

Opinnäytetyö (AMK)

Tietojenkäsittely

Tietojärjestelmä

2012

Thuy Anh Do

VIDEON ESITTÄMINEN HTML 5 VIDEO- TAGISSA



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Tietojärjestelmä

16.12.2012 | 54

Ohjaaja Tuomo Helo

Thuy Anh Do

VIDEON ESITTÄMINEN HTML 5 VIDEO-TAGISSA

Opinnäytteen tavoitteena on videon esittäminen websivulla käyttämällä HTML 5:n tarjoamaa uudistusta eli video-tagia. Opinnäytteessä tutkitaan mitkä videoformaatit sopivat esitettäväksi Internetissä ja mitkä näistä formaateista sopii parhaiten HTML 5:lle. Työssä tutkitaan näiden lisäksi HTML 5:n uudistuksia ja formaattien tuki yleisimmissä selaimissa.

Opinnäytteessä analysoitiin videoformaattien ominaisuuksia ja vertailtiin niiden saama tuki eri selaimissa ja ympäristöissä. Työssä tutkittiin myös videon käytössä olevia kuvasuhteita eri laitteissa ja ympäristöissä sekä videonpakkaukseen liittyviä keskeisiä piirteitä.

Videoformaattien testausta varten videoleikettä pakattiin haluttuihin videoformaatteihin Miro Video Converter -videonpakkausohjelmalla ja luotiin HTML 5-sivun. Sivua testattiin viidessä yleisimmissä selaimissa Windows-ympäristössä. Videoformaattien toistomahdollisuuksia eri selaimissa näkyy taulukosta 7.

ASIASANAT:

HTML 5, datasiirto, selain, tiedonpakkaus, videoformaatti.

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Data System

16.12.2012 | 54

Instructor Tuomo Helo

Thuy Anh Do

STREAMING VIDEO IN HTML 5 VIDEO-TAG

The objective of this thesis is to be able to stream video online using new video-feature offering by HTML 5. The goal is to find suitable video formats for streaming and to find which of these are supported in HTML 5. The new features of HTML 5 are also studied as well as video formats that are supported in common web browsers.

The features of video formats are analyzed and their support in different web browsers and platforms are compared. The research is also made on common used video sizes in different devices and system. Also video compression and its related vital settings have been studied.

For video testing, a Miro Video Converter –program is used to compress a video clip into desired video formats and a HTML 5 webpage is created. The webpage is then tested in five most common used web browsers. The supported playback of video formats in different web browsers can be viewed in table 7.

KEYWORDS:

HTML 5, data transmission, browser, data compression, video format.

SISÄLTÖ

1 JOHDANTO	7
2 VIDEONPAKKAUS	9
2.1 Pakkausjärjestelmä	10
2.2 Videosekvenssi	13
3 DATASIIRTO	18
4 VIDEON OMINAISUUDET	22
4.1 Kuvasuhde	22
4.2 Bittivirta	24
5 VIDEOFORMAATTI	26
5.1 Ogg Theora	28
5.2 MP4 H.264	29
5.3 WebM VP8	32
6 HTML 5	35
6.1 HTML 5 uudistuksia	36
6.2 HTML 5:ttä tukevat selaimet	38
7 VIDEON NÄYTTÄMINEN SELAIMISSA	40
7.1 Videon pakkaaminen	40
7.2 HTML 5 –sivun luominen	43
7.3 Testitulos	44
8 POHDINTA	49
LÄHTEET	51

KUVAT

Kuva 1. Tiedon pakkaus ja enkapsulointi sopivaan säiliöön (Robertson 2009).	9
Kuva 2. Yleinen videonpakkausjärjestelmä (Teuhola 2009; Yoo & Kang 2011).	11
Kuva 3. Bittivirran hierarkia (Teuhola 2009; Chen 2012).	14
Kuva 4. Delta-kuva sisältää muuttuvia liikkeitä (Ozer 2011b).	16

Kuva 5. VBR ja CBR (Ozer 2009b).	21
Kuva 6. Näyttöjen kuvasuhteet eri ympäristössä (Longtail 2012).	23
Kuva 7. Eri säiliöille sopivat kodekit (Bluejay 2010).	26
Kuva 8. Theoraa ja HTML 5:tä tukevat selaimet. (Dale & Lee 2011, 218.)	29
Kuva 9. H.264-kehitys (Ozer 2011d).	30
Kuva 10. H.264:sta tukevat selaimet. (Dale & Lee 2011, 217).	31
Kuva 11. WebM-formaattia tukevat selaimet (Dale & Lee 2011, 218).	34
Kuva 12. HTML 5 –videota tukevat selaimet. (O'Reilly 2011.)	38
Kuva 13. HTML 5 –videota tukevat mobiililaitteet. (O'Reilly 2011.)	38
Kuva 14. Videon tuottaminen WMM-ohjelmalla.	40
Kuva 15. Videon pakkaus – kodekin valitseminen.	41
Kuva 16. Videon pakkaus käynnissä.	41
Kuva 17. Pakkauksesta saadut tiedostokoot.	42
Kuva 18. Handbrake-pakkausohjelma.	43
Kuva 19. HTML 5-sivun luominen.	44
Kuva 20. Video Chromessa.	45
Kuva 21. Video Firefoxissa.	45
Kuva 22. Video Operassa.	46
Kuva 23. Video Internet Explorerissä.	47
Kuva 24. Video Safarissa.	47

TAULUKOT

Taulukko 1. Kuvasuhteen käytäntöjä (Dave 2009; Ozer 2009a; Ozer 2011e).	22
Taulukko 2. Bittivirta ja tiedostokoko (Bluejay 2010).	24
Taulukko 3. Bittivirta ja videon tyyppejä (Longtail 2012).	25
Taulukko 4. HTML 5:n uudet mediaelementit (HTML 5 Tag Reference 2012).	36
Taulukko 5. Video-tagin attribuutit (HTML 5 Tag Reference 2012).	37
Taulukko 6. Selainten käyttöosuus 2012 (Browser Statistics 2012.)	39
Taulukko 7. Videoformaattien toistomahdollisuuksia eri selaimissa.	48

LYHENTEET (TAI SANASTO)

CBR	Tasavirta siirtonopeus (engl. Constant bitrate)
DCT-kerroin	Kosinimuunnos (engl. Discrete Cosine Transform)
GOP-ryhmä	Kuvaryhmä (engl. Group of Pictures)
H.264	Kodekki
Kodekki	Pakkausmenetelmä (engl. codec)
MP4	Videoformaatti
Ogg	Videoformaatti
Theora	Kodekki
VBR	Vaihteleva siirtonopeus (engl. Variable bitrate)
VP8	Kodekki
WebM	Videoformaatti

1 JOHDANTO

Internet sai alkunsa vuonna 1969, kun BBN (alun perin Bolt, Beranek, Newman) –teknologian yritys loi ARPANet (engl. lyh. Advanced Research Projects Agency Network) –tietoverkon Kaliforniassa. Vuonna 1973 kehitettiin TCP/IP-protokollaa, joka mahdollisti koneiden yhteenliittämisen toisiinsa. Ja vuotta myöhemmin, 1974 Internet on virallisesti syntynyt. Webkäyttäjää oli vuonna 1984 vain tuhat, vuonna 1992 määrä ylitti miljoonan (Investintech.com 2012.) ja tähän päivään mennessä määrä on yli kaksi miljardia (Top 20 2012). Ensimmäinen webkamera ilmestyi vuonna 1991 Cambrigen yliopistolla ja sitä käytettiin kahvikeittimen valvontaan (Chapman 2009).

Nykyään noin 70 % Internet-käyttäjistä katsoo tai lataa videoita Internetistä ja joka viides aikuisista videonkatsojista laittaa omia videoitaan Internetiin. Käyttäjien laittamien verkkovideoiden variaatiot ovat muun muassa koti-, matkustus-, TV-, urheilu- tai musiikkileikkeitä. Näistä kotivideot vievät suurimman osan. (Branckaute 2010.) Verkkovideo on oiva ja tehokas tapa viestittää monimutkaisia sanomia. Yritysmaailmassa video on mainonnan tehokas ja nopea tapa jättää haluamansa viesti katsojien mieliin. Hyvin suunniteltu video ei vain ole tehokas viestintätapa vaan se vetää myös käyttäjiä takaisin omalle websivulleen.

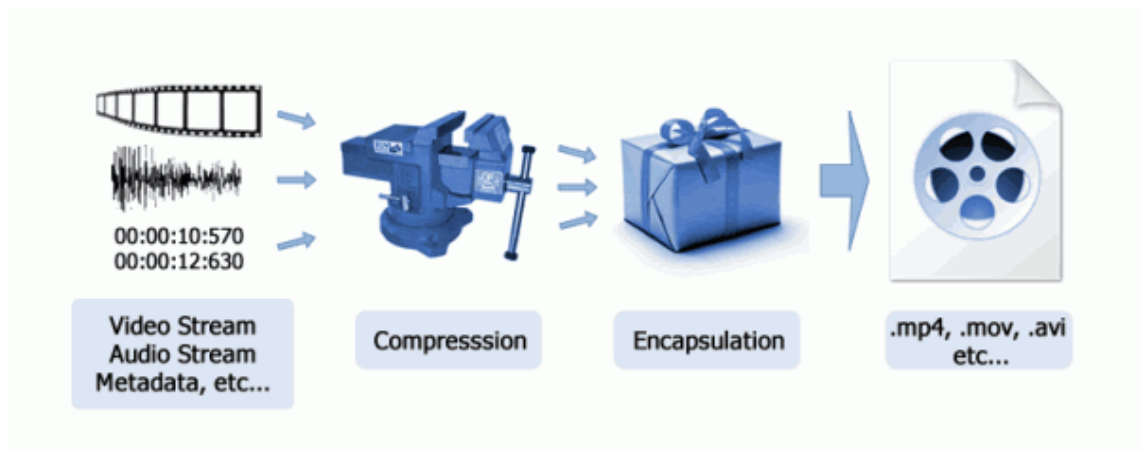
Video-aiheena on ajankohtainen ja kiinnostava. Opinnäytteen tavoitteena on videon esittäminen HTML 5 video-tagissa. Opinnäytteessä tutkitaan videoformaatteja, pakkausmenetelmiä eli kodekkeja ja näiden saama tuki HTML 5:ssä. Tarkoituksena on selvittää, mitkä videoformaatit sopivat esitettäväksi Internetissä tutor-tyyppisille videoille. Työssä esitetään myös HTML 5:n uusia

piirteitä. Toisessa kappaleessa käsitellään videonpakkausta, pakkausjärjestelmää ja videosekvenssiä.

Kappaleessa 3 käsitellään datasiirtotapoja eli videon esitystapoja Internetissä. Kuvasuhde ja bittivirta käsitellään kappaleessa 4. Nämä piirteet vaikuttavat miltä video näyttää ja kuinka nopeasti video on saatavilla eri käyttöympäristössä. Kappaleessa 5 käsitellään HTML 5:lle sopivat videoformaati. Empiirisessä osassa on videonpakkausta, HTML 5-sivun rakentamista ja videon testausta käytetyimmissä selaimissa. Opinnäytteessä ei käsitellä suoratoistoa (engl. live-streaming), audioformaatteja, videon kuvaustekniikkaa ja toisto-ohjelmia. Lähteinä on käytetty enimmäkseen Internetistä saatuja materiaaleja.

2 VIDEONPAKKAUS

Tiedon pakkaus on menetelmä, jolla saadaan tietoaimeen kuvattua tiivimmin. Tiedon pakkaus säästää kalliin tallennustilan ja helpottaa tiedon siirtoa hitailla internetyhteydellä. (Harris 2012a.) Oletetaan, että 1 kuva sisältää 8 megabittiä. Tällöin 25/sek kuvanopeudella (engl. frame rate, lyh. fps) tunninmittainen datamäärä olisi $8 \text{ Mbit} \times 25 \times 60 \times 60 = 720\,000 \text{ Mbit}$ eli 720 Gbit (audiodataa lukuunottamatta). Videota on mahdotonta siirtää pakkaamattomana nykyisillä laitteilla ja siirtonopeudella. (Teuhola 2009.) Video saattaa sisältää paitsi paljon toiminnan sisältävien kuvien ja äänien lisäksi myös metadataa, tekstitystä ja muita interaktiivisuusmahdollisuutta. Nämä eri datatyypit täytyy pakata ja enkapsuloida niille sopivaan formaattiin eli säiliöön (kuva 1).



Kuva 1. Tiedon pakkaus ja enkapsulointi sopivaan säiliöön (Robertson 2009).

