

14. Kuvien pakkausmenetelmiä

Häviötön/häviöllinen pakkaus

Binäärikuvien pakkaus

Harmaasävykuvien pakkaus

Värikuvien pakkaus

Pakkauksen taustaa

- Välttämätön suurille kuva-arkistoille
 - Säästää tilaa
 - Nopeuttaa datan siirtoa
- Informaatioteoriaa
 - Jos symbolin todennäköisyys on p , niin sen sisältämän informaation määrä = $\log_2(1/p)$ bittiä
 - *Entropia* = teoreettinen minimi keskimääräiselle bittimäärälle per symboli (S = 'aakkosto'):

$$\sum_{i \in S} p_i \log_2(1/p_i)$$

olettaen, että *symbolit ovat riippumattomia*.

Menetelmien piirteitä

- *Häviöttömyys*: alkuperäinen kuva voidaan palauttaa tarkasti (→ röntgenkuvat)
- *Häviöllisyys*: voidaan palauttaa alkuperäisen kuvan approksimaatio
- Pakkausteho (pakattu/alkuperäinen)
- Nopeus: Käytännön kannalta tärkeä
 - Koodausnopeus (→ tiedonsiirto)
 - Dekoodausnopeus (→ kuva-arkistot)
- Herkkyys tiedonsiirtovirheille
- *Progressiivinen* siirtomahdollisuus

Kuvien redundanssista

Redundanssi: käytetään enemmän bittejä datan esittämiseen kuin on tarpeen. Tyyppejä:

(1) Koodausredundanssi:

- Kaikki koodit eivät käytössä.
- Koodien esiintymistiheys vaihtelee.

(2) Pikselien välinen redundanssi:

- Naapuripikselien sävyjen välillä vahva korrelaatio.
- Pikselin sävy voidaan 'ennustaa' naapurista.

(3) 'Psykovisuaalinen' redundanssi:

- Koodaustarkkuus ylittää ihmisen havaintotarkkuuden.

Pakkauksen päävaiheet

1. Mallintaminen

- Kuvataan sävyjen jakautuma ja riippuvuus naapureista jollakin tavalla
- Muodostetaan koodattavia elementtejä (symboleja)

2. Koodaus

- Ns. 'tilastollinen' koodaus eli 'entropiakoodaus'
- Annetaan elementeille binäärikoodit
- Koodien oltava yksikäsitteisesti dekodattavissa
- Perustuu elementtien todennäköisyyksiin
- Tunnettuja algoritmeja:
 - *Huffman-koodaus*: Ei-binääriset aakkostot
 - *Aritmeettinen koodaus*: Mielivaltaiset aakkostot

