FÖRORD

Detta dokument är tänkt som en översiktlig sammanfattning av innehållet i kursen TSBK35 Kompression av ljud och bild och bygger på Harald Nautschs föreläsningsmaterial. Detta är ingen fullständig genomgång av teorin som tas upp i kursen. Jag reserverar mig dessutom för eventuella fel (av godtycklig natur).

Texten är strukturerad i ungefär samma ordning som föreläsningsserien, vilket kanske kan kännas ointuitivt för en utomstående. Grovt sett så tas källkodningsmetoder och allmänna begrepp inom komprimering upp i början, följt av kvantisering, prediktiv kodning, transformkodning, delbandskodning, ljudkodning och videokodning.

Trots dessa försiktiga inledningsord hoppas jag att dokumentet kan utgöra intressant och upplysande läsning!

INNEHÅLL

ARITMETISK KODNING....3
GOLOMBKODNING....3
SKURLÄNGDSKODNING....3
HUFFMANKODNING....3
MEDELDATATAKT....4
SJÄLVINFORMATION....4
ENTROPI....4
LEMPEL-ZIVKODNING....4
LZ77....4
LZ78....4
LOSSLESS JPEG....4
JPEG-LS....5
GIF....5
PNG....5
KVANTISERING....5
LIKFORMIG KVANTISERING....5
LLOYD-MAX-KVANTISERING....5
LLOYDS ALGORITM....6
KVANTISERING MED KOMPANDER....6
KVANTISERING MED KÄLLKODNING....6
VEKTORKVANTISERING....6
LBG-ALGORITMEN (K-MEANS)....6
PREDIKTIV KODNING....6
LINJÄR PREDIKTION....7
TRANSFORMKODNING....7
KARHUNEN-LOEVE-TRANSFORM....7
DISKRET COSINUSTRANSFORM....7
DISKRET WALSH-HADAMARD-TRANSFORM....7
BITTILLDELNING VID TRANSFORMKODNING....7
TRÖSKELKODNING....8
JPEG....8
DELBANDSKODNING....8
JPEG 2000....8
PSYKOAKUSTIK....9
HÖRNIVÅ (LOUDNESS)....9
MASKERING....9
MODIFIERAD DISKRET COSINUSTRANSFORM (MDCT)....9
MPEG-1 AUDIO....9
LAYER I....9
LAYER II....9
LAYER III (MP3)....10
DOLBY DIGITAL....10
AAC....10
VORBIS....10
SPECTRAL BAND REPLICATION (SBR)....10
VIDEOKODNING....10
HYBRIDKODNING....11
DIGITAL VIDEO (DV)....11
MPEG-1....11
MPEG-2 (H.262)....11
MPEG-4....11

ARITMETISK KODNING

Varje sekvens som källan producerat representeras med ett tal i intervallet [0, 1), vilket bekvämt kan göras med hjälp av fördelningsfunktionen. Intervallet delas in precis som fördelningsfunktionen, så att varje symbol har ett eget delintervall. När man går igenom sekvensen som ska kodas delas sedan delintervallen upp i nya delintervall som har samma proportioner som tidigare. Symbolen a_i associeras med delintervallet $[F(i-1), F(i))$. Varje sekvens identifierar unikt ett delintervall, och för att beskriva detta intervall kan man välja vilken punkt som helst på intervallet – oftast tar man antingen den nedre intervallgränsen eller intervallets mittpunkt. Det som skickas till avkodaren är förutom fördelningsfunktionen dessa identifieringspunkter, kodade binärt.

Ett praktiskt problem med aritmetisk kodning är att en dator inte kan räkna med hur fin precision som helst, och långa sekvenser kan bli väldigt små intervall. För att komma runt detta använder man en metod som bygger på fixpunktsaritmetik, som tillåter en att skicka den kodade sekvensen allt eftersom man kodar den.

Datatakten för aritmetisk kodning är teoretiskt sett litet sämre än för Huffmankodning, men i praktiken är det mycket lättare att komma nära entropin om man använder aritmetisk kodning.