Pakkaus on häviöllinen (engl. lossy), jos tietoaimeksen sisältö muuttuu pakkauksen yhteydessä, muutoin se on häviötön (engl. lossless). Videon pakkaus on mahdollista hyödyntäen datan sisältämiä redundansseja eli

vähemmän todellisia informaatioita, joita ovat esimerkiksi spatiaalisia (riippuvuus kuvan sisällä) ja temporaalisia (riippuvuus kuvien välillä, paitsi leikkauskohdissa). (Teuhola 2009.) Häviöllinen pakkaus toimii siten, että se poistaa tarpeettomat tai korvattavissa olevat informaatiot alkuperäisestä tiedostosta kokonaan, minkä tuloksena on paljon pienempi tiedostokoko. (Harris 2012b.)

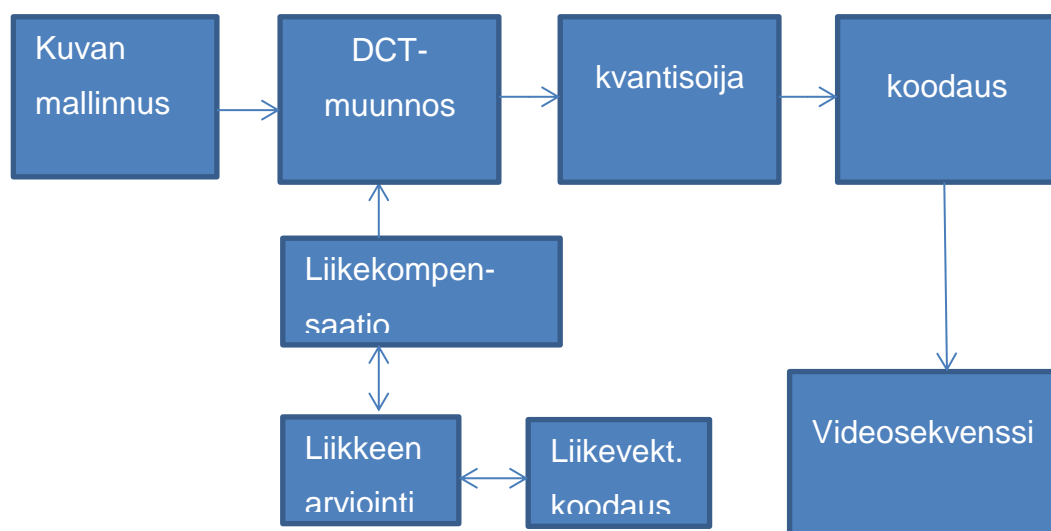
Esimerkiksi pakkauksen yhteydessä maisemakuva, jossa on käytetty erilaisia sinisiä värejä kuvaamaan taivaan, voidaan tämä korvata yhdellä sinisellä värillä, joka ei muuta taivaan olemusta. Purkamisen yhteydessä eli kuvan avatessa ihmissilmä ei huomaa pikseleiden vivahde-eroja toisistaan, silti pystytään säästämään huomattavasti paljon tallennustilaa. Häviöllistä pakkausta ei suositella käytettäväksi esimerkiksi ohjelmien, tietokantojen tai muita tarkkuutta vaativien tietojen pakkaamiseen, sillä alkuperäistä tietoa ei saa enää takaisin. (Harris 2012b.)

2.1 Pakkausjärjestelmä

Kuvanpakkauksen ensimmäinen askel mallinnusvaihe, jossa jaetaan kuvat pieniin yleensä 8 x 8 –kokoiisiin makrolohkoihin. Mallinnusvaiheessa kuvataan sävyjen jakautuma ja niiden riippuvuus toisiinsa, minkä jälkeen käytetään DCT (engl. Discrete Cosine Transform) –muunnosta, jossa muodostetaan koodattavia elementtejä. Mallintaminen perustuu todennäköisyyksiin ja koodausmenetelminä ovat muun muassa Huffman- ja aritmeettinen koodaus. (Teuhola 2009; Introduction 2012)

Useimmissa videonpakkauksessa käytössä oleva DCT-muunnos on häviötöntä pakkausta, joka perustuu kvantisointiin (engl. quantization). Toistossa käytetään InverseDCT-muunnos, joka on vastakohta DCT-muunnosta. Molemmat DCT-että IDCT-muunnos kuluttavat jopa 30 % CPU-tehosta. Kvantisointia käytetään poistamaan kuvasta tarpeettomat tiedot, joita esimerkiksi ihmissilmä ei näe. Ja lopuksi koodataan kyseiset elementit binääreiksi. Koodauksen tuloksena muodostuu videosekvenssi, joka koostuu yksittäisistä liikkumattomista kuvista. (Teuhola 2009; Introduction 2012.) Pakkauksen päävaiheet näkyvät kuvasta 2.

DCT-muunnoksen pakkausvirhe näkyy lohkomaisuutena (engl. blocking artifacts), joka johtuu liiallisesta pakkauksesta eli informaatiohukasta. Videonpakkaus on yhdistelmä kuvan- ja liikkeen pakkauksesta, missä peräkkäisten kuvien välinen korrelaatio on tärkeää. Sekä kuvan- että videonpakkauksessa esiintyy erityyppisiä pakkaushäiriöitä. Eri häiriöille tehoaa omat pakkausmenetelmänsä. Kuvanpakkauksessa videon kuvat pakataan toisistaan riippumatta. Eräs tuttu kuvanpakkausmenetelmä on esimerkiksi JPEG-kuvanpakkaus. (Teuhola 2009.)



Kuva 2. Yleinen videonpakkausjärjestelmä (Teuhola 2009; Yoo & Kang 2011).

Useimmiten videokuvissa on staattinen tausta (poislukien satunnaisvaihtelut), jos kamera on paikallaan. Liikekompensaatiota käyttäen voidaan ennustaa seuraavia kuvia melko tarkasti. Liikekompensaatio koostuu pääosin kolmesta vaiheesta: liikevektorien etsinnästä, koodauksesta ja käytöstä ennustuksessa. Liikekompensaatio toimii siten, että kuvan jokainen lohko ennustetaan naapurikuvasta, joka yleensä jaetaan 16 x 16-makrolohkoihin. Liikevektori on lohkojen sijaintiero, joka saadaan ennustajasta ja ennustetusta. (Teuhola 2009.)

Videonpakkauksessa hyödynnetään neljää seuraavaa tapaa:

- kosinimuunnos (engl. discrete cosine transform, DCT)
 - vektorikvantisointi (engl. vector quantization, VQ)
 - fraktaalitiivistys (engl. fractal compression)
 - diskreetti wavelet-muunnos (engl. discrete wavelet transform, DWT)
- (WordPress 2008.)

DCT on häviöllinen pakkausmenetelmä, joka poistaa ne tiedot, joita ihmissilmä ei pysty erottamaan tai näkemään eli tässä tapahtuu dekorrelaatio. DCT-muunnosta löytyy esimerkiksi JPEG, MPEG, H.261 ja H.263 -pakkausmenetelmissä. Vektorikvantisointi on myös häviöllinen pakkausmenetelmä, joka ottaa vastaan datajonon ja poistaa siitä redundansseja. Fraktaalitiivistys on VQ-pakkauksen muoto ja se on häviöllinen pakkaustapa. Menetelmä hakee kuvassa toisiaan muistuttavia osia ja käyttää fraktaalialgoritmia generoimaan ne osat. (WordPress 2008.)

Diskreetti Wavelet-tekniikka muistuttaa DCT-muunnosta. Diskreetti Wavelet-pakkaustapa toimii siten, että sitä sovelletaan koko kuvaan eikä pieniin haluttuun osiin kuvassa. Tiedon pakkaus säästää levytilaa ja tietoa voidaan

tallentaa ja esittää nopeammin. Toisaalta pakkausmenetelmät tuottavat enemmän tallennettavaa tietoa ja monimutkaisuutta, jotka voivat aiheuttaa virheitä esimerkiksi datasiirron aikana. (WordPress 2008.)

Pakkausmenetelmät tai –algoritmit eroavat toisistaan seuraavilla ominaispiirteillä:

- Häviötön/ häviöllinen menetelmä
- pakkausteho, eroavaisuus pakatusta alkuperäiseen materiaaliin
- nopeus, koodausnopeus on tärkeä esimerkiksi live-esityksissä ja dekkoodausnopeus toistossa
- herkkyys tiedonsiirtovirheille (engl. robustness). (Teuhola 2009.)

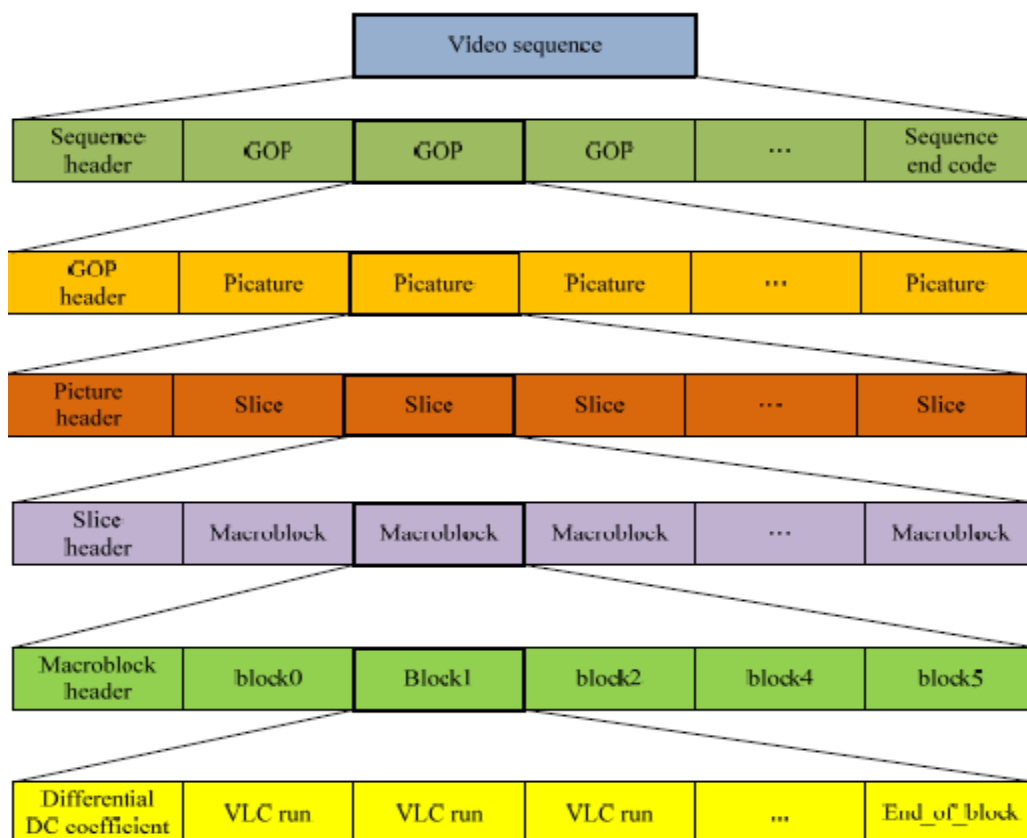
2.2 Videosekvenssi

Bittivirran hierarkia koostuu kuudesta kerroksesta (kuva 3) ja jokainen kerros alkaa aina otsikosta eli tunnisteesta (engl. header). Bittivirran ensimmäinen kerros koostuu GOP (engl. lyh. Group of Pictures) -ryhmistä eli kuvaryhmistä. Videosekvenssi alkaa sekvenssin otsikosta (engl. sequence header) ja loppuu lopputunnisteeseen (engl. sequence end code). Yhdessä videosekvenssissä voi olla monta GOP-ryhmää. Sekvenssin otsikko sisältää informaation kuvasta esimerkiksi kuvan korkeudesta ja leveydestä, resoluutiosta, kuvanopeudesta (engl. frame rate) ja puskurinkoosta (engl. buffer size). (Ozer 2011b; Chen 2012.)

Hierarkian toinen kerros on kuvakerros, joka sisältää I-, P- ja B –kuvia. GOP-ryhmässä on siis kuvajonoja, jotka alkavat aina I-kuvasta seuraavaan I-kuvaan, tätä poislukien. Esimerkiksi yksi GOP-ryhmä voi koostua I, B, P, B, P, B, P, B, I,

P, B, P-kuvajonosta. Tässä GOP-ryhmässä on tällöin kaksi jonoa. Ensimmäiseen jonoon kuuluvat I, B, P, B, P, B, P ja B –kuvat ja seuraavaan jonoon kuuluvat I, P, B ja P –kuvat, minkä jälkeen alkaa toinen GOP-ryhmä. Kuvakerroksen otsikko (engl. GOP header) sisältää aikaa ja sekvenssin alkua. (Ozer 2011b; Chen 2012.)

