AALBORG UNIVERSITET

**Det Teknisk-Naturvidenskabelige Basisår**
**Storgruppe 0233**
Strandvejen 12-14
Telefon 96 35 97 33
Fax 98 13 63 93
http://www.but.auc.dk

**Titel:**

Om fingeraftryk
– komprimering med wavelets

**Tema:**

Netværk og algoritmer

**Projektperiode:**
P2, forårssemesteret 2003

**Projektgruppe:**
0233, C224

**Deltagere:**
Arne Mejlholm
Bjarne M. Kondrup
Dan Malthesen
Dong Thanh Nguyen
Jesper Kristensen
Lars Hay Jensen

**Vejledere:**
Jon Johnsen
Claus Monrad Spliid

**Oplagstal:** 10

**Sidetal:** 99

**Bilagsantal og –art:** 1 CD-Rom

**Synopsis:**

Denne rapport er en opsummering af vores projekt, som omhandler fingeraftryk og komprimering. Der gives en forklaring på biometri, om historien bag biometrisk genkendelse og hvorledes biometrien anvendes idag. Rapporten beskriver den juridiske situation og de etiske problemstillinger der er omkring anvendelse af biometri. Udfra problemstillingen, komprimering af et digitaliseret fingeraftryk, beskriver rapporten tre teoretiske metoder til komprimeringen. De brugte metoder er wavelet transformation, spline interpolation og Huffman kodning. Rapporten beskriver hvordan vi har opbygget et C-program, ved hjælp af wavelets og Huffman kodning. Programmet er udarbejdet til komprimering og dekomprimering af et TIFF billede, og er udbygget med en test af komprimeringen.

# Indhold


**1 Forord** **5**

**2 Indledning** **6**
2.1 Problemanalyse . . . . . 6
2.1.1 Identifikationsmetoder . . . . . 6
2.1.2 Forbedring af identifikationsmetoder . . . . . 7
2.2 Problemformulering . . . . . 8
2.3 Problemafgrænsning . . . . . 9
2.4 Problemløsningsmetoder . . . . . 9

**3 Biometri og Fingeraftryk** **12**
3.1 Biometri . . . . . 12
3.1.1 Anvedelse af biometri . . . . . 13
3.2 Fingeraftryk . . . . . 15
3.2.1 Fingeraftrykket historik . . . . . 15
3.2.2 Fingeraftryk i Danmark . . . . . 17
3.2.3 Fingeraftryksscannere . . . . . 18
3.3 Lov vedrørende registre . . . . . 19
3.3.1 Behandling af personoplysninger . . . . . 20
3.3.2 Opbevaring af personoplysninger . . . . . 20
3.3.3 Opsummering . . . . . 21

**4 Teknologikritik** **22**
4.1 Etik . . . . . 22
4.1.1 Hvad er etik? . . . . . 22
4.1.2 Forskellige teknologietikker . . . . . 23
4.2 Spørgeskema - undersøgelse . . . . . 24
4.2.1 Undersøgelsens formål . . . . . 25
4.2.2 Målgruppe . . . . . 25
4.2.3 Hypoteseopstilling . . . . . 26
4.2.4 Omformning af hypoteser til spørgsmål . . . . . 26
4.3 Resultater af undersøgelsen . . . . . 27
4.3.1 Vurdering af målgrupper . . . . . 28

4.3.2 Databehandling . . . . . 28
4.4 Selve teknologikritikken . . . . . 32
4.4.1 Kriterier for et biometrisk system . . . . . 33
4.4.2 Konflikter mellem individets og statens interesser . . . 34
4.4.3 Fordele . . . . . 36
4.4.4 Ulemper . . . . . 37
4.5 Opsumering . . . . . 37

**5 Forskellige billedformater** **39**
5.1 Billedformater generelt . . . . . 39
5.2 JPEG . . . . . 39
5.3 Raster image . . . . . 40
5.4 TIFF . . . . . 40
5.4.1 Gråskala billede i TIFF . . . . . 41
5.5 FPD . . . . . 43
5.6 Opsummering . . . . . 44

**6 Komprimeringsteknikker** **45**
6.1 Grundlæggende Komprimering . . . . . 45
6.2 Huffman kodning . . . . . 47
6.2.1 Princippet bag Huffman kodning . . . . . 47
6.2.2 Huffman kodning vist ved eksempel . . . . . 48
6.2.3 Træstrukturer . . . . . 48
6.2.4 Dekomprimering . . . . . 50
6.3 Interpolation og Splines . . . . . 51
6.3.1 Interpolation . . . . . 51
6.3.2 Spline interpolation . . . . . 52
6.3.3 Splines og Fingeraftryk . . . . . 57
6.4 Wavelets . . . . . 59
6.4.1 Historien bag ved billedkomprimering . . . . . 59
6.5 Diskret wavelet transformation . . . . . 60
6.5.1 Et eksempel på wavelet transformation . . . . . 61
6.6 Transformation ved hjælp af forudsigelse . . . . . 63
6.6.1 Lifting . . . . . 63
6.6.2 Dekomprimering . . . . . 65
6.6.3 Fortolkning af signalet . . . . . 66
6.6.4 Komprimering af billeder med wavelets . . . . . 68
6.7 Opsummering . . . . . 73

**7 Udvikling af programmet** **74**
7.1 Program specifikation . . . . . 74
7.2 Programdiagram . . . . . 75
7.3 De enkelte moduler og deres funktioner. . . . . 75
7.3.1 GUI (Graphic User Interface) . . . . . 75

7.3.2 Tiff2array . . . . . 76
7.3.3 Analyse - wavelets . . . . . 76
7.3.4 syntese - wavelets . . . . . 79
7.3.5 IO - wavelets . . . . . 80
7.3.6 Gem/indlæs billede (Huffman) . . . . . 81
7.3.7 Sammenligning . . . . . 82
7.3.8 Array2tiff . . . . . 84

**8 Test af programmet** **86**
8.1 Filstørrelse . . . . . 86
8.2 Normen . . . . . 86
8.3 Delkonklusion . . . . . 89

**9 Konklusion og perspektivering** **91**
9.1 Videre perspektivering . . . . . 92

**10 Litteratur** **95**

**A Appendix** **96**
A.1 Spørgeskema . . . . . 96
A.2 CD-Rom . . . . . 98
A.3 Forklaring af diverse C termer . . . . . 98
A.3.1 Typer: . . . . . 98
A.3.2 Forkortelser: . . . . . 99

# Kapitel 1

# Forord

Vi vil gerne takke Jon Johnsen og Claus M. Spliid for deres gode og inspirerende vejledning gennem hele projektforløbet. Derudover vil vi gerne takke Kristian Rokkjær for at lade os besøge Centralbureauet for Identifikation og vise os deres arbejdsgang. Vi vil også gerne takke Arne Jensen og Steffen Lauritsen for deres tid med hensyn til nogle opklarende besvarelser af vore spørgsmål. Desuden vil vi også gerne takke alle de personer, der har svaret på vores spørgeskemaundersøgelse.

Vi har gennem Internettet fundet en stor del af den litteratur, som vi har brugt til at udarbejde projektrapporten. Faren ved at bruge Internettet som medie for den videnskablige litteratur, er at ofte er denne litteratur ikke underkastet nogen form for evaluering før den bliver publiceret. Vi har prøvet så vidt muligt at forholde os kritisk til kilderne og verificere at disse er udarbejdet af pålidelige personer.

| | |
|---|---|
| Arne Mejlholm | Bjarne M. Kondrup |
| Dan Malthesen | Dong Thanh Nguyen |
| Jesper Kristensen | Lars Hay Jensen |

# Kapitel 2

# Indledning

Der har næsten altid været et behov for at kunne indentificere enkeltpersoner. Dette kunne f.eks være ved hjælp af et kodeord, som man har brugt til at skelne ven fra fjende, eller et segl fra kongen til at vise, at man bringer hans ord. Sammen med informationssamfundet er dette krav steget, og et kodeord eller et segl er ikke længere tilstrækkeligt, hvilket bliver påpeget i afsnit 2.1.1. Dette har skabt et behov for at finde nye og mere sikre indentifikationsteknikker.

## 2.1 Problemanalyse

Vi vil undersøger hvilke identifikationsteknikker der findes og hvordan disse benyttes.

### 2.1.1 Identifikationsmetoder

Idag bruger man ofte personlige ID-kort, som f.eks pas, sygesikringsbevis, kørekort, Dankort osv. til identifikation. Mange identifikationskort benyttes, sammen med en pinkode, til at verificere at det er den rigtige person, der benytter kortet.

Samtidig med udbredelsen af Internettet er behovet for personlig verifikation steget. Desuden er brugen af ID-kort blevet utilstrækkelig, da det ikke kan verificeres, at det er ejeren, der bruger det. I nogle situationer er pinkode og ID-kort ikke sikkert nok, da disse kan forfalskes og stjæles [27]. Et andet problem med denne metode er, at man let kan komme ud for at skulle huske flere forskellige pinkoder. Der er efterhånden mange steder, hvor man skal bruge koder, for at verificere hvem man er, og så vælger mange at bruge den samme kode, hvis det er muligt. Dette medfører, at hvis der er en der har fået fat i koden, så har vedkommende adgang til mere end en ting.

Dr. Despina Polemi[1], siger i en rapport udarbejdet for Europa Kommisionen, at der findes 3 metoder til at identificer en person. Disse tre er:

- Viden personen har.
- Ting som personen ejer eller bærer på sig og karakteristika[2].
- Biometri[3].

Dr. Polemi forsætter med at fortælle, at de to første metoder ikke er tilstrækkelige nok til at modsvare den forfalskning, der finder sted idag. Dr. Polemi mener at svaret på dette problem kan være den tredje mulighed, nemlig biometri[27].

### 2.1.2 Forbedring af identifikationsmetoder

Vi tager udgangspunkt i biometrisk genkendelse af en person, hvilket leder os hen imod at bruge biometrisk genkendelse. Dette kan og bliver gjort i form af f.eks. iris scanning, stemmegenkendelse og ikke mindst fingeraftryks scanning. Den sidstnævnte har, siden starten af det tyvende århundrede, været brugt til identificering i forbindelse med retssager, og kriminel efterforskning. Da et fingeraftryk er unikt for den enkelt person, er dette en udmærket måde at identificere folk på[4].

Gennem mere end 100 år er fingeraftryk blevet brugt til identifikation, og der er endnu ikke fundet to ens fingeraftryk. I 1892 beregnede Sir Francis Galton sandsynligheden for at 2 fingeraftryk er ens til 1 til 64 milliarder[4].

For at bruge fingeraftryk til elektronisk identifikation, bliver man nødt til at digitalisere disse, så man er i stand til at kunne fortage en sammenligning. For at effektivisere dette, er det en fordel at opstille en database til arkivering af de digitaliserede aftryk. Dette stiller ikke bare nogle tekniske problemer, men også nogle etiske problemstillinger omkring overvågning og misbrug.

#### Tekniske problemer

Der vil opstå et problem med opbevaringen af de mange millioner fingeraftryk, man ville få hvis man laver en global database. Derfor skal man finde en måde at komprimere og lagre de store datamængder uden væsentligt tab af vigtig information. Hvis vi f.eks. ser på FBIs arkiver med fingeraftryk, vil dette, i ukomprimeret form, fylde mere end 2.000 Terabytes[9]. Man bliver derfor nødt til at finde en måde at gemme disse informationer og komprimere dem. Yderligere skal man have en effektiv metode til at gennemsøge denne database og lave en sammenligning af fingeraftryk.

[1] Ved Institute of Communication and Computer Systems, National Technical University of Athens.

[2] Her tænkes på vaner og personlighed, altså hvad der er karakteristisk for en person.

[3] En forklaring af biometri findes i afsnit 3.1

#### Etiske problemer

Der kan opstå problemer med, at det ikke er alle folk, som frivilligt vil lade sig registrere i en global database. En af grundene til dette kunne være, at det kan misbruges til overvågning og være en krænkelse af folks privatliv. Frygten for forfalskning af fingeraftryk er også en faktor, der taler imod brugen af fingeraftryksgenkendelse. Man kan ikke umiddelbart anskaffe sig et nyt aftryk, hvorimod pin-koder kan udskiftes, hvis koden ikke længere er sikker. Det ovenstående leder frem til overvejelsen: Hvilke etiske problemer kan der opstå i samfundet i forbindelse med implementeringen af biometrisk genkendelse og hvorledes kan man undgå problemerne.

## 2.2 Problemformulering

Udfra problemanalysen vil vi nu opstille de problemer som rapporten vil give svar på.

#### Det tekniske problem:

> *Hvordan løser man problemet med komprimering af et fingeraftryk, uden kvalitet af aftryk bliver så dårligt, at det ikke kan bruges til sammenligning*

For at opnå en løsning til dette problem, vil vi undersøge og benytte forskellige metoder kendt fra moderne matematik, billedbehandling og opbevaring af større datamængder.

#### Kontekstuelle problemstilling:

> *Hvad bruges biometrisk scanning til og hvilke etiske problemer er der knyttet til dette?*

Vi vil her lave en undersøgelse vha. et spørgeskema for at undersøge folks holdning til registrering ved hjælp af biometri. Vi vil også kigge på lovgivningen i forbindelse med denne slags teknikker og opbevaringen af disse data. På begge overordnede problemstillinger rejses der flere relevante spørgsmål.

#### Tekniske spørgsmål

- Hvordan benytter man komprimering?
  - Hvordan bruges Huffman kodning?
  - Hvordan kan splines bruges til komprimering af fingeraftryk?
  - Hvordan kan wavelets bruges til komprimering af fingeraftryk?

- Hvordan er et fingeraftryk opbygget og hvilken struktur har det?
- Hvordan gemmer man komprimeret data?
- Hvordan er enkelte billedformater opbygget?

**Kontekstuelle spørgsmål**

- Hvad er folks holdning til registrering af personlig information?
- Hvordan er loven omkring opbevaring og registering af personlig information?
- Hvor og hvordan anvendes biometrisk scanning?

## 2.3 Problemafgrænsning

For at kunne bruge fingeraftryk til identificering, vil det være hensigtsmæssigt at digitalisere fingeraftrykket, for hurtigt og effektivt, ved hjælp af en computer, at kunne identificere og genkende et aftryk. Dette må gøres ved at finde en passende algoritme, der kan digitalisere og komprimere fingeraftrykket. Digitaliseringen af fingeraftrykket skal være af en tilstrækkelig kvalitet, men må samtidig ikke optage for meget plads. Vi vil specifikt arbejde med komprimering af fingeraftryk efter at digitaliseringsprocessen er fortaget. Vi har valgt at kigge på denne del af problemet med fingeraftryk, fordi dette ikke er for omfattende til et P2 projekt og indrager alle vores PE og SE kurser. Der findes allerede en løsning på dette opbevarings problem, som er udarbejdet af FBI, men vi vil prøve om vi kan fremstille vores egen algoritme samt program til løsning af problemet. Til denne opgave har vi valgt at kigge nærmere på wavelets, som er en forholdsvis ny matematisk teknik, som kan bruges til komprimering. Da wavelet metoden ikke umiddelbart giver en komprimering i størrelsen af filen, vil vi derfor anvende Huffman kodning. Vi vil også benytte splines til at udføre komprimering, da vi mener at der er muligheder indenfor denne teknik.

## 2.4 Problemløsningsmetoder

Herefter følger en beskrivelse af de metoder vi har brugt til besvarelse af den tekniske- og den kontekstuelle problemstilling.

Vi har indledningsvis udarbejdet en beskrivelse af nutidens anvendelse af biometrisk genkendelse og tilegnet os viden om, hvor udbredt biometri er. Under denne proces udarbejder vi et spørgeskema, som danner grundlag for en etisk overvejelse om hvorvidt respondenterne ser biometri som en god eller dårlig ting. Denne undersøgelse udformes på to måder, dels udarbejdes en hjemmeside hvor man elektronisk kan give sin erfaringer og

meninger til kende, dels vil vi personligt tage ud i Aalborg og henvende os til forbipasserende for at finde deres holdninger.

Vi har tilegnet os viden omkring komprimeringsteknikker og billedformater, for at kunne udvikle et program i sproget C.

Vi beskriver teorien bag wavelets for at kunne programmere og konstruere de matematiske rutiner i et program. Vi opstiller en række teoretiske eksempler på hvorledes anvendelsen skal foregå og hvad vi forventer at opnå.

Vi beskriver teorien i Huffman kodningsmetoden og benytter denne som en forlængelse af wavelet transformeringen i programets komprimeringsfase.

Vi udvikler et program, hvor vi bygger vores komprimeringmetoder på at scanningen af et fingeraftryk er foretaget og vi har fået en strøm af rå data[4]. Vi vil benytte billedformatet TIFF, til at simulere denne data. Dette format vil vi beskrive nærmere i 5 og hvorfor vi har valgt dette format frem for andre billedformater.

Vi prøver at se på hvilke metoder, der giver bedst komprimering uden at vi laver fingeraftrykket ubrugeligt til sammenligning. For at kunne teste de forskellige komprimeringsmetoder, skal vi udarbejde en metode til at lave en vurdering af vores komprimeringsalgoritmer og deres indvirkning på billederne. Kort oversigt over hvad programmet skal indeholde:

- Konvertering af tiff billedet til et 2d-array, som vi kan arbejde med i vores komprimeringsprogram.
- Der skal også være mulighed for at gemme i vores eget format med en Huffman kodning.
- En algoritme til at komprimere og dekomprimere ved hjælp af wavelets.
- En test af hvordan komprimeringen er gået og om den er brugbar[5].

Vi har valgt at bruge dels en eksperimentel og en teoretisk metode som tilgang til problemet. Vores indgang er at bruge et program som eksperimentel metode til at analysere, hvor stor en komprimering vi kan opnå uden at kvaliteten forringes væsentligt. Til at kode programmet har vi valgt at bruge sproget C, dette er ment som forlængelse af forårets SE-kursus i C programering. Vi har valgt at beskrive tre teoretiske metoder til at opbygge komprimeringsfasen, hvor den ene er wavelet transformering, den anden er spline interpolation og den sidste er Huffman kodning. Disse metoder har relationer til kurserne vi har haft i forårssemestret. Huffman kodning kender vi fra Diskret Matematik. I Computer Støttet Beregning har vi fået kendskab til interpolation, som vi selv har bygget videre til splines som mulig metode til repræsentation af de kurver og mønstre, som et fingeraftryk består af.

[4]Dette er ukomprimeret data, som f.eks. benyttes af scanner og andre slags hardware til repræsentation af billedinformation.

[5]Vurderingen af om komprimeringen er gået godt er baseret på vores eget skøn.

Desuden benytter vores tilgang til wavelets, teorien fra lineær algebra, som vi havde i kurset Mat2A.

Huffman og transformering med wavelets udgør tilsammen den teori, der skal ligge til grund for vores programering. For at begrænse omfanget af programmet har vi valgt kun at programmere en af dem, nemlig tranformering med wavelets. Endvidere har vi benyttet en færdigkodet Huffman algoritme og implementeret denne i vores program.

Den test, der skal ligge til grund for en vudering af hvor godt komprimeringen er gået, vil være baseret på:

- Visuel inspektion.
- Filstørrelser.
- Sammenligning af pixelværdier.
  - Pixel for pixel sammenligning.
  - Gennemsitsværdier i delområder.
  - Vektor norm beregninger.

Med disse metoder kan vi så vurderer hvor godt komprimeringeren gik.

# Kapitel 3

# Biometri og Fingeraftryk

I dette kapitel giver vi en generel beskrivelse af hvad biometri er og hvad det kan bruges til. Specielt vi vil kigge nærmere på biometrisk genkendelse af fingeraftryk. Derefter vil vi betragte hvorledes lovgivningen er på dette område.

## 3.1 Biometri

Følgende er hovedsagligt baseret på kilderne [14] og [22]. Biometri er måling af biologiske mønstre på den menneskelige krop. Biometisk genkendelse er genkendelse, af et individ i en flok, baseret på individets biometriske kendetegn. Fingeraftryk er et komplekst biometrisk karakteristika, der kan aflæses og i årevis er blevet brugt til både verifikation og identifikation. Kompleksisteten af et biometrisk kendetegn betyder, i denne sammenhæng, at der er et stort antal detaljer i kendetegnet. Der findes mange biometriske træk, som kan aflæses. De kan i større eller mindre grad måles og kan bruges til identifikation eller verifikation.

| **Fysiske** | **Personlige** |
|---|---|
| DNA | Stemmen |
| Fingeraftryk | Hånd- og underskrift |
| Ansigtstræk | Skrivedynamik |
| Hånd geometri | |
| Øjet | |

Tabel 3.1: Biometriske kendetegn

Disse biometriske træk kan underinddeles i to grupper [22]: De personlige og de fysiske. De personlige er stemme, håndskrift (herunder signatur), skrivedynamik. Alle disse er personlige kendetegn, som alle forholdsvis nemt kan forfalskes. Alle gyldige dokumenter og aftaler bliver næsten altid skrevet under af involverede parter. Mange gange kan man sige at underskrifter blot

er en symbolsk handling, men alligevel bruges dette til at certificere indgåede aftaler og bindende papirer.

De fysiske træk udgør DNA, fingeraftryk, ansigtstræk, øjnene (herunder iris) og hånden. Fingeraftryk er nok bedste kendt fra kriminal efterforskning. Fingeraftryk bruges i denne sammenhæng til enten at verificere en person udfra vedkommendes fingeraftryk eller identificere en ukendt person. I denne sammenhæng bruges også DNA analyser, som med stor sikkerhed kan bekræfte, hvorvidt en person er den som vedkommende udgiver sig for, eller knytte en person til en lokation udfra efterladte DNA rester. Ulempen ved en sådan analyse er at den tager minimum et døgn, og er relativt dyr, dvs ca. 6000 kr. [14][afsnit om DNA]. Desuden er verifikation via DNA kun brugbart i meget specifikke tilfælde, hvor man blot skal bruge vejledende oplysninger. Denne vil ikke kunne bruges til endegyldig identifikation da f.eks. tvillinger har ens DNA. Yderligere efterlades DNA rester hele tiden i vores færden, hvilket både er fordelen og ulempen ved DNA analyser. Netop fordi vi efterlader rester, bruges DNA, i kriminal efterforskning, til at afsløre om en person har befundet sig på et bestemt sted.

### 3.1.1 Anvedelse af biometri

Hovedsageligt bliver biometrisk genkendelse brugt til at verificere et individ op mod en database af identiteter, som udfra resultatet, giver eller nægter adgang. Dette vil f.eks. kunne være, hvis man skulle bruge et gyldigt fingeraftryk til at starte sin bærbare computer eller opnå adgang til en bygning.

Biometrisk verifikation kan bruges til at afhjælpe problemer omkring dårlig sikkerhedsdisciplin i mange virksomheder [14][Farvel til passwords]. Det sker hyppigt at medarbejdere glemmer eller på anden måde mister deres adgangskoder. Hvis deres adgangskoder var basert på biometrisk verifikation, ville sikkkerhedsproblemet blive reduceret. Biometrisk verifikation stiller langt større krav til sikkerhed omkring behandling af data, og systemer, da man ikke kan anskaffe sig et nyt id. Biometriske træk er unikke og ud over stemmen forandres de ikke gennem ens livstid dvs. at man kan ikke anskaffe sig et nyt, hvis det bliver forfalsket. Med traditionelle adgangskoder og pin-koder kan man derimod skifte koderne, hvis disse bliver forfalsket eller opsnappet af en trejde part.

Her kommer sikkerhed i selve scanneren og systemet, ind i billedet. Flere scannere, som senere bliver beskrevet i afsnit3.2.3, er afhængig af om fingeren har elektrisk spænding, puls og blodcirkulation(varme). Disse egenskaber ændres når en finger skilles fra hånden og derved kan den ikke benyttes til snyd af biometriske skannere. Ligeledes kræves der sikkerhed i andre former for biometriske scannere, såsom at forhindre at folk snyder en iris scanner ved at bruge en kontaktlinse med anden iris på printet.

### Udbredelsen af biometrisk scanning

For nyligt er Bornholmstrafikken [11] begyndt at indføre et nyt system, som er baseret på fingeraftryksgenkendelse.

Systemet går ud på at passageren får udstedt et personligt kort, hvorpå en lille chip lagrer en skabelon af personens fingeraftryk. Når vedkommende så skal ombord på færgen aflæses kortet og fingeraftrykket. Der bliver så fortaget en sammenligning mellem kortets skabelon og det aflæste fingeraftryk. Herefter kan passageren fortsætte ombord på færgen, hvis sammenligningen blev godkendt. Systemet kan både spare tid og penge, da passagererne ikke skal hen til en billetluge. Hvilket hjælper til en nem og hurtig overfart for tilbagevendende passagere, f.eks. pendlere. Der har været problemer med at fingeraftrykkene ikke bliver genkendt, f.eks pga. skrammer på passagerernes fingrer [19]. Dette kompenserer man for ved ikke kun at tage et enkelt aftryk til sammenligning, men derimod fire[1] forskellige fingeraftryk. Så kan passageren prøve en anden finger, hvis den første ikke bliver godkendt.

Ligeledes er iris scanninger begyndt at vinde indpas, som gyldig verifikations metode i større faciliteter såsom lufthavne. Heathrow lufthavn i London skal ifølge kilderne [26] og [3] til oktober 2003 indføre et irisscanningsystem, som skal fungere som pas for passagererne. Systemet er baseret på en algoritme, der omdanner et billede af en persons iris til en 256 byte lang kode. Matematikeren John Daugman [6], som har opfundet algoritmen, kan rolig være stolt af sin opfindelse, da den for nyligt er testet med to millioner irisscanninger og systemet tog ikke fejl en eneste gang, dvs. at det endnu ikke er sket at systemet har givet en falsk positiv[2]. Nogle af disse tests har indeholdt op til ni millioner billeder i lageret [24]. Han udtaler også at snyd med systemet skulle være besværligt. Man kan lave en afbilledning af en iris på en kontakt linse og tage denne på ved scanningen. Dog skulle systemet nemt kunne afsløre snyd, da billedbehandlingen afslører de pixels som fremkommer under trykningen. Yderligere kan personalet i lufthavnen lyse i øjnene, for at se om der sker en sammentræking i pupillen og dermed konstatere om en person prøver at snyde systemet.