GOLOMBKODNING

Golomkoder är optimala för fördelningar som är monotont avtagande. Mycket mindre sidoinformation än Huffmankod – iställlet för att skicka ett helt kodträd räcker det med att överföra en parameter m. Används bland annat i JPEG-LS. Golomkodning fungerar på det sättet att man representerar heltalet n med kvoten mellan n och m, avrundat nedåt. Denna kvot kallar man för q. Resten kallas r och definieras som $r = n - qm$. Man kodar q som q ettor följt av en nolla. Resten r kodas litet olika beroende på om m är en jämn tvåpotens eller inte.

SKURLÄNGDSKODNING

Istället för att koda varje tecken för sig (eller grupper av tecken) kodar man en följd av likadana tecken som tecknet plus antalet. Fördelen med denna metod är att det kan vara lättare att utnyttja källans minne än om man till exempel Huffmankodat sekvensen direkt. En förutsättning för att det ska vara någon idé att använda skurlängdskodning är förstås att källan tenderar att producera långa delsekvenser av likadana tecken. Metoden används bland annat i faxkodning, där långa sekvenser av vitt respektive svart är det man vill komprimera.

HUFFMANKODNING

Huffmankodning är en enkel metod för att konstruera optimala koder. Man bygger upp ett kodträd där hänsyn tas till hur sannolikt det är att en viss symbol förekommer i en sekvens från källan. Det är möjligt att få bättre prestanda genom att koda flera symboler i taget, men då måste man tänka på att kodträdets storlek växer ganska snabbt, och hela kodträdet måste föras över till avkodaren som sidoinformation. När man bygger upp kodträdet så radar man upp alla symboler som löv. Sedan slår man iterativt ihop de par av noder som har lägst sannolikheter, tills man fått fram ett fullständigt träd där rotnoden har sannolikhet 1. Den binära koden fås genom att man sätter 0 och 1 till nodernas kopplingar. Koden har en variabel kodordslängd, och medelkodordslängden fås genom att man summerar de inre nodernas sannolikheter. Medeldatatakten fås genom att man dividerar kodordsmedellängden med antalet symboler man kodat åt gången.

MEDELDATATAKT

Medeldatatakten brukar betecknas som R (från engelskans *rate*) och är ett mått på hur många bitar per kodad symbol som en kodningsmetod producerar (i snitt).

SJÄLVINFORMATION

Självinformationen i för ett utfall a_i definieras som $i(a_i) = -log(p_i)$, där p_i är sannolikheten för utfallet. Logaritmen kan tas i godtycklig bas, men vanligast är att man använder bas 2. Ju lägre sannolikheten är, desto större självinformation innehåller utfallet. Notera att om sannolikheten är 1 så är självinformationen 0.

ENTROPI

Entropi för en mängd utfall är medelvärdet av självinformationen, viktat med sannolikheterna enligt $H(X) = - p_1 * log(p_1) - p_2 * log(p_2) - ... - p_N * log(p_N)$. Entropin kan ses som ett mått på osäkerheten inom källan.

LEMPEL-ZIVKODNING

Lempel-Zivkodning utnyttjar vad som hänt tidigare i sekvensen för att koda den. Metoden är mycket använd i filkomprimeringssammanhang och finns implementerad i till exempel zip och gzip samt i bildkodningsstandarderna PNG och GIF. Vid Lempel-Zivkodning behöver varken kodaren eller avkodaren känna till källans statistik – prestandan går ändå asymptotiskt mot entropigränsen. En sådan kodningsmetod kallas *universell*.

LZ77

LZ77 är en variant av Lempel-Zivkodning där sekvensen som ska kodas betraktas genom ett glidande fönster som i sin tur är indelat i två delar – *search buffer* och *look-ahead buffer* – som innehåller redan kodade symboler respektive symboler som står på tur för att kodas.

Vid kodningen försöker man hitta den längsta sekvens i search buffer som matchar den sekvens som börjar i look-ahead buffer. Det man skickar till avkodaren är en taltrippel som innehåller en pekare till hur långt in i search buffer sekvensen börjar, längden på sekvensen och nästa symbol som inte matchade. Kodningen brukar ske med kodord av fixlängd.