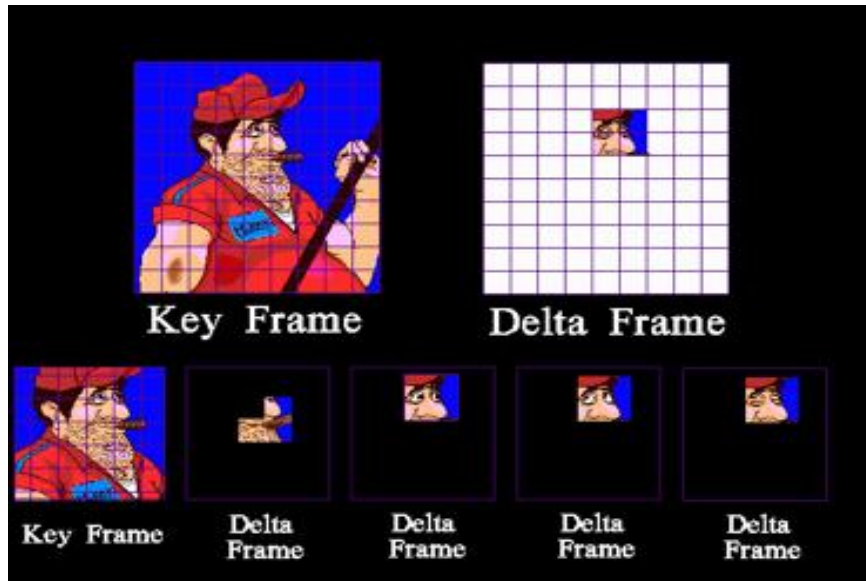
Kolmas kerros koostuu viipaleista (engl. slice), jonka otsikosta löytyy viipaleiden pituus ja sijainti. Viipaleet sisältävät makrolohkoja (engl. macroblock) ja jokainen makrolohko on jaettu 8 x 8 –kokoon. Viimeinen kerros on lohko-kerros, jossa on käytetty DCT-pakkausta. (Chen 2012.) Bittivirran kerrokset näkyvät kuvasta 5.



Kuva 3. Bittivirran hierarkia (Teuhola 2009; Chen 2012).

Häviöllinen pakkausmenetelmä hyödyntää intra- ja inter -kuvien kompressoititapaa. Intra-kuva (engl. intra-frame) on yksittäisen kuvan kompressoitua ilman viittauksia muihin kuviin. Inter-kuva (engl. inter-frame) taas kompressoii videokuvan hyödyntäen redundanteja kuvien välillä. Inter-kuva on siis tehokkaampi, sillä se sisältää vain ne tiedot, jotka muuttuvat myöhemmissä kuvissa. Useimmat pakkausmenetelmät ovat optimoituja käyttämään inter-kuvaa redundanssien etsimisessä ja erottamisessa kuvien väliltä. Häviöllisen kuvanpakkauksen huono puoli on se, että mitä parempi laatu sillä saa, sitä vaikeampi sitä on purkaa. (Ozer 2011b.)

Ennen CD-levyissä käytettyjä kodekkeja ovat Cinepak ja Indeo, jotka käyttävät kahdenlaista kuva-tyyppiä: I-kuva ja delta-kuva. I-kuva (kutsutaan myös key-kuvaksi) pitää sisällään koko kuvan ja käyttää intra-kuva-kompressoitua. Kompressoinnin yhteydessä pikselit delta-kuvassa verrataan muihin aikaisempiin kuviin, josta redundansseja poistetaan siitä. Jäljellä oleva data delta-kuvassa kompressoidaan käyttäen inter-kuva-tekniikkaa, jotta saavutettaisiin mieluisa datavirtanopeus. Kuvassa 4 esittää key- ja delta -kuvia. Delta-kuva sisältää vain muuttuvat tiedot kuvien väliltä. (Ozer 2011b.)



Kuva 4. Delta-kuva sisältää muuttuvia liikkeitä (Ozer 2011b).

Monet pakkausjärjestelmät käyttävät kaikkea kolmea kehystä pakkauksessaan: I-kuva, B-kuva ja P-kuva. I-kuva (eli key-kuva) käyttää intra-kuvan tekniikka tehden siitä suurin ja vähiten tehokas kuva-tyyppi. (Ozer 2011b.) I-kuvasta on kuitenkin hyötyä toistonaikana, sillä se nopeuttaa hakuja videossa. Näin ollen key-kuvaa suositellaan laittamaan pakkauksen asetuksen yhteydessä joka viiden tai kymmenen sekunnin välein. Esimerkiksi Youtube käyttää key-kuvaa joka toisessa sekunnissa. (Ozer 2009b.)

B-kuva voi käyttää redundanssitietoja kaikista edeltäjäkuvien tai seuraavista I-, B- tai P -kuvista. B-kuvalla on tämän takia kaksinkertainen mahdollisuus löytää redundanssia, mikä tekee siitä kaikista kuva-tyypeistä tehokkaimman pakkauksen näkökulmasta. B-kuva parantaa pakatun videon laatua, mutta sitä on vaikea purkaa toistonaikana. B-kuvat kannattaa kuitenkin jättää pois, jos haluaa videon toimivan pienitehoisissa laitteissa kuten iPodissa mutta esimerkiksi normaalissa käyttötietokoneissa tämä ei ole ongelmaa. (Ozer 2011b.)

Suuri tehoisissa laitteissa suositellaan siis hyödyntämään B-kuvaa aina kun on mahdollista, esimerkiksi pakkauksen asetuksessa kahta tai kolmea B-kuvaa yhteen jonoon. (Ozer 2009b.) Molemmat B- sekä P -kuvat ovat delta-kuvia. P-kuvat ovat näistä yksinkertaisimpia ja ne käyttävät redundansseja hyväkseen edellisistä I- tai P -kuvista. Pakkauksessa P-kuva on oletuksena eli se on käytössä, kun käytössä ei ole I- tai B-kuvaa. (Ozer 2011b.)

3 DATASIIRTO

Internetissä videota välitetään katsojille pääosin joko toistamalla videon suoraan websivulla tai tarjoamalla kokonaisina tiedostoina ladattavaksi käyttäjien koneille. Yleensä videon toisto websivulla alkaa siten, että käyttäjä napsauttaa videossa olevaa play-nappia, mikä lähettää pyynnön WWW-serverille, joka lähettää pyynnön eteenpäin mediapalvelimelle, jossa video etsitään ja lähetetään esitettäväksi katsojan koneelle. Videon esittäminen voi toteuttaa muun muassa suoratoistolla, progressiivisella tai adaptiivisella download – tavalla.

Suoratoisto

Suoratoistossa median sisältöä näytetään käyttäjille heti kun sitä on saatavilla. Live-esitys toimii esimerkiksi suoratoistona, johon käytetään yleensä tasavirtaista siirtonopeutta (engl. lyh. CBR). Suoratoisto esittää median reaaliajassa sen pituudesta huolimatta. Toimenpide vaatii siihen erikoistettuja suoratoistoservereitä. Suoratoiston huono puoli on se, että ne eivät yleensä pääse yrityksen suojamuurien läpi. Tämän takia suoratoiston toisena vaihtoehtona on käyttää TCP:n tai HTTP:n jakelua. (Milano 2008.)

Suoratoistoserverit ovat erikoisohjelmia, jotka tarjoavat suoratoiston ominaisuuksia, joita ovat esimerkiksi käyttäjälukujen seuraaminen ja käyttäjien Internet-yhteyksnopeuden havaitseminen. Suoratoistoserveriä voidaan käyttää esimerkiksi operoimalla omaa serveriä tai käyttämällä ISP-palvelua (engl. lyh. Internet Service Provider). Suoratoistossa videon kuvaus ja enkoodaus eli pakkaus tapahtuvat samaan aikaan kun jonkin esitys tai tapahtuma on käynnissä. Videon toiston yhteydessä tarvitaan tahokkaan dekodaaajan eli

purkajan, jotta se videota voisi näyttää ongelmitta katsojille. (Mediacollege.com 2012c.)

Suoratoisto on kallis menetelmä videoiden esittämiseen, ja sitä ei suositella otettavaksi käyttöön, jos siihen ei ole välttämätöntä tarvetta. Seuraavat ohjelmat tarjoavat oman serverin käytön mahdollisuuden:

- RealNetworkiltä Helix Universal Server mahdollistaa erilaisien formaattien käytön, kuten RealMedian, Windows Median ja MPEG-4-formaattien
- Applelta Quicktime Streaming Server tukee MPEG-4- ja 3GPP-formaatteja
- Macromediaalta Communication Server, joka on erikoistunut Flash-videoihin ja interaktiivisiin videosisältöihin. (Mediacollege.com 2012c.)

Downloading

Tiedoston lataus (engl. downloading) on videon jakelutapa, joka tallentaa tiedoston kokonaisuudessaan katsojan omalle tietokoneelle ennen videon toistoa. Pienillä videotiedostoilla videon lataamisessa ei välttämättä kestä kauan, mutta isojen tietostojen kanssa tämän tyyppinen jakelutapa saattaa aiheuttaa harmia katsojille. Downloading-tavan hyvä puoli on siinä, että videon eri osiin pääsee helposti käsiksi, koska koko tiedosto on jo valmiina muistissa. Tässä menetelmässä hyödynnetään HTTP-protokollaa. Downloading-tavan kehittyneempi tapa on progressiivinen lataaminen (engl. progressive downloading), joka muistuttaa suoratoistoa mutta sillä ei kuitenkaan ole kaikkea suoratoiston ominaisuuksia. (Mediacollege.com 2012a.)

Progressiivinen

Progressiivisessa latauksessa media toistetaan ja ladataan samanaikaisesti. Progressiivinen lataus käyttää standardoituja protokollia, kuten HTTP:tä tai FTP:tä median jakelussa. Suositeltava pituus tämän tyyppisille tiedostoille on vähemmän kuin 10 minuuttia. Tämä toistotapa ei tapahdu reaaliajassa, mutta sen hyvä puoli on se, että videon ja audion datavirta voidaan mukauttaa käyttäjien yhteysnopeuteen. Toinen hyvä puoli on se, että tässä on mahdollista hyödyntää VBR- (engl. two-pass variable bitrate) koodaustapaa, jolla parannetaan videon ja audion laatua. (Milano 2008.)

Toisin kuin suoratoistossa, progressiivisessa latauksessa videota esitetään HTTP-webserverin kautta. Tämä tarkoittaa sitä, että videota tallennetaan käyttäjän koneelle ja toistetaan paikallisesti suoraan käyttäjälle. Menetelmän huono puoli on se, että esitettävää materiaalia on mahdollista kopioida. (Ozer 2011c.) HTTP-toistotapa ei vaadi erikoisserveriä ja se on yksinkertaisin ja halvin tapa esittää videota kotisivulla. Tämä vaihtoehto on parempi kuin suoratoisto pieni- ja keskikokoisille yrityksille.

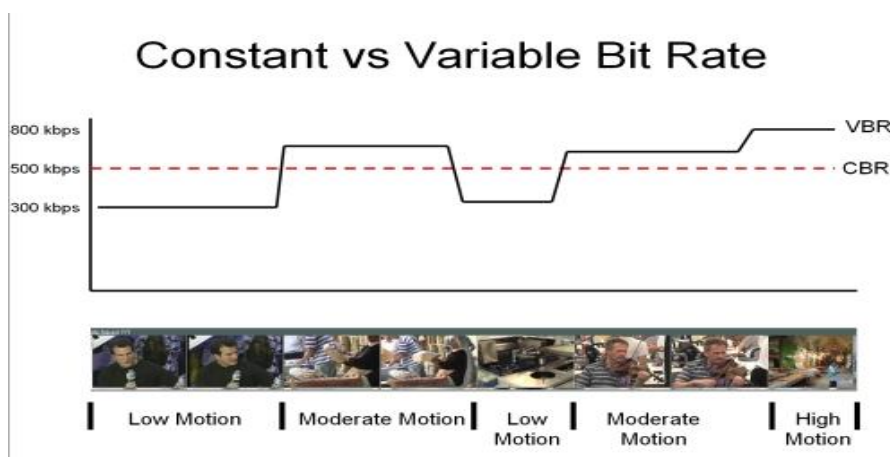
HTTP-toistolla on seuraavia rajoituksia:

- tarkoitettu pienelle käyttäjäryhmälle, noin 12 ihmisen kynnys, ei siis ole yhtä tehokasta kuin muut esitystavat
- ei pysty suoratoistoon
- ei ole mahdollista jäljittää käyttäjien yhteysnopeutta. (Mediacollege.com 2012b.)

Adaptiivinen

Adaptiivisen (engl. adaptive streaming) -toistotavan ominaisuus on sen kyky mukautua eri käyttäjien datanopeuteen serverin toimesta. Tämä on erinomainen ominaisuus ottaen huomioon pienitehoisia laitteita ja joilla on huonot yhteydet. Adaptiivista toistotapaa tarjoaa esimerkiksi Adoben Dynamic Streaming, Applen HTTP Live Streaming ja Microsoftin Smooth Streaming. (Ozer 2011c.) Kuvassa 5 näkyy adaptiivisen striimaustavan topologia.