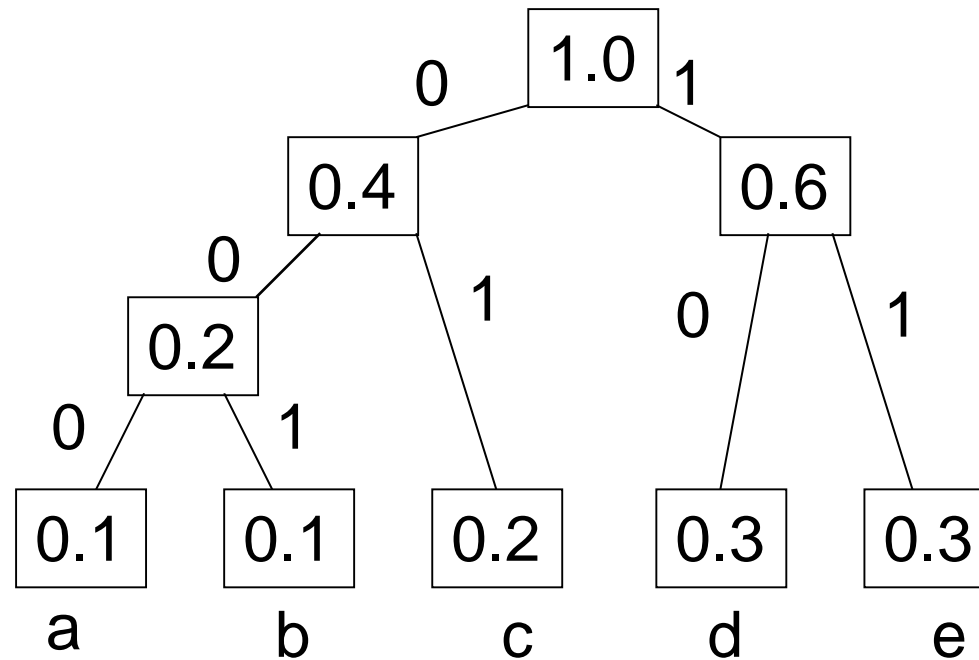
Esimerkkejä mallintamistekniikoista

- Delta-koodaus:
 - Koodattava symboli = pikselin ero edellisestä (jo koodatusta) pikseliarvosta
 - Normaalille harmaasävykuvalle delta saa arvoja väliltä $-255 \dots +255$, mutta *jakautuma on hyvin vino*, mikä tuottaa tehokkaamman koodauksen.
- Välimatkakoodaus (run-length encoding)
 - Kannattaa, jos sama sävyarvo toistuu peräkkäin.
 - Koodattava symboli = (pikseli & toiston määrä); mustavalkokuvalle riittää koodata aloitusväri, ja sen jälkeen vain toistojen määrät (vuorotellen mustia ja valkoisia pikselijonoja).

Huffman-koodauksen periaatteet

- Usein esiintyville symboleille annetaan lyhyt koodi, harvoin esiintyville pitkä. Menetelmä on optimaalinen, jos jokaiselle symbolille halutaan oma koodi.
- Koodit johdetaan rakentamalla Huffman-puu:
 - Määrätään eri symbolien (n kpl) todennäköisyydet
 - Sijoitetaan symbolit Huffman-puun lehdiksi
 - Summataan kaksi pienintä todennäköisyyttä ja luodaan parille isäsolmu. Toistetaan $n-1$ kertaa.
 - Koodaus: vasen poika = 0, oikea poika = 1
 - Symbolin koodi = bittipolku juuresta ao. lehteen.

Esimerkki Huffman-koodipuusta



Koodit: 'a' = 000, 'b' = 001, 'c' = 01, 'd' = 10, 'e' = 11

Esim. "ebecddadec" → "1100111011010000101101"

Binäärikuvien pakkaus

- Yleensä häviötön
- Valkoisia pikseleitä (= taustaväri, edustaa esim. valkoista paperia) yleensä selvästi enemmän
- Samanväriset pikselit ovat vahvasti *ryvästyneet*
- Esimerkkisovellus: telefax
- Menetelmätyyppejä:
 - Välimatkakoodaus (run-length coding)
 - Hierarkkinen lohkokoodaus
 - Ennustava koodaus

Ennustava koodaus

- Sopii erityisesti binäärikuville, mutta käytetään myös muihin (esim. lossless JPEG)
- Pikselin ennustus perustuu jo koodattujen naapurien (ylhäällä/vas.) sävyarvoihin, esim.

NW	N	NE
W	x	

- Naapuriyhdelmistä kerätään tilastoa, minkä perusteella lasketaan todennäköisyydet seuraavan pikselin sävyille. Sen jälkeen esim. aritmeettinen koodaus (Huffman ei sovi binääriaakkostolle).

Sanastopohjainen kuvanpakkaus

- Ideana koodata usein toistuvia pikseliryhmiä.
- Sovellusalueena erityisesti väripalettikuvat (eli indeksoidut värit): pikseliarvo = paletin indeksi; Mutta: Vierekkäiset väri-indeksiarvot eivät välttämättä edusta läheisiä värisävyjä.
- Kuvaa voidaan pitää 2-ulotteisena *merkkijonona*, merkkeinä indeksit väripalettiin.
- 'Sanasto' koostuu sopivista osajonoista.
- Merkkijonojen pakkausmenetelmiä:
 - LZW (GIF-kuvaformaatti käyttää)
 - LZ77 (PNG-kuvaformaatti käyttää)
- Sopii grafiikan pakkaukseen; huonommin todellisiin valokuviin.

