

# Bruk av vannmerke ved reproduksjon av gråtonebilder

EGIL BERNTSEN



**KTH Numerisk analys  
och datalogi**

Examensarbete  
Stockholm, Sverige 2004

TRITA-NA-E04112



Numerisk analys och datalogi  
KTH  
100 44 Stockholm

Department of Numerical Analysis  
and Computer Science  
Royal Institute of Technology  
SE-100 44 Stockholm, Sweden

# Bruk av vannmerke ved reproduksjon av gråtonebilder

EGIL BERNTSEN

TRITA-NA-E04112

Examensarbete i medieteknik om 20 poäng  
vid Programmet för medieteknik,  
Kungliga Tekniska Högskolan år 2004  
Handledare var Jon Yngve Hardeberg  
Examinator var Nils Enlund

## Sammendrag

Ved kopiering av et bilde blir kvaliteten ofte betraktelig forringet. Denne oppgaven har sett på mulighetene for å forbedre kvaliteten på resultatbildet. Ideen var å opprette en sanntidsprofil under kopieringsfasen. Profilen benyttes for korrigerende av de endringer som er påført bildet ved utskrift og innskanning.

En av måtene å gjøre dette på er å legge inn informasjon i bildet før det skrives ut. Informasjonen forteller hvordan bildet egentlig skal se ut. Under kopiering leses informasjonen inn når bildet skannes og inngår så i sanntidsprofilen. Bildet korrigeres så i henhold til profilen før utskrift foretas.

For å kunne bestemme hvilken type informasjon som bør tillegges bildet måtte de faktiske endringene ved utskrift og innskanning kartlegges. Studiet viste at intensiteten i bildet ble endret betraktelig. I tillegg endres geometrien i form av rotering og skalering.

Med resultatene fra denne studien som bakgrunn ble det valgt å legge inn informasjon om intensitet i bildet. Videre ble det utviklet deler av en metode som simulerer de reelle endringene som ble funnet.

Videre ble metoden benyttet til å legge inn informasjon før utskrift og finne igjen informasjonen etter innskanning. Metoden håndterer også de geometriske endringene som blir funnet slik at bildet blir rettet opp og får tilbake sin opprinnelige størrelse.

Resultatet viser at det bør være mulig å benytte en slik sanntidsprofil for å forbedre resultatet ved kopiering av bilder.

## **Abstract**

In the case of reproduction of an image the result is often depreciated. The goal of this project is to investigate methods for improving the quality of the resulting image.

The idea was to establish a real time calibration profile during the copying procedure. This profile is intended to be used for error correction of the reproduction to improve the quality of the result.

One way to do this is to embed some information about the image into the image. After scanning this information can be retrieved and used to improve the quality.

What kind of information to be used must be decided. This is information about the original image. What distortions occur in the case of printing and scanning was essential to find out. The study found that there were considerable differences in the intensities before and after printing and scanning. There were also some geometric changes like rotation and scaling.

A method was developed to embed the information about the intensity into the image. The method simulates the real printing and scanning process with changes in intensity and geometric distortions.

The method also retrieves the information after scanning and handles the rotation and scaling.

The result shows that there is a possibility to use a real time profile to improve the quality of the resulting image.

## Innholdsfortegnelse

<b>ORD OG UTTRYKK.....</b>	<b>5</b>
<b>1. INNLEDNING .....</b>	<b>7</b>
1.1. BAKGRUNN .....	7
1.2. PROBLEMSTILLING.....	7
1.3. METODE.....	8
1.4. RAPPORTENS OPPBYGNING OG BEGRENSING .....	8
<b>2. GRUNNLAG .....</b>	<b>10</b>
2.1. HALVTONERASTRERING.....	10
2.2. VANNMERKING.....	12
<b>3. FORSKNINGSFELTET I DAG.....</b>	<b>14</b>
3.1. VANNMERKING AV HALVTONEBILDER.....	14
3.2. BEGRENSNINGER .....	20
3.3. VALG AV METODE.....	21
<b>4. INNLEDENDE FORSØK.....</b>	<b>22</b>
4.1. GEOMETRISKE ENDRINGER .....	22
4.2. STØY OG ENDRING I INTENSITET .....	23
<b>5. ABSTRAKSJON AV BILDET .....</b>	<b>26</b>
<b>6. UTARBEIDELSE AV TESTMETODE.....</b>	<b>31</b>
6.1. METODEBESKRIVELSE.....	32
6.2. OPPRETTELSE AV VANNMERKET .....	33
6.3. HALVTONERASTRERING OG VANNMERKING .....	34
6.4. INNSKANNING OG DEKODING .....	38
6.5. BRUK AV METODEN.....	43
<b>7. FORSØK.....</b>	<b>50</b>
7.1. ROTASJON OG SKALERING.....	50
7.2. FORSØK OG RESULTAT FORETATT PÅ ETT BILDE .....	53
7.3. BEGRENSNING.....	57
<b>8. DISKUSJON .....</b>	<b>60</b>
<b>9. KONKLUSJON OG VIDERE ARBEID .....</b>	<b>63</b>
9.1. KONKLUSJON .....	63
9.2. VIDERE ARBEID.....	63
<b>UTSTYR SOM ER BENYTTET .....</b>	<b>65</b>
<b>REFERANSER .....</b>	<b>66</b>
<b>FIGURLISTE.....</b>	<b>67</b>
<b>TABELLISTE.....</b>	<b>68</b>
<b>VEDLEGG – RESULTATER.....</b>	<b>69</b>
<b>VEDLEGG – KODE.....</b>	<b>72</b>

## Ord og uttrykk

Opplisting av spesielle ord og uttrykk som benyttes i rapporten for å gjøre lesingen lettere.