VBR ja CBR ovat kaksi tekniikkaa, joilla kontrolloidaan bittivirtaa (engl. bit stream/ bit rate) toiston aikana tietyssä aikayksikössä (Ozer 2009b). VBR (engl. variable bitrate) on tiedonsiirto- tai tallennustapa, jossa hyödynnetään tilankäyttöä vaihtelevasti riippuen sen hetkisistä ominaisuuksista tai tarpeista (kuva 5). VBR käyttää tehokkaasti kaistanleveyttä ja se hukkaa vähemmän informaatiota kuin CBR. VBR ei kuitenkaan pienennä tiedostokokoa. CBR (engl. constant bitrate) käyttää saman verran tilaa tiedon siirtämiseen ja tässä tavassa on helpompi toteutus ja datasiirron ylläpito. Suoratoistossa käytetään yleensä CBR-siirtotekniikkaa. (AfterDawn 2012.)



Kuva 5. VBR ja CBR (Ozer 2009b).

4 VIDEON OMINAISUUDET

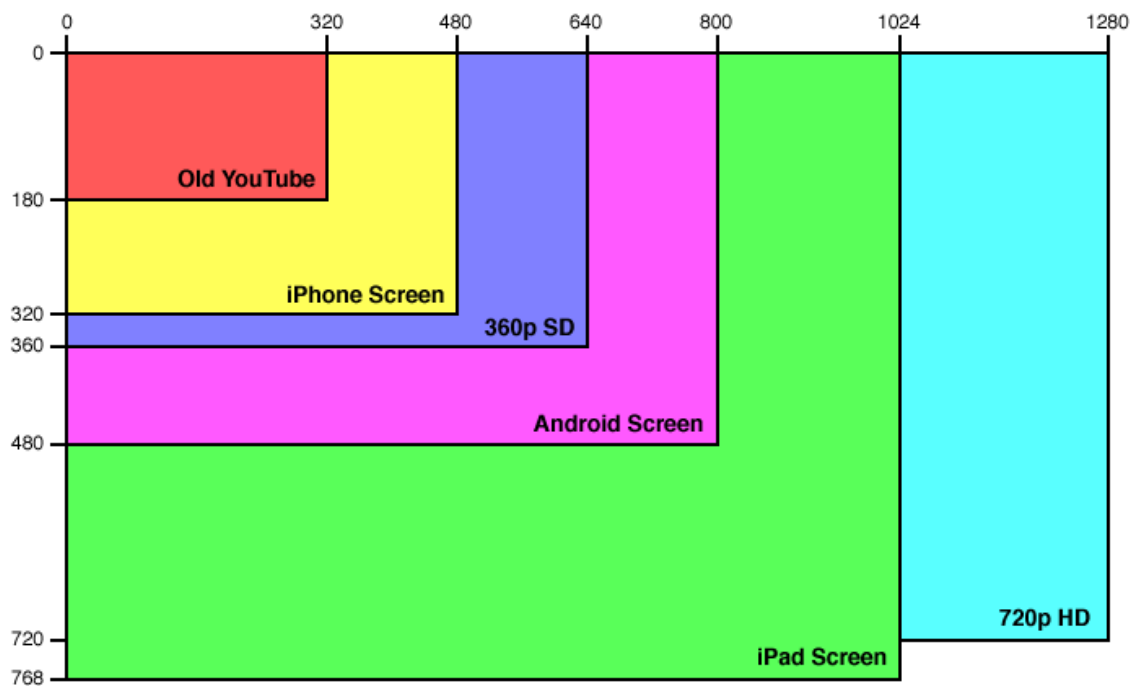
4.1 Kuvasuhde

Kuvasuhde määrittelee kuvan leveyden suhteen korkeuteen ja se on yleensä laskettavissa resoluutiosta. Resoluutio on videon korkeuden ja leveyden tulo pikseleinä. Standard definition (SD) -kuvasuhde on 4:3 ja high definitionilla (HD) on 16:9. Kuvasuhde voi riippua toistavasta laitteestakin. Esimerkiksi DVD-resoluutio on 720x576, mutta kuvasuhde voi olla sekä 4:3 tai 16:9. Tällöin toistava laite venyttää kuvaa sen asetuksen mukaan. Kuvanopeus (engl. frame rate) on yleensä 25, mikä on varsin riittävä. (Ekonoja 2011.)

Taulukko 1. Kuvasuhteen käytäntöjä (Dave 2009; Ozer 2009a; Ozer 2011e).

Standard definition (4:3)	High definiton (16:9)
640 x 480	720p (1280 x 720)
480 x 360	848 x 480
440 x 330	640 x 360
400 x 300	512 x 288
320 x 240	480 x 270
240 x 180	320 x 180
160 x 120	-
240 x 180	-

Videon oikeat mittasuhteet määrittelevät kuinka hienolta video näyttää eri näytöissä. Vaikka isot videoruudut ovat suosituimpia, niillä on haittapuolensa sillä niitä on vaikeampi purkaa. Monilla mobiililaitteilla ja/tai vanhoilla pöytätietokoneilla on vaikeuksia purkaa kuva, joka on suurempi kuin 360 pikseliä. Näytökoot vaihtelevat eri laitteissa. Kuvassa 6 havainnollistaa tärkeimpien laitteiden käyttämät kuvasuhteet. Esimerkiksi 16:9 kuvasuhteella pystytään jo näyttämään videota pöytätietokoneissa, tablet-, Android-, Windows-, iPad-, iPhone- ja iPod-laitteissa ja näiden lisäksi myös eri mediatoisto-ohjelmissa ja pelikonsoleissa. (Longtail 2012.)



Kuva 6. Näyttöjen kuvasuhteet eri ympäristössä (Longtail 2012).

4.2 Bittivirta

Bittivirta tai bittinopeus (engl. bit rate/data rate) kuvaa tiedonsiirtonopeutta tai tilan määrää audio tai video vie jossain aika yksikössä. Mittaamiseen käytetään yleensä bittejä/sekunti (engl. bits/sec tai bps) tai tavuja/sekunti (engl. bytes/sec tai Bps). (Odegard 2008; Afterdawn 2012.) Yksi bitti (engl. bit) sisältää aina kaksi mahdollisuutta. Hyvä esimerkki tästä on shakkiruudukko, jolla on kaksi värimahdollisuutta esimerkiksi shakkiruudukko. Harmaa- tai selkokuva gif-muodossa veisi yksi tavu (engl. byte), joka koostuu 8:sta bitistä, jossa on 256 värimahdollisuutta (harmaasävykuva, selkokuva gif-muodossa). Täysvärivalokuva taas vastaa kolmen tavun muistitarvetta. (Ratol 2012.)

Mitä suurempi bittivirta sitä parempi kuvalaatu ja isompi tiedostokoko (engl. file size). Bittivirtaan vaikuttavat resoluutio, video- ja audioformaatti ja videon tyyppi. Eri videon tyyppejä ovat esimerkiksi animaatio, haastattelu (engl. interview), esitelmä (engl. presentation), elokuvatrailer (engl. trailer) ja urheilu (engl. sport). (Bluejay 2010.) Taulukko 2 ja 3 havainnollistaa asian.

Taulukko 2. Bittivirta ja tiedostokoko (Bluejay 2010).

Bitrate and File size		
Bitrate	Filesize	Application
250 kbps	1.9 Mb/minute	YouTube original format
500 kbps	3.8 Mb/minute	YouTube HQ
768 kbps	5.9 Mb/minute	iPod maximum
1664 kbps	15.3 Mb/minute	TV show I downloaded from the iTunes store
2000 kbps	15.4 Mb/minute	YouTube HD

Taulukko 3. Bittivirta ja videon tyyppiä (Longtail 2012).

<i>Dimensions</i>	<i>Interviews / screencasts</i>	<i>Shows / presentations</i>	<i>Trailers / music / sports</i>
320x176 pixels	150 kbps	300 kbps	450 kbps
480x272 pixels	200 kbps	400 kbps	600 kbps
640x368 pixels	400 kbps	700 kbps	1000 kbps
800x448 pixels	600 kbps	1000 kbps	1400 kbps
1280x720 pixels	1000 kbps	1600 kbps	2200 kbps

Videon toistossa bittivirta ja resoluutio liittyvät kiinteästi toisiinsa. Esimerkiksi 320 x 240 -videolla on 76 800 pikseliä yhdessä kuvassa. Näin ollen 640 x 480 – videolla on 304 200 pikseliä, nelinkertainen määrä edelliseen verrattuna. Tällöin jälkimmäiseen versioon on sovellettava nelinkertainen pakkausmäärä, jos kumpikin versio haluttaisiin toistamaan samalla bittivirralla. Tämä vaikuttaa häviöllisiin pakkausmenetelmiin siten, että videon laatu heikkenee helposti mitä enemmän videota pakataan. (Ozer 2009a.)

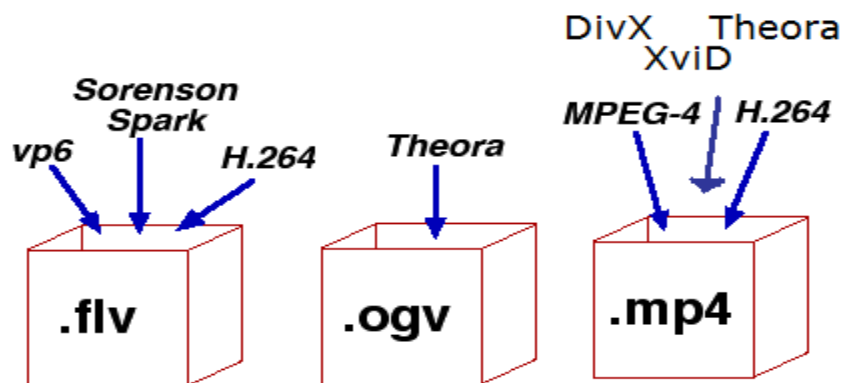
Kun videoon liittyvät ominaisuudet kuten kuvasuhde, kuvanopeus (engl. frame rate) tai pakkausmenetelmä ovat tasa-arvoiset, matalan bittivirran valitseminen pakkauksen asetuksissa tuottaisi huonompaa laatua videolle. Vaihtoehtona on kompensoida bittivirran ja resoluution väliltä parhaan laadun saamiseksi. Ensimmäinen vaihtoehto olisi valita haluttu bittivirta, josta valitaan sen tarjonnasta paras resoluutio videolle. Toinen vaihtoehto olisi resoluution valitseminen, minkä jälkeen valitaan paras mahdollinen bittivirta videolle. (Ozer 2009a.)

5 VIDEOFORMAATTI

Säiliö

Pakattu videokuva kääritään erilaisiin videoformaatteihin eli säiliöihin. Säiliö määrittää millaisia datatyyppisiä siihen voi sisällyttää. Näitä datatyyppisiä voivat olla metadata, video, audio, kappaleet, tekstitykset ja kodekki eli pakkausalgoritmi. Kehittyneet multimediasäiliöt tukevat näitä kaikkia datatyyppisiä kun taas yksinkertaisimmat säiliöt tukevat liikkuvien kuvien lisäksi vain audiota. (Videonpakkaus 2012.)

Kodekki määrää esitysnopeuden, videon laadun ja koon. (Gigafide 2009.) Kodekki on tiedonpakkausalgoritmi tai -menetelmä, joka pakkaa ja purkaa videokuvan, ja useimmiten se pakkaa myös äänen samassa yhteydessä. Samalla videoformaatilla voi olla useampia yhteensopivia kodekkeja. Esimerkiksi MPEG-4-formaatille sopivat DivX, XviD, 3ivx ja Ffmpeg MPEG-4 – kodekit (kuva 9; Videonpakkaus 2012.)



Kuva 7. Eri säiliöille sopivat kodekit (Bluejay 2010).

Divx ja Xvid tarjoavat hyvän videon koko ja laatu –suhteen ja siksi se soveltuu hyvin elokuvien pakkaukseen. Xvid on suositumpi, sillä se toimii myös Unix-järjestelmässä ja sillä saa hieman parempi laatu videolle. FFMpeg on kokoelma vapaita ohjelmistoja ja sitä käytetään laajalti Linux-ympäristössä, mutta se on myös yhteensopiva Windows- ja Mac-ympäristöille. FFMpeg pakkaa ja purkaa miltei mitä tahansa videotyyppiä. (Gigafide 2009.)

X264 tuottaa H.264-formaatteja ja sen säiliö on MP4-päätteinen, mikä on myös Youtuben käyttämä videoformaatti. Tämä formaatti on erinomainen videon esittämiseen Internetissä. Formaatin ja kodekin valinnassa on otettava huomioon toistomahdollisuus eli kuinka laajasti formaatti on tuettu. (Ozer 2011b.) Kodekit eivät toimi yksinään, niitä on käytettävä ympäristössä, jossa on mahdollisuus videon toistoon. Näitä ovat esimerkiksi Windows Media, Flash ja/tai QuickTime, jotka tarjoavat työkalut esimerkiksi pakkaukseen ja toistoon. (Gigafide 2009; Ozer 2009a.)