Formålet med dette system er ligeledes at afhjælpe kø-dannelse i lufthavnen. Man kan effektivt identificere folk og minimere personalets arbejdsbyrde ved f.eks. check in. Yderligere skulle systemet også give en ekstra sikkerhed for at passagerlisterne stemmer og at folk ikke prøver at snyde sig gennem kontrollen. Daugmans algoritme bliver anvendt over hele verden til adgangskontrol i sikrede bygninger, herunder atomkraftværker, banker og lufthavne [6]. Herunder nævner vi blot nogle få af de steder, som bruges denne algoritme:

- Heathrow Lufthavn (London)

[1] Fire er det maximale antal der kan lagres i chippen
[2] En utilsigtet godkendelse af en person, som ikke skal godkendes

- JKF lufthavn Washington
- Union Electric atomkraftværker
- Bank United of Texas

## 3.2 Fingeraftryk

### 3.2.1 Fingeraftrykket historik

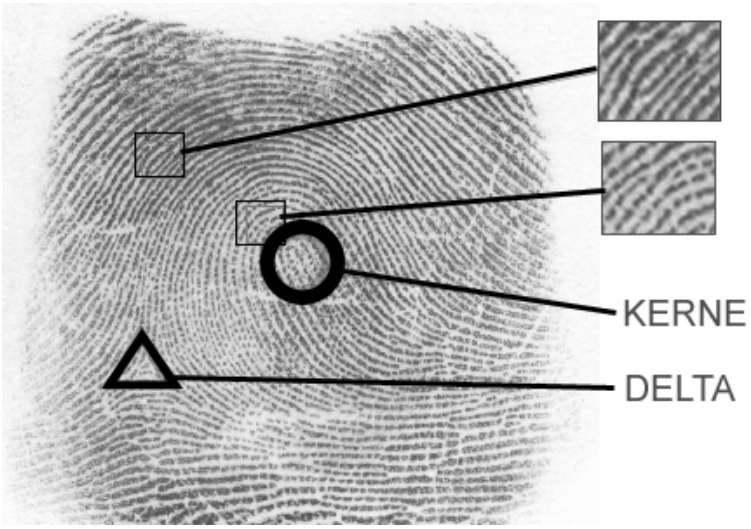
Følgende er hovedsageligt baseret på kilderne [18], [4] og [19]. De papilære linier, også kaldet fingeraftryk, opstår ca. i tredje til fjerde måned hos fostret. Hænder og fødder svulmer op hos fostret og når disse igen "falder sammen"dannes de riller, som danner fingeraftrykkets linier. Da dette sker tilfældigt, dannes der ikke ens linier hos nogen fostre, selv ikke hos identiske tvillinger[19]. Disse mønstre er uforanderlige gennem hele ens livstid, hvilket gør dem til udemærkde punkter for identifikation. Dog kan aftrykket mere eller mindre ændres ved flænger, eller andre skrammer på fingeren.

Følgende omkring fingeraftrykkets historie er baseret på kilderne: [18], [4] og [28]. I gamle kulturer, herunder Balylon, Kina og Persien, blev fingeraftryk brugt til at "underskrive" officielle dokumenter og til at skrive under på aftaler. F.eks. blev der fundet lersegl, med tommelfingeraftryk i Kina og i det 14 århundredes Persien blev dokumenter fundet med fingeraftryk på dem. En videnskabsmand i denne tid noterede desuden at ingen af de aftryk han havde set var helt ens.

I 1856 blev fingeraftryk for første gang brugt til identifikation. En forretningsmand ved navn Sir William Herschel, brugte fingeraftryk til at identificere sine ansatte og til underskrivning af kontrakter. De lokale indfødte i Jungipoor i Indien, hvor han befandt sig, var overbeviste om at den fysiske kontakt med papiret, når de afgav fingeraftryk var mere bindende en hvis de bare skrev under. Senere, efterhånden som at Sir Herschels samling af aftryk voksede, opdagede han at ikke to var ens og dette begyndte en overbevisning for ham om at fingeraftryk er unikke for alle og at de kan bruges til identifikation.

Senere i 1880 begyndte Dr. Henry Faulds at studere linierne på spidsen af fingrene, altså fingeraftrykkene. Han opdagede ikke kun at disse var unikke for alle, men lavede også et udkast til hvordan man kunne klassificere dem. Han skrev i dette år også en artikel i bladet "Nautre"[3], hvor han diskuterede fingeraftryk og deres anvendelsesmuligheder til personlig identifikation. Han beskrev også en metode til at skaffe disse aftryk, ved hjælp af blæk. Samtidig blev han krediteret for den første identificering af et fingeraftryk; Et fedtet aftryk på en flaske. I 1982 brugte Gilbert Thompson sit fingeraftryk på et dokument for sikre sig mod forfalskninger og dette er det første kendte

[3](Nature) Reference til sekundær kilde fordi original ikke er tilgængelig




Figur 3.1: Detaljerne i fingeraftryk, som også kaldes minutia

brug af fingeraftryk i USA. Allerede det efterfølgende år brugte Mark Twain i romanen "Life on the Mississippi" fingeraftryk som identifikation af en morder.

I 1892 udgav Sir Francis Galton, sin bog "Fingerprints"[4], der etablerede fingeraftryk som værende unikke kendetegn for alle. Han havde fået oplysinger omkring fingeraftryk tilsendt af sin fætter, Charles Darwin, som igen havde modtaget dem fra Dr. Henry Faulds. I bogen inkluderer han også det første system (som udkastet af Henry Faulds) til klassificering af fingeraftryk. Han identificerede de punkter som man bruger til genkendelse af fingeraftryk, såkaldte minutia også kaldet Galtons Detaljer. Detaljer er steder i fingeraftrykket, hvor de papilære linier starter, forgrenes, sammenflettes og stopper, se figur 3.1.

I 1870 opfandt den franske antropolog, Alphonse Bertillon, et system kaldet Bertillonsystemet. I dette målte og undersøgte man dimensionerne og egenskaberne på diverse kropsdele, og ud fra disse opstilles en formel. Denne skulle i teorien gælde for ethvert individ gennem dettes levetid. Systemet blev i 1903 bevist meget upålideligt, da to tvilllinger blev identificeret som værende den samme mand via systemet. De to personer blev imidlertid konstateret forskellige på deres fingeraftryk, men viste at selv tvillinger har forskellige fingeraftryk.

Omkring 1891 begyndte Juan Vucetich, fra det argentinske politi, at arkivere fingeraftryk fra personer, baseret på Galtons detaljer omkring fingeraftryk. I starten inkluderede han også Bertillons system med hans opteg-

[4]Se ovenstående fodnote

nelser, men disse har ikke megen vægt da denne metode blev bevist upålidelig. Juan Vucetich fik i 1892 brug for sin viden omkring fingeraftryk. Han identificerede en kvinde ved navn Rojas, som værende morderen til sine to børn. Kvinden havde dræbt de to børn og derefter skåret sin egen hals over i forsøg på at give skylden til en anden part. Et blodigt fingeraftryk identificerede hende dog, som værende morderen.

Imellem juli 1896 og februar 1897 etablerede Sir Edward Richard Henry [28] et modificeret klassifikations system til fingeraftryk, baseret på Galtons system. Det såkaldte Henrysystem blev hurtigt anerkendt og bliver brugt den dag i dag i de fleste engelsktalende lande. I 1901 blev Sir Henry udnævnt til kommisær i Scotland Yard som leder af afdelinger for kriminal efterforskning. Senere samme år blev fingeraftryksafdelinger i Scotland Yard afdelingen etableret.

I starten af det tyvende århundrede begyndte fingeraftryk langsomt at vinde indpas og flere og flere institutioner i USA begyndte at samle deres arkiver over fingeraftryk. I 1918 etablerede Edmond Locard 12 minutia, også kendt som punkter, som værende god standard til positiv identifikation. Selv om at indetifikationen godt kan finde sted med et mindre antal punkter, har man i USA valgt at følge denne standard. I 1924 etablerer USA's kongres en Identifikations Afdelingen af F.B.I. og i 1971 blev AFIS[5] implementeret og filerne blev delt. Kriminelles fingeraftryk blev digitaliseret og civile blev ikke digitaliserede. Det beregnes at omkring 30 millioner fingeraftryk fra kriminelle den dag i dag ligger i disse arkiver.

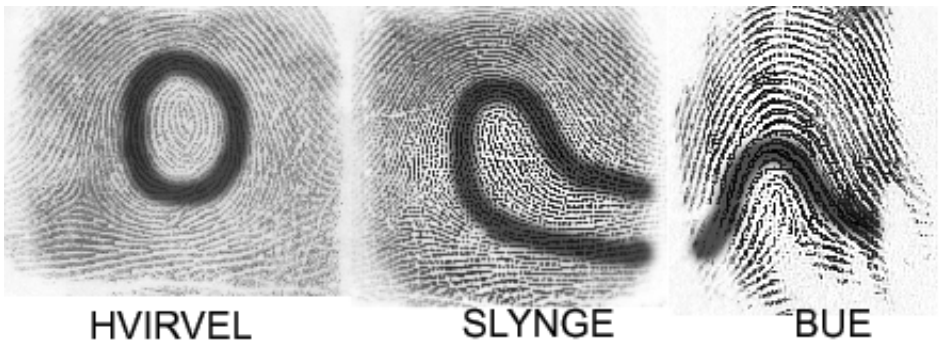
### 3.2.2 Fingeraftryk i Danmark

Den første retssag hvori fingeraftryk var det eneste bevis i sagen, fandt sted i Danmark i 1914[8]. De tiltalte Anton Broholm og Jens Herløv Jensen blev begge dømt skyldige i indbrud i 3 kirker, udfra fingeraftryk identificeret af CFI[6].

I dagens Danmark bruges ikke noget specifikt klassifikationssystem [19]. Man har i politiets fingeraftrykssektion, CFI, eksperter som identificerer fingeraftryk. De bruger AFIS til at søge i arkivet over allerede kendte kriminelle og sammenligner manuelt fingeraftryk op mod hinanden for at finde et match. Deres arkiv består af alle der har været sigtet for en forbrydelse med maksimum straf større end halvandet år. 220-240,000 personer er registreret i dette arkiv. Derudover samler man også fingeraftrykkene fra alle asylansøgere i Danmark. Fingeraftryk deles meget groft ind i tre forskellige klasser, som ses på figur 3.2, desuden er der også muligheden for en blanding.

Disse lagres så med oplysinger omkring detaljerne i aftrykkene, samt selve aftrykket. Disse detaljer bliver ved indscanningen først fundet af en computer og derefter tjekket og rettet til af en tekniker. Dette giver også

[5] Automatic Fingerprint Identification System
[6] Centralbureau for identifikation




Figur 3.2: Herover ses de 3 typer mønstre: Hvirvel, Slynge og Bue

mulighed for at tilføje ekstra punkter eller fjerne forkerte. Punkterne er detaljer i aftrykket såsom minutia, kerne og delta. Disse gemmes som vektorer, der fortæller om detaljens placering og retning. Eksempler på disse detaljer så vi i figur 3.1.

I Danmark bruges der ikke et specifikt antal punkter, men udtalelser fra CFI. Dette er i dag almen retspraksis, på baggrund af sagen om Broholm og Herløv Jensen fra 1914, som danner præcedens. Det fungerer således at hvis CFI konstaterer at to fingeraftryk er ens, bliver disse accepteret af retten. Da udtalelserne fra CFI danner grundlag for rettens syn på fingeraftryk, stilles der nogle krav for at sikre sig at deres udtalelser stemmer. Der kræves at mindst to personer, hvoraf en er afdelingsleder, til at skrive under på at to aftryk er ens. Hvis der ikke er enighed omkring en sag bliver der ikke udtalt noget før man er sikker [19].

Når man så bliver dømt, får man sine fingeraftryk, samt persondata, nedskrevet på et papir[7]. Dette bliver gemt i arkiverne og bliver liggende til sammenligning for eftertiden[8]. Skulle en person ikke være i arkivet kan en sammenligning ikke finde sted. Aftrykket bliver gemt i en speciel sektion, hvor den "venter"på at et aftryk der matcher findes. Så hvis man i forbindelse med en forbrydelse efterlader et aftryk, som ikke findes i arkivet, bliver selve aftrykket alligevel gemt i arkivet. Skulle man senere bliver taget for en anden forbrydelse, bliver det aftryk som man afgiver derefter sammenlignet med aftrykkene fra tidligere uopklarede sager.

### 3.2.3 Fingeraftryksscannere

Allerende nu findes der mange fingeraftryksscannere, som bruges til alt fra mindre systemer, f.eks. låsning af filer og computere, til større systemer,

[7]Et eksempel på dette kan findes på den vedlagte CD-Rom

[8]Loven omkring registrering og opbevaring af personlige data gennemgåes i nedenstående afsnit 3.3

som adgangskontrol i bygninger og lignende. Der er stor forskel på hvilke teknologier der anvendes, alt afhængig af hvilken producent, der laver dem. Der arbejdes for tiden på at finde en standard for fingeraftryks scannere, eller rettere de billedformater scannerne benytter. Arbejdet med standardiseringen bliver udført af Incits og Ansi [15], men de har endnu ikke lavet en standard.

Precise Biometrics er udviklerne af det system, som bruges på Bornholmslinien [11]. Chefudvikler Ola Svedin[15][afsnit: Fuldstændig sikkerhed findes ikke] fortæller, at alle deres produkter, med undtagelse af et enkelt, bruger en kapacitiv chip til aflæsning af fingeraftrykket. Modsat en optisk chip, som belyser fingeren fra neden og som kan snydes af et godt billede af aftrykket kræver den kapacative chip at fingeren er tilstede. Det fungerer ved at hvert pixel på chippen måler kapaciteten (spændingen) mellem fingeren og chippen. Større forskelle betyder riller i fingeren. Yderligere kræves en levende finger, da netop fingerens elektriske egenskaber bruges i denne form for aflæsning. Ole Svedin mener ikke at deres scannere kan snydes af en afhugget finger, eller lignende form for snyd, men deres scannere var desværre ikke med i en test omkring netop muligheder for at snyde sådanne systemer.[16]

I undersøgelsen blev forskellige former for "falske" fingre testet op mod henholdsvis kapacitive og optiske scannere. Først blev en silikone finger testet, hvor en optisk scanner godtog fingeren, mens den kapacitiv aflæser afviste den. Ved test med "gummifingre" var det derimod den kapacitive aflæser der blev snydt. Gummifingerne bestod af materialer, som er lette at anskaffe, såsom husblas. Dette betyder at en kapacativ chip ikke er nok beskyttelse imod forfalskninger, da en kunstig finger kan snyde systemet.

Det er vigtigt at skelne mellem de forskellige biometriske systemers formål. Det er ikke altid at et biometriske system er designet til at være sikkert, men derimod kan det være designet til at højne effektiviteten. F.eks. er systemet, som bliver brugt af Bornholmsfærgen [11], hovedsageligt designet til at nedskære omkostninger, fremfor at øge sikkerheden.

Med udbredelsen af biometriske scannere og den mulighed for overvågning som dette giver, er der behov at se hvordan loven er på dette område. Der skal opstilles klare rammer for, hvad der er lovligt med hensyn til brug af biometrisk identifikation. Endvidre skal vi se hvordan loven omhandler opbevaring af biometriske tegn og i hvilke tilfælde man må aftvinge folks biometriske tegn. Vi vil nu kort se på hvordan lovgivningen er med henblik på opbevaring af personlige data.

## 3.3 Lov vedrørende registre

For at kunne oprette et register er der nogle love, som skal overholdes. Der er derimod ikke nogle love, som er specifik mønstret på et fingeraftryk-

sregister, men der er love, som omhandler regler om registre og behandling af personoplysninger generelt. Som grundlag for dette afsnit ligger lov nr 429 om behandling af personoplysninger (429)[31], Bekendtgørelse om behandling af personoplysninger i det Centrale Kriminalregister (Krim)[5] og Bekendtgørelse af lov om rettens pleje (Ret)[30].

### 3.3.1 Behandling af personoplysninger

I henhold til lov nr 429, om behandling af personoplysninger, gælder der følgende. I dag kan fingeraftryk kun kræves som et led i en efterforskning, og kun hvis den pågældende er mistænkt i en sag der kan medføre fængsel i mere end halvandet år (Ret)§ 792, §792a stk 1 og stk 2. Hvis dette ikke er er tilfældet så skal personen give sit samtykke (Ret)§ 792d stk 1. Derudover skal man også fortælle vedkommende, hvad fingeraftrykket skal bruges til, og fingeraftrykket må ikke senere bruges til andre formål, (429) § 5 stk 2. Dertil kommer det at et fingeraftryk ikke må opbevares efter at personen er blevet frifundet eller sigtelsen er droppet (Ret)§ 792f stk 2. Den registrede har til enhver tid ret til, at forlange at få at vide hvilke oplysninger der behandles, hvad formålet med behandlingen er, hvem der ser og modtager oplysningerne og hvorfra disse oplysninger stammer (429)§ 31 stk 1. To mulige undtagelser fra denne lov er hvis det sker af rent videnskabelig grunde, som f.eks en undersøgelse af fingeraftryk, eller hvis det er i forbindelse med retsager[9], så kan justitsministeren fastsætte undtagelser, henholdsvis (429)§ 32 stk 4 og stk 5.

Til at holde opsyn med disse behandlinger, som bliver foretaget, har man et uafhængig datatilsyn, bestående af et råd og et nævn (429)§ 55 og § 56. Datatilsynet har ret til at stoppe en behandling hvis de finder at denne er i strid med loven, de kan endvidere også pålægge, at dem der opbevarer oplysningerne, skal øge sikkerheden omkring oplysningerne i form af tekniske foranstaltninger, (429)§ 59. Folk, ansat af datatilsynet, kan til enhver tid få adgang til lokaler, hvor der forgår behandling af oplysningerne på vegne af den offentlige forvaltning, (429)§ 62 stk 2.

### 3.3.2 Opbevaring af personoplysninger

Når man først har taget et fingeraftryk ligger der et ansvar på dem, som skal opbevare fingeraftrykket, i form af at skulle holde oplysningerne opdateret. Det vil sige at de oplysninger der ligger i registret ikke må være forkerte eller forældede (429)§ 5 stk 4. Hvis det opdages at der findes oplysninger, som viser sig at være forkerte så skal disse omgående slettes (429)§ 37 stk 1. Oplysningerne må heller ikke opbevares længere tid end det er nødvendigt og den registrede kan til enhver tid få lov til at se hvilke oplysninger der bliver opbevaret omkring vedkommende, og han/hun har ret til at gøre indsigelse

[9] næsten alle tilfælde med fingeraftryk er af denne type.

mod at disse oplysninger bliver brugt, hvis indsigelsen er berettiget (429)§ 5 stk 5 og § 35 stk 1 og 2.

I kriminalregistret må oplysningerne om en dømt person, opbevares indtil 2 år efter at denne person er død, eller når oplysningen om personens død kommer rigspolitichefen i hænde, hvis dette sker efter 2 år (Krim)§ 10.

Der pålægges også den ansvarlige for registret, at skulle sørge for at oplysningerne ikke ødelægges eller forringes, samt at de ikke misbruges eller videregives til uvedkommende(429)§ 41 stk 3.

### 3.3.3 Opsummering

Vi har i dette kapitel set på hvad biometri og biometriske kendetegn er. Vi har set hvordan et fingeraftryk er opbygget og dets karakteristika. Derefter har vi set hvad biometrisk genkendelse kan bruges til, specielt med henblik på brugen af fingeraftrykksscannere. Vi har beskrevet den historiske baggrund for metoder til genkendelse af fingeraftryk. Derudover har vi set, hvordan CFI behandler fingeraftryk og hvorfor deres ord er afgørende.

Vi betragtede de love som ligger til grund for opbevaring og registrering af personlige oplysninger. Desuden har vi set på hvordan man omgås personlige oplysinger.

# Kapitel 4

# Teknologikritik

Vi vil i dette kapitel beskæftige os med de etiske og moralske overvejlser, der opstår i forbindelse med indførelsen af en ny teknologi. Vi vil i denne forbindelse lave en teknologivurdering og en spørgeskemaundersøgelse, som skal begrunderne vores vurdering.

## 4.1 Etik

De etiske overvejelser tager udgangspunkt i at en teknologi *er* blevet eller *kan* blive problematisk [29]. Derfor falder det naturligt at bruge etik som ramme for en teknologikritik. Dette leder os hen mod at betragte hvad normer, moral samt etik egentligt er.

Som vi skrev i afsnit 2.1.1, er adgangskoder til identificering ikke tilstrækkeligt. For at følge op på vores udsagn om at der er etiske problemstillinger omkring biometri i afsnit 2.1.2, vil vi først kigge på hvad etik er, for derefter at lave en teknologikritik af biometrisk genkendelse med etik som ramme.

### 4.1.1 Hvad er etik?

Normer påvirker menneskers daglige færden og gøremål. Dette kan føre til personlige konflikter mellem normer og følelser. F.eks når man gerne vil ud og nyde det gode vejr kan der komme en konflikt med at man burde forberede sig bedre til et næste dags forelæsning. Normer kan påvirke vores adfærd, ved at diktere en standard, som man tester sine handlinger mod [29].

Moral dikterer hvad vi skal gøre i forskellige situationer, dvs. den omhandler alle vores normer, værdier og holdninger. Dette vil som oftest ikke være i en explicit formuleret form, men den viser sig som regel kun i visse situationer. Normer og moral er noget der ændrer sig med tiden i takt med samfundet. Nogle ting som ikke var accepteret for 80 års siden er nu blevet alment brug.

Med moral skal man betragte to forskellige sider, nemlig hvad man gør

og hvad man burde gøre. Moral kan fortælle os hvad vi bør gøre, men det er ikke sikkert at vi følger det. Omvendt kan vi heller ikke sige hvis man gør en ting, så er det osse det man burde gøre [29].

> "Etik er teoretiske og filosofiske overvejelser vedrørende den rette handlen." [21]

Etik handler altså om at finde moral og hvordan man kan begrunde et moralsk udsagn. Moral er etik omsat til praksis. F.eks hvis forældrene siger til sit barn at det ikke må tegne på væggene, så er det et moralsk udsagn. Fortæller forælderene hvorfor at børn ikke må skrive på væggen, så er det en etisk udredning.

### 4.1.2 Forskellige teknologietikker

Vi vil i det følgende kigge på flere former for teknologietikker, for at klargøre at den etiske indgangsvinklen til teknologikritikken ikke er ligegyldig.

#### Pligtetik

Essensen i pligtetik er at man skal kun udføre gode moralske handlinger, ikke kun fordi man har lyst eller fordi man vil være dydig, men fordi det er individets pligt. "Kants kategoriske imperativ", efter den tyske filosof Immanuel Kant, kan kaldes en etisk grundlov og er et godt eksempel på pligtetik.

> "Du skal handle sådan, at princippet i din handling kan gøres til en generel lov, som alle mennesker skal følge." [29]

Det enkelte individ skal med sin fornuft sørge for at sine handlinger overholder denne lov. Derved kommer fornuft til hænge sammen med moral og pligt.

Kendetegnende for en teknologietik baseret på pligtetikken er at den fører ofte til en teknologikritik, der er kendetegnet ved at teknologien ikke må overskride forskellige moralske grænser. Så er det ligemeget hvor mange fordele der ville være ved denne teknologi. Problemstillingen omkring fingeraftryk bliver relevant her, hvor nogle vil hævde at den personlige frihed ikke må krænkes, ligemeget hvor effektivt et system baseret på fingeraftryksgenkendelses måtte være. Pligtetik er som regel meget svær at efterleve i praksis, da den ofte vil føre til konflikter mellem interesser. Unge mænd har f.eks ofte konflikten mellem at aftjene deres værnepligt og gennemføre en videregående uddannelse hurtigt. Til en teknologikritik er pligtetikken oftest ikke repræsentativ, da den som regel ikke inddrager de positive sider.

#### Nytteetik

Nytteetikken tager udgangspunkt i fordelene ved en teknologi og det handler derefter om at minimere ulemperne.

> "Hvorvidt en handling er moralsk god eller dårlig afhænger af hvor megen glæde og gavn handlingen afstedkommer (den positive side), kombineret med den smerte og ulykke som handlingen drager med sig (den negative side)."[29]

Her er det konsekvensen af en handling der tæller, modsat pligtetikken som opprioterede handlingen. Ligeledes er det også samfundets nytteudbytte der kendetegner nytteetikken, hvor pligtetikken var baseret på individet. Denne etik er traditionelt blevet brugt til at redegøre for de positive sider ved ny teknologi. Den dikterer at det ikke betyder noget at nogle få må lide pga. en teknologi, hvis bare samfundet får det bedre. Ulempen ved denne etik er at den kommer til konflikt med vores moralske intuition. Det kan vel ikke være rimeligt at indføre ny teknologi der skader nogle for at andre får det bedre?

#### Samtaleetikken

Pligtetikken betragtes som urealistisk og nytteetikken er ikke moralsk acceptabel, derfor indfører vi samtaleetikken. Samtaleetikken baseres på at deltagerne har samme basis i form af magt og viden. Man kan gennem diskussion opnå at træffe beslutninger som alle parter kan acceptere. Der er en tendens til at man på basalt niveau kan blive enige, men jo mere detaljeret det etiske spørgsmål bliver jo mere varierer synspunkterne[21].

> "I en samtaleetik finder man frem til moralsk korrekte handlinger gennem en åben og informeret samtale, hvor alle samtaleparter har samme udgangspunkt. Det som alle de involverede parter kan blive enige om, er det moralsk rigtige." [29]

Samtaleetikken bliver altså en form for hybrid, formet på pligtetikken og nytteetikken. Pligetikken er basen, men der bliver taget højde for at individet ikke er alene, men at det befinder sig i et samfund. Handlinger har konsekvenser for individer i samfundet, derfor skal der tages hensyn til andre værdier og holdninger. Moralsk korrekte handlinger skal besluttes af de involverede personer i konsensus. Etikken betragter teknologier således at den både kan være ven og fjende.

## 4.2 Spørgeskema - undersøgelse

Da biometrisk scannning ikke er særlig udbredt i Danmark, må denne kritik tage udgangspunkt i en vurdering af hvad konsekvenserne af teknologien kan blive. Siggaard Jensen og Skovmose[25] bruger om dette termen *angst*.

> "Vi tager udgangspunkt i ≫angsten≪. Vel at mærke ikke i angsten for teknologien, men angsten for de kriser og katastrofer den teknologiske udvikling skaber, bestemte men ukendte sandsynligheder for." [25][Kapitel 7, side 110]

For at finde frem til hvad angsten hos det enkelte individ rent faktisk består af, tager vi vores udgangspunkt i en spørgeskemaundersøgelse. Denne undersøgelse er opbygget af fire hypoteser, som vi forventer at påvise ud fra de data, som vi indsamler.

### 4.2.1 Undersøgelsens formål

Formålet med undersøgelsen er, at få belyst den danske befolknings holdninger med hensyn til biomestriske scanning. Derudover belyse den almene danske borgers kendskab til biometriske scanning.

### 4.2.2 Målgruppe

Dette vil vi gøre ved elektronisk at indsamle svar fra de personer som er tilknyttet storgrupperne 31-34 ved det Teknisk-Naturvidenskabelige basisår på Aalborg universitet, også kendt som Tek-Nat Basis. Vi har valgt den elektroniske form, fordi det er nemt at indsamle data fra en stor gruppe af mennesker på kort tid. Ulempen ved denne form er at vi ikke har personlig kontakt med respondenterne og de er derfor ikke særlig forpligtiget til at give os sande data. Derfor vil vi indbygge et åbnet spørgsmål for at kunne vurdere data som vi nærer mistillid til.

For ikke at lade os forblinde af et sæt data, har vi valgt også at lave en manuel undersøgelse i området omkring Nytorv i Aalborg. Dette forgår ved at tage den direkte kontakt til personer som befinder sig i området. Fordelen ved denne metode er at respondenterne i højere grad føler sig forpligtiget til at give sande data. Ulempen er at det er meget ressource krævende.