Dot gain	Når en skriver ut et bilde vil fargestoffet som avsettes på papiret "flyte" noe utover det punktet som var ment dekket.
Terskelverdi	Benyttes når en verdi skal settes til enten 0 eller 1 (normalisert) ut fra om den er høyere eller lavere enn en terskelverdi. Hvis terskelverdi er høyere settes resultatverdi til 0, er terskelverdi lavere settes resultatverdi til 1.
Terskling	Når en benytter terskelverdi. Terskling gjøres gjerne med hele matriser, operasjonen skjer for hvert enkelt element i matrisen.
Dither celle	En dither celle er en tersklingsmatrise, en matrise som består av terskelverdier.
Bins(histogram)	Antall verdier som kan opptre i et histogram, x-akse.
Kontinuerlige toner	Toneovergangen er kontinuerlig, ikke skrittvis. Et vanlig fotografi er av denne typen.
Gråtonebilde	Et bilde der punktene har verdi som ligger på en gråtoneskala. En vanlig gråtoneskala går fra 0 til 255 med trinn på 1. Dette betyr at bilde kan inneholde 256 forskjellige gråtoner, eller et punkt ha kan en av 256 mulige gråtoner.
Halvtone	I stedet for å avsette en farge med riktig tone på papiret avsettes det flere punkter med de fargene som er tilgjengelig i utskriftsutstyret som til sammen utgjør den riktige fargen. Ved høy oppløsning

Egil Berntsen

---

	<p>vil ikke øyet oppfatte forskjellen. I denne oppgaven, som tar for seg gråtonebilder, er det bare svart fargestoff som er tilgjengelig.</p>
<p>Halvtonerastrering</p>	<p>Prosessen der terskling utføres og resultatet er et binært som består av svarte og hvite punkter.</p>
<p>Pseudo-random</p>	<p>Tilsynelatende tilfeldig. Pseudo-random genererte tall bør se tilfeldig utvalgte ut, men generatoren som har generert tallene gjør dette etter et visst mønster.</p>

## 1. Innledning

### 1.1. Bakgrunn

De aller fleste av oss kopierer bilder en gang i blant. Enten på en tradisjonell kopimaskin eller ved å benytte skanner og skriver som er tilkoblet en datamaskin. En slik kopiering vil alltid resultere i en kopi som ikke har samme kvalitet som originalbildet. Denne kvalitetsendringen er ofte markant og kan være et irritasjonsmoment.

De som kopierer et bilde vil i all hovedsak ha en så god som mulig reproduksjon. Det vil være en fordel for alle å få et så godt som mulig resultat ved kopiering, og at en vet hvordan resultatbildet vil se ut etter utskrift. Det optimale her vil være en reproduksjon helt lik originalen.

### 1.2. Problemstilling

Siden kopiering i dag kan resultere i veldig dårlig kvalitet er det ønskelig med en metode som gjør denne prosessen bedre. Ved gjentatte iterasjoner av denne reproduksjonen endres kvaliteten mye. Dette fordi kvalitetsforringelsen tas med i neste reproduksjon og endres til å bli lenger og lenger fra originalen. Dersom en finner en metode som reduserer kvalitetsforringelsen ved hver iterasjon, vil dette resultere i et bedre resultat fordi feilen som videreføres er mindre og vil redusere forringelsen deretter.

Det skal ses på om det lar seg gjøre å finne en ny og bedre metode. Ved å tillegge bildet informasjon om bildet før det skrives ut, og lese denne informasjonen ut av bildet igjen etter innskanning, så kan man bruke denne informasjonen til en prosessering av bildet for å prøve å korrigere de feil som har oppstått under prosessen. Det å legge til tilleggsinformasjon til et bilde på denne måten kalles vannmerking [1].

Informasjonen som tillegges bildet bør ikke være synlig for det blotte øyet, men samtidig må det la seg gjøre å lese det maskinelt. Det vil antakelig være vanskelig å legge inn informasjon i bildet uten å redusere kvaliteten noe. På den annen side kan en slik metode gjøre at kvaliteten ikke forverres så mye for hver iterasjon og dermed kan en muligens godta noe reduksjon i utgangspunktet.

Det finnes i dag utviklede metoder som skal kunne benyttes som grunnlag for å finne hvilken informasjon som bør overføres [2], og hvordan denne kan overføres via det analoge domenet [1] og [7]. Ingen av overføringsmetodene er gode nok fordi de ikke tar nok hensyn til de forandringene som skjer ved en utskrift/innskanningsprosess.



Ved å videreutvikle disse, og tilføre nye elementer, vil de antakelig kunne benyttes i denne oppgaven.

I utgangspunktet kunne en tenke seg en metode som bedrer kvaliteten på en reproduksjon som fungerer på både fargebilder og gråtonebilder. Fargebilder inneholder mye mer informasjon enn gråtonebilder fordi fargerommet for fargebilder har tre dimensjoner, mens det er endimensjonalt for gråtonebilder. For gråtonebilder er det bare forskjellig intensitet på punktene som utgjør et bilde, og det vil være denne intensiteten som endres ved kopiering. Selv om bildene er enklere bygd opp vil det allikevel være interessant å ta for seg slike bilder. I for eksempel skolesammenheng blir det ofte kopiert opp bilder fra artikler eller liknende som deles ut til klassen eller studenten tar en kopi i forbindelse med en oppgave. Videre kan resultatet fra denne oppgaven være til hjelp og benyttes som grunnlag dersom det ved en senere anledning skal utvikles en metode som også kan benyttes på fargebilder.

### **1.3. Metode**

I denne oppgaven er det ikke ment å utarbeide helt nye metoder, men heller benytte eksisterende som grunnlag for videreutvikling. Det finnes metoder og forskningsteori som kan benyttes. Disse vil ikke kunne benyttes direkte, men vil inngå som grunnlag for videre utarbeidelse av metoder som tar hensyn til endringene som skjer med et bilde ved en utskrift/innskanningsprosess, og som kan da benyttes i denne oppgaven.

Det ses på hvor forskningsfeltet står og hvilke metoder som kan benyttes i denne oppgaven. Det en skal komme frem til er hvordan informasjon kan legges til og bli gjenfunnet i et bilde.