Pakkausmenetelmät poikkeavat toisistaan pääasiassa seuraavilla ominaisuuksilla:

- tiedoston kokoero, esimerkiksi kahden tunnin elokuva vie AVI:lta 3 gigatavua enemmän tilaa kuin Matroskalta
- tuki kehittyneille kodekkiominaisuuksille (B-kehys tai VBR-koodaus)
- tuki kehittyneille sisällöille (kappaleet, tekstitys)
- tuki suoralähetykselle (engl. live/true streaming). (Säiliömuoto 2012.)

5.1 Ogg Theora

Ogg Theora on Xiph.org-säätiön omistama patentoimaton ja rojaltimaksuton videoformaatti. Theora jatkokehitettiin VP3-kodekista, joka kuuluu On2 Technologiesin kehittämiin kodekkeihin. Syyskuussa 2003 On2-yritys lahjoitti VP3:n Xiph.org-säätiölle. Xiph.org-säätiö kehitti Vorbis-audiokodekin. Säätiö on voittoa tavoittelematon organisaatio. Theoraa implementoidaan useimmin myös rojaltimaksuttomissa Linux-järjestelmissä (Ubuntu, Debian, Fedora). (Hyde 2009.)

Theoralla saa hyvän laatuista videoita tai jopa verrattavissa H.264:een. DiBonan tekemän vertailun mukaan 499 kbit/sec ja 327 kbit/sec molemmat kodekit pystyvät tuottamaan lähes samanlaatuista videon. (DiBona 2009.) Toisessa samankaltaisessa vertailussa Mertenkin yhteni samaan päätökseen; Theora pystyy tuottamaan lähes H.264:n tasoista videon. (Merten 2012.)

Theoran huono puoli on se, että samalla laadulla se tuottaa suuremman tiedostokokoon H.264:ään verrattuna. Tämän lisäksi on melko harvoja työkaluja, jotka pakkaavat videoita Ogg Theora -formaattiin. Sama ongelma ilmenee myös WebM-videoilla. Theoran kehityksessä on ollut katkoksia ja sitä ei ole saatu täysin valmiiksi, joten sillä ei ole laaja tuki esimerkiksi videoeditointiohjelmilta. (Vakhromova 2012.) Chrome, Firefox, Opera, SeaMonkey ja Konqueror sekä mobiililaitteissa Fennec -selaimet tukevat Theoraa ja HTML 5:tä (kuva 8).



Kuva 8. Theoraa ja HTML 5:tä tukevat selaimet. (Dale & Lee 2011, 218.)

Theora vaatii toistolaitteissa kuten Windows Media Playerissa tai Quicktime Playerissa lisäosien asentamista. Xiph.org-säätiön websivulta voi ladata esimerkiksi DirectShow-purkajan, joka tukee myös Vorbis, Speex tai Flac – kodekkeja. Kuitenkin Linux-järjestelmissä tuetaan Theoraa natiivista sekä seuraavat toisto-ohjelmat pystyvät toistamaan Theora-videoita ongelmitta ilman lisäasennuksia:

- Miro
- MPlayer
- VLC (Hyde 2009.)

5.2 MP4 H.264

H.264 on patentoitu videonpakkausstandardi. International Standards Organization (ISO) julkaisi sen ensimmäisen videostandardin MPEG-1:n vuonna 1993 ja MPEG-2:n sitä seuraavana vuonna. MPEG-4 tuli vuonna 1999. H.264/AVC sai alkunsa ISO:n ja International Telecommunications Union (ITU) yhteistyönä vuonna 2002. ISO:n ja ITU:n yhteistyöstä H.264-kodekin

kehittämisessä ja kodekin käytetyistä nimityksistä näkyy tarkemmin kuvasta 9. H.264 tunnetaan myös nimellä MPEG-4 part 10 tai MPEG-4 AVC (engl. Advanced Video Coding). AVCHD on H.264:n perusformaatti. H.264-kodekkia käytetään lähes kaikissa laitteissa iPodista satelliittitelevisioon. Ainoa alue, jossa H.264-kodekkia ei voi käyttää on videoiden editointiformaattina. H.264-videoita on pääosin Youtube-videoina, jotka muodostavat valtavan 40 % kaikista Internetin videoista. (Ozer 2011b.)

ITU –		ISO –	
International Telecommunications Union		International Standardization Organization	
Telephone, Radio, TV		Photography, Computer, Consumer Electronics	
1984	H.120		
1990	H.261 – Video Conferencing		
1993			MPEG-1 – Video CD
1994	(H.262)		MPEG-2 – Digital Cable and Satellite TV
1995	H.263 – Improved Video Conferencing		
1997			ATSC – U.S. HDTV
1999			MPEG-4
2002	AVC (H.264)		AVC (MPEG-4 Part 10)

Kuva 9. H.264-kehitys (Ozer 2011d).

H.264-lisenssimaksu voi tarkastaa lisää www.mpeg-la.com-sivuilta. H.264 on ainoa kodekki, jota Apple, Adobe ja Microsoft, kolme merkittävää suoratoiston tuottajaa, tukevat. (Ozer 2011d.)

MP4-tiedosto ja sen nimityksiä:

- .MP4, MPEG-4:n virallinen kääreformaatti
- .M4V, applen eri variantti iTunesille ja sen laitteille
- .MOV, H.264-tiedosto on tarkoitettu editoinnille tai esittämiselle QuickTimessa
- .F4V, H.264-versio Flashille
- .3GP, kännyköissä
- .MPG, H.264 MPEG-2:n siirrolle (Ozer 2011d.)

MP4-formaatti on yllämainituista sopivin HTML 5 -videota tuottaessa, sillä F4V tai MOV -formaattien kanssa selain voi vaatia Flash- tai QuickTime - mediasoittimen. (Ozer 2011d.) H.264:sta tukevat selaimet näkyvät kuvasta 10.



Kuva 10. H.264:sta tukevat selaimet. (Dale & Lee 2011, 217).

H.264:lla on neljä eri profiilia eri tehoisille laitteille verkkovideon striimaukseen: Baseline, Extended, Main ja High ja vastaavasti neljä eri tasoa: 1, 1b, 1.1 ja 1.2. Pienitehoisille iPodille on tarkoitettu Baseline-profiili (BP), hieman tehokkaammalle iPadille Main-profiili (MP) tai sitä tehokkaammalle iPhone 4S:lle High-profiili. Yleisenä suosituksena on se, että käytetään aina korkeinta profiilia ja tasoa kun on mahdollista. Esimerkiksi tuottaessa H.264-videota HTML 5:lle pöytätietokoneissa sopii High-profiili ja mikä tahansa taso, mutta jos haluaa sekä HTML 5:lle että iOS:lle niin matalin profiili on varmempi ratkaisu. (Ozer 2011d.)

H.264/AVC pyrkii tarjoamaan eri pakkausvaihtoehtoja eri kaistansiirtoleveydelle, keskusyksikön (CPU) tehokkuudelle (käsikäyttöiset vs pöytätietokoneet) ja kaikkea näiden väliltä. Kodekki on jaettu eri profiileihin ja tasoihin. Korkeampi profiili tarjoaa enemmän eri ominaisuuksia, paremman laadun ja pienemmän tiedostokoon. Huonona puolena on se, että vie enemmän aikaa koodata ja kuluttaa enemmän CPU:n tehoa. (O'Reilly 2011.) Lähteen mukaan H.264 tarjoaa parempaa laatua samalla bittinopeudella verrattuna XviD:iin ja todetaan myös, että se vie enemmän aikaa ja vaatii enemmän CPU:n tehoa sekä pakkaamisessa että purkamisessa. H.264 on hyvin suosittu HD-videona.

Mediasoittimia, jotka tukevat H.264:ää ovat muun muassa:

- iPod
- Sansa View
- Zune
- Sony NWZ video player
- PSP
- Cowon A3. (Odegard 2008.)

5.3 WebM VP8

WebM on Googlen kehittämä multimediasäiliö, jota perustuu Matroska-säiliöön. WebM käyttää VP8-kodekkia ja Vorbis-audiokodekkia. VP8 on BSD (engl. Berkeley Software Distribution) -lisenssin alla. Molemmat kodekit ovat avoimia ja rojaltimeksuttomia. Vuonna 2010 Google julkaisi VP8-kodekin, joka kuuluu

On2 Technologiesin kehittämiin kodekkeihin. Google osti kodekin kehittäneen On2-yrityksen vuonna 2010 133 miljoonalla US dollarilla (Graham 2010). Muita On2 kehittämiä kodekkejä ovat muun muassa VP3, VP6 ja Vp7. VP3:sta kehittyi myöhemmin Theora-kodekki, VP6:ta käytetään Flash-videoissa ja VP7:ää käytetään Skype-ohjelmassa. (Cohen 2010; Doig 2010.)

WebM VP8 on suunniteltu vaihtoehtoisena patentoiduille kodekeille kuten H.264:lle, MPEG4:lle, MPEG2:lle tai Flash-formaatille (WebM Files 2010.) VP8-kodekki on helppo integroida muihin olemassa oleviin järjestelmiin. VP8 pystyy hyödyntämään tehokkaasti kaistanleveyttä videonjakelussa. Se ei myöskään vaadi paljon tehoa videon toistossa, mikä sopii hyvin pienitehoisille laitteille kuten kannettaville ja käsi käyttöisille koneille ja tableteille. VP8-kodekin laatu on parempi kuin H.264 Base –profiiliin, sen muistitilantarpeen vaatimus on myös pienempi kuin H.264:n. Pakkauksessa VP8 käyttää I- ja P –kuvia, mutta ei B-kuvaa patenttisyyistä. (Cohen 2010; Doig 2010.)

VP8:aa löytyy seuraavista ohjelmista:

- Android
- Corecodec
- Skype
- Sorenson Media
- Telestream
- Wildform (Cohen 2010; Schonfeld 2010.)

VP8:aa käytetään seuraavissa laitteissa:

- AMD
- ARM
- Broadcom

VP8:aa käytetään seuraavissa laitteissa (jatkuu):

- Logitech
- Nvidia
- Qualcomm
- Texas Instruments (Cohen 2010; Schonfeld 2010.)

VP8:n sovelluslustoja ovat seuraavat:

- Brightcove
- Encoding.com
- HD Cloud
- Kaltura
- Ooyala
- Youtube
- Zencoder (Cohen 2010; Schonfeld 2010.)

WebM VP8:n hyvät puolet ovat siinä, että se on avoin ja se on optimoitu Internetiin. WebM:ää tuetaan natiivisti Google Chromessa, Mozilla FireFoxissa ja Operassa (kuva 11). Formaatin muita hyviä puoleja ovat seuraavat:

- korkealaatuinen reaaliaikaiselle videotoistolle
- tarjoaa paremmin laadun kuin H.264 Base-profiili
- pienempi muistintarve verrattuna H.264 Base-profiiliin
- helppo pakkausasetukset
- ei tarvetta käyttää I-kuvaa toistossa (Cohen 2010; WebM Project 2012.)



Kuva 11. WebM-formaattia tukevat selaimet (Dale & Lee 2011, 218).

6 HTML 5

HTML on World Wide Web Consortiumin (W3C) esittämä verkkosivujen tekemiseen käytetty kuvauskieli. W3C on kansainvälinen organisaatio, joka ylläpitää ja kehittää WWW:n standardeja tai suosituksia. W3C perustettiin Tim Berners-Leen johdolla vuonna 1994. (World Wide Web Consortium 2012.) HTML 5 on uusin versio HTML:stä ja sen standardoinnin valmistumista odotetaan vuonna 2020. Vuonna 2004 Applen, Mozilla-säätiön ja Operan työntekijät perustivat yhdessä HTML Working Grouping (HTMLWQ) -ryhmän, jonka päämääränä on kehittää HTML:ää ja API:ta Web-käyttöä varten. (Ozer 2011a.)

Tarve HTMLWQ-ryhmälle lähti siitä kun yritykset katsoivat W3C:n keskittyvän liikaa XHTML-kehittämiseen sivuuttaen HTML:n. HTML 5 sai enemmän tukea, kun Apple ilmoitti luopuvansa Flashista ja tukevansa HTML 5-tekniologiaa videon esittämiseen iPad-laitteissa. HTML 5:n käyttöönotto selaimissa on ollut hidasta ja syynä on se, että sillä ei ole yhtenevää kodekkia, joka toimisi kaikissa selaimissa, eli yhden videon esittämiseen on pakattava vähintään kahta tai kolmea eri formaattia. (Ozer 2011a.)