Den ene målgruppe er studerende, hvilket vil sige at størstedelen af besvarelserne vil komme fra yngre individer. Testpersonerne vil derfor stort set være i gang med en videregående uddannelse eller have gennemført en. Da disse personer i højere grad er knyttet til elektronisk materiel end den jævne borger kan det forventes at kendskabet til biometriske scannere er højere end i den anden målgruppe, som er et bredere udsnit af den danske befolkning som vi forventer at se på Nytorv. På Nytorv forventer vi at møde den del af befolkningen, som færdes i dette miljø midt på dagen. Dette vil sige at vi nok ikke møder den del af befolkningen der har en dagligdag præget af et 8-16 job.

### 4.2.3 Hypoteseopstilling

Vi opstiller fire hypoteser, som vi ønsker at påvise eller afvise ved at lave spørgeskemaundersøgelsen. Hypoteserne er udtryk for de forventninger, som vi har til den almindelige danskers kendskab til biometri og holdning til biometrisk genkendelse.

- **Hypotese 1 (Demografisk afhængighed)**
  Ældre frem for unge føler sig afskrækket fra at stole på automatiseret biometri validering. De er ikke vant til at lade sig registrere eller til de elektriske hjælpemidler. Med hensyn til kønnenes stillingstagen forventer vi ingen forskel.

- **Hypotese 2 (Kendskab til biometriske valideringsmetoder)**
  Vi mener at det er et fåtal af danskerne der har haft førstehåndskendskab til automatiserede biometriske identificeringsmaskinel. De er først lige på vej frem i Danmark. Desuden mener vi at manglende kendskab også giver større tendens til angst.

- **Hypotese 3 (Mistillid)**
  Folk er bange for, at de data som de afgiver til registrering kan blive misbrugt. Ens daglige færden kan kortlægges, sådan at alle folk kan overvåges hele tiden. Derudover vil der være frygt for, at ens data kan blive stjålet og reproduceret.

- **Hypotese 4 (Ubehag ved forandring)**
  Visse mennsker vil føle ubehag ved at skulle identificere sig hele dagen lang overfor fremmede mennesker. Der er betragtelige fordele ved biometrisk genkendelse, men hidtil er der også mange problemer. Nogle mennesker vil sikkert mene, at det ikke er besværet værd eller direkte ubehageligt at konvertere til genkendelse vha. biometri.

### 4.2.4 Omformning af hypoteser til spørgsmål

Det udarbejdede spørgeskema kan ses i appendix A.1. For at undgå unødig forvirring af testpersonerne, har vi valgt at lave spørgsmålene på lukket form, dvs. med et bestemt antal muligheder. Som den eneste undtagelse har vi valgt at lade testpersonen tage stilling til et åbent spørgsmål, nemlig om deres arbejde i spørgsmål tre.

- **Spørgsmål 1 og 2**
  De nødvendige demografiske oplysninger indsamles i spørgsmålene 1 og 2. Formålet med dette er at inddele respondenterne i undergrupper. For at få svar på hypotese 1 tager vi fat i alder og køn. Dette gøres ved at inddele aldersfordelingen i intervaller for at sikre enkelthed.

- **Spørgsmål 3**
  Vi laver et åbent spørgsmål for at kunne sortere eventuelle useriøse besvarelser fra. Hvis vi modtager data som vi ikke havde forventet, kan disse oplysninger bruges til at forklare abnormale tendenser i besvarelserne.

- **Spørgsmål 4, 5 og 6**
  For at belyse Hypotese 2, spørger vi hertil hvorvidt respondenten har haft førstehåndskendskab til biometriske scannere. Her forventer vi at de fleste vil svare nej, men det vil i særdeleshed være relevant at udspørge dem, hvis overhovedet nogen, der kender biometrisk apperatur, om hvilket biometrisk kendetegn det drejede sig om.

- **Spørgsmål 7, 8 og 9**
  Spørgsmålene forsøger at belyse Hypotese 3, med henblik på at finde ud af, hvor mange mennesker har noget mod biometri. Samtidigt er det nærliggende, at spørge dem der er for, hvad de ser som den egentlige fordel ved biometrisk genkendelse.

- **Spørgsmål 10 og 11**
  For at belyse Hypotese 4 har vi valgt 2 lukkede spørgsmål for se hvorledes vores respondentgruppe vil accepterere at biometrisk genkendelse bliver implementeret i deres dagligdag.

- **Spørgsmål 12**
  Til sidst har vi inkluderet et spørgsmål, der gerne skulle være med til at bestemme kvaliteten af vores spørgeskema.

## 4.3 Resultater af undersøgelsen

Ved enhver undersøgelse er der altid nogle forbehold. I forbindelse med den undersøgelse vi lavede på Aalborg Universitet d. 22 April 2003 sorterede vi selv 5% fra, da vi skønnede at de var for useriøse og utroværdige. Dette skøn foretog vi baseret på besvarerelserne i spørgsmål 3.

I forbindelse med undersøgelsen i området omkring Nytorv i Aalborg d. 4 og 5 Maj 2003 skal det nævnes at vi for, at genere så få personer som muligt, kun kontaktede personer, som vi skønnede kunne afsætte 2 min. til at udfylde vores spørgeskema. Derudover opdagede vi hurtigt at det ofte var mindre stressende for personen, hvis vi i stedet for selv at lade dem udfylde skemaet, interviewede dem således at vi læste spørgsmålene op og skrev svarene for dem. Dette kan i visse tilfælde have gjort, at vi utilsigtet har påvirket respondenten.

De besvarelser vi fik fra Tek-Nat Basis til spørgsmål 8, blev desværre pga. en fejl i vores dataindsamlingssystem værdiløse, derfor har vi valgt heller ikke at betragte de besvarelser af spørgsmål 8, der kom ind fra Nytorv.

Derudover skal det nævnes, at det omkring eftermiddagen d. 4 begyndte at regne, derfor indstillede vi arbejdet med indsamling af data indtil næste dag.

### 4.3.1 Vurdering af målgrupper

| | Tek-Nat Basis | | Nytorv | | |
|---|---|---|---|---|---|
| | **Mænd** | **Kvinder** | **Mænd** | **Kvinder** | **I alt** |
| 15-27 | 152 | 44 | 14 | 16 | 226 |
| 28-40 | 12 | 3 | 8 | 12 | 35 |
| 41-53 | 1 | 0 | 10 | 7 | 18 |
| 54-66 | 0 | 0 | 5 | 7 | 12 |
| 67-? | 0 | 0 | 4 | 1 | 5 |
| *I alt:* | 165 | 47 | 41 | 43 | 296 |

Tabel 4.1: Demografisk fordeling

Som vi ser i tabel 4.1, er undersøgelsen på Tek-Nat Basis meget skævt fordelt i forhold til alder og køn, hvorimod undersøgelsen på Nytorv er jævnt fordelt med hensyn til køn. I undersøgelsen er der desuden også flest unge, men ikke i samme grad som undersøgelsen på Tek-Nat Basis.

Undersøgelsen omfatter i alt 296 responenter, heraf er ca. to tredjedele af respondenterne fra Tek-Nat Basis. På dette kan vi groft konkludere at på trods af at undersøgelsen på Tek-Nat Basis har så stort et antal respondenter, kan den ikke betragtes som repræsentativ for unge mænd mellem 15 og 27 år. Dette er hovedsageligt fordi de studerende på Tek-Nat Basis må forventes at dele interesser indenfor naturvidenskab. Dette mener vi ikke er repræsentativt for den aldersgruppe generelt. Desuden kan vi ikke bevise at vi har sorteret alle useriøse besvarelser fra.

Undersøgelsen ved Nytorv kan heller ikke betragtes som repræsentativ da der er alt for få respondenter. Den er derimod jævnt fordel med hensyn til køn. Den kan derfor betragtes som vejledende, men ikke repræsentativ.

### 4.3.2 Databehandling

Formålet med vores spørgeskemaundersøgelse var oprindeligt at belyse den jævne danskers kendskab til biometriske scannere og deres holdning dertil.

Til behandlingen af data vil vi af praktiske årsager indele spørgsmålene i hovedgrupper:

- Demografiske oplysninger
- Kendskab til biometriske scannere
- Fortrolighed med registrering
- Holdninger til og angst for biometri

- Deres forståelse

Grundet vores hypotese 1 vil vi i det følgende primært betragte vores data fra et demografisk synspunkt.

### Kendskab til biometriske scannere

Fra tabel 4.2 ser vi besvarelsen af spørgsmål 4. Den viser at 3 ud af 20 personer, som har deltaget i undersøgelsen på Tek-Nat Basis, har prøvet eller set en biometrisk scanner. I en undersøgelse af ca. 200 personer er dette faktisk en del mere end vi havde forventet. Tallet er mindre for de folk vi mødte på Nytorv. Tendensen er, som vi ser i tabel 4.3, at ældre folk har mindre kendskab til biometriske scannere. Det udsving vi ser i aldersgruppen 41-53 på Nytorv, skyldes statistisk tilfældighed, da det drejer sig om 3 personer i den aldersgruppe, som har prøvet biometrisk scanning. Ved udformningen af spørgeskemaet havde vi ikke tænkte på, at der faktisk er indbygget biometrisk genkendelse[1] i de fleste nye mobiltelefoner. På Nytorv oplevede vi faktisk kun en af de 85 adspurgte, som hæftede sig ved dette. Vi forklarer derfor den højere kendskabsrate hos unge, med at de i højere grad er bevidste om den indbyggede stemmegenkendelse i nyere mobiltelefoner. Konklusionen på hypotese 2 må altså være, at en del har kendskab til biometrisk scanning.

| | Tek-Nat Basis | | Nytorv | |
|---|---|---|---|---|
| | **Ja** | **Nej** | **Ja** | **Nej** |
| *Mænd* | 17% | 83% | 15% | 85% |
| *Kvinder* | 17% | 83% | 7% | 93% |

Tabel 4.2: Kendskab til biometri, køn

| | Tek-Nat Basis | | Nytorv | |
|---|---|---|---|---|
| | **Ja** | **Nej** | **Ja** | **Nej** |
| *15-27:* | 18% | 82% | 13% | 87% |
| *28-40:* | 7% | 93% | 7% | 93% |
| *41-53:* | 0% | 100% | 18% | 82% |
| *54-66:* | 0% | 0% | 0% | 100% |
| *67-?:* | 0% | 0% | 0% | 100% |

Tabel 4.3: Kendskab til biometri, alder

[1] Stemmegenkendelse

### Fortrolighed med registrering

Af tabel 4.4 ser vi umiddelbart at mænd fremfor kvinder har noget imod registrering af biometriske kendetegn, men kigger vi nærmere, så ser vi, at der faktisk er en tendens til at kvinder i højere grad er i tvivl i stedet for imod registrering. Generelt set over alle aldersgrupper og køn er det ca. en tredjedel af respondenterne, der stiller sig imod at få sine biometriske kendetegn registreret.

| | Tek-Nat Basis | | | Nytorv | | |
|---|---|---|---|---|---|---|
| | **Ja** | **Nej** | **Ved ikke** | **Ja** | **Nej** | **Ved ikke** |
| *Mænd* | 30% | 64% | 6% | 30% | 65% | 5% |
| *Kvinder* | 17% | 68% | 15% | 16% | 72% | 12% |

Tabel 4.4: Fortrolighed med registrering, køn

### Holdninger til og angst for biometri

I hypotese 3 formulerede vi vores forventninger til, hvad folks meninger egentlig ville være omkring sikkerheden i de biometriske systemer. Ud fra tabel 4.5 og 4.6 ser vi at ud af 296 personer, er der ikke nogen, der vil være bekymret omkring sikkerheden i et biometrisk system. De folk vi mødte på gaden, argumenterede hovedsageligt med, at det var de egentlig ikke bange for da de egentlig ikke kunne se, hvorfor folk skulle stjæle deres biometriske kendetegn og udgive sig for at være dem. De kunne ikke se nogen trussel mod deres person. Ca. 25% af de adspurgte ville være delvist bekymrede over sikkerheden i biomtriske systemer. Det er noget lavere end hvad vi havde forventet. Vi havde forventet at halvdelen af responenterne ville være bekymrede ved at afgive deres data.

| | Tek-Nat Basis | | | | |
|---|---|---|---|---|---|
| | **Høj grad** | **En del** | **Mindre grad** | **Slet ikke** | **Ved ikke** |
| *15-27:* | 0% | 23% | 54% | 13 % | 10% |
| *28-40:* | 0% | 14% | 72% | 7 % | 7% |
| *41-53:* | 0% | 0% | 100% | 0 % | 0% |
| *54-66:* | 0% | 0% | 0% | 0 % | 0% |
| *67-?:* | 0% | 0% | 0% | 0 % | 0% |

Tabel 4.5: Angst for at sikkerheden ikke vil være tilstrækkelig, alder (Tek-Nat Basis)

Af tabel 4.7 og 4.8 ser vi at fordelingen af svar, groft set, fordeler sig jævnt over alle svarmuligheder. Desværre åbner dette ikke op for ret mange fortolkningsmuligheder. Konklusionen er derfor at der er mange holdninger

| | Nytorv | | | | |
|---|---|---|---|---|---|
| | **Høj grad** | **En del** | **Mindre grad** | **Slet ikke** | **Ved ikke** |
| *15-27:* | 0% | 0% | 68% | 29 % | 3% |
| *28-40:* | 0% | 19% | 37% | 44 % | 0% |
| *41-53:* | 0% | 24% | 12% | 35 % | 29% |
| *54-66:* | 0% | 0% | 50% | 33 % | 17% |
| *67-?:* | 0% | 0% | 50% | 50 % | 0% |

Tabel 4.6: Angst for at sikkerheden ikke vil være tilstrækkelig, alder (Nytorv)

til overvågning. En af tendenserne i svarene er, at der ikke er mange, som har svaret ved ikke. Derfor kan vi konkludere, at det trods alt er noget som de fleste folk har en mening om. Dette fænomen kan skyldes, at den almene borger knapt nok kender teknologien og derfor ikke kan vurdere konsekvenserne af den.

| | Tek-Nat Basis | | | | |
|---|---|---|---|---|---|
| | **Høj grad** | **En del** | **Mindre grad** | **Slet ikke** | **Ved ikke** |
| *15-27:* | 21% | 35% | 32% | 7 % | 5% |
| *28-40:* | 7% | 29% | 57% | 7 % | 0% |
| *41-53:* | 100% | 0% | 0% | 0 % | 0% |
| *54-66:* | 0% | 0% | 0% | 0 % | 0% |
| *67-?:* | 0% | 0% | 0% | 0 % | 0% |

Tabel 4.7: Angst for overvågning, alder (Tek-Nat Basis)

| | Nytorv | | | | |
|---|---|---|---|---|---|
| | **Høj grad** | **En del** | **Mindre grad** | **Slet ikke** | **Ved ikke** |
| *15-27:* | 20% | 27% | 33% | 17 % | 3% |
| *28-40:* | 16% | 5% | 27% | 47 % | 5% |
| *41-53:* | 18% | 24% | 29% | 29 % | 0% |
| *54-66:* | 17% | 0% | 33% | 42 % | 8% |
| *67-?:* | 50% | 0% | 25% | 25 % | 0% |

Tabel 4.8: Angst for overvågning, alder (Nytorv)

Af tabel 4.9 og 4.10 ser vi, at cirka hver anden ser positivt på biometrisk scanning og specielt i tabel 4.10, ser vi at der er nogle enkelte, der ikke ville være positivt stillet overfor indførelsen af biometrisk scanning, men stadig kan se en fordel for samfundet. Desuden viser det, at folk er mere bange for at blive overvåget, end at miste eneretten på deres indentitet.

| Tek-Nat Basis | | | Nytorv | | |
|---|---|---|---|---|---|
| **Ja** | **Nej** | **Ved ikke** | **Ja** | **Nej** | **Ved ikke** |
| 42% | 38% | 20% | 54% | 20% | 26 % |

Tabel 4.9: Positivt stillet overfor biometrisk scanning ved f.eks. Dankort

| Tek-Nat Basis | | | Nytorv | | |
|---|---|---|---|---|---|
| **Ja** | **Nej** | **Ved ikke** | **Ja** | **Nej** | **Ved ikke** |
| 57% | 32% | 11% | 59% | 27% | 14 % |

Tabel 4.10: Kan se en fordel ved biometrisk scanning

**Deres forståelse**

Vi tilføjede spørgsmål 11 for hurtigt og nemt at kunne kontrollere kvaliteten af vores undersøgelse, ihvertfald respondenternes bedømmelse. Deres bedømmelse er faktisk meget positiv, vi ser at i tabel 4.11, er det kun 1 ud af 10, der har givet udtryk for at de kun "i lav grad" forstod vores formulering af spørgeskemaet. Der er en tendens til at de respondenter, der har deltaget elektronisk på Tek-Nat Basis, oftere har svaret "i lav grad", dette falder meget naturligt, da vi på Nytorv havde den direkte kontakt med respondenten, sat i kontrast til at respondenterne på Tek-Nat Basis kun havde vores skrevne ord at støtte sig til. Med disse holdninger og tendenser kigger vi nu på den egentlige teknologikritik.

| | Tek-Nat Basis | | | Nytorv | | |
|---|---|---|---|---|---|---|
| | **Lav** | **Mellem** | **Høj** | **Lav** | **Mellem** | **Høj** |
| *Mænd* | 6% | 25% | 69% | 0% | 22% | 78% |
| *Kvinder* | 11% | 34% | 55% | 5% | 33% | 62% |

Tabel 4.11: Grad af forståelse af spørgeskema, køn

## 4.4 Selve teknologikritikken

Med lanceringen af fingeraftryksvalidering som betalingsmiddel på Bornholmerfærgen[2], tager vi så småt hul på diskussionen omkring et muligt overvågningssamfund i Danmark. Dette spås til blot at være et af de første steder hvor elektronisk validering med biometri får indpas. I dag kan vi opleve det på Bornholmerfærgen, flyvevåbnet overvejer at indføre det som indentifikationssystem og markedet indenfor biometri vil ifølge Preben Mejer, leder af Innovation Lab, tyvedobles inden 2006 [32].

Anvendelsesmulighederne er uoverskuelige, i stedet for et dankort kunne man sætte fingeren på en scanner, og derved hæve fra automat eller betale

[2]Se afsnit 3.1.1

for sine varer. Kort sagt kan biometri anvendes overalt, hvor der kan være brug for identifikation, givet selvfølgeligt at det også er bakket op af den nødvendige sikkerhed.

Biometrisk scanning har potentialet til at blive et problem, hvis det ikke bliver håndteret ordentligt. Derfor er det på plads med en teknologikritik af biometrisk genkendelse. I det følgende vil vi forsøge at kortlægge de væsentligste fordele og ulemper ved automatiseret biometrisk genkendelse.

### 4.4.1 Kriterier for et biometrisk system

I denne teknologikritik er det væsentligt at skelne mellem individet og samfundet, fordi personlige interesser konflikterer med samfundets interesser. Der er desuden to forskellige anskuelser af termen sikkerhed. Der er sikkerheden for, at der er sket den rigtige identifikation og sikkerhed for, at den biometriske data ikke bliver kompromitteret.

Skulle man overgå til systemer baseret på biometrisk genkendelse ville dette system have nogle høje krav at leve op til. Netop fordi der er nogle begrænsninger indenfor biometri som man ikke har med traditionelle indentifikationsmetoder, skal dette system være meget mere sikkert. Eksempelvis må selve systemet ikke være sårbart for uvedkommende, eller på anden måde omgås uden at man afgiver korrekt identifikation. Da man umuligt kan skifte biometriske identifikatorer må systemet **aldrig** tage fejl. Hvis systemet producerer en positiv falsk, se tabel 4.12, ville hele systemet være ubrugeligt, da denne fejl ikke tolereres. Konsekvensen af dette ville være at hele systemet skal udskiftes og der skal findes et nyt.

| | **Korrekt person** | **Falsk person** |
|---|---|---|
| **Korrekt kode** | Positiv positiv | Positiv falsk |
| **Falsk kode** | Falsk positiv | Falsk falsk |

Tabel 4.12: Oversigt over positiv falsk begrebet

Med traditionelle pin-koder kan systemet ikke lave en forkert match, da det er en række tal som sammenlignes. Dvs der ikke er nogen vurdering i processen. Tager man derimod genkendelse af biometriske kendetegn vil systemet analysere, sammenligne og give et bud på om der er match. Da input kan variere fra gang til gang, vil systemet lave et veldefineret "gæt" på om der er match. Med adgangskoder er outputtet være en boolsk sandhedsværdi, hvorimod biometri giver en sandsynlighed.

Hvis man fra start har antagelsen, at inputtet er perfekt hver gang, kan man stille kravet, at der heller ikke må forefindes falske positive. Dette ville svare til at afvise en autoriseret person, hvilket naturligvis heller ikke er hensigten. Man kan under realistiske omstændigheder sige, at positive falske resultater er uacceptable, hvorimod at falske positive hovedsageligt vil være på grund af dårligt input fra brugerens side. F.eks. kunne dette være, at

fingeren ved fingeraftryksaflæsning er drejet, eller man udtværer aftrykket under aflæsning. Altså hvis input fra bruger ikke overholder en vis kvalitet.

Et system er selvfølgeligt ikke mere sikkert end dets skabere, og det kan derfor altid lade sig gøre at "knække" de fleste sikkerhedssystemer. Problemstillingen om at gøre systemet sikkert er en helt anden debat, som vi ikke finder det relevant at bringe ind i denne sammenhæng. Dog vil vi blot konstatere, at der vil blive brug for et meget højt sikkerhedsniveau, hvis den digitale repræsentation af det pågældende biometriske kendetegn bliver kendt for andre, så er det meget ubejlejeligt, omend i nogle tilfælde umuligt, at skulle lave om på kendetegnet. Sker dette så falder hele systemet til jorden og bliver ubrugeligt. Dette kan sidestilles med arbejdsgangen på CFI [19], hvor deres ord er altafgørende. Hvis der sker en fejlidentificering, en positiv falsk eller falsk positiv, så vil det have uoverskuelige konsekvenser.

### 4.4.2 Konflikter mellem individets og statens interesser

Amerikanerene er, traditionelt set, modstandere af enhver form for registrering. De diskuterer stadig deres ret til at beskytte deres egen ejendom. Derfor har man kunne forvente, at de ikke vil acceptere biometrisk identifikation. Men 11. September 2001 rykkede den generelle holdning mod en accept af større grad af registrering. Det mest markante eksempel på biometrisk scanning er et overvågningskamera, der hele tiden aflæser de forbipasserendes ansigter og sammenligner dem med et register af kriminelle, teorister ol. Dette er en diskret form for scanning, da den er 100% automatiseret, men det er faktisk en meget høj grad af overvågning. Eksperter har kunnet konkludere, at to af de terrorister der kaprede fly 11. september 2001, kunne have været stoppet, inden de boardede flyet hvis ansigtsscannere var blevet anvendt i lufthavnen [20]. Sådanne argumenter har tilsyneladende haft stor inflydelse på den generelle amerikanske holdning. Denne holdning forventer vi vil brede sig til resten af den vestlige verden.

I dagligdagen sker konflikter mellem individet og statens interesser hyppigt. Biometri kan som nævnt i kapitel 3.1.1 bruges til adgangskontrol. Hvis dette bliver alment brug, kan et centralt register følge en enkelt borger daglige færden. Som vi så i tabel 4.7 og 4.8, var det et fåtal, der i høj grad var bekymrede over denne fare. De fleste begrundede deres mening med, at det ikke ville blive aktuelt, da de enten ikke havde noget at skjule eller stolede på, at de der administrede teknologien ikke ville bruge den til overvågning.

Staten har større interesse i sådan form for overvågning. En kontrollerende instans vil klart kunne se en fordel i at kunne bevise, hvor en kriminel eller mistænkt har befundet sig på visse tidspunkter på dagen. Dette skaber en konflikt med nogle menneskers følelse af personlig frihed, og dermed åbner det klassiske spørgsmål: Hvem skal holde øje med dem der overvåger?

Hvad er forskellen på at indsamle informationer til et personligt pas

og indsamle informationer til biometrisk validering? Der er en lille, men ikke ubetydelig forskel, som består af det skillerum, der er mellem personen og identifikationsmidlet[35][side 301]. Et pas holder informationen externt fra individet, individet kan glemme passet. Hvorimod biometrisk validering sikrer at individet ikke kan glemme passet, han bliver faktisk lig med passet. For nogle personer kan det føles som et indhug på den personlige frihed.

En anden konflikt er mellem kriminelle og staten. Kriminelle vil naturligvis ikke være glade for større overvågning. Ligeledes kan det argumenteres at staten gerne vil overvåge tidligere forbrydere. Hvorimod disse formentlig ikke vil overvåges resten af deres liv, pga en tidligere fejl de har begået.

| **Biometri**<br>Fordele | **Adgangskoder**<br>Ulemper |
|---|---|
| Man kan ikke glemme sine biometriske kendetegn | Man kan glemme sine adgangskoder |
| Et biometrisk kendetegn er komplekst og man skal ikke huske på det | Komplekse adgangskoder er svære at huske |
| Et biometrisk kendetegn beviser at et individ har været på stedet | Et password beviser at der er et individ der har kendt passwordet |
| En kombination af flere biometriske kendetegn til verificering giver højere sikkerhed | To eller flere adgangskoder giver ikke væsentlig højere sikkerhed |
| Med biometri har man et genetisk pas, som man er identisk med, man kan altså ikke undslippe sin identitet | Personer kan skaffe falske pas og derved snyde myndigheder |
| Kan ikke glemmes i en fraværende tilstand | Kan glemmes eller fejlindtastes pga. stress og fraværende tilstand |

Tabel 4.13: Fordele ved biometri fremfor adgangskoder

Meningerne overfor anvendelsen af biometrisk verifikation er delte. Nogle mener ikke, at det har de store fordele fremfor nuværende, traditionelle verifikationsmetoder. Andre taler for implementationen af den nye teknik og siger at denne vil afhjælpe sikkerhedsproblemer, nedskære omkostninger eller på andre måder være behjælpelig i vores dagligdag. Udover fordelsmæssige overvejelser omkring den nye teknologi, spiller etik også ind i billedet. Vil den nye teknologi blive brugt ansvarligt og korrekt, eller i stedet til at krænke folks personlige frihed? I tabel 4.13 og 4.14 ses en opstilling af fordele og ulemper ved biometrisk genkendelse og adgangskoder.

| **Adgangskoder**<br>Fordele | **Biometri**<br>Ulemper |
|---|---|
| Bliver sikkerheden kompromitteret så kan man få et nyt password | Bliver sikkerheden kompromitteret så er det ikke til at få et nyt biometrisk kendetegn |
| Med et centralt register kan man kun konstantere at en kode er blevet brugt | Med et centralt register bliver det nemmere at overvåge et individ |
| Man kan have flere identiteter til f.eks arbejde og personlige indkøb | Der er kun mulighed for en identitet |

Tabel 4.14: Fordele ved biometri fremfor adgangskoder

### 4.4.3 Fordele

En af de største fordele ved automatiseret biometrisk genkendelse er kompleksiteten i det biometriske kendetegn. Problemet ved adgangskoder i dag er at jo mere komplekse de bliver, jo sværere bliver de også at huske. F.eks. vil det almene individ $A$ ofte vælge sin fødselsdato, barns navn eller lignende som adgangskode. Dette gør det nemt for en angriber $E$ at bryde koden og udgive sig for at være $A$ . Såfremt $E$ ikke har mulighed for at tilegne sig den elektronisk lagrede udgave af adgangskoden, så afhænger sikkerheden udelukkende af kompleksiteten af adgangskoden.