LZ78

I LZ78 byggs en ordbok av unika sekvenser upp medan kodningen genomförs. I början är ordboken tom, förutom index 0 som betyder ”ingen match”. Varje sekvens som kodas skickas som en taltupel som innehåller index i ordboken för den längsta matchande sekvensen man hittat samt nästa tecken i indata som inte matchade. Den matchade sekvensen tillsammans med nästa icke-matchade tecken i indata läggs till som en ny post i ordboken. Ingen sidoinformation behöver skickas, eftersom avkodaren kan bygga upp en precis likadan ordbok.

LOSSLESS JPEG

Lossless JPEG är en variant i JPEG-standarden för distortionsfri kodning av bilder. Bildpunkterna kodas radvis uppifrån och ned. Färgvärdet på en given pixelposition i bilden predikteras baserat på kringliggande pixlars värden och prediktionsfelet kodas antingen med adaptiv aritmetisk kodning eller med huffmankod. Om man väljer att använda huffmankodning så byggs trädet upp på ett litet speciellt sätt, som inte tas upp här.

JPEG-LS

JPEG-LS är en standard som hanterar både distortionsfri och ”nästan distortionsfri” kodning av bilder. Bildpunkterna kodas radvis uppifrån och ned, och när en given pixels färgvärde ska kodas bildar man först en *kontext* genom att beräkna gradienter för de kringliggande pixlarna, kvantisera dessa och ta fram ett tal, som alltså utgör aktuell kontext. Sedan görs prediktion av pixelvärdet och prediktionsfelet kodas med en golombkod, vars parameter *m* beror av kontexten. Om långa skurar av värden förekommer i bilden kommer skurlängskodning att användas istället.

GIF

GIF står för *Graphics Interchange Format* och är en standard för bildkodning. Metoden bygger på att man definierar en ”virtuell skärm” där mindre bilder läggs in, och för varje liten bild skickas en position och storlek. Färgtabeller som kan innehålla max 256 färger används. Varje delbild kan ha sin egen färgtabell, eller så används istället en global tabell. Index till färgtabellen kodas med LZW.

GIF stöder *interlace*, vilket innebär att man skickar en lågupplöst bild i början som man sedan kompletterar med fler bildpunktsrader allt eftersom.

PNG

PNG är en förkortning för *Portable Network Graphics*. Standarden är ett alternativ till GIF, och använder en variant av LZ77 som kallas *Deflate* (som även används i zip och gzip) med search buffer på 32768 och matchlängder mellan 3 och 258. PNG stöder prediktion, och där går det att välja mellan fem olika prediktorer (varav en motsvarar ”ingen prediktion”.

KVANTISERING

Kvantisering innebär att man går över från ett kontinuerligt alfabet till ett diskret (eller från ett större diskret alfabet till ett mindre sådant). Kvantisering är nödvändigt för att vi ska kunna använda någon källkodningsmetod på signalen i fråga.

En *M*-nivåers kvantiserare har *M+1* stycken *beslutsgränser* och *M* stycken *rekonstruktionspunkter*. Hur dessa är placerade beror på kvantiseringsmetoden

LIKFORMIG KVANTISERING

Vid likformig kvantisering är alla beslutsområden lika stora (utom möjligtvis ändintervallen) och rekonstruktionspunkterna ligger i varje beslutsområdes mittpunkt. Distortionen vid likformig kvantisering fås genom sambandet $D=\Delta^2 / 12$, där Δ motsvarar längden på ett beslutsområde. Detta gäller om antalet kvantiseringsnivåer är stort (*fin kvantisering*).

LLOYD-MAX-KVANTISERING

En Lloyd-Max-kvantiserare är uppbyggd så att distortionen i den kvantiserade signalen minimeras. Man kan komma fram till att den optimala placeringen av rekonstruktionspunkterna är i sannolikhetsmassans tyngdpunkt för varje intervall. Den bästa placeringen av beslutsgränserna är mitt emellan rekonstruktionspunkterna. Minimeringsproblemet som man måste lösa för att hitta dessa punkter och gränser går bara att hantera om det rör sig om relativt enkla fördelningar som till exempel likformiga- eller laplacefördelningar. För att hitta en lösning för godtyckliga fördelningar får man istället använda någon numerisk metod. Lloyds algoritm är ett exempel på en sådan.