For å tilnærme simuleringene som må foretas i denne oppgaven den reelle verden, må det foretas innledende forsøk som forteller hva som skjer med en reproduksjon ved bruk av skriver og skanner.

Resultatene fra de innledende testene må så benyttes videre for å se på om det finnes muligheter for å utarbeide metoder som kan ta hensyn disse endringene.

### **1.4. Rapportens oppbygning og begrensing**

I kapittel 2 ses det på hvilke grunnleggende arbeidsfelt som er tenkt benyttet for å se videre på problemstillingen. Kapittel 3 ser på hvor forskningsfeltet står i dag med de begrensninger som eksisterende metoder innehar. Det velges også hvilke metoder en har tenkt å benytte som grunnlag for å gå videre med oppgaven. Kapittel 4 beskriver de innledende forsøk som måtte gjøres for å finne ut av hvilke hensyn en

må ta for å kunne foreta forsøk ved simulering som er nær det reelle. I kapittel 5 bestemmes det hvilken informasjon om bildet som skal benyttes som en abstraksjon av bildet, og skal legges inn i bildet som et vannmerke. I kapittel 6 utarbeides og beskrives en testmetode der eksisterende metoder ligger i bunn som grunnlag. I kapittel 7 beskrives de forskjellige forsøkene som er foretatt av utarbeidet testmetode for å sjekke om den takler de endringene som skjer ved en utskrift/innskanningsprosess. Videre vises det et spesifikt eksempel på hvordan et bilde kan korrigeres ved hjelp av informasjonen som er tillagt til bildet. Det ses også på begrensningen av hvilke typer bilder metoden kan benyttes på og når metoden feiler. I kapittel 8 diskuteres det hva som er kommet frem under oppgavens gang, og i kapittel 10 konkluderes det og ser på hva slags videre arbeid som kan gjøres for å forbedre resultatene i denne oppgaven.

På slutten av rapporten kommer en oversikt over utstyr som er benyttet under forsøk, figur- og tabelliste.

Som vedlegg til rapporten ligger resultatene fra forsøk og kildekode for testmetoden som er utarbeidet og benyttet under forsøkene.

På grunn av oppløsning på skjerm og skriver vil mange av visualiseringene av bilde kvalitet ikke bli korrekt vist i denne rapporten. Dette gjelder spesielt på papir. Dette må det tas hensyn til når en leser rapporten og vurderer bildene. Bildene blir derfor kun illustrerende.

## 2. Grunnlag

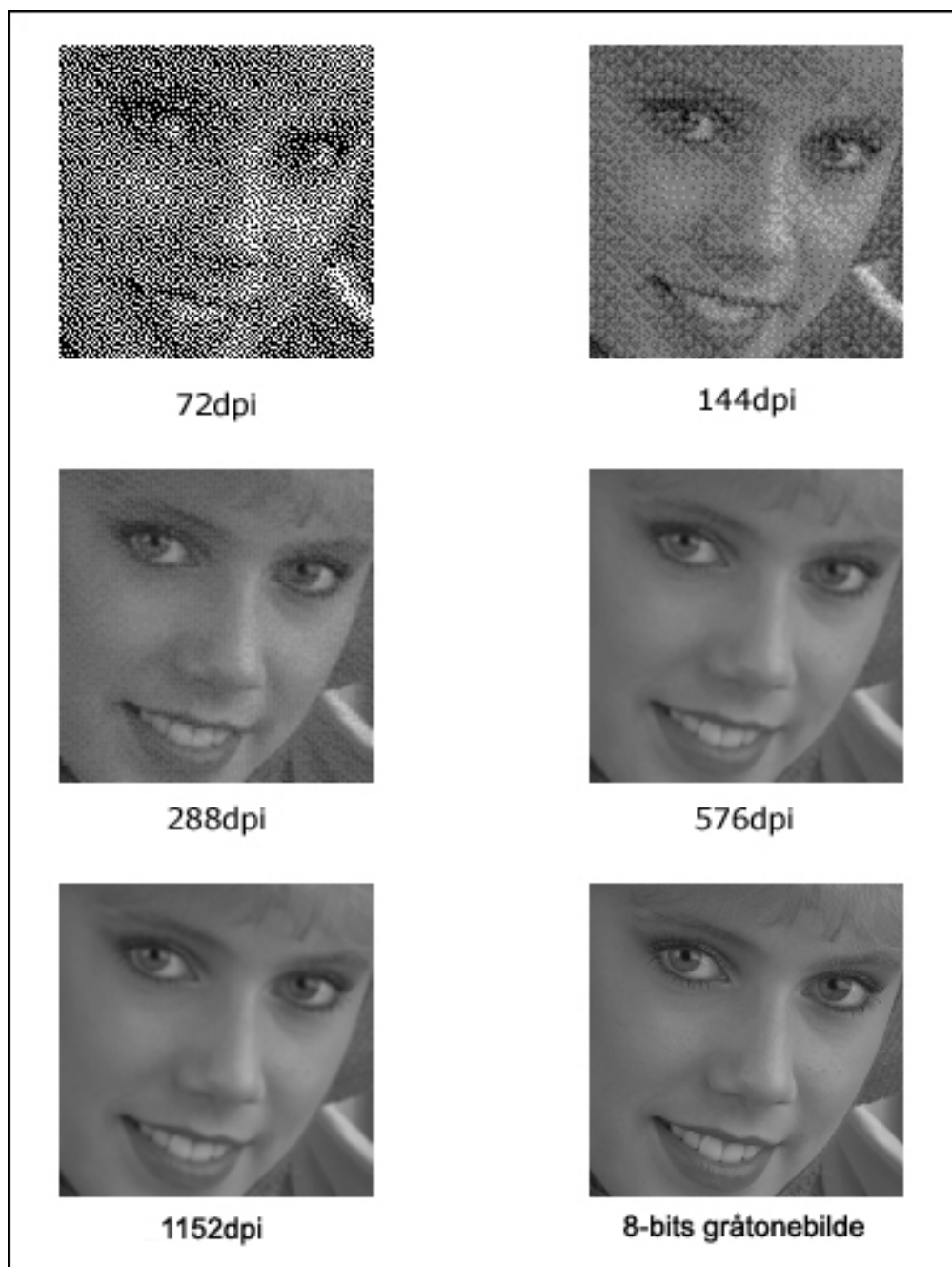
Som grunnlag for denne oppgaven ligger halvtoneastreringen som normalt gjøres av driveren til skriveren. Det er ved å ta over denne delen av en utskriftsprosess som er tenkt benyttet for å legge inn informasjon i bildet, vannmerkingen.