Egenskaberne for de biometriske data er, at den er kompliceret, dvs. der er mange detaljer og derfor store mængder af informationer. Ved brug af biometrisk scanning, er der ikke noget at huske på, du bærer altid dine biometriske data. Forskellige grader af mentalt fravær er i dag grund til mange falske positiver. Disse vil med biometrisk scanning i større grad kunne elimineres. Fordelen for individet er altså, at han ikke behøver koncentrere sig om at huske mange adgangskoder, hvorimod fordelen for den kontrollerende instans er at der er større sikkerhed for, at den person der bliver identificeret rent faktisk er en positiv positiv.

Flere forskellige biometriske data kan med fordel kombineres i forbindelse med en identificering, dvs hvis man kombinerer f.eks. stemmegenkendelse med iris scanning, er der større sandsynlighed for en positiv positiv. Dette ser man ikke med adgangskoder, hvor flere forskellige adgangskoder ikke højner sikkerheden væsentligt, men derimod forvirrer eller er til gene for individet.

### 4.4.4 Ulemper

Flere identiteter baseret på biometrisk kendetegn er ikke mulig. Et individ kan i dag have en identitet til brug på arbejde og en til brug ved postforsendelser til hjemmet. Med biometrisk genkendelse bliver det ikke muligt, fordi der kun er en central identitet. Dette kan understreges ved f.eks. login til systemer. Selvom folk kan have flere forskellige login navne, vil de alle have samme adgangskode, og derfor være unødvendige. Dette åbner op for en alvorlig potentiel sikkerhedsrisiko ved biometri fremfor traditionelle adgangskoder. Eksempelvis hvis person $A$ har adgang til tre forskellige systemer baseret på biometriske kendetegn. Lykkedes det så for angriberen $E$ at forfalske hans kendetegn, giver dette ikke kun adgang til et, men alle systemer hvor $A$ har adgang, forudsat at $E$ ved hvilke systemer $A$ har adgang til.

Hvor et system bygget på adgangskoder kan generere nye adgangskoder i tilfælde af at systemets data bliver kompromitteret, kan det ikke lade sig gøre at udskifte de biometriske data.

Et skift fra nuværende teknologi, baseret på pin-koder og adgangskoder, til biometrisk identifikation vil skabe nye problemer, da disse systemer endnu er både forholdsvis dyre og tidskrævende at skifte til. Skulle man f.eks. skifte fra pin-koder til biometrisk identifikation i en bank, skulle man både ombygge automater og bankens systemer. Yderligere skulle folk have muligheden for at bestemme om de ville bruge biometri, eller fortsætte med pin-koder. Implementering vil dog efter al sandsynlighed ske langsomt og løbende i dette årti.

## 4.5 Opsumering

Efter at have undersøgt biometrisk genkendelse og forhørt os om den almene danske borgers holdninger hertil, har vi fundet ud af, at hovedparten er åbne over for implementeringen af biometri i deres hverdag. Vi kan ligeledes konkludere, at den generelle holdning til registering af personlig data er, at den danske borger ikke frygter, at biometri vil skabe et overvågningssamfund. Manglerne med hensyn til juridske grundlag for biometri kan skabe unødvendig angst i befolkningen. Dette mener vi fører frem til en række succeskriterier, som vi mener skal opfyldes før at biometrien bliver implementeret i den danske dagligdag. Vi mener at et juridisk grundlag skal udarbejdes. Dette skal behandle alle aspekterne omkring anvendelse af biometri. En anden af de ting som biometrien står og falder på, er om sikkerheden kan garanteres. Vi mener at sikkerheden for at ens biometriske data ikke falder i forkerte hænder og sikker identifikation er væsentlige faktorer, der spiller ind i billedet. Hvis der findes en biometrisk løsningsmodel, er det vigtigt at de fordele der er i modellen opvejer ulemperne. Envidere skal befolkningen være villig til at lave denne overgang. Dette kan opnås ved at man sørger for at lave

en langsom overgang til biometri, for at vænne befolkningen til brugen af de nye systemer. Endnu en ting, som bør undgås, er problemer i systemets overgangsfase, da dette kan skabe mistillid til modellen og dermed konflikt. Vi er selvfølgelig bevidste om at der altid vil være problemer i en opstartsfase, men vi mener at det er vigtigt at disse problemer bliver løst hurtigt. Når disse ting bliver opfyldt, mener vi at biometrien har en lys fremtid som afløser for pin-koder og adgangskoder.

# Kapitel 5

# Forskellige billedformater

Vi har i tidligere kapitler set hvad biometri kan bruges til, men for at teknologien kan udbredes skal problemet omkring komprimering af fingeraftrykkene, som vi omtalte i afsnit 2.1.2, løses. Derfor vil vi i de følgende kapitler kigge på metoder til at komprimere data. Disse data vil i vores tilfælde være gråskala fingeraftryk. Derfor vil vi først kigge på forskellige billedformater og derefter komprimering af billeder.

## 5.1 Billedformater generelt

Vi skal som sagt benytte et billedformat til at opbevare informationer omkring det originale fingeraftryk, så vi kan lave sammeligning mellem vores eget komprimerede billede og det originale aftryk. Dette format skal være i stand til at gemme det originale aftryk uden betydelig forringelse af billedkvalitet. For at finde et format med denne egenskab, har vi undersøgt følgende billedformater.

- JPEG - Joint Photographic Experts Group Format.
- TIFF - Tag Image File Format.

Vi vil i dette kapitel kort beskrive de forskellige billedformater og derefter forklare væsentlige dele af TIFF formatet, eftersom vi er kommet frem til, at dette format lagrer det originale aftryk i den kvalitet vi ønsker uden at komprimere. Vi vil også komme ind på vores eget format, som vi vil benytte til at gemme det komprimerede fingeraftyk. Så vi kan lave en test på om aftrykket fylder mindre efter komprimeringen.

## 5.2 JPEG

JPEG er et billedformat lavet til at optimere billeder til brug på Internettet og i sammmenhæng, hvor der ønskes en lille filstørrelse. JPEG for-

matet bruger normalt en kombination af forskellige komprimeringsteknologier for at opnå en meget høj komprimeringsrate, samtidig med at man stadig bibeholder en god billedkvalitet. JPEG har fået sin store udbredelse sammen med Internettet, og er derfor blevet det mest benyttede filformat til udveksling af billeder over Internettet. På nuværende tidspunkt kan langt de fleste programmer læse og skrive i JPEG formatet.

JPEG formatet er bedst egnet til billeder med stor farvedybde[1], da det er her, at man kan udnytte de ekstreme gode komprimerings egenskab bedst muligt. JPEG benytter en farve repræsentation på 3 bytes til at gengive billeder. Denne farvegengivelse er lavet efter RGB skalaen[2] og kan derfor vise farver i de enkelt pixel meget præcist. Der findes situationer, hvor JPEG formatet ikke er velegnet. Dette er når man har billeder med meget skarpe kanter og nuancer mellem to kontrastfarver. Det er derfor vi fravælger dette format til behandling af fingeraftryk.

## 5.3 Raster image

Raster image er nærmere en metode, end det er et billedformat. Incitamentet til at vi har dette afsnit med er, at TIFF bygger på raster image data. Et grafisk raster billede er en datafil, som består af et rektangulært område af pixels på en computermonitor eller en anden form for billed fremviser. Hver pixel i dette billede består af en kombination af farverne rød, grøn og blå, som er grundfarver i skalaen. Altså arbejder raster billeder med RGB farveskalaen.

Kvaliteten af et billede, gemt ved hjælp af raster image data, er bestemt af det samlede antal pixels som billedet består af, og hvor mange bits pr. pixel man har scannet billedet i. Tager vi f.eks et billede gemt i 640 x 480, vil dette indholde 307.200 pixels. Dette vil se grynet og smudset ud i forhold til det samme billede i 1024 x 768, som består af 786.432 pixels. Man skal bare huske jo højere opløsning jo mere fylder dataene også, når de skal opbevares.

Dette betyder at når først et billede er scannet ind og gemt med raster image data metoden, kan billedet ikke scaleres uden at kvaliteten bliver ringere. Når man først har digitaliseret et billede med x antal pixels, kan man ikke ændre på dette. Også selv om man benytter en bedre monitor eller et grafikkort med større opløsning.

## 5.4 TIFF

Alt information i dette afsnit er baseret på [1]. TIFF formatet er lavet specielt til at håndtere billeddata fra f.eks en scanner eller lignede udstyr.

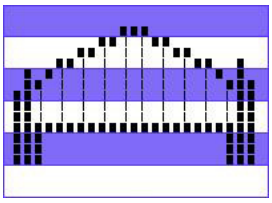
[1] F.eks. 24bit

[2] Rød, Grøn og Blå farveskalen er generelt meget benyttet indenfor digital billedbehandling

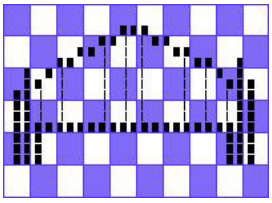
De grundlæggende egenskaber ved TIFF er som følgende:

- TIFF understøtter bi-level (ren sort/hvid), gråskala, palet farver[3] og RGB farver.
- TIFF understøtter en rækker forskellige komprimeringsteknikker, som man kan bruge til sine applikationer.
- TIFF er et meget rigt format med meget billedinfomation og højt detalje niveau.
- TIFF er ikke *afhængigt* af et bestemt hardwaresystem så som scanner og grafikkort.
- TIFF er uafhængigt af operativsystem.

Vi har valgt at fokusere på TIFF formatet i forhold til dets struktur i gråskalaen, efter som det er dette vi skal bruge i projektet. Vores billeder er opbygget i striber og ikke tern. Dette gør vi fordi vi så kan læse billedet stribe for stribe, hvilket vi ser som fordelagtigt, i modsætning til at lave en algoritme der læser billedet tern for tern. Disse to typer er illustreret på figur 5.1 og figur 5.2.



Figur 5.1: Billede med striber



Figur 5.2: Billede med tern

### 5.4.1 Gråskala billede i TIFF

Når man bruger TIFF til at bearbejde gråskala billeder kan dette gøres med 4- eller 8-bit, hvilket svarer til 16 eller 256 grå farver.

[3]Brugerdefineret farveskala

### Strukturen i TIFF gråskala

Et TIFF billede består af en række byggesten, hvor den første er IFH (Image File Header). Denne består af 8 bytes, som indholder følgende informationer:

- Byte 0-1
  Giver information om rækkenfølgen bytene skal læse i, i resten af filen.
- Byte 2-3 Indeholder tallet 42, som angiver at det er en TIFF fil.
- Byte 4-7 Angiver hvor i filen byteoffsetet befinder sig altså hvor billededata starter.

Derefter kommer IFD (Image File Directory), som angiver hvad billedet indholder og infomation om billedet, som f.eks dimension. Der findes en lang række egenskaber, som er gemt her, men da vi kun kigger på gråskala, vil vi kun beskrive de funktioner, som er nødvendig for dette. Komplet dokumentation kan findes på [1].

| Tagname | Decimal | Hex | Type | Value |
|---|---|---|---|---|
| ImageWidth | 256 | 100 | SHORT or LONG | |
| ImageLength | 257 | 101 | SHORT or LONG | |
| BitsPerSample | 258 | 102 | SHORT | 4 or 8 |
| Comperssion | 259 | 103 | SHORT | 1 or 32773 |
| PhotometricInterpretation | 362 | 106 | SHORT | 0 or 1 |
| StripOffsets | 273 | 111 | SHORT or LONG | |
| RowsPerStrip | 278 | 116 | SHORT or LONG | |
| StripByteCounts | 279 | 116 | LONG or SHORT | |
| XResolution | 282 | 11A | RATIONAL | |
| YResolution | 283 | 11B | RATIONAL | |
| ResolutionUint | 296 | 128 | SHORT | 1,2 or 3 |

For at disse skal give mening kommer her en forklaring på de enkelte tags:

**ImageWidth**
Angiver hvor bredt billedet er, altså hvor mange pixels en række indholder.

**ImageLength**
Angiver højden, altså hvor mange pixels en kolonne indholder.

**BitsPerSample**
Angiver hvor mange farver hver pixel kan antage; 4 eller 8 bit.

**Comperssion**
Angiver om billedet skal komprimeres, og hvordan. Da vi ikke skal bruge komprimering sætter vi denne til 1 hvilket betyder ingen koprimering.

**PhotometricInterpretation**
Bruges til at angive hvilken vej farveskalaen skal løbe. Om den skal gå fra hvid til sort eller sort til hvid. Hvis vi sætter PhotometricInterpretation til 1, vil den løbe fra sort til hvid og sættes den til 0, vil det være hvid til sort. Dette er ikke uvæsentligt da det ændrer på hvilken farve værdierne repræsenter.

**StripOffsets**
Angiver hvor det første billeddata befinder sig i filen.

**RowsPerStrip**
Angiver hvor mange linier billeddata strækker sig over, det anbefaldes ikke at laver rækker over 8Kb længde.

**StripByteCounts**
Angiver hvor mange byte en stribe er efter en eventuel komprimering.

**XResolution**
Antallet af pixel per ResolutionUnit i bredden.

**YResolution**
Antallet af pixel per ResolutionUnit i højden.

**ResolutionUnit**
Angiver hvilken måleendhed billedet er angivet i, hvor 1 er ingen, 2 er tommer og 3 er centimeter.

Der findes mange flere tags end overstående, men disse skal være i IFD for at fremstillet et billede under TIFF formatet. Resten af en TIFF fil består af data, som angiver farverne til de enkelte pixels i billedet.

## 5.5 FPD

Da vi ikke har været i stand til at finde et standard format, som kan bruges til at opbevare fingeraftryks informationer. Har vi bestemt os for at lave vores eget format, som vi kalder FPD formatet[4]. Dette bliver beskrevt nærmere i afsnit 7.3.5.

[4]Finger Print Data.

## 5.6 Opsummering

I dette kapitel har vi beskrevet billedformaterne JPEG og TIFF. Vi har fundet ud af, at det format vi vil bruge, til det originale fingeraftryk, skal være TIFF, da dette virker som det bedst egnede.

# Kapitel 6

# Komprimeringsteknikker

I det følgende vil vi beskrive de metoder vi bruger til at komprimere med. Vi starter med en grundlæggende komprimering for derfor at betragte Huffman kodning. Derefter følger en beskrivelse af splines med udgangspunkt af i fingeraftryk som eksempel. Til sidst i kapitlet er der en længere gennemgang af grundprincipperne i wavelet komprimering.

## 6.1 Grundlæggende Komprimering

Dette afsnit er baseret på [10] Efterhånden som datamængderne bliver større og større bliver behovet for komprimering også større. Dette har resulteret i, at der forekommer mange forskellige komprimeringsværktøjer på marked så som 'rar', 'tar', 'bz2', 'gz', 'zip' osv. Algoritmerne i komprimeringsprogrammerne er forskellige, fra program til program, og effektiviteten af disse afhænger af, hvad det er man vil komprimere.

En af metoderne til at komprimere med er Lossless komprimeringsmetoden. Denne metode søger efter ord og sætninger, der forekommer igen i en data fil. Disse gentagelser repræsenteres ved hjælp af tal og der opbygges en "ordbog", hvor tallene afspejler de forskellige gentagelser. Derefter modificerer komprimeringsalgoritmen teksten udfra ordbogen og det resulterer i en tekst bestående af tal og bogstaver. Uden ordbogen giver den komprimerede tekst ingen mening. Dette medfører ofte en mindre filstørrelse. Som eksempel tager vi sætningen:

*fiskere fisker kun om dagen, for om dagen fisker fiskere bedst.*

Ved at se på sætningen kan vi observere, at der er nogle ord, som optræder flere gange. For at komprimere denne sætning behøver man kun at betragte selve sætningen, da størrelsen af sætningen er overkommelig, og selv danne en ordbog. Hvis det havde været en en hel bog eller artikel i stedet vil man anvende en komprimeringsalgoritme. Ud fra sætningen danner vi ordbogen:

*1. fisker*

*2. om dagen*

Udfra ordbogen omskriver vi så sætningen til:

*1e 1 kun 2, for 2 1 1e bedst.*

Nu kan vi sammeligne størrelsen på den oprindelige sætning og den komprimerede. Den oprindelige sætning bestående af 63 tegn med mellemrum og den komprimerede sætning bestående af 29 tegn plus ordbogens 16 tegn, hvilket sammenlagt giver 45 tegn. Vi har altså fået reduceret den oprindelige sætning med 18 tegn. Hvis vi antager at et tegn svarer til en byte, så vil vi have opnået en komprimering på 18 bytes fra de oprindelige 63 bytes. Dette giver en komprimeringsrate på

$$\begin{aligned}(63-45)/63 &= 0,286 \\ &= 28,6\%\end{aligned}$$

Det er muligt at bestemme forskellige ordbøger til den samme komprimeringsprocces. En ordbog kan give en komprimeringsrate på 40% mens en anden ordbog, måske kan give en komprimeringsrate på 65%. En god udvælgelse af ordbog er nøglen til en effektiv komprimeringsrate.

Da billede, musik, videosekvenser og lignende filtyper ikke altid har særlig mange gentagelser virker Lossless metoden derfor ikke særlig godt på denne type filer. Lossless metoden kan dog benyttes, men vil ikke nødvendigvis give en god komprimeringsrate. Derfor er der behov for at anvende andre metoder til at komprimere videosekvenser, billede og lyd. Til formålet bruger man noget, der kaldes Lossy komprimering. Lossy komprimering fungerer ved at der sættes en grænse[1] for hvor meget information, der er behov for. Alt under grænsen bliver så fjernet. Jo højere grænsen er placeret jo bedre bliver komprimeringen. Ud fra disse kriterier udvælger man grænsen ud fra behovet for kvaliteten og komprimeringen. Det er muligt at komprimere et billede til at fylde næsten ingenting, men det giver anledning til misvisende information, da man så ikke kan se hvad det forestiller.

Vi vil nu se på Huffman koding, som er en komprimerings metode, af typen Lossless. Vi vil benytte denne metode senere hen til komprimering af vores eget billedformat, som bliver beskrevet i afsnit 5.5

[1] Også kaldet threshold

## 6.2 Huffman kodning

Følgende er hovedsageligt baseret på kilderne [34], [12], [7] og [13]. Da vi i vores eget billedformat, se afsnit 5.5, gerne vil have filen til at fylde så lidt som muligt, har vi valgt at komprimere filen ved hjælp af Huffman koding. Hvis man gemmer data i ASCII[2] tegntabellen fylder alle tegn det samme. Dvs.at man har en fast længde til at repræsenter et tegn. Vores problem er, hvis repræsentationen af tallet 0 og 255 (0 og 255 er henholdsvis repræsentationen af sort og hvid) fylder det samme som alle andre tal, vil vi ikke opnå en væsentlig komprimering ved udelukkende at bruge wavelets[3] Derfor vil vi bruge Huffmans metode til at repræsentere disse tal for at opnå en bedre komprimering.

I 1950'erne stillede Claude Shannon og R.M. Fano sig selv spørgsmålet: Ville det ikke være bedre, hvis hyppigt fremkommende tegn havde en mindre længde og mindre hyppige en større længde?. De udviklede den første komprimeringsalgoritme efter dette princip, i 1950. Allerede i 1951 udgav D.A. Huffman en artikel hvori han forbedrede Shannon-Fano algoritmen, og Huffman algoritmen vandt hurtigt indpas [12].

### 6.2.1 Princippet bag Huffman kodning

I den traditionelle ASCII kode vil hvert tegn bestå af 8 bit, da man i ASCII bruger en fast størrelse for alle tegn. Man kan derfor i ASCII, med den faste størrelse af 8 bit pr. tegn, have 256 tegn, hvilket svarer til $2^8$ tegn. Hvorimod Huffman kodning tildeler hvert tegn et variabelt antal bit afhængig af deres hyppighed. Derved opnås betragtelige reduktioner i størrelsen af den samlede fil.

Udgangspunktet er at en fil består af en række tegn, hvor hvert tegn består af en samling bits. Antallet af bits, der skal repræsentere et tegn, afhænger af hvor mange tegn, der ialt skal repræsenteres. F.eks. vil man med 1 bit kunne repræsentere 2 tegn. Et tegn for værdien 0 og et for værdien 1. Ligeledes kan man med 2 bit repræsentere 4 tegn, hvilket svarer til $2^2$, som kan ses på nedenstående tabel.

| | |
|---|---|
| første tegn: 00 | andet tegn: 01 |
| tredje tegn: 10 | fjerde tegn: 11 |

Mere generelt kan $2^n$ tegn repræsenteres ved brug af $n$ bit.

[2] American Standard Code for Information Interchange.

[3] Efter wavelet transformationen er der ikke opnået nogen særlig komprimering, men outputfilen består hovedesagligt af 0 værdier, se afsnit 6.5.

### 6.2.2 Huffman kodning vist ved eksempel

Vi vil her vise et simpelt eksempel på hvordan Huffman komprimering kan reducere størrelsen på en fil. Vores fil vil i eksemplet bestå af en række tegn:

AAAAAAAAAABBBBBBBBCCCCCCDDDDDEE

Som vi kan se dukker nogle tegn op flere gange end andre. Her følger en liste over de forskellige tegns hyppighed, senere omtalt som tegnets "frekvens"[4]:

$$\begin{array}{ccc} \text{A} = 10 & \text{B} = 8 & \text{C} = 6 \\ \text{D} = 5 & \text{E} = 2 & \end{array}$$

Som i normal ASCII kode vil filen fylde 8 bit pr. tegn, hvilket giver:

$$10 \cdot 8 + 8 \cdot 8 + 6 \cdot 8 + 5 \cdot 8 + 2 \cdot 8 = 248 \text{ bits}$$

Bruger vi Huffman kodning til at komprimere dataene med, ser det således ud. Vi har allerede talt frekvenserne på tegnene op, så skal man blot sætte hvert tegn til en værdi. Det vil sige at hvis vi sætter værdierne til:

$$\begin{array}{ccc} \text{A} = 11 \text{ (2 bit)} & \text{B} = 10 \text{ (2 bit)} & \text{C} = 01 \text{ (2 bit)} \\ \text{D} = 001 \text{ (3 bit)} & \text{E} = 000 \text{ (3 bit)} & \end{array}$$

Vi ganger herefter, som med ASCII, antal bit, med hvert tegns "frekvens", og adderer:

$$10 \cdot 2 + 8 \cdot 2 + 6 \cdot 2 + 5 \cdot 3 + 2 \cdot 3 = 69 \text{ bits}$$

Som det fremgår har vi repræsenteret samme mængde tegn, men bruger mindre plads.

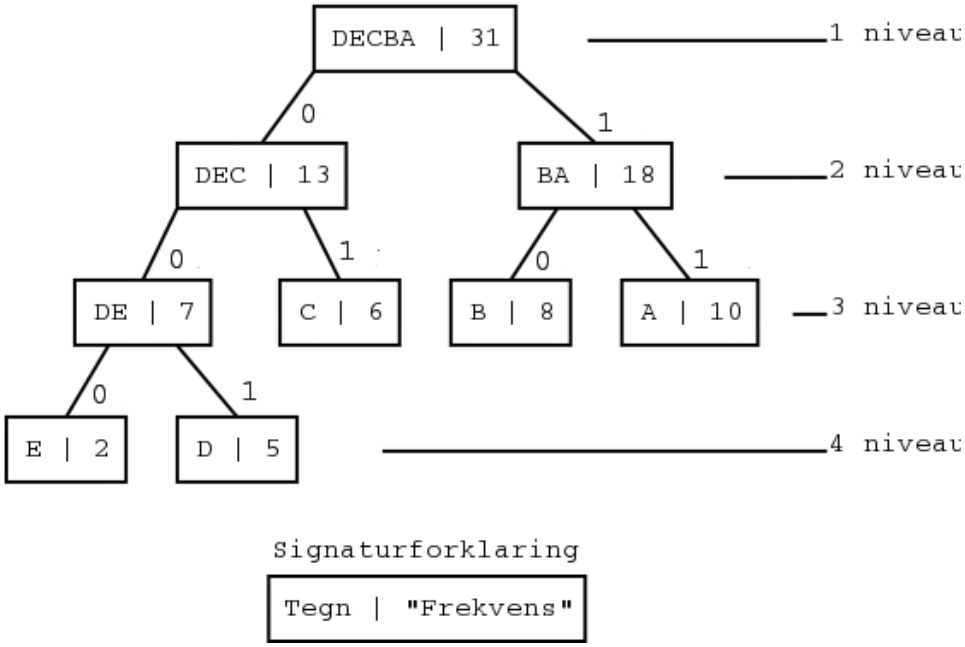
### 6.2.3 Træstrukturer

Tildelingen af bitværdier til tegn kan også opstilles som en binær træstruktur. Dette fungerer således at man starter med at have et knudepunkt, der maksimum har 2 "børn". Hvert barn har, ligesom sin forælder, mulighed for at have to børn, og således fortsætter træet.

Opbygningen af en Huffman træstruktur foregår bagfra. Man opstiller sin liste over tegn, og deres frekvenser. Udfra denne opremsning følger man denne løkke:

1. Tag de to tal med laveste frekvens.
2. De to tal skal danne et børnepar.

[4] reel frekvens er f.eks. 10/31, men bruges her til at tælle antal forekomster af tegnet




Figur 6.1: Huffman træstruktur opbygget af vores eksempel

3. Deres forælder har summen af børnenes frekvenser som frekvens.
4. Forældren erstatter nu de 2 børns plads i listen.
5. Man begynder forfra med de 2 laveste frekvenser igen.
6. Løkken slutter ved, at man kun har en samlet sum af frekvenser tilbage i listen. Denne bliver startknuden i træet.

Ved denne metode får man ikke kun opbygget en træstruktur, men faktisk også den struktur, som er mest optimal, med hensyn til filens bitvise størrelse, ifølge kilderne [13] og [12]. Vi opbygger i figur 6.1 vores tidligere eksempel som en træstruktur.

Vi har givet at D og E har de to laveste frekvenser. De danner et par af børn til forældren DE, som har deres samlede sum frekvenser. DE erstatter så de to børns plads i vores liste, og vi finder igen de to laveste frekvenser. I næste omgang bliver det C og DE som skal danner par, deres forældre er så DEC. Vi fortsætter med denne metode, indtil vi til sidst danner forældren DECBA i toppen af vores træ. Ud fra en træstruktur kan man se hvordan tildelingen af værdier til hvert tegn er fundet. Et skridt til højre giver værdien 1, og et skridt til venstre værdien 0. F.eks. vil A kræve 2 skridt til højre fra toppen og får derfor værdien 11.

