INF2310 - Stikkord over resten av pensum

Kristine Baluka Hein (krisbhei@ifi.uio.no)

9

2D Diskret Fourier transform (DFT) Bytte representasjon av et bilde.
Går fra fra romlige koordinater til frekvenspar: $f(x,y) \overset{\mathcal{F}}{\mapsto} F(u,v)$.

$$F(u,v) = \sum_{\text{over alle (x,y)}} (\text{cos-bilde med frekvens u,v}). * f + j \sum_{\text{over alle (x,y)}} (\text{sin-bilde med frekvens u,v}). * f$$

der $.*$ representerer punktvis multplikasjon.

Nyttige egenskaper - DFT Matematiske egenskaper (kan utledes direkte ved definisjon)

- Periodisitet
- Element i posisjon $(0,0)$, DC, har verdi lik summen av alle gråtoneverdiene
- **Konvolusjonsteoremet:** Punktvis mupltiplikasjon i frekvensdomenet tilsvarer konvolusjon i bildedomenet. Tilsvarende vil konvolusjon i frekvensdomenet tilsvare punktvis multplikasjon i bildedomenet.
- Symmetri om aksene som deler spekteret i fire like store deler (først speling om horisontal/vertikal akse, så speiling om vertikal/horisontal akse, mao kompleks konjugert).

Hvordan bildets struktur påvirker respons i frekvensdomenet
- Bred struktur gir smal respons i frekvensdomenet
frekvenser 'forsvinner' under den brede strukturen

- Smal struktur gir bred respons i frekvensdomenet
flere frekvenser 'treffer' på den smale strukturen

Design av filter i frekvensdomenet
Filter med reelle koeffsienter -> konjugert symmetri. Verdier gjerne mellom 0 (demper/'fjerner' en frekvens) og 1 (bevarer frekvens).

Filtre Ideell, Butterworth, Gaussisk, Notch

Vindu Fjerne effekter som følge av periodisiteten.

10

Redundans Data som kan fjernes uten å miste viktig informasjon. Fire ulike typer: Psykovisuell, interbilde, intersampel og kodings redundans.

Kompresjonsrate
$C = \dfrac{\text{Ukomprimert størrelse}}{\text{Komprimert størrelse}}$ eller $C = \dfrac{\text{Gjennomsnittlig antall bits per symbol, ukomprimert}}{\text{Gjennomsnittlig antall bits per symbol, komprimert}}$

Entropi $H = -\sum_i p(i) \log_2(p(i))$

Shannon-Fano
Generering av kode for symbolene:

1) Sortér frekvensene i synkende rekkefølge.

2) Del opp de sorterte frekvensene til hvert av symbolene i to like store grupper (hvis odde antall frekveknser -> velg skille s.a gruppe blir så like sannsynlige som mulig). Legg den ene gruppen til venstre i et binærtre, og den andre til høyre.

   Oppstår det en gruppe bestående av én frekvens, så blir symbolet tilhørende frekvensen en løvnode i binærtreet.

3) Fortsett 2) til alle symboler er plassert i treet.

Huffman
Generering av kode for symbolene:

1) Sortér frekvensene i stigende rekkefølge.

2) Dann en gruppe (node) av de to minste frekvensene . Frekvensen til denne gruppen er summen av de to minste frekvensene. La den nye gruppen bli en rotnode der de to minste frekvensene er barnnoder. 0 til venstre gren, 1 til høyre gren.

   Legg den nye gruppen i den sorterte rekken av frekvenser, og sortér på nytt.

3) Fortsett 2) til alle symboler er plassert i treet.

Optimal i den forstand at gir minste gjennomsnittlige kodelengde.
Hvis alle symbolssannsynlighetene er toerpotenser, så er den gjennomsnittlige kodelengden lik entropien.

Aritmetisk koding
Koder en sekvens av symboler istedenfor pr. symbol.
Dele opp 'current-intervall' rekursivt i like store deler som det kumulative histogrammet til symbolene deler opp intervallet mellom 0 og 1.
Når funnet intervallet, så finn et tall i intervallet med kortest bitseksvens.
Dekoding utføres ved å omgjøre gitt bitsekvens til desimaltall. Gjør samme rekursive oppdeling av intervall helt til treffer en del av intervallet der spesifisert sluttsymbol befinner seg . Lagrer symbolene som tilhører hver del av intervallene underveis.