Kodekkien implementoinnissa on ollut erimielisyyksiä, sillä Apple ei suostu käyttämään Ogg Theoraa, koska sen mukaan kodekilta puuttuu laitteistotukea ja patenteista on epäselvyyksiä. Hakukonejätti Google tukee H.264:sta sekä Ogg Theoraa, mutta ilmoitti kuitenkin ettei se voi tarjota H.264-lisenssiä kolmansille osapuolille. Google myös otti esille Theoran heikon videonlaadun. Sekä Opera että Mozilla kieltäytyivät implementoimasta H.264:ää lisenssistä. Microsoft ei ole ottanut kantaa HTML 5-tuesta. (Ozer 2011a.)

6.1 HTML 5 uudistuksia

HTML 5:ssä dokumenttityypin ilmoittaminen on paljon yksinkertaisempaa `<!DOCTYPE HTML>` -rivillä. HTML 5 tuo paljon uutuuksia ja uudistuksia muun muassa lomakkeisiin (engl. form) ja parempaan websivun jäsentelyyn ja rakenteeseen (engl. semantic/structural elements) sekä helppoja piirtomahdollisuuksia `<canvas>`-elementillä. HTML 5:ssä videon esittäminen on mahdollista, koska uudistukseen kuuluu video-tag, joka mahdollistaa videon suoraan esittämistä selaimissa selainten tarjoamalla natiivilla toisto-ohjelmalla. Videolle on mahdollista asettaa jatkuvaa toistoa loop-attribuutilla tai automaattista esittämistä autoplay-attribuutilla. Äänileikkeiden soittamista on mahdollista audio-tagilla. Muita videosisällön luomisen kannalta merkittävimmät uudet elementit näkyvät taulukosta 4 ja taulukosta 5. (HTML 5 Tag Reference 2012):

Taulukko 4. HTML 5:n uudet mediaelementit (HTML 5 Tag Reference 2012).

Tagi	Kuvaus
<code><audio></code>	Pelkkä audio-sisällölle
<code><video></code>	Video- tai elokuva-sisällölle
<code><source></code>	Määrittelee median sijainnin
<code><poster></code>	Posteri-kuvan videolle
<code><embed></code>	Lisäosien upottamiseen

Taulukko 5. Video-tagin attribuutit (HTML 5 Tag Reference 2012).

Attribuutti	Kuvaus
Src	Määrittää median URL:n
Autoplay	Aloittaa toiston automaattisesti
Controls	Näyttää peruskontrollit (play,pause)
Loop	Median jatkuvatoisto
Muted	Äänetön-mahdollisuus
Height	Määrittää toisto-ohjelman korkeuden pikseleinä
Width	Määrittää toisto-ohjelman leveyden pikseleinä

Muita HTML 5:n elementtejä ja ominaisuuksia ovat muun muassa:

- <canvas>-elementti 2D-piirtämiselle
- Paikallinen tallennus (engl. local storage) -mahdollisuus
- <article>, <footer>, <header>, <nav>, <section> -elementit jäsentelyyn
- Kalenteri-, päiväys-, aika-, email-, URL- ja haku-toimintojen lisääminen (HTML 5 Tag Reference 2012.)

iOS-järjestelmä tunnistaa vain ensimmäisen <source>-tagin ja se jättää muut <source>-tagit huomioitta. Tämä tarkoittaa sitä, että ensimmäinen <source>-tagi on varattava iOS-laitteille, jos haluaa median toistavan iOS-laitteissakin. Poster-attribuutti ei myöskään toimi aiemmissa iOS 3:issa. Onneksi nämä ongelmat korjattiin iOS 4:ssä. Android 2.2 tai sen vanhempi versio <source>-tagissa oleva type-attribuuttia tai <video>-tagin controls-attribuuttia ei tueta. Nämä ongelmat korjattiin Android 2.3 -versiossa. (O'Reilly 2011.) Sekä iOS että Android tukevat HTML 5 videota seuraavanlaisessa muodossa:

H.264-videokodekki + AAC-audiokodekki + MP4-säiliö.

6.2 HTML 5:tä tukevat selaimet

HTML 5 <video> ja <audio>-elementtiä tukevat ainakin kuvassa 12 esitetyt selaimet ja kuvassa 13 mobiililaitteet tai niiden uudemmissa versioissaan. (O'Reilly 2011.)



Kuva 12. HTML 5 –videota tukevat selaimet. (O'Reilly 2011.)



Kuva 13. HTML 5 –videota tukevat mobiililaitteet. (O'Reilly 2011.)

Taulukosta 6 näkyy vuoden 2012 tilasto selainten käyttösuudesta kesä- ja syyskuun väliltä. Taulukosta selviää, että Chromen käyttösuus on suurin, toiseksi tulee Firefox ja kolmanneksi Internet Explorer (lyh. IE). Tosin IE-selaimen käytössä on ilmennyt pientä jatkuvaa laskua. Safarin ja Operan käytössä ei ole tapahtunut paljoa muutosta. (Browser Statistics 2012.) Tämän tiedon perusteella WebM, MP4 ja Ogg –videoformaatteja on testattu näissä mainituissa selaimissa.

Taulukko 6. Selainten käyttösuus 2012 (Browser Statistics 2012.)

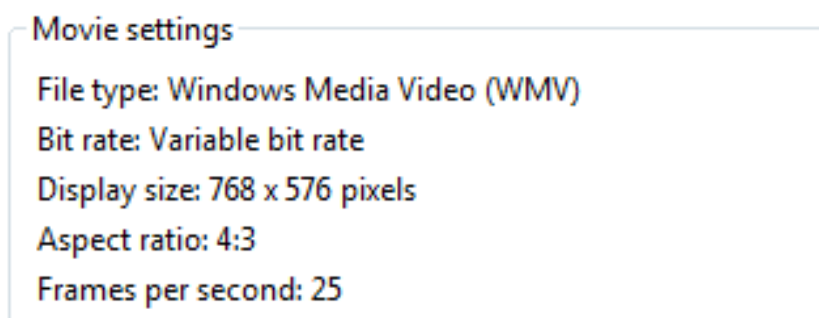
Browser Statistics Month by Month

2012	Internet Explorer	Firefox	Chrome	Safari	Opera
September	16.4 %	32.2 %	44.1 %	4.2 %	2.1 %
August	16.2 %	32.8 %	43.7 %	4.0 %	2.2 %
July	16.3 %	33.7 %	42.9 %	3.9 %	2.1 %
June	16.7 %	34.4 %	41.7 %	4.1 %	2.2 %

7 VIDEON NÄYTTÄMINEN SELAIMISSA

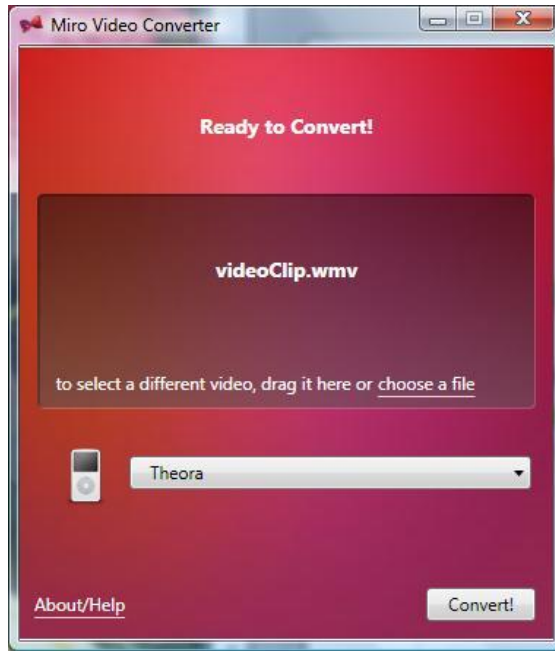
7.1 Videon pakkaaminen

Videoeditointiin on käytetty Windows Movie Maker –ohjelma, joka tuotti *.wmv-päätteisen videotiedoston eli säiliön. Videon asetuksena on käytetty VBR:ää, kuvasuhde on 4:3 ja kuvanopeus 25 (kuva 14). Videosta on poistettu audiodata.

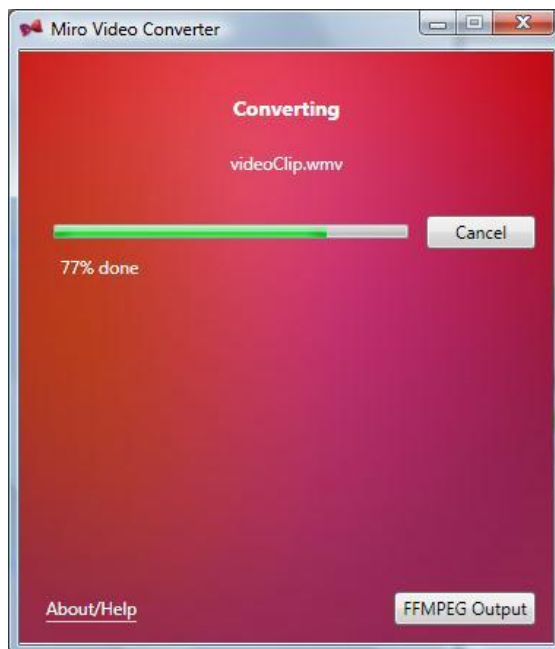


Kuva 14. Videon tuottaminen WMM-ohjelmalla.

Tämän jälkeen on käytetty Miro Video Converter –ohjelmaa pakkaamaan videota kolmeen eri formaattiin, joita ovat MP4, Ogg ja WebM. Ohjelmalla ei pysty vaikuttamaan paljon pakkausasetukseen, mutta se tarjoaa yleisimmät kodekit ja se sisältää valmiina Ffmpeg:n ja Ffmpeg2theoran, joiden avulla voi muuttaa asetuksia tarpeen mukaan komentoriviä käyttäen sekä Mac-, Linux että Windows-koneissa. Kuvassa 15 ja 16 muunnetaan *.wmv-videon Ogg-formaattiin. Tässä työssä Ogg-formaatissa on käytetty Theora-kodekkia, WebM VP8-kodekkia ja vastaavasti MP4 H.264-kodekkia.







Kuva 15. Videon pakkaus – kodekin valitseminen.



Kuva 16. Videon pakkaus käynnissä.

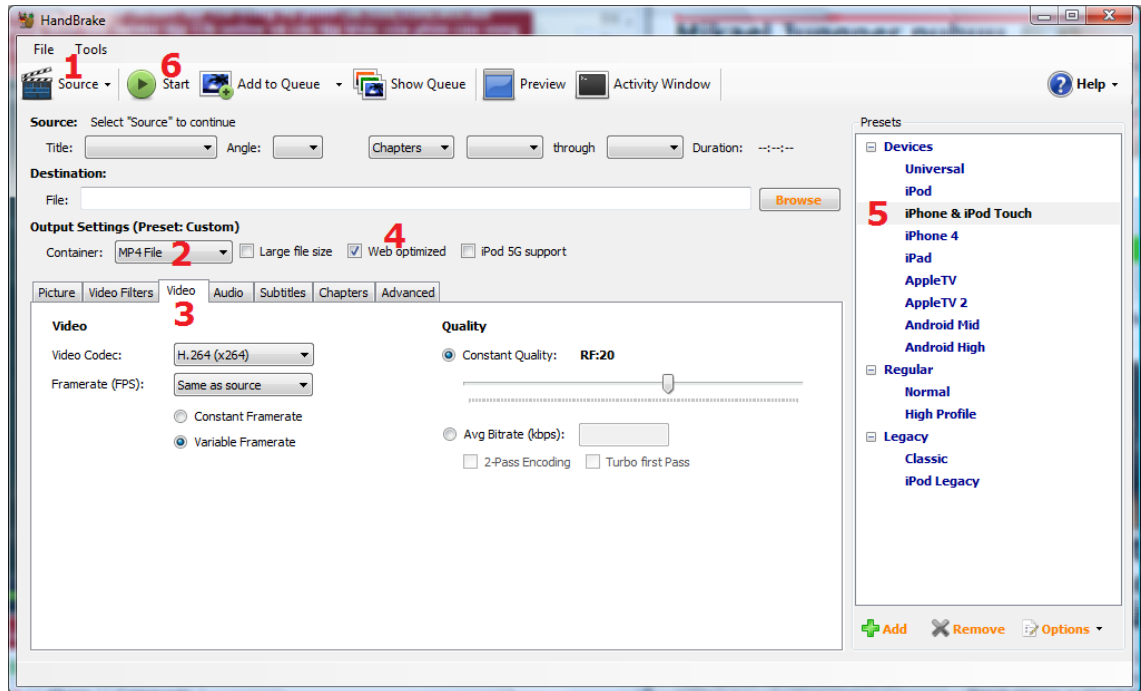
Pakkausohjelmalla saatiin halutut videoformaatit. Pakkauksesta huomattiin, että tuotettujen tiedostojen koko ei ole sama (kuva 17). Samasta *.WMV-videoleikestä WebM-formaatti tuotti pienimmän ja Ogg-formaatti suurimman tiedostokoon. Pakkausohjelman vaikutusta pakkaukseen ja tiedostokokoon ei ole tässä opinnäytteessä laajemmalti tutkittu. Videoita on testattu VLC- ja Quicktime-toisto-ohjelmissa ennen niiden testaamista selaimissa. VLC-toisto-ohjelmassa kaikki formaatit toimivat paitsi WebM-formaatti. MP4-formaatti näyttää lohkomaiselta VLC:ssä. Ogg/ WMV/ WebM-videoita ei pystytty toistamaan Quicktimessa. Ainoa formaatti, joka toimii Quicktimessa on MP4.

 miroMp4.mp4	1,747 KB
 miroWebm.webm	295 KB
 miroOgg.ogv	3,928 KB
 videoClip.wmv	4,568 KB

Kuva 17. Pakkauksesta saadut tiedostokoot.