### 2.1. *Halvtoneastrering*

Ordet rastring kommer fra latin, rastrum, som betyr å dele opp [5], og det er nettopp det som skjer ved halvtoneastrering. Et digitalt bilde som er tenkt skrevet ut inneholder normalt fargetoner. Dette kan de fleste skrivere ikke skrive ut.

Ved utskrift avsettes det fargestoff i forskjellige mengder punktvis. Hvordan dette blir gjort bestemmes av programvaren til skriveren. For gråtonebilder vil dette si at enten settes punktet med svart fargestoff ellers gjøres ingenting og punktet forblir hvitt, dette forutsetter hvitt substrat. Til dette benyttes et raster som terskler verdiene fra gråtonebilde, og forteller om punktet skal være svart eller hvitt.

Ved å sette sammen et mønster av svarte og hvite punkter kan man simulere en gråtoneverdi. Slik som Figur 1 viser, vil man med høy oppløsning og ved å se på bildet på litt avstand, oppnå at øyet ikke oppfatter at bildet egentlig består av hvite og svarte punkter. Det vil oppfattes som at bildet har reelle gråtoner. I binære bilder kan man således gi inntrykk av mange gråtoner ved å lage forskjellig mønster som består av punkter som enten samles tett eller mer spredt. Det første gir inntrykk av mørk gråtone, det andre mer lyst. Rastring er således en preprosessering av bilde før det blir skrevet ut.



**Figur 1: Høy oppløsning på binært bilde vil se ut som et gråtonebilde.**

Ved halvtoneastrering i denne oppgaven må det benyttes en metode som på engelsk kalles "Ordered Dither" [6]. Dette vil si at en prosesserer hvert punkt hver for seg og uavhengig av nabopunkter og hvordan utfallet ble av prosesseringen der. Dette må gjøres slik fordi vi ved dekoding av vannmerket ikke kjenner det originale gråtonebildet eller vannmerket, og kan derfor ikke se på hva som er gjort tidligere i prosesseringen.

## 2.2. Vannmerking

Vannmerking er å tilføre ekstra informasjon til et medium. Mediet kan være digitalt eller analogt og mediet kalles i denne sammenheng informasjonsbæreren. Det vil være store forskjeller på hvordan vannmerkingen foretas ut fra om mediet er analogt eller digitalt. Så lenge et bilde holdes digitalt vil det ikke endres utover det som er bestemt av metodene som benyttes på det, her lagres all data og denne endres ikke om den sendes fra en maskin til en annen via internett eller på annen måte overføres digitalt. På denne måten vil en alltid kunne finne igjen informasjon som er lagt til bildet så lenge en kjenner metoden som er benyttet ved vannmerkingen. I analoge medier, det trykte, er dette annerledes. Her vil verdiene på punktene endre seg grunnet begrensninger i det tekniske utstyret og papiret som benyttes, og man vil ikke lenger ha full kontroll slik tilfellet er i det digitale.

Eksempler på bruk av vannmerking er kopirettigheter, ekthetsbevis, skjulte eller åpne meldinger, osv.

Det er flere mål og hensyn det må tas stilling til ved valg av metode for vannmerking [13] og [14].

- Hvor godt skal informasjonen skjules?
- Hvor mye forringelse av kvalitet godtas?
- Hvor mye informasjon skal legges inn i bildet?
- Skal vannmerket være robust eller skjørt?
- Skal vannmerket tåle geometriske endringer av bildet?

Svaret på disse spørsmålene vil avgjøre hva en metode for å legge til og lesing av vannmerke må innbefatte.

I Photoshop fra Adobe finnes det muligheter for å legge inn digitale vannmerker i bilder. Det er Digimarc® som står bak denne funksjonaliteten. Her kan en legge inn kopirettighet og andre attributter. I tillegg velger en hvor synlig vannmerket skal være og hva slags utenhet bildet skal vises på.

Digital vannmerking er det å legge til informasjon til et digitalt medium slik som bilder, film og lyd. Denne informasjonen legges til ved hjelp ett sett gitte regler, en algoritme. Den "omvendte" algoritmen kan så benyttes til å finne igjen denne informasjonen. Vannmerket kan være synlig/hørbart eller ikke, avhengig av hva som er meningen eller grunnen til at vannmerket er lagt til. Videre kan dette vannmerket være skjult slik at ikke andre enn de som skal ha tilgang til denne informasjonen har dette. Dette vil som regel være at man trenger en privat digital nøkkel som inngår i en algoritme for å finne igjen denne informasjonen. Digital vannmerking blir mye brukt for blant annet opphavsrett, spore endringer, opprettelsesdato og liknende.

Selve vannmerkingen i denne oppgaven vil foregå digitalt, men siden vannmerket skal overleve i et analogt medium vil metoder som kan benyttes være en annen enn ved digitale medier. Med analog menes her spesielt papir av forskjellig art. Den mest kjente analoge vannmerkingen er nok av verdipapirer. Dette gjøres for å vanskeliggjøre forfalskning og for å kontrollere ektheten av verdipapiret. Videre vil det enkelte ganger være aktuelt å merke for eksempel bilder med opphavsrett eller annen informasjon.