Træstrukturen har yderligere den nyttige egenskab at der ikke er to tegn, som starter med den samme værdi. Når alle tegn er slutknuder, kan man ikke lave flere værdier, som har den samme start bit. F.eks. giver $B$ værdien 10. Da $B$ ikke har nogen børn kan vi ikke finde et tegn med værdien 101, eller på anden måde starter med 10. Dette er utrolig vigtigt for dekomprimeringen af vores fil. Sådanne knudepunkter, som ikke har nogen børn, kaldes også for træets blade.

Når man har oprettet en træstruktur, kan man udfra dens værdier, for hvert tegn, opstille en tabel. Denne tabel indeholder så hvert tegn, og dets værdi. Tabellen skal gemmes med i den komprimerede fil for at vi kan dekomprimere filen igen. Når vi har skrevet tabellen i filen, erstattes alle de tidligere tegn med de nye fundne værdier, og som resultat heraf fylder filen mindre.

### 6.2.4 Dekomprimering

Når vi har en komprimeret fil, med tilhørende dekomprimeringstabel, kan vi dekomprimere filen igen. Dekomprimeringen er baseret på det princip , som træet er opbygget efter. Som det kan ses på figur 6.1 er alle vores tegn bladene på træet. Vi tager nu blot vores række af bitværdier og begynder at sammenligne bit for bit med tabellen over tegn. F.eks. vil vores fil starte med 111111... hvilket i den originale fil er en række A'er. Først ser vi et 1. Dette kan kun blive 2 tegn, A eller B, da ingen andre starter med 1. Næste bit er også 1, hvilket så snævrer mulighederne ned til kun en mulig tegnrepræsentation, nemlig A. Derefter fortsætter man denne metode og erstatter løbende bitværdierne med korrekte tegn.

Ud fra vores afkodningsnøgle[5], kan man også genskabe træstrukturen. Dette vil i nogle tilfælde være nyttigt for overblik og dekomprimering. Man tager udgangspunkt i filens samlede antal tegn. Så tager man tegnene et efter et og erstatter disse med deres korsponderende bitværdi. F.eks. vil A kræve to skridt til højre fra roden, da hvert 1-tal kræver et skridt til højre. Man fortsætter således blot denne udskridtning med alle tegn og får derved hele træet. Bagefter kan man også optælle de forskellige "frekvenser" og således udfylde træet helt. Dette kræver, at man dekomprimerer hele filen og på ny optæller hvert tegns "frekvens". Det er utroligt vigtigt for dekomprimering at filen bliver gemt korrekt. Hvis man under oprettelsen af filen, ved en fejl, kommer til at bytte rundt på nogle bits, vil dette skabe fejl i dekomprimeringen og i alvorlige tilfælde skabe en fil der ikke ville give mening efter dekomprimering. Efter at have set på en Lossless komprimeringsmetode vil vi nu se på splines, som vil vi benytte som lossy komprimering.

[5]De tegn tabellen indeholder og tilhørende bitværdi

## 6.3 Interpolation og Splines

Vi vil bruge splines til at finde en metode, hvormed vi kan tilnærme os en linie i et fingeraftryk. Med denne metode kan vi således lave et helt fingeraftryk ved hjælp af splines. Metoden til dette finder vi ved først at kigge på lineære splines og derefter bygge videre til kubiske splines, som vi har tænkt os at bruge til at repræsentere fingeraftrykket med. Dette kapitel er baseret på [33].

### 6.3.1 Interpolation

Der er flere forskellige metoder, hvorved man kan prøve at tilnærme sig et sæt data. Interpolation er en process, der bruges til at finde en funktion eller en samling af flere funktioner for at beskrive et sæt givne data. Gennem disse data skaber man den funktion, der passer bedst til situationen. Den skabte funktion har den egenskab, at man med en forholdsvis lille afvigelse kan approksimere den oprindelige funktion ud fra et givet antal punkter. Derved skaber man en metode til at beskrive komplicerede datasæt på en enkel og effektiv måde. Hvis man skal approksimere givne data med grafen for en funktion, kan man med fordel kigge på polynomier som værktøj hertil. Weierstrass's sætning beskriver i [33][Kapitel 4, side 75] denne situation:

> Lad $f$ være en kontinuert funktion i intervallet $[a, b]$, givet et $h > 0$, så eksisterer der et polynomium $p_{N(h)}$ af grad $N(h)$ så:
>
> $$|f(x) - p_{N(h)}(x)| < h \tag{6.1}$$
>
> for alle $x$ der tilhører intervallet $[a, b]$, derfor eksisterer der en følge af polynomier således at
>
> $$||f(x) - p_n||_\infty \to 0 \quad \text{for} \quad n \to \infty \tag{6.2}$$

Sætningen viser, at der kan bestemmes en følge af polynomier $p_n$, således at forskellen mellem funktionen $f$ og det interpolerede polynomium går mod 0, hvis $n$ (angiver trinet i følgen) går mod uendeligt.

Når man har valgt polynomier som værktøj, er det af to grunde. Først, givet en kontinuert funktion, kan man med vilkårlig nøjagtighed approksimere et polynomium, så det lægger sig så præcist, som det er nødvendigt op af funktionen. Dvs. man kan opbygge en funktion ud af polynomier, der er vilkårlig tæt på den oprindelige funktion. Jo flere man tager med, desto større nøjagtighed opnår man. Da det ikke er muligt i praksis[6] at bruge et uendeligt antal polynomier til at beskrive en funktion, findes der nogle metoder til at denne proces; En af dem er spline interpolation.

[6]Her tiltænkt at man skal kunne udregne interpolationen i en computer

### 6.3.2 Spline interpolation

Der findes forskellige metoder til at approksimere et bestemt polynomium til en funktion, men et enkelt polynomium er sjældent nok og ved at dele funktionen op i intervaller ender man med en samling af polynomier, som skal sættes sammen til en beskrivelse af funktionen. Denne kombination af polynomier på bestemte intervaller, kan lede til problemer i og omkring sammensplejsningerne mellem de forskellige polynomier. Der kan fremkomme små uregelmæssigheder på den beskrivende funktion, så den mister sin kontinuitet og regelmæssighed. Ideen bag splines er at udglatte sådanne uregelmæssigheder og skabe en kontinuert "glat"funktion.

#### Lineære Splines

Lineære splines, eller splines af første grad, er en simpel udgave af spline interpolation. Lineære splines skal være kontinuerte og består af rette liniestykker. Givet at vi har nogle kendte punkter i intervallet $[a, b]$ beskrevet ved:

$$a = x_0 < x_1 < \ldots < x_m = b$$

og vi vil have en lineær spline $s$, således at et givet punkt i splinen svarer til et givet punkt i funktionen, så kan den linære spline mellen to punkter findes. For at gøre dette skal $s$ være et polynomium af grad et og gå gennem punkterne $(x_k, f(x_k))$ og $(x_{k+1}, f(x_{k+1}))$. Liniens ligning for en ret linie gennem førnævnte punkter er

$$y = f_k + \frac{f_{k+1} - f_k}{x_{k+1} - x_k}(x - x_k) \tag{6.3}$$

hvor vi i stedet for $f(x_k)$ skriver $f_k$ og $k = 0, 1, \ldots, m$.

Dette udtrykt med vores termer er

$$s_k(x) = a_k + b_k(x - x_k) \tag{6.4}$$

hvor

$$a_k = f_k$$

og

$$b_k = \frac{f_{k+1} - f_k}{x_{k+1} - x_k} = f[x_k, x_{k+1}]$$

Vi vil nu vise anvendelsen af ligning 6.3 med et eksempel.

#### Eksempel 1: Liniær spline

Vi vil finde den lineære spline der går gennem punkterne i tabel 6.1

| $x$ | 0 | 3 | 4 | 5 |
|---|---|---|---|---|
| $f(x)$ | 0 | 1 | 2 | 4 |

Tabel 6.1: Funktionsværdier

Hvor $a = x_0 = 0,\ x_1 = 1,\ x_2 = 2,\ x_3 = 4 = b$ og

$$a_0 = f_0 = 0$$
$$a_1 = f_1 = 1$$
$$a_2 = f_2 = 2$$

og

$$\begin{aligned} b_0 &= f[x_0, x_1] = \frac{1-0}{3-0} = \frac{1}{3} \approx 0.33 \\ b_1 &= f[x_1, x_2] = \frac{2-1}{4-3} = 1 \\ b_2 &= f[x_2, x_3] = \frac{4-2}{5-4} = 2 \end{aligned}$$

Hvilket giver os

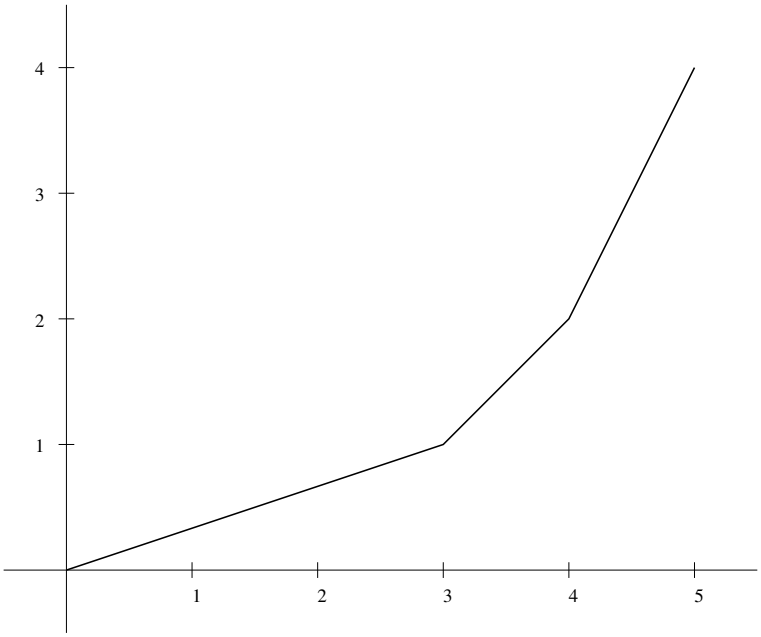
$$s(x) = \begin{cases} 0 + 0.33(x-0) & = 0.33x & ,0 \leq x \leq 3 \\ 1 + 1(x-1) & = x & ,3 \leq x \leq 4 \\ 2 + 2(x-2) & = 2x - 2 & ,4 \leq x \leq 5 \end{cases}$$

### Kubiske splines

Da vi ikke kan nøjes med at repræsentere et fingeraftryk ved hjælp af lineære splines uden at have mange punkter, har vi valgt at kigge på kubiske splines, der er splines af tredje grad. Disse er bedre til at tilnærme sig liniestykker, der ikke er rette. Vi vil have en "glat" funktion der går gennem $m + 1$ punkter. Badsberg siger i [2][Udledning] at:

> Erfaringen viser, at hvis m (redigeret, var n) er mere end ca. 5, kan man få et polynomium, der oscillerer voldsomt, selv om punkterne ligger pænt.

Derfor er det hensigtsmæssigt at sammensætte en funktion af en række polynomier af lavere grad. Dermed kan man lettere opnå en glat overgang fra det ene polynomium til det andet. Vi ønsker også at hver spline i de $m$ intervaller højest repræsenteres af et polynomium af tredje grad. For at sikre sig at funktionen gennem punkterne bliver glat, gælder der, at den skal være to gange differentiabel, og de afledte af anden orden skal være kontinuerte.




Figur 6.2: Grafen for den lineære spline i eksempel 1

**Eksempel på naturlig kubisk spline**

Naturlig kubisk splines er en metode at beregne splines på. Det er denne metode vi har valgt at benytte. Vi vil gennemgå et eksempel på disse og vise de nødvendige formler til at udføre beregningerne[7]. Vi vil lave en tilnærmelse af funktionen $f(x) = x^{-3} + \ln x$ for $x > 0$ ved hjælp af naturlige kubiske splines og det givne datasæt i tabel 6.2

| $x$ | 0.75 | 1 | 3 | 7 | 9 |
|---|---|---|---|---|---|
| $f(x)$ | 2.08 | 1 | 1.14 | 1.95 | 2.20 |

Tabel 6.2: Funktionsværdier

Ligningerne for kubiske splines kan skrives på formen:

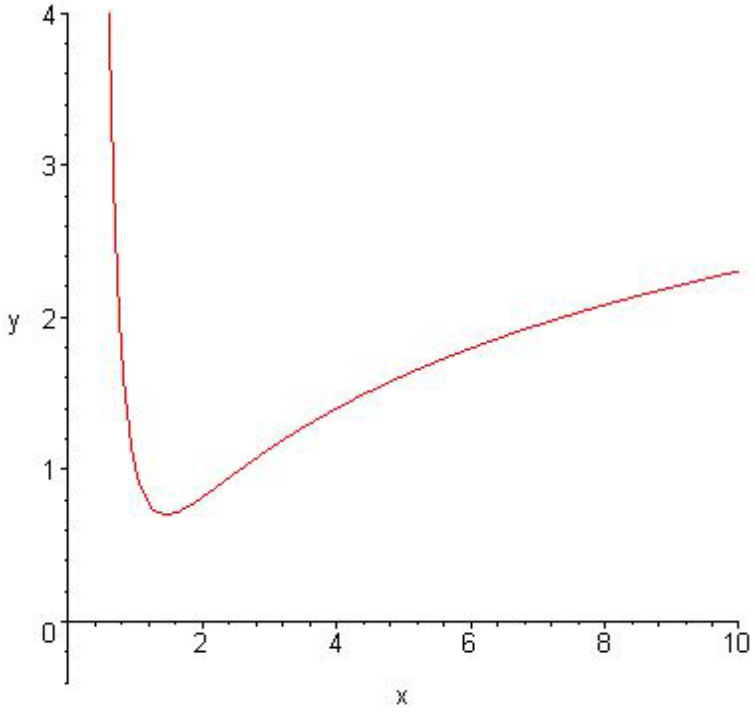
$$s_k(x) = p_i(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3, \qquad (6.5)$$

hvor

$$x_k < x < x_{k+1}, \quad k = 0, 1, ..., m - 1.$$

i intervallet $[a, b]$ hvor

[7]En nærmere udledning af disse kan ses i [33][side 104 - 106]




Figur 6.3: Funktionen $f(x) = x^{-3} + \ln x$ for $x > 0$
i intervallet $[0.5, 10]$

$$a = x_0 < x_1 < \ldots < x_m = b$$

Vi skal så have fundet koeffecienterne $a, b, c$ og $d$. $a_k$ kan findes på samme måde som i ligning 6.4, så bestemmes $a_k$ direkte:

$$a_k = f_k \tag{6.6}$$

hvilket i dette tilfælde giver:

$$a_0 = 2.08 \quad a_1 = 1 \quad a_2 = 1.14 \quad a_3 = 1.95$$

For at finde $c_k$ skal vi bruge ligningen

$$h_k c_k + 2(h_k + h_{k+1})c_{k+1} + h_{k+1}c_{k+2} = 3(\delta_{k+1} - \delta_k)), \quad k = 0, 1, \ldots, m-1 \tag{6.7}$$

Vi kan så opstille ligning 6.7 til det tridiagonale matricesystem som ses af matrice 6.8.

$$M \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{m-1} \end{bmatrix} = 3 \begin{bmatrix} \delta_1 - \delta_0 \\ \delta_2 - \delta_1 \\ \vdots \\ \delta_{m-1} - \delta_{m-2} \end{bmatrix} \tag{6.8}$$

hvor

$$M = \begin{bmatrix} 2(h_0 + h_1) & h_1 & 0 & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & 0 \\ 0 & \ddots & \ddots & \ddots \\ 0 & 0 & h_{m-2} & 2(h_{m-2} + h_{m-1}) \end{bmatrix} \tag{6.9}$$

$h_k = x_{k+1} - x_k$ og $\delta_k = \frac{f_{k+1} - f_k}{x_{k+1} - x_k}$
Vi begynder med at finde $h_k$ til at være

$$h_0 = 0.25 \quad h_1 = 2 \quad h_2 = 4 \quad h_3 = 2$$

og $\delta_k$ til at være

$$\delta_0 = -4.33 \quad \delta_1 = 0.07 \quad \delta_2 = 0.20 \quad \delta_3 = 0.12$$

Når vi har bestemt værdierne for $h_k$ og $\delta k$, sættes disse ind i matricen 6.8 og man får:

$$\begin{bmatrix} 4.6 & 2 & 0 \\ 2 & 12 & 4 \\ 0 & 4 & 12 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 13.20 \\ 0.41 \\ -0.24 \end{bmatrix} \tag{6.10}$$

Ud fra matrice systemet 6.10 kan vi finde løsningerne til:

$$c_1 = 3.18 \quad c_2 = -0.55 \quad c_3 = 0.16$$

Når vi så har fundet løsningerne til $c_k$ kan vi gå videre og finde løsningerne til $b_k$ ved hjælp af formel 6.11

$$b_k = \delta_k - \frac{h_k}{3}(c_{k+1} + 2c_k) \tag{6.11}$$

$$b_0 = -4.60 \quad b_1 = -3.80 \quad b_2 = 1.45 \quad b_3 = -0.09$$

Løsningerne til $d_k$ kan findes på tilsvarende måde ved at sætte ind i ligning 6.12

$$d_k = \frac{c_{k+1} - c_k}{3h_k} \tag{6.12}$$

hvilket giver løsningerne:

$$d_0 = 4.24 \quad d_1 = -0.62 \quad d_2 = 0.06 \quad d_3 - 0.03$$

Nu har vi løsningerne til alle koefficienterne og splinen kan så findes ved at sætte disse ind i formel 6.5 hvilket giver de følgende fire ligninger:

$$\begin{aligned} s_0(x) &= 2.08 - 4.60(x - 0.75) + 0(x - 0.75)^2 + 4.24(x - 0.75)^3 \\ s_1(x) &= 1 - 3.80(x - 1) + 3.18(x - 1)^2 - 0.62(x - 1)^3 \\ s_2(x) &= 1.14 + 1.45(x - 3) - 0.55(x - 3)^2 + 0.06(x - 3)^3 \\ s_3(x) &= 1.95 - 0.09(x - 7) + 0.16(x - 7)^2 - 0.03(x - 7)^3 \end{aligned}$$

Ligningerne i 6.13 kan omskrives til den naturlige kubiske spline

$$s_k(x) = \begin{cases} 2.08 - 4.60(x - 0.75) + 4.24(x - 0.75)^3 & 0.75 \leq x \leq 1 \\ 1 - 3.80(x - 1) + 3.18(x - 1)^2 - 0.62(x - 1)^3 & 1 \leq x \leq 3 \\ 1.14 + 1.45(x - 3) - 0.55(x - 3)^2 + 0.06(x - 3)^3 & 3 \leq x \leq 7 \\ 1.95 - 0.09(x - 7) + 0.16(x - 7)^2 - 0.03(x - 7)^3 & 7 \leq x \leq 9 \end{cases}$$

Denne spline giver tilnærmelsen der kan ses i figur 6.4.

Sammenligner man figur 6.4 med den oprindelige funktion i figur 6.3 kan man se, at vores spline tilnærmer sig funktionen godt. Denne kan godt blive bedre ved at lave afstanden mellem de valgte $x$ værdier mindre.

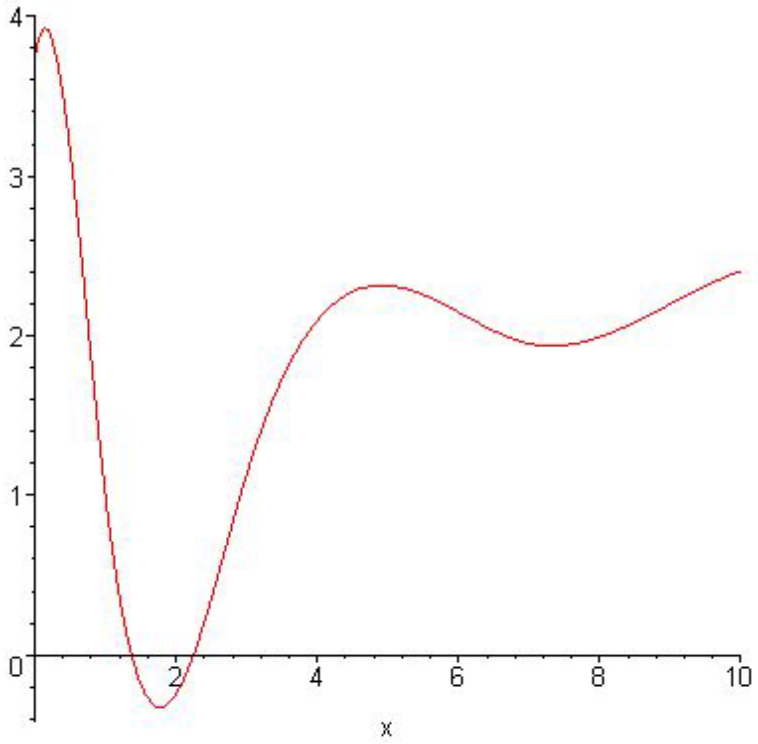
### 6.3.3 Splines og Fingeraftryk

Da vi har tænkt os at benytte kubiske splines til at beskrive et fingeraftryk, skal vi have en metode til at gøre dette. Den mest oplagte måde er at se på fingeraftrykket og vælge nogle punkter, som ved hjælp af kubiske splines kan bruges til at beskrive et karakteristisk træk ved aftrykket. Disse punkter kan man finde ved at lægge et koordinatsystem ned over mønsteret og vælge punkterne, se figur 6.5.

Ved hjælp af de punkter kan vi udregne en række splines til at beskrive mønsteret. Denne metode vil ikke kunne gengive det nøjagtige billede af fingeraftrykket, uden at skulle bruge en masse punkter, men den kan lave en acceptabel tilnærmelse.

#### Eksempel på fingeraftryk

Vi tager udgangspukt i et udsnit af et fingeraftryk og vil så tilnærme linierne ud fra de punkter man kan se på figur 6.6




Figur 6.4: Tilnærmelse af funktionen $f(x) = x^{-3} + \ln x$ for $x > 0$

| $x$ | 96 | 115 | 137 | 154 | 176 |
|---|---|---|---|---|---|
| $f(x)$ | -445 | -432 | -416 | -409 | -414 |

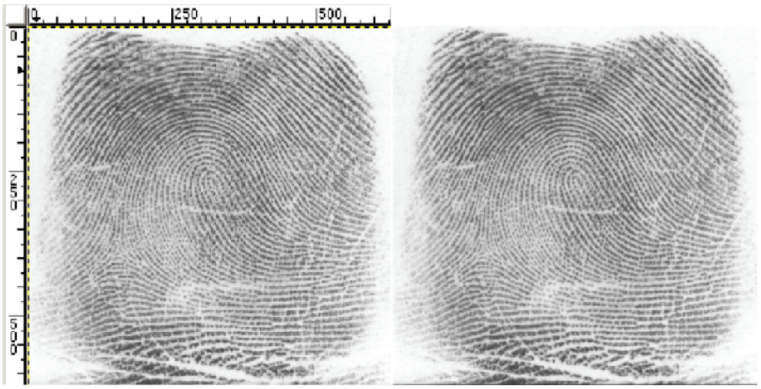
Tabel 6.3: Funktionsværdier for linie 1

med datasættene:

Med samme fremgangsmåde som i afsnit 6.3.2 kommer vi frem til følgende splines:

$$s_{l1}(x) = \begin{cases} -567.14 + 2.50x - 1.91 \cdot 10^{-2} + 6.64 \cdot 10^{-5}x^3 & ,96 \leq x \leq 115 \\ -189.10 - 7.37x + 6.67 \cdot 10^{-2}x^2 - 1.82 \cdot 10^{-4}x^3 & ,115 \leq x \leq 137 \\ 75.90 - 13.17x + 0.11x^2 - 2.85 \cdot 10^{-4}x^3 & ,137 \leq x \leq 154 \\ -2226.70 + 31.68x - 0.18x^2 + 3.45 \cdot 10^{-4}x^3 & ,154 \leq x \leq 176 \end{cases}$$

$$s_{l2}(x) = \begin{cases} -479.03 - 1.19x + 2.20 \cdot 10^{-2}x^2 - 7.63 \cdot 10^{-5}x^3 & ,96 \leq x \leq 115 \\ -256.95 - 6.98x + 7.24 \cdot 10^{-2}x^2 - 2.23 \cdot 10^{-4}x^3 & ,115 \leq x \leq 137 \\ -1856.74 + 28.05x - 0.18x^2 + 4.00 \cdot 10^{-4}x^3 & ,137 \leq x \leq 154 \\ -320.66 - 1.87x + 1.10 \cdot 10^{-2} - 2.07 \cdot 10^{-5}x^3 & ,154 \leq x \leq 176 \end{cases}$$



Figur 6.5: Venstre: Fingeraftryk efter koordinatsystemet er tilføjet, højre: Fingreaftryk før koordinatsystemet er tilføjet

| $x$ | 96 | 115 | 137 | 154 | 176 |
|---|---|---|---|---|---|
| $f(x)$ | -458 | -441 | -427 | -425 | -424 |

Tabel 6.4: Funktionsværdier for linie 2

$$s_{l2}(x) = \begin{cases} -1258.41 + 20.74x - 0.18x^2 - 6.59 \cdot 10^{-4}x^3 & ,115 \leq x \leq 137 \\ 1747.22 - 45.08x + 0.30x^2 - 6.59 \cdot 10^{-4}x^3 & ,137 \leq x \leq 154 \end{cases}$$

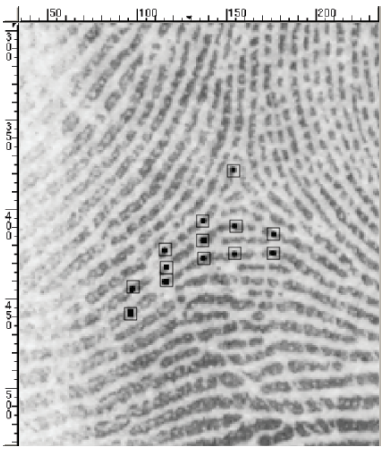
Grafen for disse splines kan ses på figur 6.7. Sammenlignes disse med linierne i det oprindelige fingeraftryk 6.6, kan man se at naturlige kubiske splines er en god tilnærmelse. Vi vil nu se på Wavlets som er den metode, vi har valgt at bruge til at komprimere fingeraftryk.

## 6.4 Wavelets

Vi vil i dette afsnit beskrive wavelets transformationer og hvordan disse kan benyttes til at komprimere billeder. Dette kapitel er i store træk et resumé af [23].

### 6.4.1 Historien bag ved billedkomprimering

Teorien bag wavelets blev udviklet uafhængig inden for matematik, fysik, geologi og elektronik i starten af 80'erne. Sammenspillet mellem disse områder har i de senere år ført til mange spændende anvendelses muligheder, f.eks. billedkomprimering og lydfiltrering. Fordelene ved at benytte wavelets til billedkomprimering er, at man kan opnå en høj komprimerings rate, og samtidig bevarer detaljerne.