Handbrake-pakkausohjelma (kuva 18) toimii Linux, Mac sekä Windows – ympäristöissä. Handbrake-pakkausohjelmalla pystyy vaikuttamaan pakkausasetuksiin ainakin seuraavasti (O'Reilly 2011):

- 1 = source, lähde
- 2 = container, säiliö
- 3 = video, audio, subtitles-välilehdet (kuvassa video-välilehti). Video-välilehdessä pystyy vaikuttamaan videon kodekkiin, kuvannopeuteen tai laatuun
- 4 web optimized, valinta mahdollistaa web-optimoinnin
- 5 iPhone & iPod Touch, optimoi videon näyttämisen näille laitteille
- 6 start, pakkaa videon



Kuva 18. Handbrake-pakkausohjelma.

7.2 HTML 5 –sivun luominen

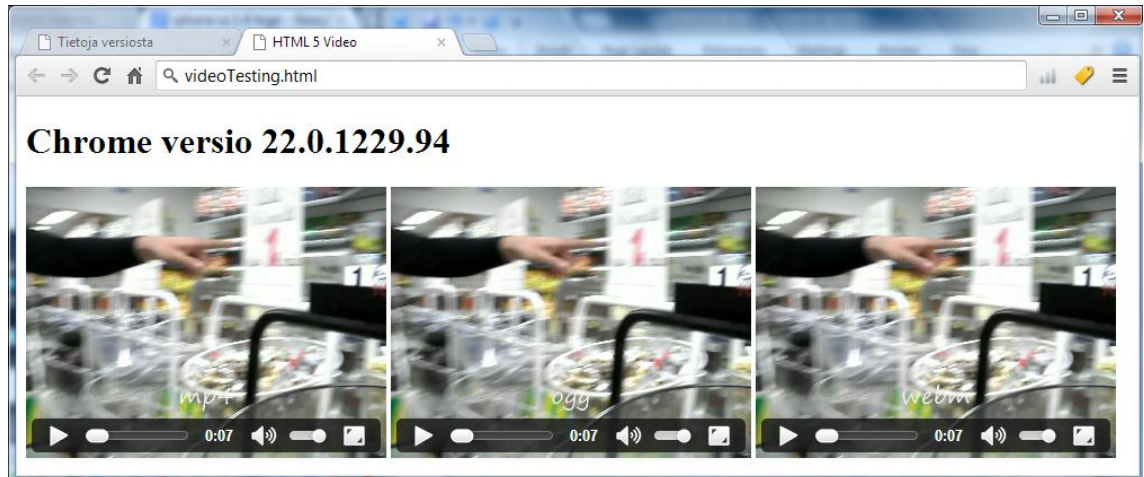
Notepadissä rakennettiin yksinkertainen HTML 5-sivu (kuva 19). MP4 kannattaa laittaa ensimmäisenä, jos haluaa videon toistavan myös Apple-laitteissa. Apple-laitteet tunnistavat vain ensimmäisen source-tagin ja jättää muut tagit huomioitta.

```
1 <!DOCTYPE HTML>
2 <html>
3 <body>
4
5 <video width="320" height="240" controls="controls">
6 <source src="miroMp4.mp4" type="video/mp4">
7 <source src="miroOgg.ogv" type="video/ogg">
8 <source src="miroWebm.webm" type="video/webm">
9 <p>Your browser does not support the video tag.</p>
10 </video>
11
12 </body>
13 </html>
```

Kuva 19. HTML 5-sivun luominen.

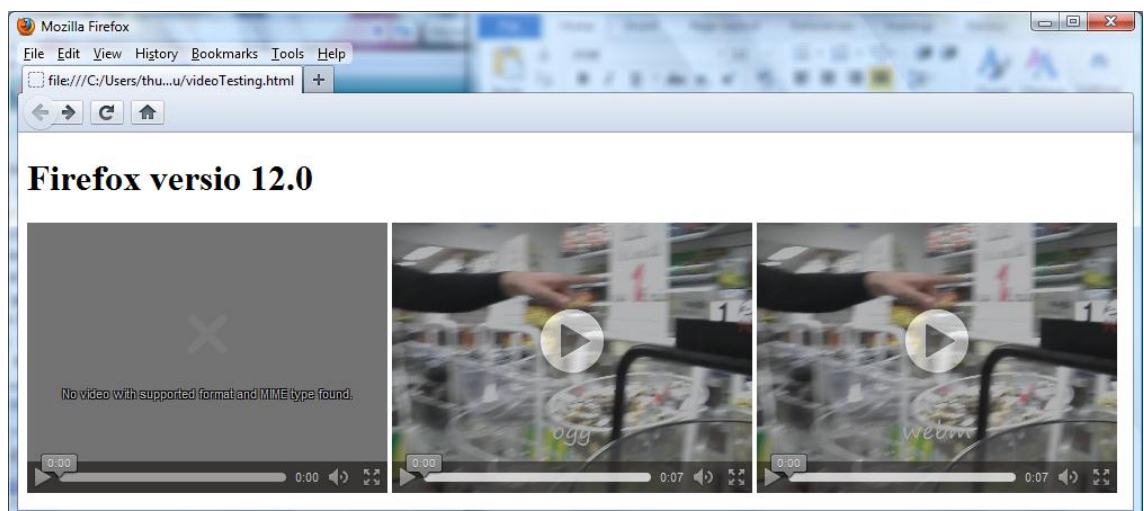
7.3 Testitulokset

Seuraavaksi kaikkia kolmea videoformaattia testattiin eri selaimissa. Selainten toisto-ohjelma tarjoamat kontrollit poikkeavat jonkin verran toisistaan, kuitenkin kaikilta löytyy seuraavat perusnapit: play, volume ja fullscreen. Tulokset näkyvät kuvista 20 - 24.



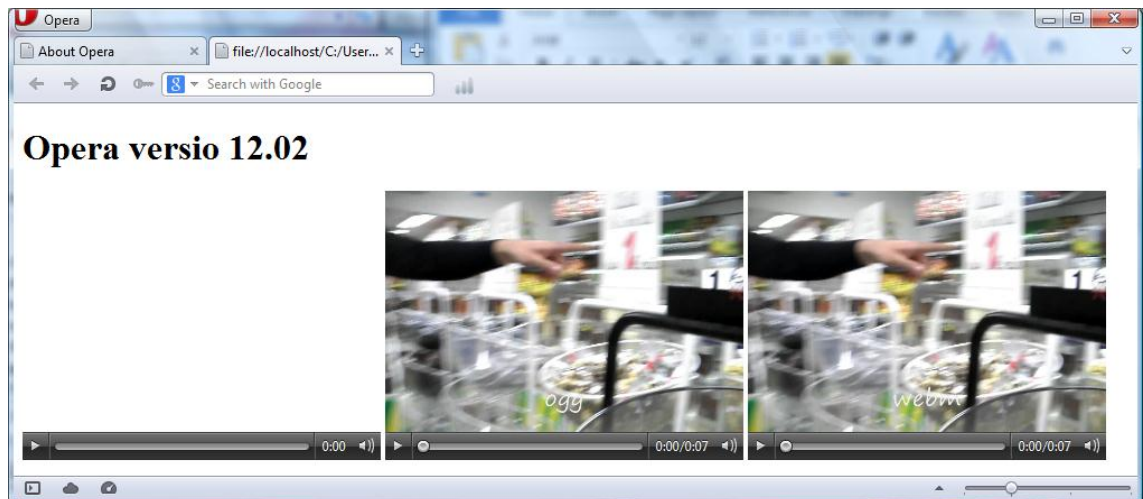
Kuva 20. Video Chromessa.

Chromessa näkyvät kaikki kolme videota ongelmitta esimerkiksi ilman lisäasennuksia. Formaattit ovat vasemmasta oikeaan MP4, Ogg ja WebM. Kuvalaadussa ei ole paljoa eroa. Kaikki kolme toimivat hyvin ja peruskontrollitkin löytyvät (kuva 20).



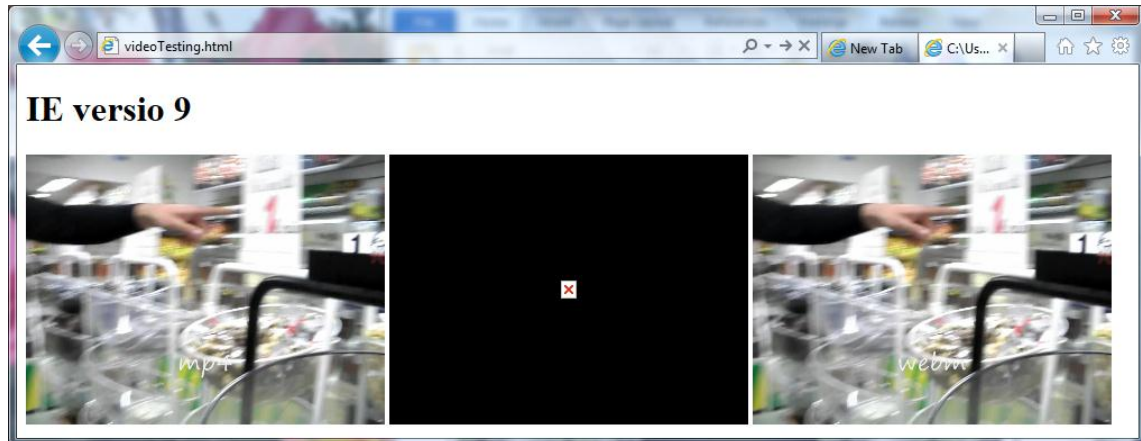
Kuva 21. Video Firefoxissa.

Firefoxissa Ogg- ja WebM-videot toimivat hyvin. Kontrollit ovat samantyyppiset kuin muissakin selaimissa, ainoa ero on kun kuvaa näpättyessä video pysähtyy ja/tai toistuu. MP4 H.264 -formaatti ei toimi Firefoxissa, koska Firefox tukee vain avointa formaattia. (kuva 21).



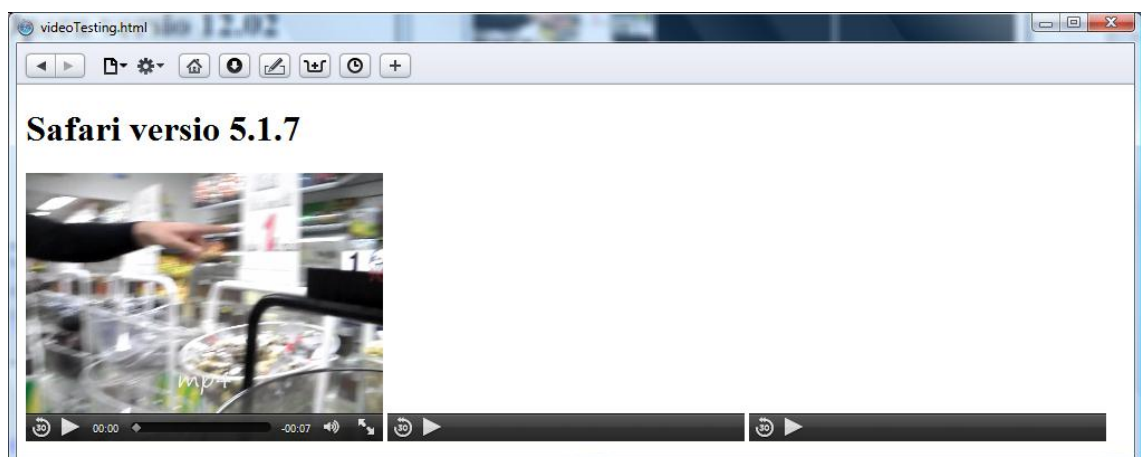
Kuva 22. Video Operassa.

Operassa Ogg- ja WebM-video toimivat hyvin. Operassa toisto-ohjelman kontrollit ovat valmiina näkyvissä (vrt esimerkiksi Chromeen). Kuten Firefox, Operakaan ei tue MP4 H.264 -formaattia. Sekä Firefoxissa että Operassa ei tarvittu lisäosien asentamista videoiden toistossa (kuva 22).