I teorien er det mulig å tenke seg at en vil klare å skrive ut nøyaktige pikselverdier, det samme gjelder å lese disse inn igjen ved hjelp av en skanner. Dette er ikke tilfellet. Grunnet "dot gain" ved utskrift vil blekk flyte utover den egentlige destinasjon. Denne "dot gain" vil være forskjellig hos de forskjellige skrivers og papirkvalitet. Skanneren vil heller ikke klare å registrere helt korrekte pikselverdier. Til sammen kan denne reproduksjonsprosessen endre kvaliteten på bildet til å bli veldig dårlig.

Når bildet skal gå fra å være digitalt, til å bli digitalt igjen etter å ha vært analogt vil flere faktorer spille en rolle for hva som blir de registrerte pikselverdier grunnet utskrift/innskanningsprosessen. Vi hadde et bilde med kjente gitte verdier så lenge det forble digitalt. Det som vi nå har er et bilde med helt andre pikselverdier enn det opprinnelige.

### 3. Forskningsfeltet i dag

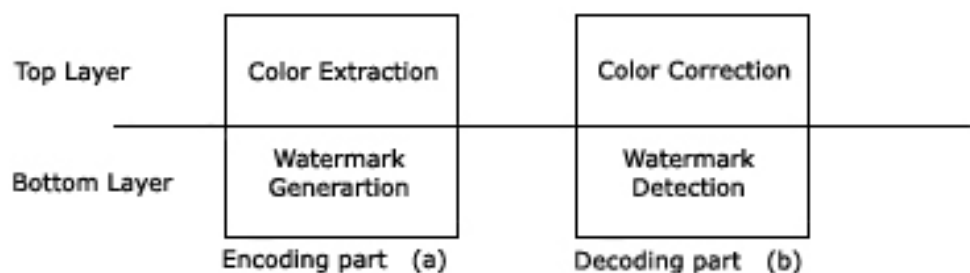
Der er foreslått mange metoder for vannmerking, og forbedring av metoder, av bilder som er ment å forbli i det digitale domenet. De fleste har som mål å ikke være synlige, men de kan ha forskjellige andre mål. Kong et.al [8] foreslår en metode for å legge inn et robust vannmerke for i forbindelse med kopirettigheter. Det samme gjør Liu og Chen [9], Kong et.al. [11] og Pereira et.al [12]. Wu og Tsai [10] benytter vannmerking for overføring av den informasjon brukeren måtte ønske. Når det gjelder metoder for analog vannmerking der informasjonen skal la seg gjenskape etter utskrift/innskanningsprosessen er ikke det samme tilfelle. Det er gjort noe på dette området, men de er mangelfulle.

#### 3.1. Vannmerking av halvtonebilder

Slik vannmerking er tenkt benyttet i denne oppgaven har man kun det innskannede bildet til rådighet når vannmerket skal bli gjenfunnet. Dette betyr at man ikke kan sammenligne de to bildene for å finne differanse mellom dem, eller endringen som har skjedd. Dette vil være avgjørende for hva slags metoder som kan benyttes ved dekoding.

Det er foreslått en metode som skal løse problemet med at feil ved reproduksjon av fargebilder akkumulerer for hver iterasjon av reproduksjonen [2]. Dette er tenkt løst ved å få til en sanntidskalibrering ved hjelp av informasjon som er tillagt bildet.

Metoden som forslås består to deler som vist i Figur 2. Den ene delen tar for seg det å finne informasjonen som vannmerket skal bestå av og legge dette til bildet (a), den andre delen har som oppgave å finne igjen denne informasjonen etter innskanning, og endre bildet ut fra denne (b). Den første delen er plassert i skriveren, den andre i skanneren.



Figur 2: Arkitektur for foreslått metode [2].

Hver av de to hoveddelene i arkitekturen i Figur 2 er igjen delt i to lag, topp- og bunnlag. Topplaget er ansvarlig for fargehåndtering mens bunnlaget er ansvarlig for å legge til og finne igjen vannmerket. Topplaget finner informasjonen om bildet som

skal inngå i vannmerket. Bunnlaget legger inn denne informasjonen i bildet. Ved innskanning finner bunnlaget igjen vannmerket og denne informasjonen benyttes i topplaget til å korrigere fargene i bildet slik at de blir mer lik originalen.

Metoden som er foreslått tar ikke for seg bunnlaget som omhandler implementering av vannmerket. Det skal finnes flere eksisterende teknologier som er robuste nok til å overleve utskrift/innskanning prosessen.