Differansetransform utnytter at horisontale nabopiksler har ofte nesten lik intensitet.
For hver rad i bildet: start på andre piksel i raden, og ta differansen mellom den og pikslen som ligger før, altså første. Fortsett videre på tredje piksel,finn differansen mellom den og andre piksel. Fortsett slik på alle piksler i raden.

Løpelengdetransform
For hver rad: lag 2-tupler der første element er hvilken verdi som sees på, og andre element er hvor mange ganger verdien forkommer i raden helt til det oppstår en ny verdi.

Lempel - Ziv - Welch Lager kodebok underveis. Initialiserer med kode til symbolene i alfabetet, så utvider kodeboken med sekvenser av symbolene. Gunstig når samme sekvenser av symbolene forekommer ofte i meldingen.

For hvert symbol i meldingen:
Se etter lengste sekvens som er inneholdt i kodeboken fra og med symbolet som sees på. Send

tilhørende kode. Legg til i kodeboken: sekvensen der tilhørende kode nylig har blitt sendt + neste symbol. Hopp til symbolet som er etter den sendte sekvensen. Gjør det samme ved dette symbolet. Fortsett til hele meldingen har blitt sendt.

JPEG-koding
8x8 DCT,punktdividerer med kvantiseringsmatrise, komprimerer og koding av blokkene.
Invers: dekoder blokker, inverstransformerer og punktvis multpliserer blokkene, tar IDCT på hver av dem.

11

Dilasjon Utvider forgrunnselementer.
Fyller opp hull og legger til forgrunnspiksler (legger til: $\oplus$).

$$(f \oplus s)(x,y) = \begin{cases} 1, & \text{dersom 180 grader rotert } s \\ & \text{har noe overlapp med } f \text{ om punktet (x,y)} \\ 0, & \text{ellers} \end{cases}$$

Erosjon Forminsker forgrunnselementer og forstørrer hull.
Fjerner støy som er mindre enn strukturelementet s (trekker fra: $\ominus$).

$$(f \ominus s)(x,y) = \begin{cases} 1, & \text{dersom } s \text{ overlapper fullstendig } f \text{ om punktet (x,y)} \\ 0, & \text{ellers} \end{cases}$$

Finne kanter til forgrunnsobjekt
To måter:

- Dilatere bildet med et strukturelement, så trekke fra originalbildet
- Trekke fra originalbildet med erodert bilde med et strukturelement

Naboskap til strukturelement vil påvirke naboskapet til kantene.

Hente ut sammenhengende forgrunnskomponenter $c_k = (c_{k-1} \oplus f) \cap f, \qquad k = 1, 2, \ldots$
c_0 har samme størrelse som f og 1 på en koordinat der komponeneten befinner seg.
Forstørrer området og sjekker hvilke av de nyintroduserte pikselene er med i f. Fortsetter helt til $c_k = c_{k-1}$.

Hente ut hull $c_k = (c_{k-1} \oplus f) \cap f^c, \qquad k = 1, 2, \ldots$
Henter ut region som tilsvarer hull i en forgrunnskomponent. c_0 er nå 1 på en koordinat der hullet befinner seg.

Dualitet

- Dilasjon: $f \oplus s = (f^c \ominus s)^c$
- Erosjon: $f \ominus s = (f^c \oplus s)^c$

Morfologisk åpning $f \circ s = (f \ominus s) \oplus s$
Erosjon, så dilasjon. Erosjon for å fjerne støy, dilasjon for å få tilbake omtrentlige originale størrelser.

Morforlogisk lukking $f \bullet s = (f \oplus s) \ominus s$
Dilasjon, så erosjon. Dilasjon for å tette hull, erosjon for å få tilbake omtrentlige originale størrelser.

Hit-or-miss $f \circledast s = (f \ominus s_1) \cap (f^c \ominus s_2)$ Løser for hver av parantesene, så finner de pikslene som er felles. Brukes til å finne spesifikke mønstre, fjerne enkeltpiksler og tynne og utvide områder i forgrunnen

Morfologisk tynning $f \otimes S = f \cap (f \circledast S)^c$
der S er mengde av ulike strukturelementer. For hver iterasjon, utfør hit-or-miss for hvert strukturelement i S og oppdater bildet som jobbes på med resultatet. Fortsett iterasjonene til ingen av strukturelementene gir noen forandring.

Morfologisk lukking $f \odot S = f \cup (f \circledast S)$
Utføres på samme måte som for morfologisk tynning.