Sanastopohjainen koodaus Javassa

- Pakkaus *java.util.zip*
 - Kooderi: *Deflater*-luokka
 - Dekooderi: *Inflater*-luokka
- Voidaan käyttää oletusarvoisia (de)kooderi-olioita tai luoda ne ja säätää parametrit itse.
- Pakkaus/purku toimii eräänlaisena suodattimena syöttö/tulostus-operaatioissa:
 - Koodatun lukeminen: *InflaterInputStream*
 - Kirjoitus koodaten: *DeflaterOutputStream*

Häviöllinen pakkaus

- Perustuu psykovisuaaliseen redundanssiin:
 - Tasaisilla alueilla ei saa olla suuria virheitä, mutta tasainen alue pakkautuu muutenkin tehokkaasti.
 - Jos kuvassa on tiheitä vaihteluita, ihmissilmä ei ole niin herkkä sävyjen tarkkuudelle, jolloin voidaan 'fuskata' enemmän.
- Tässäkin perustana vahva riippuvuus naapurisävyistä
- Kohina hankaloittaa pakkausta
- Tyypillinen pakkaussuhde: 1:10 – 1:20
(vrt. tyypillinen häviötön suhde: 1:2 – 1:3)

Häviöllisiä pakkausmenetelmiä harmaasävykuville

- **Kvantisointi:**
 - Esimerkiksi sävyjen approksimointi pienemmällä bittimäärällä
 - Sovelletaan harvoin yksinään, yleensä osa jotain muuta menetelmää
- **Ennustus:**
 - Esim. naapurien perusteella; kuten vastaava häviötön menetelmä, mutta ennustusvirhe kvantisoidaan ennen koodausta.
 - Kulkee nimellä DPCM = *Differential Pulse Code Modulation*

Häviöllisiä pakkausmenetelmiä harmaasävykuville (jatk.)

- **Otanta** (subsampling):
 - Koodataan vain osa pikseleistä.
 - Dekoodauksessa puuttuvat pikselit interpoloidaan.
 - Yleensä osa muita menetelmiä.
- **Pyramiditekniikka:**
 - Toistetaan otantaa hierarkkisesti
 - Lasketaan ja koodataan virhekuvat (= erot alkuperäisen ja otoksesta interpoloidun kuvan välillä)
 - Mahdollistaa kuvien *progressiivisen siirron*: Ensin karkea versio nopeasti, sitten laadukkaampi.

Häviöllisiä pakkausmenetelmiä harmaasävykuville (jatk.)

- **Vektorikvantisointi**

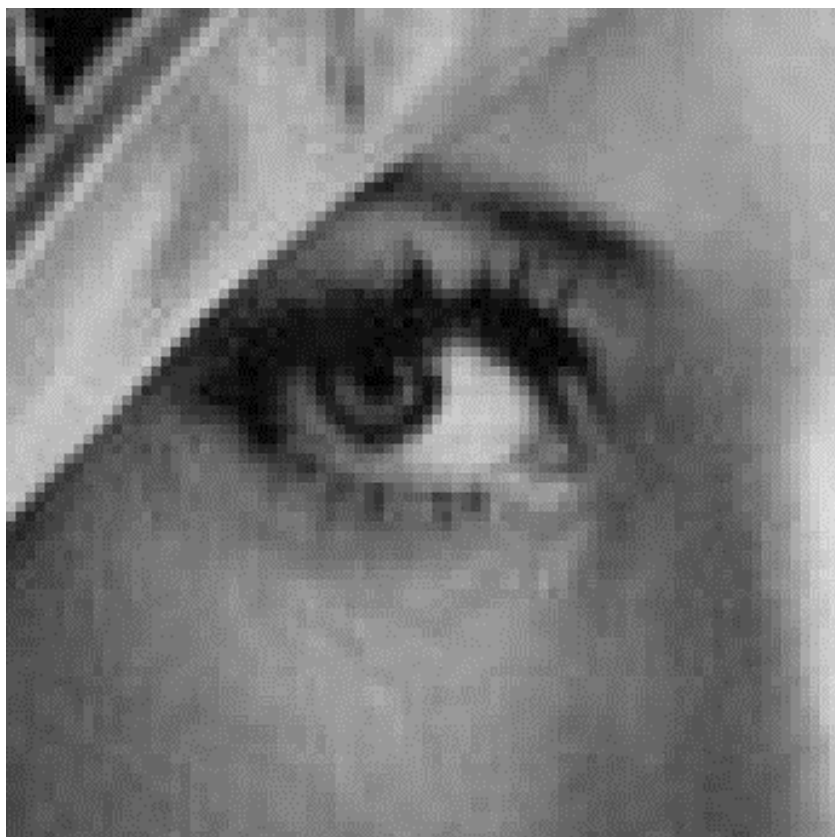
- Perustuu ns. *koodikirjaan*, jossa joukko pieniä lohkoja (esim. 4x4); vrt. väripaletti
- Etsitään kullekin kuvan lohkolle lähin vastine koodikirjasta
- Koodi = vastineen indeksi
- Ongelmana edustavan koodikirjan muodostaminen (yleinen / johonkin kuvaan sidottu)
- Koodikirjan tulisi olla riittävän edustava, jotta jokaiselle esiintyvälle lohkolle löytyy läheinen koodikirjan lohko; useita heuristiikkoja olemassa.

