

Ukeoppgaver uke 17 INF2310, våren 2010.

Tema: Kompresjon I

1. Vi har gitt følgende bilde:

1	0	1	2	2	2	3	3	3
1	1	1	2	1	1	3	3	3
1	0	1	1	2	2	2	3	3
2	1	2	2	3	2	3	4	4
2	1	2	3	2	2	3	4	4
2	2	2	3	3	3	4	3	4

a) Finn Huffman-kodingen av dette bildet. Hvor mange biter blir det per piksel i gjennomsnitt etter koding – hvis vi ser bort fra at vi trenger plass til å lagre kodeboken?

b) Ved differansetransform tar vi differansen mellom et piksel og dets nabo til venstre. Siden pikslene lengst til venstre i bildet ikke har noen venstre nabo, beholder vi pikselverdien her. Finn differanse-transformen av bildet ovenfor.

c) Finn så Huffman-koden for det differansetransformerte bildet, slik at du kan beregne det gjennomsnittlige antall bits per piksel for det differansetransformerte bildet.

d) Entropien i det bildet vi startet med er 2.06. Hvorfor ble det gjennomsnittlige antall biter per piksel større enn entropien i deloppgave a), men mindre enn 2.06 i deloppgave c)?

e) Hvis det bildet du fikk oppgitt i starten av oppgaven var det andre bildet i en bildesekvens, og det første bildet var slik:

1	1	1	2	2	2	3	3	3
1	1	1	2	2	2	3	3	3
1	1	1	2	2	2	3	3	3
2	2	2	3	3	3	4	4	4
2	2	2	3	3	3	4	4	4
2	2	2	3	3	3	4	4	4

Hvilke to bilder ville du da komprimere, og hvor mange biter vil du i gjennomsnitt trenge per piksel for hvert av de to bildene?

2. Anta at det er $G=2^8$ forskjellige gråtone-nivåer i hvert sample, og at når vi sorterer dem etter hvor ofte de forekommer i et bilde, så finner vi i et spesielt tilfellet at sannsynlighetene er $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots, \frac{1}{128}, \frac{1}{256}, \frac{1}{256}$.

a. Hvor mange slike bilder kan vi overføre i parallell på en 64 kbits/s linje med Huffman-koding av amplitudene?

Hint 1: Entropien er gitt ved

$$H = -\sum_{i=0}^{G-1} p_i \log_2(p_i)$$

Dessuten: $\log(\text{teller/nevner}) = \log(\text{teller}) - \log(\text{nevner})$

Og til slutt: $\log_2(2^n) = n$

Hint 2: Summen $\frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{4}{16} + \frac{5}{32} + \dots$ konvergerer raskt mot 2.

3. Optimal Huffman-koding:

Finn kodeboken for en Huffman-koding av "DIGITAL OVERALT!".

Hvorfor kan vi uten å gjøre noen logaritme-beregninger si hva entropien til denne teksten er?

4. Implementasjon av enkel Huffman-koding

Skriv et matlab-program som implementerer Huffman-koding av et bilde basert på det normaliserte histogrammet til bildet.

Du trenger IKKE implementere selve lagringen av kodeboken med kodeord av ulik lengde. Det du skal implementere er kode for å finne kodeboken slik at du kan beregne gjennomsnittlig antall bit per piksel, samt entropi.