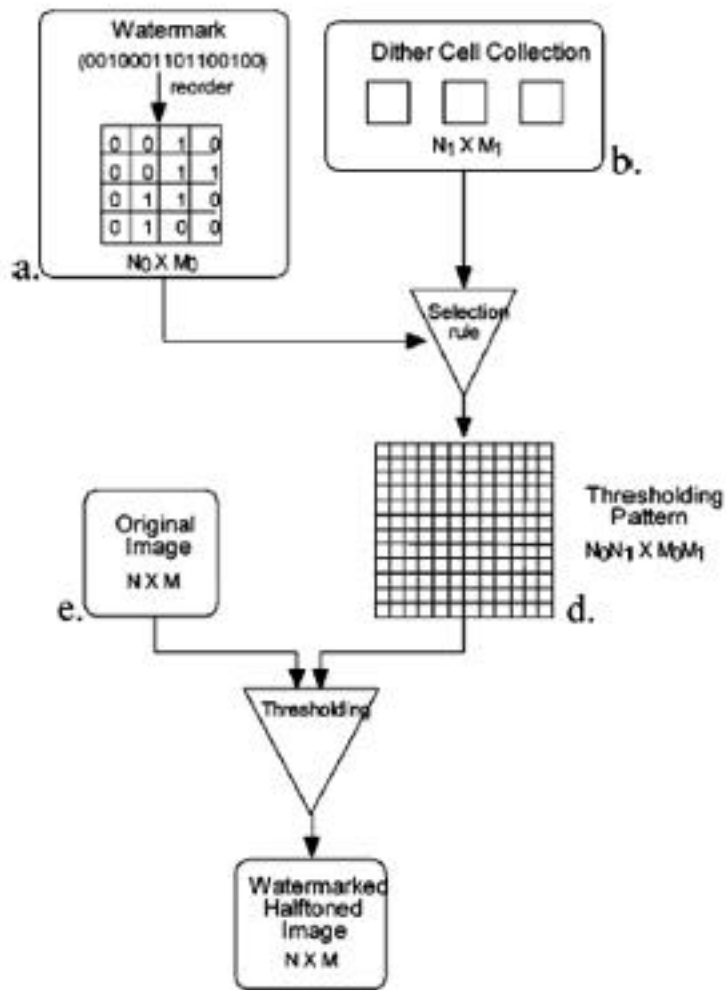
For veldig enkle oppbygde bilder der området bare består av noen uniforme flater, slik som grafer, kan hele bildet komprimeres uten tap til et begrenset sett av binære data som kan legges inn i bildet som et vannmerke. Denne metoden går ikke på mer generelle bilder der oppbygningen vil være mer komplisert. På de enkle bildene vil det derfor være mulig å legge inn informasjon om alle piksler med både farge og plassering, mens det for de mer generelle bildene vil bli for mye informasjon til at dette går. Av denne grunn foreslås det å opprette en abstraksjon av fargene som er benyttet i bildet og la dette inngå i vannmerket. Når bildet skannes inn opprettes det en ny abstraksjon av det innskannede bildet. Disse to abstraksjonene sammenliknes for å opprette en sannhetsprofil. Deretter foretas en korreksjon av pikselverdiene i henhold til den nå opprettede profilen. Hvilken type abstraksjon som skal benyttes er avgjørende for resultatet. En åpenbar mulighet er å benytte histogram, men det viste seg å være meget vanskelig å sammenlikne to fargehistogram for å opprette en fargeprofil. Av denne grunn foreslås det å velge ut ett sett med piksler etter visse regler og legge inn informasjon om farge og lokalisering i vannmerket. Samtidig vil det være nødvendig å legge inn informasjon som sikrer at bildet etter innskanning kan rettes opp slik at det står likt som originalen. Så opprettes en profil ved hjelp av fargepar ved å ta verdiene fra vannmerket og verdier fra det innskannede bildet med samme plassering. Korreksjon foretas så med profilen som er opprettet.

Reglene som benyttes for utvelgelse av informasjon som skal tillegges vannmerket er utarbeidet slik at så få som mulig binære data skal inngå i vannmerket. Metoden som foreslås ender opp med å legge inn fargen og posisjon til 100 representative piksler for et bilde som dekker et A4 ark. Dette vil utgjøre 500 bytes med informasjon hvis en går ut fra en oppløsning på 300dpi.

Hel-Or [1] har beskrevet og testet en metode for vannmerking av bilder som skal overleve utskrift/innskanning med bruk av halvtonestrering. Utgangspunktet for metoden er et gråtonebilde og informasjon en ønsker å overføre i bildet. Figur 3 viser en overordnet oversikt over hvordan metoden blir benyttet. En har informasjon (a) som en ønsker å overføre via det analoge bildet, og selve bildet som skal bære denne informasjonen (e). Informasjonen som skal overføres gjøres binær om den ikke er det i utgangspunktet. Informasjonen som nå er binær ordnes slik at de danner en todimensjonal matrise med en gitt størrelse (a).

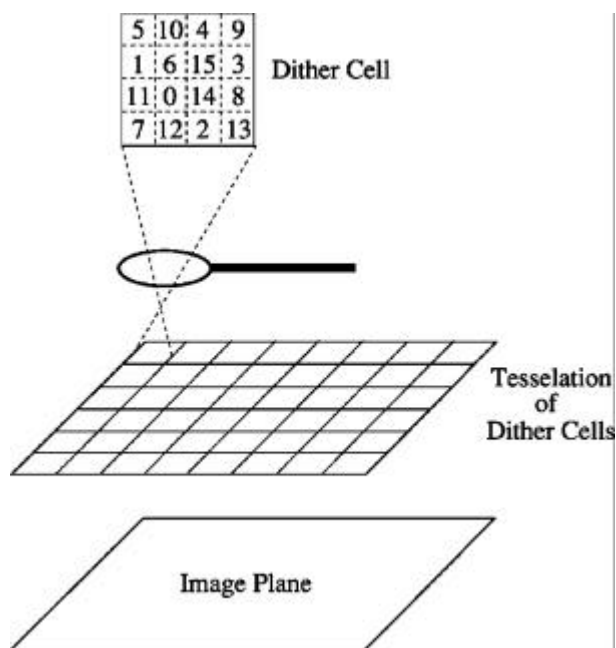


Det utformes deretter en tersklingsmatrise (d) for terskling av originalbildet. Dette gjøres ved at en for hver bit i vannmerket velger dither celle (b) ut fra verdi på vannmerkebit. Disse dither cellene stilles opp slik at de til sammen danner en større matrise (d) og Figur 4. Dette blir tersklingsmatrisen som originalbildet skal terskles på grunnlag av.



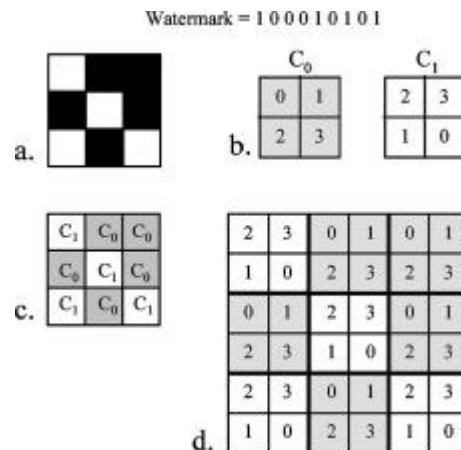
Figur 3: Skjematisk oversikt over vannmerkingsteknikk [1].