Häviöllisiä pakkausmenetelmiä harmaasävykuville (jatk.)

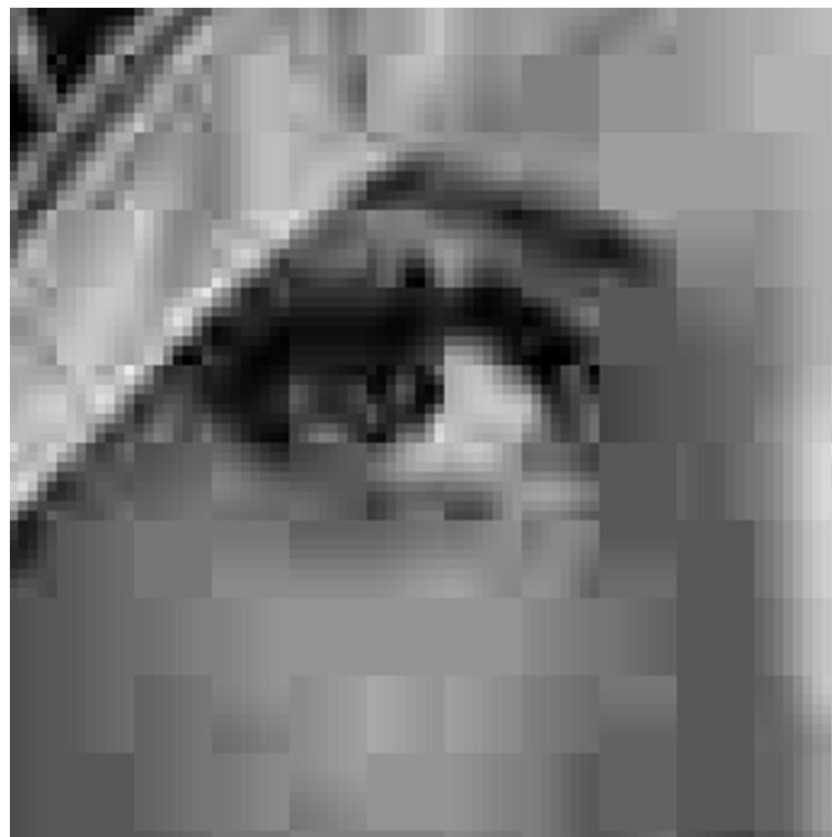
- **Muunnospohjainen pakkaus: JPEG**
 - Muunnos frekvenssialueelle: Diskreetti kosinimuunnos (DCT) → ei kompleksilukuja
 - Erikseen 8 x 8 –lohkoille → kerroinmatriisit
 - Matalat taajuudet matriisissa vas. ylhäällä, korkeat taajuudet oikealla alhaalla
 - Taajuuskertoimet kvantisoidaan ja koodataan; matalat taajuudet tarkemmin
 - Pakkauksen perusta: suurin osa kertoimista nolliä
 - Kertoimet koodataan frekvenssijärjestyksessä (ns. sik-sak-järjestys), käyttäen yhdistettyä välimatka- & Huffman-koodausta.