LLOYDS ALGORITM

I Lloyds algoritm börjar man med en startuppsättning av rekonstruktionspunkter. Man beräknar sedan de optimala beslutsgränserna utifrån dessa, och tar fram den resulterande distortionen. Om denna överstiger en bestämd tröskel beräknar man nya optimala rekonstruktionspunkter baserat på sina nuvarande beslutsgränser och upprepar processen från början.

KVANTISERING MED KOMPANDER

Istället för att utforma själva kvantiseraren på ett komplext sätt kan man ta fram en kompressorfunktion och dess invers, expanderfunktionen. Kompressorfunktionen appliceras på signalen och resultatet kvantiseras likformigt. På mottagarsidan används expanderfunktionen för att återskapa signalen. Kvantisering med kompander används bland annat vid talkodning i telenätet.

KVANTISERING MED KÄLLKODNING

Vanligen är det olika sannolikhet att hamna i olika intervall vid kvantisering. Detta går att utnyttja genom källkodning. Det går att visa att den optimala kvantiseraren vid fin kvantisering är likformig, så om man vet att man ska källkoda den kvantiserade signalen är det ingen större idé att krångla till själva kvantiseringssteget.

VEKTORKVANTISERING

Ofta är korrelationen mellan närliggande sampel ganska betydande, vilket man utnyttjar i vektorkvantisering. Principen går ut på att man bildar vektorer av ett visst antal sampel, och kvantiserar dessa istället för varje sampel för sig. Rekonstruktionspunkterna är alltså vektorer i detta fall.

Fördelen med vektorkvantisering är att man utnyttjar källans minne redan i kvantiseringssteget, och behöver därför sällan applicera någon källkodning på den kvantiserade signalen. Dessutom visar det sig att distortionen vid en given datatakt alltid kommer att minska när man ökar antalet dimensioner – även för minnesfria källor. En nackdel är att metoden är både långsam och minneskrävande när vektorernas dimension är stor.

För att hitta optimala rekonstruktionsvektorer och beslutsgränser vid vektorkvantisering kan man använda Lloyds algoritm i flera dimensioner. I praktiken brukar man använda en variant som kallas *LBG-algoritmen* eller *K-means*. Denna kräver inte att man känner till fördelningsfunktionen.

LBG-ALGORITMEN (K-MEANS)

I LBG-algoritmen börjar man med en startkodbok och en uppsättning träningsvektorer. Man beräknar optimala beslutsområden, det vill säga man tilldelar varje rekonstruktionsvektor de träningsvektorer som ligger närmast. Sedan tar man fram den resulterande distortionen, och om den överstiger ett visst tröskelvärde så beräknar man nya rekonstruktionsvektorer som medelvärdet av träningsvektorerna i respektive område. Efter detta börjar man om från början.

PREDIKTIV KODNING

Prediktiv kodning är ännu en kodningsprincip som utnyttjar korrelationen mellan närliggande sampel i en signal. Signalens värde i en viss tidpunkt predikteras baserat på värden tidigare i signalen. Prediktionsfelet (skillnaden mellan det verkliga värdet och det predikterade) kvantiseras och skickas till avkodaren. Detta är den generella idén, men tyvärr fungerar den inte i verkligheten. Problemet är att avkodaren kommer att återskapa en distorderad version av prediktionsfelet och därför blir den resulterande signalen också distorderad. För att komma till rätta med det här låter man även prediktorn jobba med återskapde värden. På det sättet förekommer samma beräkningar på både kodarsidan och avkodarsidan.

LINJÄR PREDIKTION

En vanligt förekommande prediktionsmetod är den linjära, där det predikterade signalvärdet i en viss tidpunkt är en linjärkombination av tidigare sampelvärden. Koefficienterna i denna linjärkombination kan beräknas genom att man minimerar uttrycket för prediktionsfelets varians. Detta är önskvärt eftersom distortionen hos den resulterande signalen är direkt beroende av prediktionsfelets varians, vid fin kvantisering.

