta videosta otettiin vertailun vuoksi myös kaksi versiota käyttäen eritasoista pakkausta, jotta nähtäisiin, kuinka pakkausten taso vaikuttaisi lopulliseen laatuun.

Videon satunnaisille kehyksille suoritettiin myös PSNR-mittauksia MATLAB-sovelluksella, jotta pystyisimme vertailemaan aistinvaraisen ja matemaattisen laadun-tarkkailun välisiä tuloksia. Kuvassa 14 esitetään referenssikuva yhdessä kahden pakatun kuvan kanssa. Kuvat ovat kaappauksia samasta kehyksestä, ja niitä on suurennet-

tu 3200 % pikselien näkemisen mahdollistamiseksi. Vasemmanpuoleisin kuva on referenssikuva, jota ei ole pakattu ollenkaan. Keskimmäinen kuva on videosta, jota on pakattu minimaalisesti käyttäen 50–60 Mb/s -bittinopeutta. Oikeanpuoleisin kuva on videosta, jota on pakattu runsaammin käyttäen 10–16 Mb/s -bittinopeutta. Kuvat ovat kooltaan 24 x 24 pikseliä.



Kuva 14. Kolme kuvaa samasta videon kehyksestä käyttäen erisuuruisia pakkauksia. Vasemalla referenssikuva, keskellä 50–60 Mb/s ja oikealla 10–16 Mb/s.

Aistinvaraisesti kuvat näyttävät hyvin samanlaisilta, eikä niiden kuvanlaadullisia eroja voida ihmissilmin havaita.

Kehykselle suoritettu PSNR-mittaus antoi seuraavat tulokset:

Bittinopeus	PSNR	SNR
50–60 Mb/s	22,7015 dB	19,5913 dB
10–16 Mb/s	22,6603 dB	19,5501 dB

Häviöllisesti pakattujen kehysten välillä tyypillinen tulos on 30–50 dB, mikä tarkoittaisi, että matemaattisesti tarkasteltuna kuvanpakkauksen laatu on normaalia huonompi. Aistinvaraisesti tarkasteltuna kuvan laatu näyttää hyvältä, mikä on ristiriidassa PSNR-mittausten kanssa. Kuvassa 14 esimerkkinä käytetyt kuvakaappaukset on esitetty suuremmissa koossa liitteessä 1.

Kuvassa 15 on esitetty referenssikuva kahden pakatun kuvan rinnalla. Nämä kuvat ovat eri kehyksestä, ja myös niille suoritettiin PSNR-mittaukset. Tämän kehyksen kohdalla tulokset olivat seuraavanlaiset:

Bittinopeus	PSNR	SNR
50–60 Mb/s	48,5398 dB	43,6634 dB
10–16 Mb/s	44,1526 dB	39,2762 dB



Kuva 15. Kolme kuvaa samasta videon kehyksestä käyttäen erisuuruisia pakkauksia. Vasemalla referenssikuva, keskellä 50–60 Mb/s ja oikealla 10–16 Mb/s.

Kuvan 15 tapauksessa PSNR-mittaus antoi erittäin hyvän tuloksen, mikä kertoisi pakkauksen olevan onnistunut kyseisen kehyksen kohdalla. Myös aistinvaraisesti kuva näyttää hyvältä, joten tulokset ovat linjassa. Kuvassa 15 esimerkkinä käytetyt kuva-kaappaukset on esitetty suuremmassa koossa liitteessä 2.

Vaikka kuvanlaatua voidaankin mitata matemaattisesti laskemalla, on silti tärkeää luottaa omaan arviointikykyynsä kuvanlaadussa. Vaikka matemaattinen mittaus antaisikin huonon tuloksen, ei se välttämättä tarkoita huonoa visuaalista tulosta.

7 Yhteenveto

Insinööriyöraportissa on esitelty videonpakkauksen toimintaperiaate ja käytössä olevat menetelmät. Menetelmien tarkoituksena on poistaa videosta redundanttia informaatiota, jota ei pystytä ihmissilmin havaitsemaan. Koska video koostuu yksittäisistä kuvista, moni pakkausmenetelmä perustuu samoihin toimintaperiaatteisiin still-kuvan pakkauksen kanssa.

Videonpakkausta siis tapahtuu sekä yksittäisissä, että peräkkäisissä videon kehyksissä. Yksinkertaistettuna koko prosessia voidaan kuvata yksittäisen I-kehyksen lähettämisenä, minkä jälkeen lähetetään ohjeita kuinka kuvalohkojen tulee käyttäytyä seuraa-

vien P- tai B-kehysten kohdalla. Tällaisia käskyjä voivat olla esimerkiksi seuraavat: Älä tee mitään (erodata), liiku (liikevektori), vaihda väriä.