Figur 6.6: Punktvalg illustreret på udsnit af fingeraftryk

| $x$ | 115 | 137 | 154 |
|---|---|---|---|
| $f(x)$ | -423 | -406 | -378 |

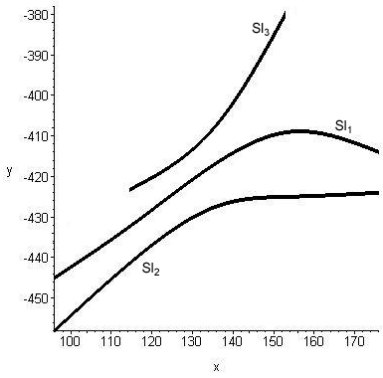
Tabel 6.5: Funktionsværdier for linie 3

Wavelets fik sit gennembrud indenfor billedkomprimering, da FBI, i 1995, udskrev en konkurrence om at finde en komprimeringsalgoritme til at komprimere deres fingeraftryksarkiver. På dette tidspunkt var standarden for billedkomprimering, JPEG formatet som vi beskrev i kapitel 5.2. Der var flere forskellige forskningsgrupper, som kom med forslag til en ny komprimeringsalgoritme. Det var en forskningsgruppe fra Los Alamos National Laboratory, styret af Jonathan Bradley og Christopher Brislawn, der vandt konkurrencen. Projektet blev ledet af Tom Hopper, som arbejdede for FBI. Deres forslag gik ud på at benytte moderne matematik til komprimering [9].

## 6.5 Diskret wavelet transformation

Man skelner normalt mellem kontinuert wavelet transformation og diskret wavelet transformation. Kontinuert wavelet transformation benytter man til kontinuerte signaler, f.eks i analoge signaler. Diskret wavelet transformation benytter man f. eks til digitaliserede signaler, som digitale lyde eller billeder. Da vi i dette projekt arbejder med billedkomprimering, der kan anskues som en endelig og tællelig mængde[8] af datasignaler, tager vi derfor udgangspunkt i diskret wavelet transformation. Rent praktisk er et *endeligt* datasignal et

[8]En tællelig mængde er en mængde bestående af tælleligt mange elementer, f.eks. mængden $\mathbb{N}$. Godt nok er $\mathbb{N}$ uendelig stor, men hvis man har uendelig tid, så kan man tælle mængden igennem. En ikke tællig mængde er mængden $\mathbb{R}$




Figur 6.7: Tilnærmelsen ved hjælp af splines ud fra de valgt punkter

specielt tilfælde af et *uendeligt* datasignal, da man ved et endeligt datasignal kan tilføje ekstra nul-værdier før og efter signalet for at opnå et uendeligt signal. Dette kan illusteres således:

$$1, 4, 5, 6, 3, 2, 7, 8 \qquad \rightarrow \qquad \ldots, 0, 0, 0, 1, 4, 5, 6, 3, 2, 7, 8, 0, 0, 0, \ldots$$

Et endeligt diskret signal, som skal wavelet transformeres, skal bestå af indekserede[9] reelle- eller kompleksværdier. Vi vil nu gennemgå, hvordan man kan benytte wavelets til at komprimere data med. Vi starter med at vise hvordan wavelets kan benyttes til at komprimere data i enkeltdimensionel form, og derefter bevæger vi os over til det todimensionelle tilfælde.

### 6.5.1 Et eksempel på wavelet transformation

Antag at der er givet følgende indekserede datasignaler, for værdierne $x[1], x[2], ..., x[8]$ for følgen[10] $x[n]$.

$$4, \quad 12, \quad 32, \quad 36, \quad 16, \quad 8, \quad 20, \quad 28$$

Vi tager værdierne parvis og udregner middelværdien og differensen for signalet. Følgende formler benyttes:

$$s = \frac{a+b}{2} \tag{6.13}$$

$$d = b - a \tag{6.14}$$

Her er $a$ og $b$ henholdsvis den første og den anden værdi i et par. $s$ er middelværdien og $d$ er differensen. Vi vil så få et nyt signal hvor de første 4

[9] Begrebet indebærer at rækkefølgen for elementerne i et signal har betydning for fortolkningen

[10] Vi har valgt at følge Arne Jensens [23] notation, $x[j]$, for en følge frem for den traditionelle måde $\{x_j\}_{j=0}^{\infty}$

indgange er middelværdier og de resterende 4 er differenser. Derefter udregnes tilsvarende middelværdierne og differenserne for de første 4 indgange for det nye signal. Processen gentages indtil man har et signal med en værdi bestående af middelværdien af de 8 originale værdier og 7 differens-værdier. Den øverste række er vores originale signal, mens den nederste række er

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 12 | 28 | 36 | 16 | 8 | 20 | 28 |
| 8 | 32 | 12 | 24 | 8 | 8 | −8 | 8 |
| 20 | 18 | 24 | 12 | 8 | 8 | −8 | 8 |
| 19 | −2 | 24 | 12 | 8 | 8 | −8 | 8 |

Tabel 6.6: Wavelets transformation for et ordnet signal med længden $2^j$, $j = 3$.

vores komprimerede signal se tabel 6.6. Hvis man tæller efter så ses det at det originale signal fylder 14 tegn, mens det komprimerede signal kun fylder 13 tegn. Komprimeringsraten for dette eksempel er ikke særlig stor, men hvis man tillader at miste nogle informationer i signalet, kan komprimeringsraten blive større. Den teknik man benytter sig af kaldes for thresholdning, som vi beskrev i starten af kapitel 6.1. Man vælger en numerisk værdi $c$, og alle værdier, numerisk, i det transformerede signal mindre end $c$ sættes til 0 (nul). Tager vi f.eks. $c = 9$, så vil vores komprimerede signal se således ud:

$$19, \quad 0, \quad 24, \quad 12, \quad 0, \quad 0, \quad 0, \quad 0$$

For at give et bedre billede af hvordan processen foregår, tager vi udgangspunkt i processen generelt. Vi antager at vi har givet et diskret, indekseret og endeligt datasignal som følgen $x[n]$ med en given længde $n = 2^j$:

$$x[0], \quad x[1], \quad \ldots, \quad x[2^j - 1], \quad x[2^j]$$

For at det bliver muligt at foretage en komprimering, skal signalet være af længden $2^j$ for et givet $j \in \mathbb{N}$. Grunden til at længden skal være $2^j$, er at man efter komprimering vil opnå et signal, der er halvt så langt som det signal, der skal transformeres. Hvis man har et signal med længden forskellig fra $2^j$ , søger man for at længden bliver $2^j$ ved at tilføje signalet ekstra nul-værdier enten foran selve signalet eller efter signalet. Så opnås komprimeringen ved at opdele signalet, således at man har delfølgen $s_{j-1}$ for lige indgange ($\text{lige}_j$) og delfølgen $d_{j-1}$ for ulige indgange ($\text{ulige}_j$). Det nye signal består af følgerne $s_{j-1}$ med længden $2^{j-1}$ og $d_{j-1}$ med længden $2^{j-1}$. Tilsammen har det nye signal længden $2^{j-1} + 2^{j-1} = 2^j$, som også er det vi startede med.

$$s_j \rightarrow s_{j-1}, d_{j-1}$$

Indgangene i det nye signal kan man opnå ved at benytte følgende formler:

$$d_{j-1}[n] = s_j[2n+1] - s_j[2n] \tag{6.15}$$

$$s_{j-1}[n] = \frac{s_j[2n] + s_j[2n+1]}{2} \tag{6.16}$$

De to ovenstående formler 6.15 og 6.16 er en mere generel udgave af formlerne i afsnit 6.14 og 6.13. Man benytter igen formlerne 6.15 og 6.16 for følgen $s_{j-1}$ mens man lader følgen $d_{j-1}$ være uændret indtil man ender med en følgen:

$$s_0, d_0, d_1, \ldots d_{j-2}, d_{j-1}$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $s_3[0]$ | $s_3[1]$ | $s_3[2]$ | $s_3[3]$ | $s_3[4]$ | $s_3[5]$ | $s_3[6]$ | $s_3[7]$ |
| $s_2[0]$ | $s_2[1]$ | $s_2[2]$ | $s_2[3]$ | $d_2[0]$ | $d_2[1]$ | $d_2[2]$ | $d_2[3]$ |
| $s_1[0]$ | $s_1[1]$ | $d_1[0]$ | $d_1[1]$ | $d_2[0]$ | $d_2[1]$ | $d_2[2]$ | $d_2[3]$ |
| $s_0[0]$ | $d_0[0]$ | $d_1[0]$ | $d_1[1]$ | $d_2[0]$ | $d_2[1]$ | $d_2[2]$ | $d_2[3]$ |

Tabel 6.7: Wavelet transformation for signalet $s_j[n]$, $j = 3$

Tabel 6.7 illustrerer wavelet transformationen for et givet signal med længden $2^j$, $j = 3$. Man starter med den første række, og efter den første transformation kommer man til 2. række, osv. Ved den sidste række i tabel 6.7 er differenserne, dvs. d'erne omrokeret, da vi ikke har benyttet "In place"egenskaben[11], desuden giver tabellen også et bedre overblik, af det transformerede signal. Man benytter ofte begrebet skalering, som siger noget om hvor mange gange der udføres wavelet transformation. For et givet signal med længden $2^j$ er det maksimale antal skaleringer, der kan foretages $log_2(2^j) = j$. For ovenstående tabel 6.7 er skalering 3.

## 6.6 Transformation ved hjælp af forudsigelse

Der findes flere måder at udføre wavelet transformationer på. I stedet for at betragte datasignalet som parvis, kan man betragte d'erne som en forudsigelsen ved hjælp af de omkringliggende elementer. Vi tager udgangspunkt i en lineær funktion, $s_j[n] = a \cdot n + b$, hvor $s_j[n]$ er vores signal, med et givet $a$ og et givet $b$. Vi har tre elementer efterfølgende hinanden $s_j[2n]$, $s_j[2n+1]$ og $s_j[2n+2]$ er det muligt at forudsige hvor stor en afvigelse $s_j[2n+1]$ er fra den lineære funktion, se figur 6.8. Følgende formler benyttes til at udregne elementer i en wavelet transformation ved hjælp af en forudsigelse.

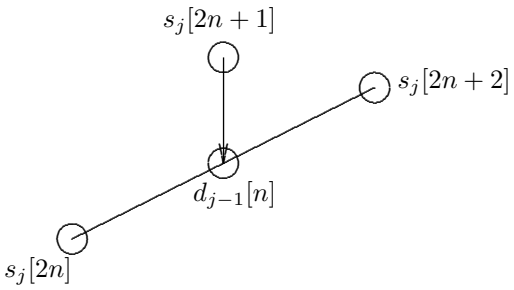
$$d_{j-1}[n] = s_j[2n+1] - \frac{1}{2}(s_j[2n] + s_j[2n+2] \quad (6.17)$$

$$s_{j-1}[n] = s_j[2n] + \frac{1}{4}(d_{j-1}[n-1] + d_{j-1}[n]) \quad (6.18)$$

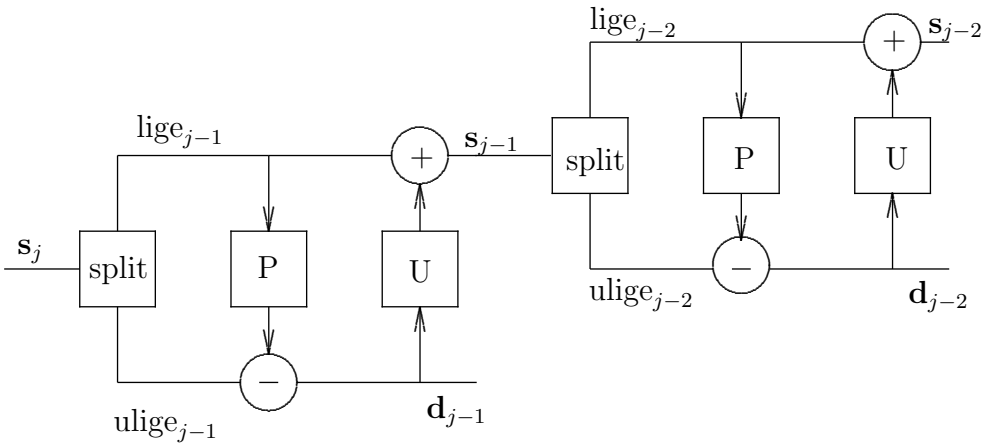
### 6.6.1 Lifting

I det følgende afsnit vil vi beskrive begrebet lifting.

[11] dette beskrives nærmere i afsnit 6.6.1




Figur 6.8: Wavelet transformation ved hjælp af forudsigelse




Figur 6.9: To trins lifting

På figur 6.9 ses hvordan signalet $s_j$ bliver splittet, til lige og ulige delfølger $\text{lige}_{j-1}$ og $\text{ulige}_{j-1}$. Minuserne på figur 6.9 betyder i dette tilfælde at det øvre signal trækkes fra det nedre signal, dvs. at $\text{lige}_{j-1}$ trækkes fra $\text{ulige}_{j-1}$. Tilsvarende betyder plusserne i figur 6.9 at det signal lægges til det øvre signal. I det næste afsnit beskrives Operationerne P (forudsigelse)[12] og U (opdatering)[13].

### Forudsigelse og Opdatering

$P$ er en forudsigelses operation og $U$ er en opdatering operation. Forudsigelses operationens primære opgave er at forudsige hvordan signalets værdier er placeret i forehold til hinanden. Hvilket betyder at der er en sammenhæng mellem værdierne i signalet. Opdatering operationen opdaterer signalet, baseret på prediction operationens forudsigelse om signalets værdier, se afs-

[12] Vi bruger betegnelsen P, da det hedder prediction på engelsk
[13] Kaldes U da det hedder update på engelsk

nittet 6.6. Vi har tidligere benyttet forudsigelses operationen, se formel 6.15, og opdatering operationen se formel 6.16. Det skal nævnes, at der findes forskellige måder at implementere forudsigelses og opdatering operationen.

**Transformation med "in place"**

Man kan benytte forudsigelses og opdaterings operationerne til at foretage en "in place" wavelet traformation. Dette indebærer, at man benytter den samme lagerplads til transformationerne. Derfor behøver man ikke at oprette nyt lagerplads hver gang man udfører en transformation. Hvis vi benytter formlerne 6.15 og 6.16, så er det ikke ligegyldig hvilken formel man anvender først. En korrekt brug kan give "in place"egenskaben. Benyttes formlen 6.16 først bliver man nødt til at benytte lagerplads til både $a$ og $b$ hele vejen gennem transformationen. Benytter man formlen 6.15 først kan mellemresultaterne lagres i enten a eller b, og man undgår yderligere allokering af lagerplads. Dette ses ved at sammenligner tabel 6.7 med tabel 6.8 ved at observere differenserne, $d$'erne. Efter udregning af differensen kan vi passende placere differenserne i de ulige indgange i det oprindelige signal, og dermed undgår vi yderligere allokering af lagerplads, samt omrokering af $d$'erne.

| | $s_3[0]$ | $s_3[1]$ | $s_3[2]$ | $s_3[3]$ | $s_3[4]$ | $s_3[5]$ | $s_3[6]$ | $s_3[7]$ |
|---|---|---|---|---|---|---|---|---|
| $P$ | $s_3[0]$ | $d_2[0]$ | $s_3[2]$ | $d_2[1]$ | $s_3[4]$ | $d_2[2]$ | $s_3[6]$ | $d_2[3]$ |
| $U$ | $s_2[0]$ | $d_2[0]$ | $s_2[1]$ | $d_2[1]$ | $s_2[2]$ | $d_2[2]$ | $s_2[3]$ | $d_2[3]$ |
| $P$ | $s_2[0]$ | $d_2[0]$ | $d_1[0]$ | $d_2[1]$ | $s_2[2]$ | $d_2[2]$ | $d_1[1]$ | $d_2[3]$ |
| $U$ | $s_1[0]$ | $d_2[0]$ | $d_1[0]$ | $d_2[1]$ | $s_1[1]$ | $d_2[2]$ | $d_1[1]$ | $d_2[3]$ |
| $P$ | $s_1[0]$ | $d_2[0]$ | $d_1[0]$ | $d_2[1]$ | $d_0[0]$ | $d_2[2]$ | $d_1[1]$ | $d_2[3]$ |
| $U$ | $s_0[0]$ | $d_2[0]$ | $d_1[0]$ | $d_2[1]$ | $d_0[0]$ | $d_2[2]$ | $d_1[1]$ | $d_2[3]$ |

Tabel 6.8: Wavelet transformering med "In place"

Generelt for formlerne 6.17 og 6.18 kan elementerne i en følge for et givet signal beskrives ved hjælp af forudsigelses og opdatering operationerne:

$$s_{j-1} = \text{lige}_{j-1} + U(d_{j-1}) \quad (6.19)$$

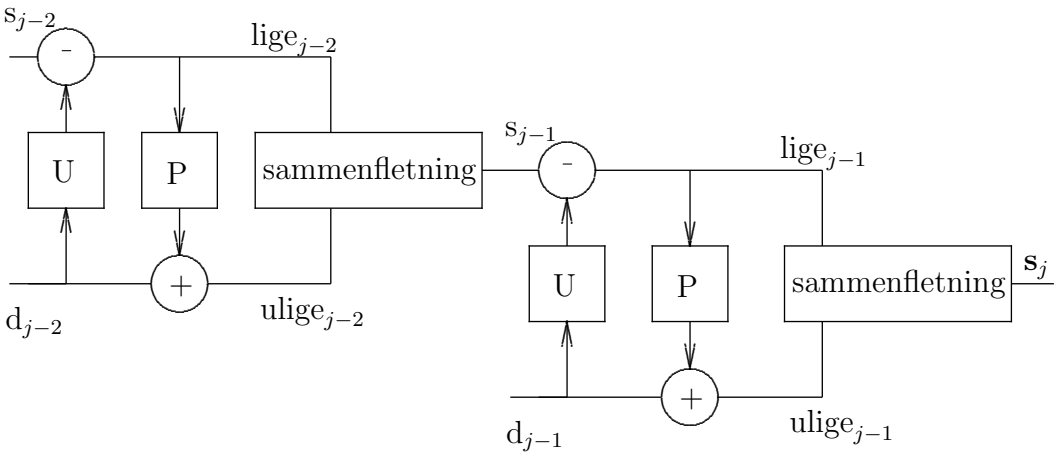
$$d_{j-1} = \text{ulige}_{j-1} - P(\text{lige}_{j-1}) \quad (6.20)$$

En algoritme, som indeholder både en forudsigelses og en opdatering operation, kaldes for en "et trins lifting". Man kan sammenkæde flere trin for at få en flere trins komprimering, se figur 6.9.

### 6.6.2 Dekomprimering

Vi har beskrevet, hvordan man opnår et komprimeret signal. På figur 6.10 vises hvordan dekomprimeringen sker for en to trins lifting. Hvis man sam-

menligner billederne 6.9 og 6.10, kan man observere, at der er sket et fortegnsskift.




Figur 6.10: Den inverse to trins wavelet transformation

Wavelet transformeringsprocessen laver det originale signal, til det komprimerede signal, dvs. det nye signal man vil ende med, kan beskrives som:

$$W_a^j : s_j \rightarrow s_0, d_0, d_1, \ldots, d_{j-2}, d_{j-1}$$

Processen kaldes også for *analyse*. Den anden vej, som bringer det komprimerede signal tilbage til det originale kaldes for *syntese*, og kan beskrives som:

$$W_s^j : s_0, d_0, d_1, \ldots, d_{j-2}, d_{j-1} \rightarrow s_j$$

De operationer som vi har benyttet os af er invertible, og komprimeringen er lossless Dvs. at man altid kan vende tilbage til det oprindelige signal, hvis blot man kender forudsigelses og opdaterings operationenerne.

### 6.6.3 Fortolkning af signalet

Lad os tage udgangspunkt i det forrige eksempel med et givet signal $s_j$, med længden $2^j$ for $j = 3$. Betragt den sidste linje i tabellen. I stedet for at transformere signalet tilbage til det oprindelige signal, kan vi modificere signalet således at præcis en af indgangene i signalet er 1, mens resten erstattes med værdien 0 (nul). Vi får så den første transponerede enhedsvektor[14] $\mathbf{v}[1]^T$

[14] fra kilde [17]

for $\mathbb{R}^8$ til at være $[1,0,0,0,0,0,0,0]$. Vi udfører den inverse wavelet transformation på vektoren $\mathbf{v}[1]^T$, se tabel 6.9. Tilsvarende gøres for den anden transponerede enhedsvektor $[0,1,0,0,0,0,0,0]$ se tabel 6.10, Der skal gøres opmærksom på at i tabel 6.9 og tabel 6.10 sker transformation nedefra og opefter. Dette fortsættes indtil vi har udført den inverse wavelets transformation for alle enhedsvektorerne $\mathbf{v}[1]^T, \mathbf{v}[2]^T, \ldots, \mathbf{v}[8]^T$. Vi ser så på hvad der sker med enhedsvektorerne under den inverse transformationen.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Tabel 6.9: Tilbage transformering for $[1,0,0,0,0,0,0,0]$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | −1 | −1 | −1 | −1 |
| 1 | 1 | −1 | −1 | 0 | 0 | 0 | 0 |
| 1 | −1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Tabel 6.10: Tilbage transformering for $[0,1,0,0,0,0,0,0]$

De enhedsvektorer som vi før inverterede og transponrede skal nu transponeres endnu en gang. Disse opstilles således, at den første enhedsvektor er placeret som den første søjle på matricen og den anden enhedsvektor i den anden søjle osv., så fås følgende:

$$W^{(3)}_{s_{enhed}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{6.21}$$

Vi har nu fundet en måde at finde den inverse transformation for et givet signal, med længden $2^j$, ved at betragte signalet som nogle vektorer i $\mathbb{R}^{2^j}$. Betragter vi igen matricen:

$$W^j_{a_{enhed}} = [\quad \mathbf{v}[1] \quad \mathbf{v}[2] \quad \ldots \quad \mathbf{v}[2^j-1] \quad \mathbf{v}[2^j] \quad ]$$

så kan et givet transformeret datasignal $\mathbf{x}[2^j]$ opstilles som en lineærkombination af vektorerne $\mathbf{v}[1] \quad \mathbf{v}[2] \quad \ldots \quad \mathbf{v}[2^j-1] \quad \mathbf{v}[2^j]$:

$$\mathbf{y}[2^j] = \mathbf{v}[1]\mathbf{x}[1] + \mathbf{v}[2]\mathbf{x}[2] + \cdots + \mathbf{v}[2^j - 1]\mathbf{x}[2^j - 1] + \mathbf{v}[2^j]\mathbf{x}[2^j]$$

Dvs. har vi den rigtig base, kan vi også findes det transformerede signal eller det inverse transformerede signal. Ved at multiplicere matricen $W^j_{s_{enhed}}$ med et givet komprimeret datasignal $\mathbf{x}[2^j]$, opstillet som søjle, så fås det oprindelig datasignal $\mathbf{y}[2^j]$: $W^j_{s_{enhed}}\mathbf{x}[2^j] = \mathbf{y}[2^j]$. På tilsvarende måde finder man $W^j_{a_{enhed}}$ ved at wavelets transformere de 8 enhedsvektorer.

$$[1,0,0,0,0,0,0,0] \quad [0,1,0,0,0,0,0,0] \quad \ldots \quad [0,0,0,0,0,0,0,1]$$

og efter transponering ender man op med følgende.

$$W^{(3)}_{a_{enhed}} = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$$

endvidere gælder der at

$$W^j_{s_{enhed}} \cdot W^j_{a_{enhed}} = I \qquad \text{og} \qquad W^j_{a_{enhed}} \cdot W^j_{s_{enhed}} = I$$

da, $W_{s_{enhed}}$ er invertibel, hvor matricen $I$ er en $j \times j$ enhedsmatrice. Vi har i dette afsnit benyttet lineær algebra til at give en fortolkning af wavelet transformation. Det er ikke effektivt at implementere wavelet transformation ved brug af matrix multiplication i et program, på baggrund af matriceregneoperationerne, da man vil få en kvadratisk udførselstid, dvs. $O(n^2)$ fremfor en lineær udførselstid, dvs. $O(n)$, for en udførelse af lifting.

### 6.6.4 Komprimering af billeder med wavelets

Indtil videre har vi kun omtalt signaler i en dimension, men da billeder er repræsenteret i to dimensioner, skal vi udvide wavelets til at kunne håndtere to dimensioner. Til det formål findes der to metoder. Man kan vælge at anvende to endimensionelle transformationer eller man kan bruge ægte todimensional transformation.

#### Separat transformation

For endimensionelle transformationer repræsenterer man et givent billede ved hjælp af en matrice. Lad os nu antage at vi har en $m \times n$ matrix:

$$X = \begin{bmatrix} x[1,1] & x[1,2] & \dots & x[1,n] \\ x[2,1] & x[2,2] & \dots & x[2,n] \\ \vdots & \vdots & \ddots & \vdots \\ x[m,1] & x[m,2] & \dots & x[m,n] \end{bmatrix} \tag{6.22}$$

Man kan repræsentere datasignalet ved at tilføje de efterfølgende rækker i matricen efter hinanden. Med denne teknik vil man komme ud for, at nogle af punkterne er "fejlplacerede", således at den sidste søjle vil ligge op af den første søjle i matricen og det kan give forskydninger. Således bliver matrixen 6.22 opstillet som:

$$X_{m \times n} = x[1,1], \dots, x[1,n], x[2,1], \dots, x[2,n], \dots, x[m,1], \dots, x[m,n]$$

To på hinanden følgende enkeltdimensionelle transformationer foregår ved, at man wavelet transformerer matricen med alle søjlerne uafhængig af hinanden og derefter wavelet transformerer man alle rækker i matricen. Det er underordnet om man vælger at benytte wavelet transformationen med rækkerne først eller søjlerne først. For at undgå at niveauet af kontrasterne mellem kanterne af billedet og de omkringliggende punkter, uden for billedet, bliver alt for stor, forsøger man at udjævne kanterne ved at tilføje udvidelsesværdier ved hjælp af en spejling ved kanterne af billedet. Vi vil nu vise hvordan spejlingen sker. Denne sker i to trin, spejling langs søjlerne og spejling langs rækkerne. Antag at vi har et givet todimensionelt signal, f.eks. et billede som i matrix 6.22, så udfører vi en spejling så matrix 6.22 bliver til:

$$X = \begin{bmatrix} x[1,-1] & x[1,0] & x[1,1] & \dots & x[1,n] & x[1,n+1] & x[1,n+2] \\ x[2,-1] & x[2,0] & x[2,1] & \dots & x[2,n] & x[1,n+2] & x[2,n+2] \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x[m,-1] & x[m,0] & x[m,1] & \dots & x[m,n] & x[m,n+1] & x[m,n+2] \end{bmatrix}$$

Spejlingen sker langs søjlerne 1 og n. Søjlerne $-1$, 0 er en kopi af søjlerne 2 og 1 og søjlerne $n+1$, $n+2$ er en kopi af søjlerne $n$, $n-1$. På tilsvarende måde sker spejlingen langs rækkerne 1 og $m$, hvor rækkerne $-1$, 0, er en kopi af 2, og 1, samt rækkerne $m+1$, $m+2$ er en kopi af rækkerne $m$ og $m-1$.