TRANSFORMKODNING

Idén bakom transformkodning är att uttrycka signalens värden i en ny bas, så att de dekorreleras så mycket som möjligt och så att det blir lättare att kvantisera dem skalärt med ett gott resultat. Arbetsgången är ganska rättfram: man tar in ett block av sampel, applicerar en (reversibel) transform på det och får en ny sekvens. Denna kvantiseras och kodas på något trevligt sätt.

Önskvärda egenskaper hos transformen är att den ska koncentrera signalenergin till så få komponenter som möjligt. Dessutom bör den dekorrelera transformkomponenterna, vilket motsvarar att man tar bort beroendet mellan dem. Helst ska den också vara okänslig för förändringar i källans statistik, och förstås vara enkel och snabb att räkna ut. Hittills har man inte lyckats ta fram någon transform som uppfyller alla dessa egenskaper.

KARHUNEN-LOEVE-TRANSFORM

Karhunen-Loeve-transformen (KLT) är den transform som ger maximal energikoncentrering och dekorrelerar transformkomponenterna mest. Basvektorerna i KLT är de normerade egenvektorerna till signalens korrelationsmatris. Följdaktligen är en nackdel med KLT att den är signalberoende och därför måste man skicka transformmatrisen som sidoinformation, och dessutom räkna ut en ny matris om källans statistik ändas.

DISKRET COSINUSTRANSFORM

Den diskreta cosinustransformen (DCT) liknar den diskreta fouriertransformen ganska mycket och kan på motsvarande sätt beräknas snabbt i en dator. DCT har nästan lika bra egenskaper som KLT för källor med hög korrelation mellan närliggande sampel. DCT används mycket inom bildkodning och finns i bland annat JPEG och MPEG.

DISKRET WALSH-HADAMARD-TRANSFORM

I den diskreta Walsh-Hadamardtransformen (DWHT) består transformmatrisen av en hadamardmatris, normerad med en faktor som beror av matrisens storlek. Detta innbeär att den förutom skalningsfaktorn bara innehåller 1 och -1, vilket gör den mycket enkel att beräkna. Resultatet av transformkodning med DWHT är dock inte jättebra – basfunktionernas utseende ger upphov till att eventuella kvantiseringsfel blir väldigt synliga eller hörbara.

BITTILLDELNING VID TRANSFORMKODNING

När man transformerat signalen kommer förhoppningsvis den mesta energin vara koncentrerad till några få transformkomponenter. Detta bör man ta hänsyn till när man kvantiserar transformkomponenterna, och man måste därför tänka efter litet när man bestämmer hur många bitar man ska kvantisera respektive komponent med. Generellt gäller att man vill hitta en bittilldelning som minimerar den resulterande distortionen. Detta går att göra på flera sätt. Ett ganska bekvämt sätt är att iterativt dela ut bitar till de komponenter som har högst distortion, och sedan räkna om distortionerna. Denna typ av metod kallas *zonkodning*. Zonkodning fungerar tyvärr inte särskilt bra om statistiken varierar mycket mellan de transformerade blocken, eftersom man bara gör en bittilldelning som sedan gäller för samtliga block.

TRÖSKELKODNING

Man kan förenkla kodningen av transformkomponenterna genom att sätta alla komponenter under ett visst tröskelvärde till 0. Detta brukar gälla för ganska många komponenter, vilket man kan utnyttja genom att skurlängdskoda den resulterande sekvensen. I två dimensioner får man fram en endimensionell sekvens för skurlängdskodning genom att gå igenom komponenterna i ett sick-sack-mönster (*zigzag-scanning*).

JPEG

JPEG är en ISO-standard för bildkodning. Den använder en DCT på block av storlek 8 x 8 bildpunkter åt gången och tillåter 8-12 bitar per färgkomponent. JPEG använder en kombination av likformig kvantisering och tröskelkodning, där steglängden för kvantiseringen kan väljas olika för olika transformkomponeneter – vanligen kvantiseras högfrekvenskomponenterna mycket hårdare än de med lågfrekvensinnehåll. Källkodningen är skurlängskodning av nollor följt av huffmankodning. Huffmankodträdet för DC-nivåerna (DC-nivån är den lägsta frekvenskomponenten) byggs baserat på skillnaden från DC-nivån i föregående block. Övriga komponenter ordnas i sick-sack-ordning, skurlängdskodas och huffmankodas.