Käskyjä ei kuitenkaan voida lähettää loputtomasti peräkkäin, sillä virhe käskyn tulkinassa aiheuttaa virheellisen toiminnon myös seuraavassa kehyksessä. Tämän takia välillä on lähetettävä uusi puhdas I-kehys käskyjonon nollaamiseksi. Uuden puhtaan I-kehysten lähettämisen jälkeen jatketaan taas käskyjen lähettämistä, ja koko prosessi toistaa itseään. Kvallohkoille tarkoitetut käskyt on kuitenkin tallennettava videodataan jotta ne voidaan suorittaa käänteisesti dekooderissa kokonaisen videokuvan muodostamiseksi.

Videokuvan pakkaamiseen on tarjolla runsaasti vaihtoehtoja. Niistä vaihtoehtoista onkin valittava omalle videolleen järkevin vaihtoehto. Lähdemateriaalin koko ja laatu vaikuttavat huomattavasti valinnan tekemiseen, sillä ei ole järkevää pakata erittäin laadukasta lähdemateriaalia mahdollisimman pieneen kokoon. Videonpakkaus onkin tasapainottelua videotiedoston laadun ja koon välillä. Ei ole mahdollista saavuttaa erinomaista kuvanlaatua erittäin pienellä tiedostokoolla. Näin ollen onnistuneessa videonpakkauksessa saavutetaan parhaat, videon kannalta tärkeät ominaisuudet.

EEMontti-kehityshankkeeseen valmistetut esittelyvideot vastasivat laadultaan, visuaalisuudeltaan ja käyttötarkoitukseltaan sekä asiakkaan että omia tavoitteitamme. Koska aistinvaraista kuvanlaaduntarkkailua pystyttiin suorittamaan videoiden koostamisen yhteydessä, se osoittautui matemaattista PSNR-kuvanlaaduntarkkailua tehokkaammaksi toimintamalliksi.

Laadullisesta näkökulmasta videoille ei ole varsinaista standardia. Ainoa sitova tekijä on kuvan resoluutio, joka määrittää, mihin kuvakategoriaan video kuuluu. Laadun tarkkailussa on tärkeää luottaa omiin aisteihinsa: jos videon laatu näyttää hyvältä, se yleensä onkin hyvä.

Lähteet

- 1 Fischer, Walter. 2010. Digital Video and Audio Broadcasting Technology, Signals and Communication Technology, 3rd edition. Berlin Heidelberg: Springer-Verlag.
- 2 Watkinson, John. 1995. Compression in video & audio. Focal Press.
- 3 Meyer, Mark-Paul. 2000. Restoration of motion picture film. Conservation and Museology. Butterworth-Heinemann.
- 4 Pääkkönen, Matti. 1991. A:sta ö:hön. Suomen yleiskielen kirjaintilastoja. Kielikello, nro 1, s. 3.
- 5 Richardson, Iain E. 2010 The H.264 Advanced Video Compression Standard. John Wiley & Sons.
- 6 Encoding Youtube videos at highest quality. 2013. Verkkodokumentti. Videomaker. <<http://www.videomaker.com/article/17034-encoding-youtube-videos-at-the-highest-quality>>. Luettu 10.1.2015.
- 7 Video formats explained. 2012. Verkkodokumentti. Videomaker. <<http://www.videomaker.com/article/15362-video-formats-explained>>. Luettu 10.1.2015.
- 8 Streaming video - how it works. 2002. Verkkodokumentti. Videomaker. <<http://www.videomaker.com/article/8553-streaming-video-how-it-works-and-why-it-sometimes-doesnt>>. Luettu 10.1.2015.
- 9 EEMontti. 2012. Verkkodokumentti. EEMontti. <<http://www.eemontti.fi>>. Luettu 23.11.2014.
- 10 Blu-ray FAQ. Verkkodokumentti. Blu-ray.com. <<http://www.blu-ray.com/faq/>>. Luettu 15.1.2015.
- 11 Sullivan, Gary. J; Ohm, Jens-Rainer; Han, Woo-Jin; Wiegand, Thomas. 2012. Overview of the High Efficiency Video Coding (HEVC) Standard. Transactions on Circuits and Systems for Video Technology, nro 12, s.1649–1668.
- 12 VP9 encoder. Verkkodokumentti. Heywatch. <<http://www.heywatchencoding.com/vp9>>. Luettu 18.1.2015.
- 13 The Codecs That Make UHD Video Possible: HEVC Vs. VP9. 2014. Verkkodokumentti. Streaming media. <<http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/The->

Codecs-That-Make-UHD-Video-Possible-HEVC-Vs.-VP9-96926.aspx>. Luettu 18.1.2015.

- 14 Fernando, C. N. Pereira; Touradj, Ebrahimi. 2002. The MPEG-4 book. Prentice Hall.
- 15 Video File Container Formats, Compression and Codecs. 2010. Verkkodokumentti. Reelseo. <<http://www.reelseo.com/file-formats-containers-compression/>>. Luettu 11.1.2015.
- 16 Welstead, Stephen. 1999. Fractal and wavelet image compression techniques. SPIE Publication.
- 17 Haskell, Barry. G; Puri, Atul; Netravali, Arun. N. 2002. Digital video: An introduction to MPEG-2. Kluwer Academic Publishers.