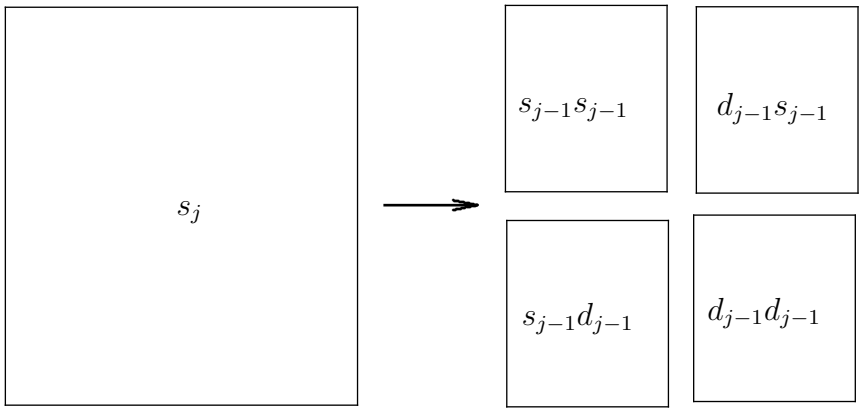
$$X = \begin{bmatrix} x[-1,1] & x[-1,2] & \dots & x[-1,n] \\ x[0,1] & x[0,2] & \dots & x[0,n] \\ x[1,1] & x[1,2] & \dots & x[1,n] \\ \vdots & \vdots & \ddots & \vdots \\ x[m,1] & x[m,2] & \dots & x[m,n] \\ x[m+1,1] & x[m+1,2] & \dots & x[m+1,n] \\ x[m+2,1] & x[m+2,2] & \dots & x[m+2,n] \end{bmatrix} \quad (6.23)$$

For hver indgang i det udvidede billede i matrix 6.23 udføres følgende lifting operation:

$$x[m][2n+1] = x[m][2n+1] - \frac{x[m][2n] + x[m][2n+2]}{2} \quad (6.24)$$

$$x[m][2n] = x[m][2n] + \frac{x[m][2n-1] + x[m][2n+1]}{4} \quad (6.25)$$

Efter søjlerne og rækkerne er blevet transformeret vil indholdet af data se således ud:




Figur 6.11: Data repræsentation af billedet efter transformering

Efter den første transformation består billedet af 4 lige store kvadrater. I det nedeste højre kvadrat har vi udregnet differenser for rækkerne og søjlerne. For nederste venstre kvadrat, har vi udregnet middelværdien for rækkerne og differensen for søjlerne. Øverste højre kvadrat er et omvendt tilfælde af nederste venstre kvadrat af billedet. Den sidste kvadrat indeholder kun middelværdierne, som benyttes til den nye wavelets transformation. Efter dette ender man med et nyt billede, bestående af mindre forskellige rektangulære billeder. I venstre side af figur 7.2 ses en en-skalering af billedet til højre.

Figur 6.12: Venstre: 1 skalering med seperate wavelets transformation. Højre: Det originale billedet før seperate wavelets transformation

Denne metode bibeholder kun de væsentlige oplysninger, og de unødvendige oplysninger bliver så kasseret. Med denne form for transformation, kan man komme ud for, at informationerne ikke bibeholdes, som de burde. Da transformationen sker i 2 seperate-transformationer kan en lille fordrejning i det originale signal giver en meget stor afvigelse af det komprimerede signal Hvis vi derimod benytte en ægte $2D$ transformation vil vi ikke komme ud for disse situationer.

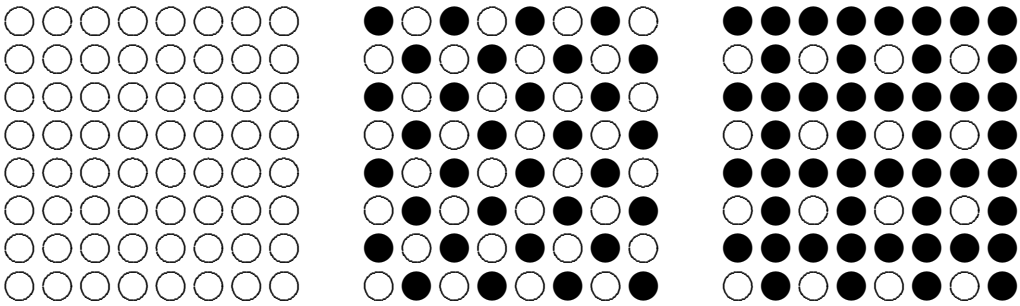
### Ægte $2D$ transformation

Lad os nu tage udgangspunkt i matricen $X$, og lad os antage, at der er uendelig mange tomme indgange udenfor matricen, se ligning 6.26.

$$X = \begin{bmatrix} \ddots & \vdots & \vdots & \dots & \vdots & \ddots \\ \dots & x[1,1] & x[1,2] & \dots & x[1,n] & \dots \\ \dots & x[2,1] & x[2,2] & \dots & x[2,n] & \dots \\ \dots & \vdots & \vdots & \ddots & \vdots & \dots \\ \dots & x[m,1] & x[m,2] & \dots & x[m,n] & \dots \\ \ddots & \vdots & \vdots & \dots & \vdots & \ddots \end{bmatrix} \tag{6.26}$$

På figur 6.13(a) har vi det originale billede, hvor alle pixels er repræsenteret som en matrix med cirkulære punkter som elementer.

Praktisk set behøver matricen ikke at være uendelig stor, da en $9 \cdot X$ matrix er tilstrækkeligt stor til at udføre en ægte 2D transformation. Vi



(a) (b) (c)

Figur 6.13: Processen for en 2 skalering

vil komme ind på senere hvorfor det er nok. Som tidligere beskrevet foretages der også her spejling langs kanterne. Man udfører en ægte $2D$ wavelet transformation, ved at markere hver indgang som enten hvid eller sort. Den mere systematiske metode er, at vælge et punkt, der er placeret i øverste venstre hjørne, og farver punktet sort. Derefter vælges de fire nærliggende punkter og disse farves hvide. For hver af de hvide punkter vælges fire andre nærliggende punkter og disse farves sorte, således at hver anden af indgangene er enten sort eller hvid, dette illustreres i figur 6.13(b). Fortolkningen for det endimensionelle tilfælde kan overføres til $2D$. De hviden cirkulære punkter er koefficienterne for følgen $s_{j-1}$, og de sorte cirkulære punkter er koefficienterne for følgen $d_{j-1}$. For de sorte punkter benytter man følgende forudsigelses operation til udregning af punktets nye værdi, som er baseret på de *nærmeste* punkter.

$$x_{\bullet}[m,n] = x[m,n] - \frac{1}{4}\left(x[m-1,n] + x[m+1,n] + x[m,n+1] + x[m,n-1]\right)$$

For at finde de nye værdier for de hvide punkter, benyttes følgende opdaterings operation, dog benyttes de *nye* $x_{\bullet}$ værdier fra tidligere udregninger, fra formel 6.27.

$$x_{\circ}[m,n] = x[m,n] + \frac{1}{8}\left(x_{\bullet}[m-1,n] + x_{\bullet}[m+1,n] + x_{\bullet}[m,n+1] + x_{\bullet}[m,n-1]\right)$$

Dette er en 1 skalering, se figur 6.13(b) På tilsvarende måde bestemmes anden skaleringen ved at bestemme de sorte og hvide punkter først og derefter udføres forudsigelses operationer og opdaterings operationer i den nævnte rækkefølge. Det maksimale skalering, for de ægte wavelets transformationer, kan blive dobbelt så mange som ved separate wavelet transformation, da antallet af $s$'erne i billedet bliver reduceret til det halve for

hver skalering i den ægte wavelet transformation, mens antallet af $s$'erne i billedet bliver reduceret til det kvarte for hver skalering for den separate wavelets transformation. Tidligere har vi nævnt at det var tilstrækkeligt at have en $9 \cdot X$ stor matrix for at udføre en ægte 2D transformering. For hver skalering, der foretages, bliver de punkter vi skal wavelet transformere, skrympet mod nederst venstre hjørne, se figur 6.13(c). Punktet kalder vi for udbreddelsespunktet, og da den nabo der ligger længst væk, højst kan ligge m punkter (billedets højde) fra udbredelsespunktet, er det derfor tilstrækkeligt, at de 8 undermatricer har samme højde og bredde som den oprindelige matrix.

## 6.7 Opsummering

Vi har beskrevet to forskellige teknikker til at komprimere data med, nemlig Lossy og Lossless. Lossless metoder er komprimeringsalgoritmer, som ikke taber informationer under komprimeringesprocessen, hvorimod lossy mister al information under en hvis grænse, betegnet threshold. Alt efter hvordan dataen er sammensat, kan den ene komprimeringsteknik være mere fordelagtig end den anden. Vi har set på Huffman kodning, som er en Lossless komprimeringsalgoritme. Derudover har vi beskrevet splines som en mulig kandidat til en Lossy komprimerings metode. Vi har desuden beskrevet den diskrete wavelet transformation i både den en- og todimensionelle situation. Hertil kommer en beskrivelse af, hvordan vi kan bruge wavelet tranformation til at komprimere billeder enten ved hjælp af seperate transformationer eller ægte 2D tranformation. Vi har set at wavelets både kan være en Lossless og Lossy metode.

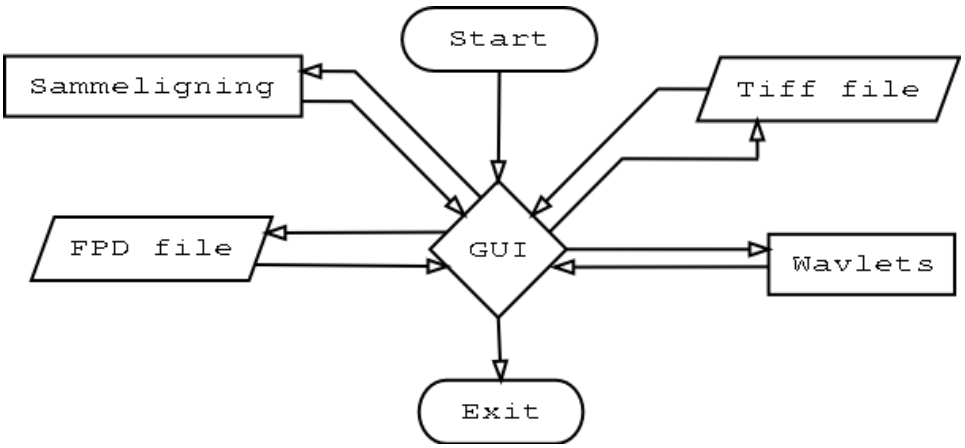
# Kapitel 7

# Udvikling af programmet

Vi har valgt at benytte sproget C til at lave et program, der komprimerer fingeraftryk ved hjælp af wavelet transformation. Grunden til at vi bruger C er, at vi har haft kursus i dette på anden semester. Programmet er opdelt i flere forskellige moduler, så det ikke bare er en i gruppen, som laver alt koden. Ved at opdele programmet i mindre dele, bliver det også mere overskueligt og derved lettere at lave. Formålet med dette kapitel er at skabe overskuelighed og struktur i programmeringsprocessen. Som ekstra vejledning har vi vedlagt en liste over nogle af de tekniske termer i programeringssproget C, disse kan finde i appendix A.3

## 7.1 Program specifikation

Programmet skal kunne indlæse et billede af TIFF formatet. Hertil skal vi benytte libtiff biblioteket. Det kræves at billedets opløsning er $1024 \times 1024$. Det skal kunne foretage en wavelet transformation på billedet. Når man har foretaget en transformation, har man to muligheder. Man kan gemme billedet i FPD, se afsnit 5.5, eller man kan foretage en dekomprimering med det samme. Når man har foretaget en dekomprimering af dataene, kan man så foretage en sammenligning mellem det orginale og det dekomprimere aftryk, for at undersøge om man har mistet for mange information i billedet. Resultaterne af denne sammenligning består af tre forskellige teknikker, som er forklaret nærmere under beskrivelse af det tilhørende modul. Man kan også vælge at gemme det dekomprimerede billede i TIFF, så man visuelt kan se, om billedet er i orden og brugbart. Der er yderligere den mulighed, at man kan indlæse et billede, som før er blevet gemt i FPD formatet. Herefter kan foretages en dekomprimering af billedet, hvorefter det kan gemmes som et TIFF billede.




Figur 7.1: Modulstrukturen i programmet

## 7.2 Programdiagram

## 7.3 De enkelte moduler og deres funktioner.

Vi vil i det følgende beskrive de enkelt moduler og deres funktion i det samlede program. Vi har valgt at opstille de centrale modulbeskrivelser på følende måde:

- Modulnavn
- Funktioner der indgår i modulet
- Beskrivelse af funktionerne i modulet, herunder en beskrivelse af input og output for de enkelte funktioner

### 7.3.1 GUI (Graphic User Interface)

**Funktioner**

Vi har fremstillet et interface, som kan køre i en standard terminal på en Linux eller Unix maskine. Vores GUI består af en menu, hvori man kan vælge mellem følgende punkter:

1. **Indlæs billede:** Indlæser et TIFF billede, som brugeren angiver.
2. **Komprimering:** Bruger wavelets til at foretage en komprimering/transformering.
3. **Gem billedet:** Laver en fil og en Huffman kodning på denne og gemmer billedet i FPD.
4. **Indlæs komp. billede:** Indlæser tidligere komprimerede billeder.

5. **Dekomprimering:** Dekomprimerer det wavelet transformerede billede.
6. **Sammenligning:** Foretager en sammenligning med det orginale billede.
7. **Gem billede som TIFF:** Gemmer billedet i TIFF format.
8. **Exit:**

Menuen er opbygget sådan, at man skal starte fra oven og gå ned igemmen menu, for at foretage en komprimering, dekomprimering, gemning og en sammenligning. Det er GUIen, der sørger for at holde sammen på de forskellige dele af programmet.

### 7.3.2 Tiff2array

#### Funktion

Konverterer en billedfil af TIFF format til et 2D array.

#### Beskrivelse af funktioner i modulet

Dette modul indeholder kun en funktion, tiff2array, som ikke bruger nogen hjælpefunktioner. Funktionen tager et filnavn og to variabler af typen int som input. Den åbner TIFF filen for læsning og indlæser informationer omkring højden og bredden. Udfra dette beregner den hvor mange pixels billedet indeholder. Det næste skridt er at frigøre hukommelsen til arrayet i forehold til billedets størrelse. Derefter checker den om TIFF billedet indeholder mere end en stribe (5.4). Checker fotometrisk information for hvilken vej farverværdierne løber i farveskalaen. Ligeledes skal der være styr på byterækkefølgen. Når dette er gjort indlæses billedet ind i en buffer, og herefter lægges dataen i 2D-arrayet. Billedet bliver lukket igen og arrayet sendes retur til GUI'en.

Modulet tager et TIFF billede som input og laver et dynamisk 2D array, som har samme størrelse som billedet. Arrayet indeholder værdier mellem 0 og 255, som repræsenterer farvetonerne i billedet. Dette modul kan kun læse TIFF billedere i gråtoner med 256 farver. Grunden til dette er, at vi ikke finder det nødvendigt at anvende flere farver, i et billede, til at repræsentere et fingeraftryk.

### 7.3.3 Analyse - wavelets

#### Funktioner

Formålet med dette modul er at komprimere et givet 2D array med højden $2^j$ og bredden $2^j$. Det er derfor ikke muligt at bruge dette modul på et billede med andre dimensioner. Følgende funktioner findes i dette modul.

- komprimering
- array_opret
- h_udvid
- v_udvid
- d_udvid
- analyse_image
- tresholding

### Komprimering

Denne funktion er interfacet til resten af programmet, dvs. når denne funktion kaldes vil andre funktioner blive kaldt for at udføre en wavelet transformation. Den kalder følgende funktioner:

- array_opret
- h_udvid
- v_udvid
- d_udvid
- analyse_image
- tresholding

Som parameter medtager den et 2D-array, billedets højde og bredde, antal skaleringer og den ønskede thresholding. Funktionen bliver returneret en struct med det udvidede array.

### array_opret

Da vi kun får selve arrayet med informationer om billedet, og da en ægte 2D transformation kræver yderligere lagerplads, bliver vi derfor nødt til at oprette et nyt array med ekstra lagerplads. Desuden har vi også brug for at sammenligne det ukomprimerede array med det komprimede array. Det udvidede array bliver 9 gange så stort som det oprindelig array se figur 7.2.

Derefter kopieres værdierne fra det oprindelige array, som vi har fået udefra, over i område 5. Som parameter medtager funktionen et 2D-array, billedets højde og bredde, antal skaleringer og den ønskede thresholding. Returnerer en struct med det udvidede array.

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Figur 7.2: Den udvidede array med opdelinger op i 9 store områder

**v_udvid**

Funktionen spejler værdierne således at værdierne bliver spejlet fra område 5 til område 2, samt en spejlning fra område 5 til område 8, se figur7.2. Som parameter medtager funtionen en pointer til en struct at type billede.

**h_udvid**

Funktionen spejler værdierne således at værdierne bliver spejlet fra område 5 til område 4, samt en spejlning fra område 5 til område 6, se figur7.2. Som parameter medtager funtionen en pointer til en struct af type billede.

**d_udvid**

Denne funktion udfører 4 spejlinger. Den spejler således at område 2 spejles over til område 1 og område 3. Derefter spejles fra område 8 til område 7 og område 9. Som parameter medtager funtionen en pointer til en struct af typen billede.

**analyse_image**

Denne funktion er selve analysen. Når vi har fået lavet det udvidede array, kan vi foretage en transformering. Vi initialiserer farverne i vores udvidede array ved at farve dem BLANK. Dette vil få en betydning i vores farvning senere. Desuden farves område 5, se figur 7.2, i vores udvidede array med farven HVID. Herefter skal vi finde de felter, der skal farves sorte. Dette sker ved at finde ud af hvilken skalering der skal udføres, da vi benytter forskellige søgemetoder til at lokalisere felterne. Efter at vi har lokaliseret de søgte felter og farvet dem sorte, benytter vi formel 6.27. Desuden farver vi de punkter der har været involveret i formel 6.27, og som ligger udenfor område

5, GUL. Nu behøver vi ikke at udregne de søgte hvidfarvede felter længere, da område 5 nu kun består af to farver SORT og HVID . Vi løber område 5 igennem og opdatere felterne. Igen farver vi de punkter, der har været involveret i formel 6.27, og som ligger uden for område 5, GUL. Efter hver skalering opdateres de felter, der indeholder farven SORT, således at disse farves ROED, da disse ikke skal benyttes i den efterfølgende transformation. Således fortsættes indtil det ønskede antal skaleringer er opnået.

Som parameter medtager den en 2D-array, billedets højde og bredde, antal skaleringer og den ønskede thresholding. Returnerer en struct med det udvidede array.

#### thresholding

Søger det udvidede array igemmen for GULe og ROEDe felter, hvori den numeriske værdi er mindre end thresholdingen, disse felter sættes til 0. Funktionen tager en pointer til en struct af typen billede.

### 7.3.4 syntese - wavelets

#### Funktioner

Funktionen dekomprimerer et givet 2D array. Følgende funktioner findes i dette modul.

- dekomprimering
- syntese_image
- bil_trans

#### dekomprimering

Denne funktion er interfacet til resten af programmet, dvs. når denne funktion kaldes vil andre funktioner blive kaldt for at udføre en invers wavelet transformation. Den kalder følgende funktioner:

- syntese_image
- bil_trans

#### syntese_image

Denne funktion er næsten den inverse af analysen. Da vi har farvelagt områderne 1 til 9 i analysen, er algoritmen for den inverse transformation nu nemmere. For hvert af de hvide felter lokaliseres de 4 tilhørende felter, der indgår i formlen 6.27. Disse 4 benyttes tilsammen til at udregne værdien i det hvide felt. Herefter farves de 4 felter SORT. For alle SORTfarvede felter

i område 5 udregnes værdien, ved hjælp af de 4 søgte felter udfra formel 6.27. De 4 søgte felter farver vi HVID. Dette er tilsammen en skalering.

Efter hver skalering farver vi de SORTe felter i området 5 HVIDe for at gøre klar til en ny transformation. Funktionen kører indtil den ønskede skalering er opnået. Som parameter tager funktioen en pointer til en struct af typen billede og returner det område 5, se figur 7.2.

**bil_trans**

Denne funktion kopierer data fra område 5 over i et nyt array, ved hjælp af dynamisk allokering, med samme størrelse som område 5.

### 7.3.5 IO - wavelets

**Funktioner**

Skriver det komprimerede data i en fil med vores egen notation, og læser filen ind og gør opsætningen klar til en dekomprimering. Følgende funktioner findes i dette modul.

- write_out_file
- read_in_file

**write_out_file**

Da vores udvidede array er 9 gange så stort, som det oprindelige ikke udvidede array, finder vi derfor en måde at gemme det komprimerede data til senere brug med. Alle værdierne i område 5 skal gemmes, da de udgør selve billedet. Desuden har nogle af felterne uden for område 5 også en betydning for dekomprimeringen. Derfor medtager vi de felter, som er farvet GUL, se afsnit 7.3.3. Vi har valgt at gemme informationerne fra område 5 ved først at gemme farven af selve feltet efterfulgt af værdien i feltet. Da der bliver mange gentagelser af farven ROED efterfulgt af værdien 0, har vi valgt at beskrive mønsteret med betegnelsen A. For andre værdier og farver i området 5 benytter vi tegnet 'R' og 'H' som separatorer for henholdsvis farverne ROED og HVID efterfulgt af en værdi. Nedenstående afsnit viser hvordan vores output filformat er udformet.

**Forberedelses filoutput til Huffman**

Fil-output : hoej bred skal tresholding F1V1F2V2F3V3....FsVs m1 n1 vaerdi1 m2 n2 vaerdi2.... ms ns vaerdis R R R.

- hoej og bred — er højden og bredden af vores oprindeligt billede.

- skal — er den antal skaleringer vi har benyttet til wavelet transformation.
- tresholding — den thresholding der er benyttet til komprimeringen.
- F1V1F2V2F3V3.... — F1 er farven til V1 og F2 er farven til V2 osv. Hvis farven og dens tilhørende værdi, f.eks, F1V1 så erstatter vi det med tegnet A, ellers markerer vi F'erne enten som 'R' eller 'H'. V'erne er værdierne. Disse farver og værdier er hentet fra området 5. (se figur 7.2)
- m1 n1 vaerdi1 m2 n2 vaerdi2.... ms ns vaerdis — m1 og n1 er række og søjle i det udvidede array, altså koordinaterne for den pågældende værdi, her har indgangen værdien vaerdi1, osv. mellemrummerne mellem m'erne, n'erne og vaerdi'erne skal med, da de er separatorer. Disse beskriver de GUL'e felters placering og værdi i det udvidede array
- R R R — Er vores termineringsmønster

1024 1024 20 10 AAAAR-14H54AAAAR33H22.....H34 300 200 25 ... 340 500 35 R R R

Overstående ses et eksempel på output af en fil. Parameterne til funktionen: pointer til struc'en af typen billede, navnet på den indlæste fil og tilstanden på filen, dvs. om den er åbnet til skrivning, læsning eller begge dele. Vores fil format FPD fremkommer, ved at man anvender Huffman kodning på ovenstående filoutput.

#### read_in_file

Starter med at allokere hukommelse til vores udvidede array som i afsnittet 7.3.3. Indlæser tegnene så de bliver placeret i de rigtige positioner inde i det udvidede array, og derefter returnerer den det udvidede array, der nu er klar til dekomprimeringsoperationen. Parameterne til funktionen er navnet på den indlæste fil og tilstanden på filen, dvs. om den er åbnet til skrivning, læsning eller begge dele. Returnerer en pointer til en struct af typen billede.

### 7.3.6 Gem/indlæs billede (Huffman)

#### Funktioner

Dette program tager en fil som input og laver en ny fil, som er Huffman kodet. Så det vi gør er at gemme de nødvendige informationer i en ASCII fil, som beskrevet i afsnit 7.3.5. Vi kører så det eksterne program på ASCII filen. Programmet som vi bruger til Huffman kodning har vi fra Michael Dippersteins hjemmeside se kilde [7].

### 7.3.7 Sammenligning

#### Funktioner

Dette modul foretager en sammenligning mellem det komprimerede og det originale billede, for at undersøge om der er nogle store forskelle mellem dem. Til dette bruger vi tre forskellige metoder:

- Pixel vs. Pixel
- Modul sammenligning
- Vektor norm opstilling

Alle disse metoder giver hver et bud på, hvorledes billedet har ændret sig under komprimerings processen.

#### Beskrivelse af funktionerne i modulet

Vi vil nu beskrive funktionerne af de ovennævnte metoder i modulet. De forskellige sammenlignings teknikker benytter næsten de samme variable som input til deres funktioner. Vi har derfor valgt at sætte dem sammen i en struktur (struct), som vi har kaldt "sammenligning". Den indeholder de to billeder, som skal sammenlignes, informationen om deres størrelse, filenavn på det originale og det komprimerede billede. De værdier vi benytter til sammenligningsteknikkerne er de nummeriske værdier, af de to billeders differens i de enkelte indgange. Vi får et nyt array, som indeholder differenserne fra de to billeder. Efter forklaringen af de enkelte funktioners formål beskrives hvad funktionen tager ind som input, og hvad den returnerer.

#### total_match

Denne sammenligningsteknik er den mest simple af vores sammmenligningsalgoritmer. Funktionen sammenligner alle pixels i billederne. Det vil sige at der foretages en total sammenligning af billederne pixel for pixel.

$$a[j,k] = b[j,k] \tag{7.1}$$

Hvor $a$ og $b$ er de to arrays, som repræsenterer pixelen i billedet og $j, k$ er index tal i arrayet. Vi opdeler antallet af pixels i to variabler alt efter, om de har den samme værdi. Den optæller altså antal af pixel, som ikke passer overens med det originale billedes pixels. På den måde får man en variabel, hvori man kan se hvor mange pixels, der har ændret værdi og dermed deres farve. Denne test siger ikke noget om hvad værdien er ændret til.

Funktionen tager en pointer, der refererer til en struct af type sammenligning, som henholdsvis indeholder det komprimerede billede, det originale,

samt højde og bredde på disse. Vi benytter også to ints til at tælle sammen på antallet af pixels, som er uændret, og dem som har ændret sig. Modulet returnerer ikke noget, men bruger "call by reference" på de to ints.

#### px_match

Denne sammenligningsprocess er den mest avancerede sammenligningsalgoritme, vi benytter i dette sammenligningsmodul. Funktion tager et delområde af billedet og bruger det til en sammenligning. For at finde ud af hvor store disse delområder skal være benytter vi modulus. Vi beregner en talrække, som både går op i den halve højde og halve bredde af billedet. Vi udskriver en talrække på skærmen, så brugeren kan vælge et af tallene til sammenligningen. Alt efter hvilket tal brugeren vælger, bestemmer dette hvor stort et område vi skal kigge på af gangen.