DELBANDSKODNING

Delbandskodning fungerar så att man delar upp signalens frekvensinnehåll i olika band med hjälp av bandpassfilter. De nya signalerna kan nersamplas utan att man förlorar information, eftersom de har mindre bandbredd än originalsignalen (enligt Nyquist). Vid avkodningen körs processen baklänges, så att de olika banden samplas upp innan de sätts ihop igen.

De filter man använder kan antingen vara ett antal bandpassfilter som körs på signalen, eller så har man bara två (ett lågpass- och ett högpassfilter) som man applicerar rekursivt.

Filtrering med en *flat filterbank* resulterar i en likformig uppdelning av frekvensaxeln. Man kan även använda sig av så kallad *dyadisk* uppdelning, där man bara fortsätter att dela upp den ena grenen som bildats i varje filtrering (typiskt väljer man att rekursivt dela upp lågfrekvenssignalen).

Tvådimensionella signaler (bilder) filtreras horisontellt och sedan vertikalt med filter utformade för detta. Normalt fortsätter man att dela upp endast den del som innehåller lågpassinformation i båda ledderna.

Kvantisering och källkodning kan gå till på i princip samma sätt som vid blockbaserade transformer. Högfrekvenskomponenterna kvantiseras i regel hårt i jämförelse med lågfrekvensinnehållet. Bittilldelning fungerar precis som vid zonkodning.

JPEG 2000

JPEG 2000 är en ISO-standard för kodning av stillbilder som stöder upp till 38 bitar per färgkomponent. En dyadisk delbandstransform delar upp bildens frekvensinnehåll i 0 till 32 steg. Den transformerade bilden delas in i mindre block som kvantiseras och källkodas. Man använder likformig kvantisering följt av aritmetisk kodning, där varje bitplan kodas för sig. Koefficienterna kodas med hänsyn taget till omgivande koefficienter, men ett eventuellt beroende mellan olika delband utnyttjas inte. JPEG 2000 producerar en progressiv bitström, så man kan få fram hela bilden nästan direkt, men med låg kvalitet. Standarden tillåter distortionsfri kodning, men denna är inte lika bra som till exempel JPEG-LS.

PSYKOAKUSTIK

Man har studerat människans hörsel förhållandevis ingående, och det har visat sig att det finns ett antal egenskaper som kan utnyttjas vid kodning av ljudsignaler. Huvudproblemet är att placera kvantiseringsbruset så att det hörs så litet som möjligt, även om mängden brus totalt sett är densamma.

HÖRNIVÅ (LOUDNESS)

Hörnivån (*loudness* på engelska) för ett ljud definieras som nivån på en ton vid 1 kHz som uppfattas som lika starkt som ljudet. Hörnivån är beroende av intensitet, frekvensinnehåll och längd. Ljud med en hörnivå på under 0 dB kan vi inte uppfatta – gränsen kallas för *hörseltröskeln*. Denna kan förstås utnyttjas i ljudkodningssammanhang genom att alla frekvenskomponenter som ligger under den kan tas bort helt utan att lyssnaren märker någon skillnad.

MASKERING

Maskering är ett fenomen som innebär att starka ljud ”dränker ut” svaga ljud, så att bara det starka ljudet hörs. Detta medför att de svaga ljuden kan tas bort helt eller kvantiseras hårdare. Maskning uppträder både i tid och frekvens. Maskningen varierar med ljudets nivå och frekvens, och i tidsplanet uppträder maskningseffekter både före och efter det maskerande ljudet (fast maskningen efter har längre varaktighet).

MODIFIERAD DISKRET COSINUSTRANSFORM (MDCT)

Den modifierade diskreta cosinustransformen (MDCT) är mycket populär inom ljudkodning. Skillnaden mot vanliga DCT är att MDCT opererar på överlappande block av sampel, vilket gör att man bättre kan utnyttja någon psykoakustisk modell vid kodningen.