Den samme matrisen benyttes flere ganger for rastering av hele bildet ved at matrisen stilles opp ved siden av seg selv, i både vertikal og horisontal retning, så mange ganger det er plass til i originalbildet. Kravet vil være at matrisen ikke er større enn bildet det skal tillegges. I så fall vil ikke et helt vannmerke bli overført.



**Figur 4: Tersklingsmatrisen oppbygning [1]. Matrisen er satt sammen av alle dither cellene som er valgt ut fra vannmerket som skal legges inn i bildet [1].**

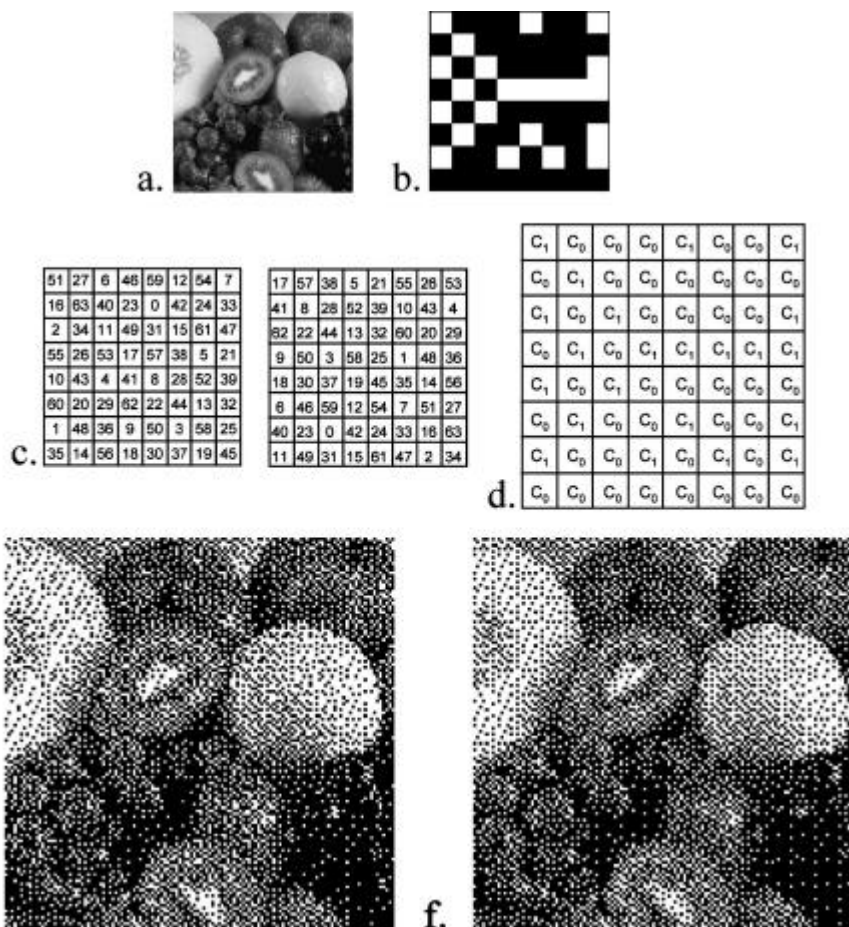
Figur 5 viser et eksempel på opprettelse av tersklingsmatrise. Først ordnes den binære informasjonen som vannmerket skal bestå av som en todimensjonal matrise (a). Når informasjonen som skal legges inn er binær må det være to forskjellige dither celler tilgjengelig (b). For hver bit i vannmerket velges en type dither celle når bit er "1" og en annen når bit er "0" (c). Hver av disse dither cellene er tersklingsmatriser som skal kode inn informasjon om ett bit, de er derfor forskjellige slik at terskling for "0" gir et annet resultat enn for "1". Ved å stille opp dither cellene slik at de danner en større matrise, vil de sammen danne tersklingsmatrisen som skal benyttes ved halvtoneastreringen (d).



**Figur 5: Opprettelse av tersklingsmatrise [1]. Vannmerket med 9 bit stilles opp som en todimensjonal matrise der svart piksel representerer "0" og hvit representerer "1". To dither celler benyttes for terskling der tersklingsverdiene er normalisert til område 0 – 3. For hvert bit i vannmerket velges korresponderende dither celle, C0 for "0" og C1 for "1". De valgte dither cellene stilles opp som en større matrise(d), dette er tersklingsmatrisen som skal benyttes på bildet som skal terskles.**

Figur 6 viser et eksempel på vannmerking av et bilde etter beskrevet metode [1]. En har et gråtonebilde (a) og informasjon (b) som skal legges inn i bildet. Informasjonen er en serie binære data som er stilt opp slik at de danner en matrise. For hver av informasjonsbitene velges riktig dither celle (c) og til sammen vil alle utvalgte dither celler danne en tersklingsmatrise (d) som originalbildet skal terskles med. Resultatet av halvtonestreringen (e) sammenliknes med samme bilde som er rastret uten at det er lagt inn et vannmerke (f).

Når vannmerket skal gjenfinnes må størrelsen på vannmerket og dither cellene være kjente. Da vil oppgaven være å finne ut hvilke dither celler som er benyttet ved vannmerking. Ved å finne hvilke dither celler som er benyttet vil en samtidig finne hvilket bit som er kodet inn siden det er en kjent sammenheng mellom dither celle og bit.



Figur 6: Eksempel på vannmerking av et bilde [1]. Originalbildet som skal vannmerkes(a). Vannmerket på 64 bit(b) er oppstilt som en 8x8 matrise. For hvert bit velges korresponderende dither celle(c) og sammen danner de tersklingsmatrisen(d). Resultatbildet etter rastring(e). For sammenlikning skal (f) vise samme bilde rastret uten vannmerke. Hvilken metode som da er benyttet kommer ikke frem i artikkelen.