Derefter laver vi plads i hukommelsen til det array, som funktionen skal bruge til at holde styr på hver pixel i det enkelte delområde vi kigger på. Vi allokerer også plads i hukommelsen til det array, som bliver returneret til sidst i funktionen. De to arrays er af typen int og double, grunden til at den sidste er af typen double er at den skal indholde flydende tal, som vi får ved at regne gennemsnittet ud af de forskellige delområder.

Det næste der sker i denne funktion er selve kernen i funktionen. Det er her sammenligningen finder sted. Denne del består af 4 "for" løkker, som holder styr på hele billedet og det aktuelle delområde. Selve udregningerne af fejlen, som er opstår i billedet beregnes i de indre "for" løkker. Her finder vi de gennemsnitlige forskelle mellem de to billeder i det aktuelle delområde. Herefter tages de numeriske værdier og disse bliver gemt i et array, som bliver returneret til sidst i denne funktion. Når vi har beregnet gennemsnittet i et delområde hopper vi ud af den indre løkke. Den ydre løkke sørger for, at vi kommer ind i det næste delområde.

Funktionen tager en pointer til en struktur af typen sammenligning og en pointer af typen double, som skal indeholde det tal, vi finder ved hjælp af modulus. Denne funktion returner et array af typen double til systemet sammen med pointer af typen double. Grunden til at vi bruger double er at vi benytter flydende tal til nogle af udregningerne i denne funktion.

#### vektor_match

Denne funktion benytter tre forskellige normer vektorer til at beskrive forskellene i billedet. Vi kigger her på max normen, første normen og anden normen. Vi starter i denne funktion med at beregne forskellen mellem de to billeder ved at trække de to arrays fra hinanden.

$$v = a - b \tag{7.2}$$

, hvor $a$ og $b$ er de to 2D arrays, som repræsenterer de to billeders pixels værdier. Vi opstiller tre normer for v:

$$||v||_{max} = max\{|a[j,k] - b[j,k]| \quad | \quad j = 1, ..., N, k = 1, ..., N\} \tag{7.3}$$

Denne norm vektor kaldes for max normen. Den beskriver den maximale værdi i arrayet. Denne test vil give et meget lille udslag hvis de to billeder er meget tæt på at være ens. Den fortæller ikke noget om hvor fejlen er, men bare hvor stor den er. Den næste norm er første normen som vi udregner på følgende måde:

$$||v||_1 = |v_1 + v_2 + ... + v_n| \tag{7.4}$$

$$= \left| \sum_1^n (v_n) \right| \tag{7.5}$$

Første normen reagerer meget kraftigt på fejl, da den ligger størrelsen af alle fejl sammen. Er der mange små fejl, får vi et stort tal. Ligeledes hvis der er en stor fejl får vi også et stort tal. Så denne norm værdi er bedst til at påvise om der er mange fejl i billedet.

$$||v||_2 = \left| \sqrt{v_1^2 + v_2^2 + ... + v_n^2} \right| \tag{7.6}$$

$$= \sqrt{\sum_1^n |v_n^2|} \tag{7.7}$$

Anden normen reagerer knapt så kraftigt på fejl i billedet, som første normen gør, men den stiger meget når der opstår en større fejl i billedet. Den bedste måde at bruge disse normer på, kigger vi nærmere på under testen af programmet8.

Denne funktion tager en struktur pointer af typen sammenligning som input. Desuden tager den også to pointere af typen int og en af typen double. Disse bruger vi til at returnere de tre normer, som vi beregner udfra billedet.

Vi vil nu kigge nærmere på array2tiff, som opbygger et TIFF billede udfra de dekompremerede data.

### 7.3.8 Array2tiff

Konverterer et 2D array til et TIFF i gråskala med 256 farver.

#### Funktion

Dette modul består kun af en funktion, som gemmer et TIFF billede.

**array2tiff**

Array2tiff starter med at allokere plads til raster image dataene i hukommelsen i form af et array. Dette skal bruges til at repræsentere pixels værdierne i billedet. Vi tager alle værdier i 2D arrayet og indsætter dem i det andet array. Da dette er et almindeligt array, bliver værdierne opstillet på en lang række. For at vi kan skrive et TIFF billede, bliver vi nødt til at sætte en række tags, der er det mindste antal tags, som der skal til for at skrive et TIFF billede. De angiver nogle af de grundlægende informationer, som skal til for at åbne billedet igen. Når tag værdierne er sat kan vi gemme billedet. Når billedet er skrevet til disken, lukker vi filen og og returnerer kontrollen tilbage til GUI'en.

Funktionen modtager en pointer af type "sammenligning" og en char pointer, som indeholder et filenavn, hvori billedet skal gemmes. Den returnerer ikke noget, udover at den gemmer billedet på systemet.

# Kapitel 8

# Test af programmet

Vi har foretaget en test af vores programs komprimeringsengenskaber, med 10 forskellige billeder af fingeraftryk. Alle de fingeraftryk vi kørte igennem vores program, havde en filstørrelse på 1024Kb og dimensionerne 1024 x 1024 med en opløsning på 500dpi. Vi prøvede tre forskellige thresholds på henholdsvis 10, 20 og 30. Et eksempel på hvordan billeder ser ud før og efter at have komprimeret kan ses på billederne 8.1 og 8.2.

Samtlige billeder fra testen, kan findes på den vedlagte CD-Rom. Filerne er navngivet 1.tif - 10.tif og filnavnene på de billeder, som har været udsat for komprimering er opbygget på følgende måde: 1.tif hedder 1_out_10.tif, når den har en threshold på 10, 2.tif, med threshold på 20, hedder 2_out_20.tif. Alle filnavne er opbygget på denne måde. Vi har valgt at kigge på de to bedste komprimerede fingeraftryk og de to dårligeste komprimerede. De to bedste er nr. 10 og 7, mens de dårligste er nr. 1 og 9.

## 8.1 Filstørrelse

Vi har udfra resultaterne, af komprimeringen af de ti fingeraftryk, fundet et gennemsnit af komprimeringsraten af fingeraftrykket, som har været udsat for vores komprimering. Disse gennemsnitsværdier kan se i tabel 8.1.

Vi har valgt at sætte threshold til 20, da billederne ved højre threshold bliver stærkt forringet. Vi kan se fra tabel 8.1, at den bedste komprimering, i forhold til billedets kvalitet, er ca. 79.13%.

## 8.2 Normen

Vi vil også kigge på de forskellige værdier af normer, som vi udregner under sammeligninsprocessen. Disse kan give os en idé om billedets kvalitet, uden at kigge på billedet i TIFF. Vi har i denne tabel brugt de to bedst komprimerede og de to dårligst komprimerede billeder. En oversigt over de forskellige normværdier kan ses i 8.2. Det man kan bruge max normen til,

Figur 8.1: Det orginale fingeraftryk(venstre) og billedet udsat for en threshold på 10(højre)

Figur 8.2: Fingeraftrykket udsat for en threshold på henholdsvis 20(venstre) og 30(højre)

| Filnavn | Threshold | Filstørrelse | Komprimering |
|---|---|---|---|
| 10_out_10 | 10 | 240Kb | 76.56% |
| 10_out_20 | 20 | 192Kb | 81.18% |
| 10_out_30 | 30 | 178Kb | 82.60% |
| 7_out_10 | 10 | 249Kb | 75.69% |
| 7_out_20 | 20 | 194Kb | 81.01% |
| 7_out_30 | 30 | 178Kb | 82.58% |
| 1_out_10 | 10 | 376Kb | 63.30% |
| 1_out_20 | 20 | 254Kb | 75.15% |
| 1_out_30 | 30 | 230Kb | 79.18% |
| 9_out_10 | 10 | 342Kb | 66.54% |
| 9_out_20 | 20 | 236Kb | 76.91% |
| 9_out_30 | 30 | 201Kb | 80.37% |

Tabel 8.1: Test af de forskellige billeders komprimeringsniveau

| Threshold | Størrelse | Komprimering |
|---|---|---|
| 10 | 296.1Kb | 71.08% |
| 20 | 213.7Kb | 79.13% |
| 30 | 188.5Kb | 81.59% |

Tabel 8.2: Gennemsnitlig komprimeringsrate

er at se den maximale pixel forskel i billedet. Første norm fortæller noget om forskellen i samtlige pixels i billedet. Hvis vi tager første normen og dividerer med antallet af pixels i billedet, kan vi se den gennemsnitlige differens mellem det originale og det dekomprimerede billede. Eksempel på disse udregninger kan ses i udregning 8.1.

$$\begin{aligned} \text{forskel} &= 3921088/(1024 * 1024) && (8.1)\\ &= 3921088/1048576 && (8.2)\\ &\approx 3.739 && (8.3) \end{aligned}$$

Fra udregningen kan vi se, at fingeraftryk 10 med en threshold på 10, har en gennemsnitlig afvigelse på ca. 3.739 pr. pixel. Denne test siger ikke noget om, hvori billedet fejlen ligger. Anden normen repræsenter længden af forskellen mellem de to billeder. Vi har ud fra vores test konkluderet, at hvis anden normen er over 12000 så bliver kvaliteten af billedet for dårligt til, at det kan bruges.

Ud fra vores test har vi valgt at sige at den maksimale threshold er 20. Ved denne threshold kan vi stadig tyde detaljer i billedet. Vi kan også bruge normer til at se om komprimeringen er god, eller om der er gået for mange detaljer tabt.

| Threshold | Max norm | Første norm | Anden norm |
|---|---|---|---|
| 10 | 23 | 3921088 | 4871.73 |
| 20 | 64 | 8916906 | 11100.57 |
| 30 | 87 | 14733114 | 17773.29 |
| 10 | 28 | 3926767 | 4963.67 |
| 20 | 56 | 8182477 | 10184.67 |
| 30 | 85 | 11771465 | 14755.31 |
| 10 | 27 | 4085406 | 5080.84 |
| 20 | 60 | 8726003 | 11100.35 |
| 30 | 78 | 12027651 | 15371.58 |
| 10 | 171 | 3826308 | 5300.29 |
| 20 | 171 | 8453603 | 10826.41 |
| 30 | 173 | 12909056 | 16350.03 |

Tabel 8.3: Test af de forskellige billeders normer.

## 8.3 Delkonklusion

Vi har vist at TIFF formatet er velvalgt som base for vores komprimeringsfase. Dette er også valgt fordi vi med TIFF formatet har LibTIFF biblioteket at støttes os op af. Komprimeringen kan foregå på to måder: Lossless og Lossy. Vi har betragtet hvordan man kan bruge Huffman kodning til at komprimere Lossless. Dette bruger vi til at komprimere de data vi får ud af wavelet transformationen.

Splines kan benyttes til at komprimere fingeraftryksbilleder, men vi valgte ikke at implementere dette i vores program.

Vi har vist hvordan man kan benytte wavelets til at komprimere data, ved hjælp af to operationer, forudsigelse og opdaterings operationerne, både med og uden thresholding. Ved et fornuftigt valg af thresholding kan man både opnå en høj komprimeringsrate og samtidig bibeholde de vigtigste informationer. En af de afgørende kriterier og nøglen for hvor god ens komprimeringrate er, bestemmes af de rigtige forudsigelse- og opdaterings operationer. Vi har beskrevet i detaljer hvordan seperat wavelet transformation samt ægte wavelet transformation fungerer. Desuden har vi argumenteret for hvad der vil ske med billedet, hvis vi havde valgt at benytte separate wavelet transformationer. Derfor er det ikke ligegyldigt hvordan vi anvender wavelet transformationerne. Vi har desuden været inde på at implementeringen af wavelets, ved hjælp af lineær algebra, ikke er velegnet, men det giver derimod en anden fortolkning af hvordan wavelet transformerer signalerne.

Vores program er bygget op omkring en wavelet transformation hvoraf outputtet bliver komprimeret med en Huffman kodning. Dette gemmer vi i vores eget filformat FPD. Vi kan decomprimere billedet igen hvilket giver os muligheden for at teste komprimeringen.

Testen af programmet forløb på visuel og filstørrelsesmæssig vurdering

samt beregninger på forskellene i billederne. Af denne test så vi, at vi gennemsnitlig, kunne komprimere med 79,13% med en threshold på 20. Dette fandt vi acceptabelt. Vi havde ønsket en højere komprimering, men taget i betragtning at komprimeringsmetoden er udviklet til gråskalabilleder generelt og ikke specialiseret til fingeraftryksbilleder er det ikke nogen dårlig komprimeringsrate.

# Kapitel 9

# Konklusion og perspektivering

Vi har i dette projekt beskæftiget os med biometri og komprimering af fingeraftryk. Vi har undersøgt historien bag fingeraftryk og hvordan disse er blevet anvendt op gennem tiden.

Biometri kan bruges til verificering og indentifikation af personer ud fra deres biometriske karakteristika. Vi har kigget på forskellige former for biometriske scannere og hvordan disse bliver anvendt. Sikkerheden omkring disse scannersystemer er en faktor der skal prioteres højt og ikke må fejle. I Danmark har de første biometriske scanningssystemer fået deres indtog og dette kan skabe nogle etiske problemer. Disse problemer kan føre til, at befolkningen kan blive angst for teknologien, hvis denne ikke håndteres ordenligt. Vi har set at nogle folk har modvilje mod at lade sig registrere. For at gribe hånd om dette er det nødvendigt at foretage nogle lovmæssige tiltag. En anden faktor som er vitig for anvendelse af biometrisk genkendelse, er at der generelt skal være flere oplysning om anvendelsen af disse. Derfor vil det tage tid, at få indført dette da folk skal vænne sig til det. Hvis man løser de problemer, der er ved biometri, har det en lys fremtid som hurtig og effektiv indentifikationsmetode.

Splines fandt vi egnet til komprimering, men har valgt ikke at implementere den.

Vi har brugt TIFF billedformatet som basis for komprimering og vi har lavet vores eget format, FPD til at gemme de komprimerede fingeraftryk. Til at komprimere med har vi set på Lossless og Lossy komprimerings metoder. Vi har konkluderet, at ved fingeraftryk, er det en god ide at kombinere de to metoder. Vi bruger først wavelets, som er en Lossy komprimeringsmetode og derefter Huffman kodning, som er en Lossless komprimeringsmetode. Vi bruger en ægte 2D wavelet transformation, i stedet for seperate transformationer til komprimering af fingeraftryk. Den ægte tranformation, med en threshold på 20 og en maksimum skalering, fandt vi mere velegnet til kom-

primering af fingeraftryk. Med vores implementation af disse to komprimerings metoder har vi opnået en komprimeringsrate på ca. 79.13%, hvilket vi anser for at være en relativt god komprimeringsrate i det omfang, at den er udarbejdet i en projekt periode. Dermed gør vi os ikke nogle forventninger om, at dette program kan bruges i nogen kommerciel sammenhæng. Vi spår at udviklingen og ubredelse af disse scannere vil stige kraftig i den næste årti.

I en videre perspektivering af projekt kunne man arbejde med en implementering af splines og derved opnå en bedre komprimeringsrate. Man kunne også arbejde med at optimere wavelets transforamtion til at arbejde med andet end en lineære transfomation. Med omfattende arbejde kan man specialisere waveletkomprimeringen til fingeraftryk. Huffman kodningen kunne forbederes eller måske udbyttes med en anden kodning, som kunne lave en bedre komprimering. Alt i alt er der en mange mulige måder at bringe komprimeringsrate højere op, men de sidste få procent vil blive sværere og sværere at hente.

## 9.1 Videre perspektivering

På nuværende tidspunkt er det kun retsplejeloven, der tager højde for biometri og der er derfor et behov for at der bliver lavet en lov, der specifikt håndterer behandling af biometriske oplysninger. Da biometri er et område der giver mulighed for at overvåge personer skal der strammes op over for sikkerheden ved opbevaringen og behandlingen af oplsyningerne. Der skal også laves faste opsyn med registret, for at sikre at lovene bliver overholdt. Sidstnævnte problemstilling vil blive belyst i forbindelse med teknologikritikken i kapitel 4.

# Litteratur


[1] Adobe. Tiff documentation rev. 6.0. http://www.libtiff.org/document.html. Set d. 19-05.03.

[2] Jens Henrik Badsberg. Franske kurver: Interpolation og approximation. http://www.math.auc.dk/~jhb/Undervisning/Interpolation/Franske/. Afsnit om Kubiske Splines, set 12-05-03.

[3] CNN. Airport security in a blink. http://www.cnn.com/2002/TECH/science/02/08/airports.eyes/index.html. Link fundet gennem John Dauhmans hjemmeside, set 13-04-03.

[4] Special Agent Ed German US Army Criminal Investigation Command. The history of fingerprints. http://onin.com/fp/fphistory.html. Fingeraftrykket historie findes her: http://onin.com/fp/fphistory.html. Kilde set 26-03-03.

[5] Datatilsynet. Bekendtgørelse om behandling af personoplysninger i det centrale kriminalregister. http://www.datatilsynet.dk/include/show.article.asp?art_id=499. set 25-04-03.

[6] John Daugman. John daugman's webpage, cambridge university, computer laboratory, cambridge uk. http://www.cl.cam.ac.uk/users/jgd1000/. John Daugman's personlige hjemmeside, omhandlende delvist hans iris algoritme, set 26-03-03.

[7] Michael Dipperstein. Michael dipperstein's huffman code page. http://mdipper.digitalrice.com/huffman/. Set d. 21-05.03.

[8] Ugeskrift for retsvæsen. Ugeskrift for retsvæsen 1914. Kopi af s. 669-672, modtaget d. 25-04-03, leveret af biblioteket, da original kilde ikke er til udlån.

[9] Michael W. Frazier. *An Intorduction To Wavelets Through Linear Algebra*. Springer Verlag, 1999.

[10] Tom Harris. How file compression works. http://www.howstuffworks.com/file-compression.htm/printable. Set d. 22-03-03.

[11] Kristian Herlufsen. Et fingeraftryk for en færgebillet. *Tænk + test*, Febuar 2002. Set d. 18-03-03. Artikel omkring nyt fingeraftryks system på bornholmsfærgen.

[12] Mcgraw Hill. Discrete mathematics and it's applications, 2003. Afsnit omkring Huffman kodning, siderne 650 - 651.

[13] HowToDoThings.com. Huffman compression algorithm. `http://www.howtodothings.com/showarticle.asp?article=313#Code:`. Side som beskriver Huffman algoritmen i detaljer. Set 14-05-03.

[14] Bo Juni. Din krop er din kode. `http://www.bitconomy.dk/magasin.asp?article=2942&showmenu=3#`. Website artikel som omhandler biometri generelt, set 26-03-03.

[15] Kent Krøyer. Fingeren erstatter pin-koderne. *Ingienøren - Elektronik og tele*, 03 2003. 2 siders artikel i Inginøren omhandlende fingeraftryk og biometri.

[16] Yokohama Nat. University Matsumoto Laboratory. Undersøgelse af sikkerhed for fingeraftryks genekendelse. `http://www.itu.int/itudoc/itu-t/workshop/security/present/s5p4.pdf`. Rapport omkring sikkerhed i fingeraftryks genkendelse, set 26-03-03.

[17] David C. Lay. *Linear Algebra - and Its Applications*. Addision Wesley, 2003, 3. ed.

[18] Dr Simon W. Lewis. History of fingerprints. `http://www.deakin.edu.au/forensic/Chemical\%20Detective/fingerprint_his%tory.htm`. Site omrking kriminal efterforskning og teknikker brugt deri, set 26-03-03.

[19] Interview med Kriminalkommissær Kristian Rokkjær. Sted: Centralbureau for identifikation, slotsherrensvej 113 kbh. Tlf: 33910910 -¿ 9422. Ekskursion til København d 08-04-03.

[20] Mette Bom og Ida Leisner Morten Jastrup. Kroppen som identifikation. `http://www.tekno.dk/subpage.php3?article=585&language=dk&category=6&top%ic=k`, Oktober 2001. set 25-04-03.

[21] Anne Skare Nielsen. Etik for begyndere. `www.cifs.dk/scripts/artikel.asp?id=282`. set 29-04-03.

[22] Department of Computer Science at the University of Saskatchewan. Network security technology - introduction to biometrics. `http://www.cs.usask.ca/undergrads/der850/project/biometrics/`. Omhandler biometri generelt, set 26-03-03.

[23] Arne Jensen og Anders la Cour-Harbo. *Ripples in Mathematics - The Discrete Wavelet Transform*. Springer, 2001.

[24] British Telecom og Joh.Enschedé. Test reports on the daugman algoritms. `http://www.cl.cam.ac.uk/users/jgd1000/iristests.pdf`. Link fundet fra John Daugman hjemmeside, set 14-04-03.

[25] Hans Siggaard Jensen og Ole Skovmose. *Teknologikritik*, volume 142 sider. Systime, 1986.

[26] Niels Berg Olsen. Iøjnefaldende adgangskode. `http://www.ing.dk/apps/pbcs.dll/article?Avis=IG&Dato=20010811&Kategori=%ARKIV&Lopenr=11005001&Ref=AR`. Omhandler iris scanning via John Dauhmans metode, set 26-03-03.

[27] Dr. Despina Polemi. Biometric techniques: Review and evaluation of biometric techniques ... `ftp://ftp.cordis.lu/pub/infosec/docs/biomet.doc`. Omkring biomteriske teknikker og deres brug, set 05-04-03.

[28] London Metropolitan Police. Sir edward henry. `http://www.met.police.uk/so/100years/henry.htm`. Kilde om sir Edward Henry fra London Metropolitan Police hjemmeside, set 07-04-03.

[29] Søren Barelebo Rant, Henrik Nerup og Wenneberg. *Info- Samfundet, mellem teknik og etik [Kapitel 8]*. øknom, 1997 2 udg.

[30] Retsinfo. Bekendtgørelse af lov om rettens pleje, kapitel 72. `http://www.retsinfo.dk/_GETDOC_/ACCN/A20020077729`. set 23-04-2003.

[31] Retsinfo. Lov om behandling af personoplysninger. `http://www.retsinfo.dk/_GETDOCI_/ACCN/A20000042930-REGL`. set 18-4-03.

[32] Skib. Fingeraftryk skal identificere flyvevåbnets ansatte. `http://politiken.dk/VisArtikel.sasp?PageID=256995`. Online artikel omkring flyvevåbenets fremtidige brug af fingeraftryk, set 19-03-03.

[33] Peter R. Turner. *Guide to Scientific Computing*. Macmillan Mathematical Guides, 2000, 2 udg. Kapitel 4.

[34] ukendt. The squeeze page. `http://www.cs.sfu.ca/cs/CC/365/li/squeeze/Huffman.html`. Side som beskriver forskellige Loss Less komprimeringer. Set d. 14 - 05 - 03.

[35] Irma van der Ploeg. The illegal body: 'eurodac' and the politics of biometric identification. *Kluwer Academic Publishers*, pages 295–302, 2000.

# Bilag A

# Appendix

## A.1 Spørgeskema

Vi er en gruppe datalogi- og matematik-studerende ved basisuddannelsen på Aalborg Universitet, som arbejder på et projekt omhandlende biometrisk identificering af individer. Dvs. vi arbejder med at genkende folk på f.eks. deres fingeraftryk.

I denne forbindelse vil vi gerne høre Deres mening om denne praksis og beder dem besvar følgende:

1. Køn:

   | Mand | Kvinde |
   |---|---|
   | ☐ | ☐ |

2. Alder:

   | 15-27 | 28-40 | 41-53 | 54-66 | 67-? |
   |---|---|---|---|---|
   | ☐ | ☐ | ☐ | ☐ | ☐ |

3. Hvad er deres beskæftigelse:_____

4. Har De set eller prøvet en biometrisk scanner før? (F.eks. Iris scanner, fingeraftryk scanner eller stemmegenkendelse)

   | Ja | Nej | Ved ikke |
   |---|---|---|
   | ☐ | ☐ | ☐ |

   *Hvis nej gå til punkt 7*

5. Hvis Ja - Hvorhenne?:

☐ I Danmark
☐ I udlandet

6. Hvilken del af Dem blev scannet?

☐ Fingeraftryk
☐ Iris
☐ Stemme
☐ Andet:_____

7. Vil De føle ubehag ved at lade Deres fingeraftryk registrere?

| Ja | Nej | Ved ikke |
|---|---|---|
| ☐ | ☐ | ☐ |

8. Hvilken af følgende udsagn vil De syntes er den største fordel ved et system baseret på en biometrisk scanning?

☐ At man hurtig kan identificer en person.
☐ At man med sikkerhed kan identificer en person.
☐ At man slipper for at huske pinkoder.
☐ At man undgår at en tredjepart kan bruge ens kort mm.
☐ At man ikke kan få sin "kode" stjålen.
☐ Andet:_____

9. I hvor høj grad finder De følgende udsagn rigtige:

   (a) Mine fingeraftryk kan nemt blive stjålet og reproduceret fra registret

| I høj grad | En del | I mindre grad | Slet ikke | Ved ikke |
|---|---|---|---|---|
| ☐ | ☐ | ☐ | ☐ | ☐ |

   (b) Hvis jeg afgiver mine fingeraftryk kan de bruges til at overvåge min daglige færden

| I høj grad | En del | I mindre grad | Slet ikke | Ved ikke |
|---|---|---|---|---|
| ☐ | ☐ | ☐ | ☐ | ☐ |

10. Er De positivt stillet overfor at biometrisk genkendelse bliver indført som f.eks. en erstatning for pinkode på Dankort?

| Ja | Nej | Ved ikke |
|---|---|---|
| ☐ | ☐ | ☐ |

11. Vil De synes at det er en fordel at overgå til et sådant system?

| Ja | Nej | Ved ikke |
|---|---|---|
| ☐ | ☐ | ☐ |

12. I hvor høj grad forstod De spørgsmålene?

| Høj | Mellem | Lav |
|---|---|---|
| □ | □ | □ |

**Tak for hjælpen!**

## A.2 CD-Rom

På den vedlagte CD-Rom finder i følgende:

- Materiale fra vores besøg hos CFI
- Programfilerne og kildekoden der udgør vores program
- Det materiale der er blevet brugt til testfasen

Derudover er der link til vores hjemmeside, hvor I finder store dele af vores arbejde og vores logbog. Til logbogen kræves der login, dette er følgende:

```
Bruger: vejleder
Kode:   popstar
```

## A.3 Forklaring af diverse C termer

Her er en kort og hurtig oversigt over de termer vi har brugt i forklaringen af vores applikation.

### A.3.1 Typer:

- Int - Et heltal tilhørende $\mathbb{N}$.
- Float - Et tal tilhørende $\mathbb{R}$. Den har en længde som er afhængig af den computer der bruger den.
- Double - Det samme som en Float, men med længden er det dobbelte.
- Char - Et ASCII tegn.
- Struct - En struktur bestående af typer.
- Array - En liste af ens typer.
- Pointer - En pointer peger på en addresse i hukommelsen.
- 2D-Array - En liste af lister.

### A.3.2 Forkortelser:

- GUI - Graphical User Interface - Grafisk bruger grænseflade
- IO - Input og Output