MPEG-1 AUDIO

MPEG-1 audio är ljudkodningsdelen i videokodningsstandarden MPEG-1. Den stöder samplingsfrekvenser på 32, 44.1 och 48 kHz och en eller två ljudkanaler (mono, dubbla monokanaler eller stereo, där stereo kan vara antingen vanlig stereo eller *joint stereo*). Datatakten kan ligga på mellan 32 och 224 kbit/s per kanal. Grunden i standarden är en delbandskodare med 32 stycken lika breda frekvensband.

Det finns tre nivåer (*layers*) av komprimering i MPEG-1 audio-standarden.

LAYER I

Denna nivå är den enklaste typen och fungerar bra för datatakter över 128 kbit/s per kanal. Signalen kodas i *frames* om 384 sampel och en psykoakustisk modell används för att fördela bitar till de olika delbanden. Likformig kvantisering används.

LAYER II

Mer komplex än layer I och låter bra för datataker omkring 128 kbit/s per kanal. Signalen kodas i frames om 1152 sampel och en psykoakustisk modell används för att fördela bitar till de olika delbanden. Även här används likformig kvantisering.

LAYER III (MP3)

Layer III har störst komplexitet av de tre nivåerna och ger bäst kompression. Layer III (som oftast kallas mp3) ger en okej kvalitet vid 64 kbit/s per kanal, ungefär, och riktigt bra kvalitet vid 192 kbit/s per kanal. Liksom layer II kodas signalen i frames om 1152 sampel, men här görs en MDCT inom delbanden för att förfina frekvensuppdelningen ytterligare. En olikformig kvantisering används, till skillnad från layer I och II. Efter kvantiseringen appliceras en huffmankod (med fixa kodord). Standarden tillåter den momentana datatakten att variera mellan blocken – det är möjligt att låta omgivande block få bitar från ett block som inte behöver alla bitar för att uppnå önskad ljudkvalitet.

DOLBY DIGITAL

Dolby digital är en standard för ljudkodning som stöder samplingsfrekvenser på 32, 44.1 och 48 kHz och upp till 5+1 ljudkanaler. Datatakten ligger mellan 32 och 640 kbit/s per kanal. Ljudet kodas i frames om 1536 sampel och en MDCT används. Det är möjligt att utnyttja beroendet mellan ljudkanalerna i Dolby digital.

AAC

AAC står för *Advanced Audio Coding* och är en ljudkodningsmetod som används i MPEG-2 och MPEG-4. Den använder en MDCT i kombination med en olikformig kvantisering (kompander). En slags fix huffmankod appliceras på de kvantiserade komponenterna. Den version av AAC som används i MPEG-4 kan använda aritmetisk kodning och vektorkvantisering.

VORBIS

Vorbis är en open-source-metod för ljudkodning som alltså inte innehåller några patenterade delar och därmed får användas fritt av vem som helst. Den utnyttjar en MDCT, och det är *envelopen* till transformdata som beräknas och skickas till mottagaren som filterkoefficienter. *Residualen* (transformdata dividerat med envelopen) vektorkvantiseras och huffmankodas. Det fina är att de flesta värdena i residualen blir 0 eller ±1.

SPECTRAL BAND REPLICATION (SBR)

Spectral band replication är en metod där man tar bort högfrekvensinnehållet ur ljudsignalen innan själva kodningen. Vid avkodningen återskapas högfrekvensinnehållet baserat på lågfrekvensinnehållet. Extra information som underlättar vid återskapningen skickas med som sidoinformation från kodarsidan. SBR minskar datatakten kraftigt utan att försämra den upplevda kvaliteten särskilt mycket.

SBR kan användas tillsammans med nästan vilken kodningsmetod som helst. Det finns implementerat tillsammans med mp3 och kallas då mp3PRO och tillsammans med AAC heter det AACPlus.

VIDEOKODNING

Vid videokodning lagras färginformationen med en luminans- och två krominanssignaler. Krominanssignalerna kan samplas grövre än luminanssignalen utan att kvaliteten blir för dålig, och detta utnyttjas ofta i videokodning.

En videoström kan betraktas som en signal i antingen två eller tre dimensioner. Det finns dock praktiska problem med att koda video som en 3D-signal, eftersom statistiken ofta är helt annorlunda i tidsplanet jämfört med i bildplanet och det krävs stora buffertar vid såväl kodning som avkodning. Fördröjningar är ytterligare en nackdel med 3D-tänkandet.