Kuva 23. Video Internet Explorerissä.

Explorerissa MP4 ja WebM toimivat hyvin. Jotta WebM-video näkyisi IE 9:ssä, käyttäjän on itse asennettava VP8-kodekin (O'Reilly 2011). Videon kontrollit tulevat vasta näkyviin kun hiiri on päällä (kuva 23). IE tarjoaa 30 sekunnin eteenpäinkelausmahdollisuus, muissa tämä ominaisuus ei ole tarjolla. IE:ssä video tulee näkyviin vasta kun on poistanut block content –ominaisuuden selaimelta.



Kuva 24. Video Safarissa.

Safarissa MP4 toimii hyvin sen jälkeen kun Quicktime on asennettuna koneeseen (kuva 4; O'Reilly 2011). Videossa on 30 sekunnin taaksepäinkelausmahdollisuus (vrt IE).

Taulukosta 7 näkyvät tulokset. Tähti * = selain vaatii Quicktime-asentamista, ** = selain vaatii VP8 asentamista. Taulukossa 7 näkyvät tulokset.

Taulukko 7. Videoformaattien toistomahdollisuuksia eri selaimissa.

	Chrome	Firefox	Opera	IE	Safari
MP4	OK	-	-	OK	OK*
Ogg	OK	OK	OK	-	-
WebM	OK	OK	OK	OK**	-

8 POHDINTA

Opinnäytteessä tutkittiin HTML 5:n uudistuksia, videoformaatteja ja kodekkeja, videon ominaisuuksia ja formaattien tukea selaimissa. Videota laitettiin HTML 5-sivulle ja sivua testattiin Windows-järjestelmässä Chrome-, Firefox-, Opera-, Internet Explorer ja Safari-selaimissa. Tuloksista näkyy (taulukko 7), että Chrome-käyttäjillä kaikki H.264-, Ogg- sekä WebM-formaatti näkyvät hyvin. Operassa ja Firefoxissa toimii hyvin avoin lähdekoodi Ogg- ja WebM-formaatti. Internet Explorerissä näkyy MP4-formaatti ongelmitta, mutta WebM-formaatin käytössä on vaadittu lisäosan asentamista koneelle.

Safarissa näkyy ainoastaan MP4-formaatin, joka vaatii Quicktimen asentamista. Testattujen formaattien lisäksi tarjotaan yleensä vielä Flash-formaatti siltä varalta, että jotkut käyttäjät käyttävät edelleen vähän vanhempia selaimia. Ja Flash-formaatti on sijoitettava object-tagiiin, koska vanhat selaimet eivät ymmärrä video-tagia. (Bluejay 2010.) WebM-formaattia tuetaan kaikissa neljässä selaimissa, lukuunottamatta Safarissa. Ogg- ja MP4-formaatti saavat yhtä paljon tukea, mutta tuloksesta näkyy, että Ogg on hieman suositumpi sillä MP4:n toisto Safarissa vaatii hieman lisätyötä käyttäjille.

Tehokas MP4 H.264 -formaatti vaatii paljon CPU-tehoa toistonaikana ja se sopii enemmänkin videoille, jotka sisältävät paljon toimintaa. Tutor-tyyppisissä videoissa MP4-formaatissa ei ole paljoa hyötyä. Työssä on käytetty Windows Movie Maker –videoeditoriohjelmalla videon editointiin. Miro-pakkausohjelmaa on käytetty muuntamaan videopätkät eri formaatteihin. Videoformaatteja ei ole testattu eri selaimissa Mac-järjestelmässä tai mobiililaitteissa. Tämä olisi täydentänyt opinnäytteen tavoitetta.

HTML 5:n uudistukset helpottavat videon laittamisen suoraan omalle websivulleen ilman toisen videopalvelun käyttöä tai toisto-ohjelmien asentamista. HTML 5:n huono puoli on kuitenkin siinä, että tällä hetkellä sitä ei ole standardoitu. Toiseksi HTML 5:ssä ei voida vielä käyttää yhtä ainoa videoformaattia, joka toimisi kaikissa selaimissa ja järjestelmissä. Tämä tarkoittaa videoiden Internetiin lataaville käyttäjille enemmän työtä, sillä yhdestä videoleikkeestä on pakattava ainakin kolmeen eri formaattiin, jos haluaa ottaa suurimman videokatsojaryhmän huomioon. Kohderyhmän valitessa on mietittävä käyttäjien käyttötottumukset ja mieltymykset, näihin taas vaikuttavat tarjolla olevien ohjelmistojen ja laitteistojen saatavuus, tuki ja suosio.

LÄHTEET

Bluejay, M. 2010. Getting video onto your website: Web video demystified. Viitattu loka/2010
<http://websitehelpers.com/video/>.

Branckaute, F. 2010. History of Online Video. Viitattu 27.10.2010
<http://www.blogherald.com/2010/10/27/history-of-online-video/>.

Browser Statistics 2012. W3schools.com. Viitattu 1.9.2012
http://www.w3schools.com/browsers/browsers_stats.asp.

Chapman, C. 2009. The History of the Internet in a Nutshell. Viitattu 15.11.2009
<http://sixrevisions.com/resources/the-history-of-the-internet-in-a-nutshell/>.

Chen, H. 2012. Video Compression Tutorial. Viitattu 1.12.2012
http://djj.ee.ntu.edu.tw/Video_Compression_陳信慧.pdf.

Cohen, Y. 2010. VP8 for Embedded Video. Viitattu 16.12.2012
<http://www.slideshare.net/DSPiP/google-vp8#btnNext>.

Dale, C. & Lee, J. 2011. HTML 5 Multimedia Development Cookbook. Olton Birmingham:
Packt Publishing Ltd.

Dave 2009. What size should your web video be? Viitattu 3.7.2009
<http://webvideouniversity.com/podcast/video/2009/07/03/what-size-should-your-web-video-be/>

DiBona, C. 2009. YouTube/ Ogg/Theora comparison. Viitattu 13.6.2009
<http://people.xiph.org/~greg/video/ytcompare/comparison.html>

Doig, J. 2010. Introducing WebM, an open web media project. Viitattu 19.5.2010
<http://blog.webmproject.org/2010/05/introducing-webm-open-web-media-project.html>.

Ekonoja, A. 2011. Kaappausvideot ja interaktiivinen taulu – Luento 5. Viitattu 19.10.2011
<http://appro.mit.jyu.fi/ope/luennot/luento5/>

Gigafide 2009. The Basics of Video Encoding. Viitattu 21.8.2009
http://www.youtube.com/watch?v=_nPsBwMivCk.

Harris, T. 2012a. How File Compression Works. Viitattu 13.12.2012
<http://computer.howstuffworks.com/file-compression.htm>.

Harris, T. 2012b. Lossy and Lossless Compression. Viitattu 13.12.2012
<http://computer.howstuffworks.com/file-compression3.htm>.

HTML 5 Tag Reference 2012. W3Schools.com. Viitattu 13.9.2012
http://www.w3schools.com/html5/html5_reference.asp.

Introduction 2012. Introduction to Video Compression. Viitattu 16.12.2012
http://www.cse.iitb.ac.in/synerg/lib/exe/fetch.php?id=public%3Aproj%3Amultimedia-content-adaptation-to-mobile-devices&cache=cache&media=public:proj:ganesh_video_adaptation:introduction-to-video-compression.pdf.

Investintech.com 2012. The History of the Internet. Viitattu 13.12.2012
<http://www.investintech.com/content/historyinternet/>.

Longtail 2012. Web Video Compression. Viitattu 1.9.2012
<http://www.longtailvideo.com/support/jw-player/26/web-video-compression>.

Mediacollege.com 2012a. Introduction - How to Create Streaming Video. Viitattu 24.6.2012
<http://www.mediacollege.com/video/streaming/overview.html>.

Mediacollege.com 2012b. HTTP Streaming Video. Viitattu 25.6.2012
<http://www.mediacollege.com/video/streaming/http.html>.

Mediacollege.com 2012c. Streaming Video Servers. Viitattu 25.6.2012
<http://www.mediacollege.com/video/streaming/server.html>.

Merten, M. 2012. Another online-video comparison. Viitattu 1.9.2012
<http://people.xiph.org/~maikmerten/youtube/>.

Milano, D. 2008. Choosing the Right Web Video Codec. Viitattu 20.7.2012
www.rhozet.com/whitepapers/choose_right_web_video_codec.pdf.

Odegard, A. 2008. Video Format Guide. Viitattu 1.11.2008
<http://anythingbutipod.com/2008/11/video-format-roundup/>.

Hyde, A. 2009. Ogg Theora Cook Book. Viitattu 15.8.2009
<http://www.flossmanuals.net/ogg-theora/>.

O'Reilly 2011. Video on the Web. Viitattu 19.7.2012
<http://diveintohtml5.info/video.html>.

Ozer, J. 2009a. Streaming 101: The Basics - Codecs, Bandwidth, Data Rate and Resolution. Viitattu 5.2.2009
<http://www.streaminglearningcenter.com/articles/streaming-101-the-basics---codecs-bandwidth-data-rate-and-resolution.html>

Ozer, J. 2009b. Streaming 102: Codecs, plus VBR and CBR, and I, B and P Frames. Viitattu 9.2.2009
<http://www.streaminglearningcenter.com/articles/streaming-102-codecs-plus-vbr-and-cbr-and-i-b-an-p-frames.html>

Ozer, J. 2011a. What is HTML 5? Viitattu 10.3.2011
<http://www.streaminglearningcenter.com/articles/what-is-html5-2.html>.

Ozer, J. 2011b. What is a codec. Viitattu 21.3.2011
<http://www.streaminglearningcenter.com/articles/what-is-a-codec.html>.

Ozer, J. 2011c. Book Excerpt: Streaming vs. Progressive Download vs. Adaptive Streaming. Viitattu 26.5.2011
<http://www.streaminglearningcenter.com/articles/book-excerpt-streaming-vs-progressive-download-vs-adaptive-streaming.html>.

Ozer, J. 2011d. Encoding for HTML 5. Viitattu 18.7.2011
http://www.streaminglearningcenter.com/attachments/download/105/encoding%20for%20HTML5_SME2012.pdf.

Ozer, J. 2011e. What Video Formats Should You Stream? Here's How To Decide. Viitattu 22.9.2011
<http://www.onlinevideo.net/2011/09/what-video-formats-should-you-stream-heres-how-to-decide/>.

Ratol 2012. Bittikarttagrafiikka: Bitit ja kuvan värimäärä. Viitattu 1.9.2012
<http://www.ratol.fi/opensource/foto6/bitti/bittivarit.htm>.

Robertson, M. 2009. Video File Container Formats, Compression and Codecs – Oh My! Viitattu 29.11.2009.
<http://www.reelseo.com/file-formats-containers-compression/#axzz2ElcsHq6t>.

Säiliömuoto 2012. Wikipedia. Viitattu 6.8.2012
<http://fi.wikipedia.org/wiki/Säiliömuoto>.

Schonfeld, E. 2010. Google, Mozilla, And Opera Take on H.264 With The WebM Project, A new Royalty-Free Video Codec. Viitattu 19.5.2010
<http://techcrunch.com/2010/05/19/webm-google-h-264/>.

Sevitt, D. 2012. The Three Types of Online Video For Business. Viitattu 1.9.2012
<http://www.reelseo.com/types-online-video-business/>.

Teuhola, J. 2009. Videon pakkausmenetelmiä. Viitattu 12.7.2012
http://staff.cs.utu.fi/kurssit/digitaalisen_videonkasittelyn_perusteet/syksy-2009/slides/6-Pakkaus.pdf.

Top 20 2012. Internet World Stats. Viitattu 19.6.2012
<http://www.internetworldstats.com/top20.htm>.

Vakhromova 2012. What do you need to know about HTML 5 video. Viitattu 12.5.2012
<http://www.sitepoint.com/what-do-you-need-to-know-about-html5-video/>.

Videonpakkaus 2012. Wikipedia. Viitattu 13.11.2012
<http://fi.wikipedia.org/wiki/Videonpakkaus>

WebM Project 2012. About WebM. Viitattu 9.9.2012
<http://www.webmproject.org/about/>.

WebM Files 2010. WebM – The new Video File Format. Viitattu 31.7.2010
<http://www.webmfiles.org/>.

WordPress 2008. Video Compression Technology advantage and disadvantage. Viitattu 9.12.2008
<http://ravipowar1.wordpress.com/2008/12/09/video-compression-technology-advantage-and-disadvantage/>.

World Wide Web Consortium 2012. Wikipedia. Viitattu 17.8.2012
<http://fi.wikipedia.org/wiki/W3C>.

Yoo, C.; Kang H. 2011. Enhanced compression of integral images by combined use of residual images and MPEG-4 algorithm in three-dimensional integral imaging. Viitattu 15.9.2011
<http://www.sciencedirect.com/science/article/pii/S0030401811006663>.