JPEG: Pakkaushäviöiden havainnollistus

Lievä pakkaus (1:7)



Voimakas pakkaus (1:30)



Häviöllisiä pakkausmenetelmiä harmaasävykuville (jatk.)

- **Wavelet-pakkaus:**

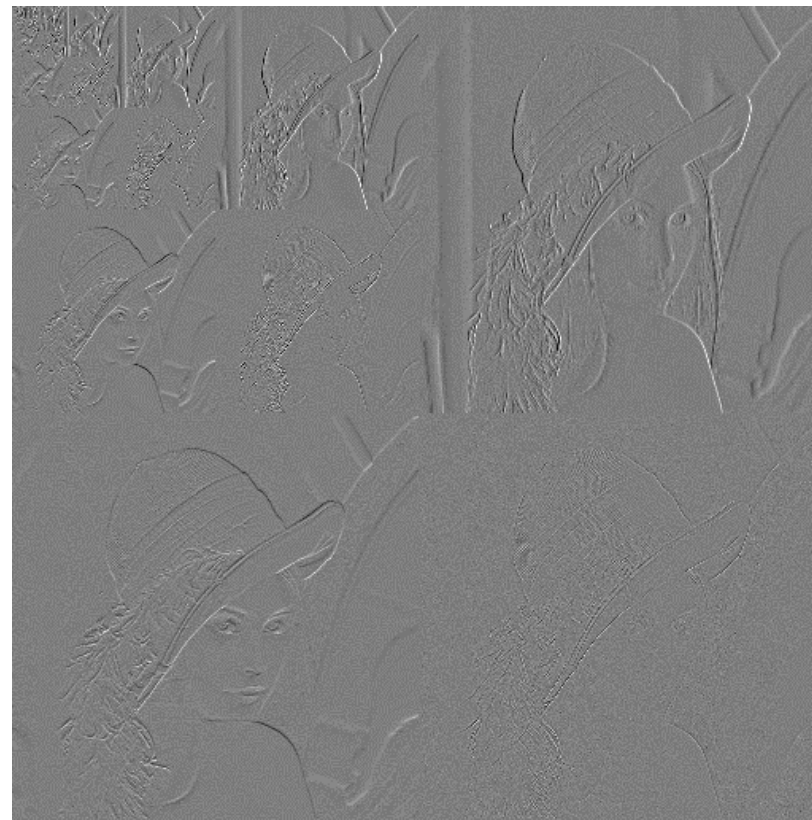
- Kuva hajoitetaan eritaajuisiin komponentteihin, jotka ovat myös spatiaalisesti lokalisoituja (toisin kuin DCT).
- Wavelet-muunnos erikseen vaaka- ja pystysuunnissa
- Rekursio matalille taajuuksille: Muistuttaa pyramidikoodausta
- Erilaisia kantafunktioperheitä ('mother wavelets', esim. Haar, Daubechies)
- Toteutus suodattimilla
- JPEG 2000 –menetelmän perusta; selvästi parempi kuin 'vanha' JPEG

Wavelet-muunnoksen havainnollistus

1. muunnosvaiheen tulos



Täyden rekursion tulos



Häviöllisiä pakkausmenetelmiä harmaasävykuville (jatk.)

- **Fraktaalipakkaus:**
 - Etsii kuvasta itseään toistavia (self-similar) osia
 - Koodaustulos = fraktaalien generointifunktiot (parametrit)
 - Pakkaussuhde erittäin hyvä
 - Syntyvä virhe usein melkoinen
 - Erittäin hidas pakkaus, mutta suhteellisen nopea purku.

Värikuvien pakkauksesta

- Suoraviivainen yleistys harmaasävykuvien pakkaukselle
- Ensimmäinen idea:
 - Kullekin RGB-värikanavalle erikseen
 - Ei tehokas, koska värit korreloivat
- Parempi idea:
 - Erotetaan kirkkaus (luminanssi) väri-infosta
 - JPEG: muunnos YUV-järjestelmään (tai $Y C_B C_R$); krominanssi (U, V) voidaan koodata epätarkemmin kuin kirkkaus (Y).