Betydligt vanligare är att man ser videoströmmen som en sekvens av tvådimensionella bilder. Man kodar varje bild med någon känd metod – tilll exempel transformkodning – och kan dessutom utnyttja tidsberoendet mellan närliggande bilder i sekvensen. Det sistnämnda kallas *hybridkodning*.

HYBRIDKODNING

De flesta moderna videokodningsmetoder använder hybridkodning, vilket innebär att man gör prediktiv kodning i tidsled och transformkodning i bildplanet. För att kompensera för kamerarörelser, zoomingar och rörliga objekt i bilden använder man *rörelsekompensering*, vilket innebär att man för varje block i bilden letar efter ett liknande block i föregående bild inom ett begränsat område. Translationen mellan dessa skickas som en *rörelsevektor* till mottagaren. Prediktionen utnyttjar blockens rörelse för att skapa en skillnadsbild, som kodas och skickas. För att undvika att eventuella fel fortplantar sig bör man med jämna mellanrum skicka bilder som kodas helt oberoende av andra bilder.

DIGITAL VIDEO (DV)

Digital video, eller *DV*, är en konsumentstandard för just komprimering av digital video. Block om 8 x 8 bildpunkter kodas med DCT, och det exakta tillvägagångssättet kan varieras beroende på om det är en videoström med mycket rörelser eller om det är något mer stillsamt som kodas. Transformkomponeneterna kvantiseras likformigt och resultatet zigzag-scannas och skurlängdskodas. De resulterande taltuplarna kodas med en förbestämd variabellängdskod. Datatakten är 25 Mbit/s vilket motsvarar ungefär fem gångers kompression. Ljudet lagras okomprimerat med antingen två eller fyra kanaler (båda med den totala datatakten på 1.5 Mbit/s).

MPEG-1

MPEG-1 är en standard för videokodning som motsvarar VHS-kvalitet (datatakten ligger på 1.5 Mbit/s). Rörelsekompensering och DCT används. I MPEG-1 har man infört en prediktion som är beroende av bilder både före och efter aktuell bild, vilket innebär att man måste skicka bilderna i en annan ordning än de visas.

Kvantiseringen går till på ett liknande sätt som i JPEG, men här har luminans- och krominansblock skilda kvantiseringsmatriser. De kvantiserade komponenterna zigzag-scannas och eventuella nollor skurlängdskodas. De resulterande taltuplarna kodas med en fix variabellängdskod.

MPEG-1 används bland annat i VideoCD (VCD).

MPEG-2 (H.262)

MPEG-2 (kallas även H.262) är väldigt lik MPEG-1, men tillåter högre upplösningar och datatakter. Används i SuperVideoCD (SVCD), DVD, digital-TV och HDTV.

MPEG-4

MPEG-4 är en standard för multimediakodning. Man tänker sig en scen med flera möjliga bild- och ljudkällor, där varje källa kodas med en lämplig metod. Avkodaren ska sätta samman alla källorna till motsvarande scen.

Videokodningen liknar övriga MPEG-standarder med en DCT-kodare och rörelseestimering. En skillnad är dock att videon kan ha vilken form som helst – inte bara rektangulär. Detta innebär att information om formen måste skickas som sidoinformation.

Stillbilder kodas med en delbandskodare. En *sprite* är i MPEG-4 en särskild form av stillbild som fungerar som bakgrund i en hel videosekvens. Genom att använda sprites kan man skicka över bakgrunden en enda gång, istället för att göra det i varje bildruta.

Syntetiska objekt stöds av standarden. Ett exempel på tillämpning är att ett mänskligt ansikte kan beskrivas som en enklare trådmodell med tillhörande textur. Trådmodellens rörelse kodas effektivt och texturen skickas bara en gång.

MPEG-4 innehåller flera olika typer av ljudkodningar som man kan välja mellan beroende på vilken typ av signal det är som ska kodas. AAC används för generella vågformer, men det finns även talkodning, text-to-speech-kodning (att syntetisera tal från text) och en metod som syntetiserar musik från en beskrivning av instrument och toner (ungefär som MIDI).