Fu og Au har foreslått en ekstremt simpel metode som kalles "Data Hiding Self-Toggling" (DHST) [7]. Denne går enkelt ut på å identifisere noen forhåndsbestemte pseudo-random plasseringer i et allerede halvtonerastret bilde og sette pikselverdien. Disse "tilfeldige" plasseringene kan genereres ved hjelp av en generator med en kjent nøkkel. En eksakt gang i metoden er: 1) Velg ut N "tilfeldige" plasseringer, 2) For hver av disse plasseringene ser en på pikselverdien. 3) Hvis denne er lik vannmerkebit som skal legges inn gjøres ingenting, hvis de er ulike endres verdi til det omvendte.

Fordelen med DHST er at den er veldig enkel og krever lite prosessering. Videre vil det være lett å bestemme mengden informasjon som legges inn ved å kun justere antall "tilfeldige" plasseringer det skal legges inn vannmerkebit. En åpenbar ulempe med denne metoden er at plasseringene som benyttes er helt uavhengig av

bildeinnhold. Siden det ikke tas hensyn til plassering vil det statistisk sett bety at halvparten av plasseringene som benyttes får endret verdi på piksel. Dette vil bety en endring i lokal middelvei og kan gi synlige artefakter.

For å bedre kvaliteten på DHST er det utarbeidet en forbedret metode som kalles "Data Hiding Pair-Toggling" (DHPT). Denne skal rette opp problemet med at lokal middelvei endres fordi ett bit endres. Dette gjøres ved å også endre ett annet piksel slik at middelvei for et område forblir likt. Dette andre pikselet må da endres omvendt av det første. Hvis skifter fra svart til hvit på den "tilfeldige" plassen, skal et annet piksel innenfor et område på 3x3 piksler, skiftes fra hvit til svart. En eksakt metode blir: 1) Velg ut N "tilfeldige" plasseringer, 2) For hver av disse plasseringene ser en på om dette er likt med vannmerkebit som skal kodes inn. 3) Hvis det er likt gjøres ingenting. Hvis det ikke er likt endres pikselverdi på plasseringen og en av kandidatene rundt denne plasseringen som kan skiftes slik at middelvei opprettholdes. Hvis det ikke finnes noen kandidater som kan skiftes foretas ingenting. Dette siste vil skje svært sjeldent og dermed vil lokal middelvei i all hovedsak opprettholdes.

### **3.2. Begrensninger**

Ved metodene som er nevnt der oppgaven er å føre et vannmerke via det analoge domenet er det minst én innlysende begrensning, ingen av dem tar hensyn til at det vil være geometriske endringer i den digitale kopien. Dermed vil det ikke være så rett frem å finne igjen vannmerket i bildet. Fan med flere [2] nevner at bildet må rettes opp før fargekorreksjon kan foretas, men det fortelles ikke hvordan dette er tenkt utført.

Det finnes flere problemer med metoden som er beskrevet av Hel-Or [1]. Uniforme flater som ligger i ytterpunktene av gråskalaen vil ikke bli rastret. Dette er fordi når tersklingsmatrisen "legges over" gråtonebildet vil alle punktene i denne flaten falle på samme side av verdi i tersklingsmatrisen. Dette fører til at alle punktene settes til enten svart eller hvit og dermed blir det ikke noe mønster som en kan lete etter når vannmerket skal gjenskapes. Et bilde som har slike nevnte uniforme flater vil kanskje stort sett bestå av annen type informasjon. Da kan man rette opp problemene ved å legge inn flere vannmerker og benytte seg av en flertallsavgjørelse som bestemmer hvilken bitverdi som med mest sannsynlighet er det korrekte. Dette vil kreve at bildet har et visst minstemål i utstrekning i forhold til størrelsen på vannmerket.

I [1] vises differanse mellom to rastrede bilder, det ene produsert ved hjelp av vannmerkingsteknikken som er beskrevet, og det andre uten vannmerke. Dette gir lite mening da det ikke fortelles om rastreingsmetode som er benyttet for rastreing

uten vannmerke. Det er mulig det her er benyttet den ene av de to dither cellene som brukes ved vannmerking, men det står ikke noe om dette.

### **3.3. Valg av metode**

Som utgangspunkt for videre arbeid velges det å benytte arkitekturen som er foreslått [2]. Denne deler opp oppgavene på en fornuftig måte som består to deler. Den ene plassert i skriveren og den andre i skanneren. Hver av disse delene deles så i topp- og bunnlag. Topplaget på skriveren er ansvarlig for å finne en abstraksjon av bildet som skal legges inn i vannmerket, mens bunnlaget står for selve vannmerkingen. På skanner siden er bunnlaget ansvarlig for å finne igjen vannmerket, mens topplaget står for korreksjon av det innskannede bildet ut fra informasjonen som ble funnet i vannmerket av bunnlaget.

Videre velges metoden foreslått av Hel-Or [1] for selve vannmerkingen. Her skjer vannmerkingen ut fra innhold i bildet, og ikke tilfeldige plasseringer slik som det gjøres i en annen foreslått metode [7]. Det er visse begrensninger i metoden som skal benyttes, men målet er å videreutvikle metoden slik at man kommer over noen av disse hindringene.

## **4. Innledende forsøk**

Utstyret som er benyttet i forsøkene er av forskjellige typer og årgang. Noe er ment for hjemmebruk og annet er ment for kontoret der bruken er hyppigere.

Det skjer en del endringer i bildet ved en reproduksjon. For å gå videre med oppgaven må en finne ut hvilke endringer det er snakk om og i hvilken grad. Det er begrensningene i det tekniske utstyret og substratet (papiret) det skrives ut på som er grunn til endringene, samt rotering og skalering.

### **4.1. Geometriske endringer**

Når en skal skanne inn et bilde må brukeren ofte plassere arket på skannerflaten selv, dette krever da at brukeren plasserer bildet rett for at resultatet skal bli rett. Det finnes tre hovedmåter for brukeren å plassere arket med bildet på skannerplaten. En kan legge arket inn i hjørnet. Dette er antakelig den mest brukte metoden hvis papiret bildet er på har rette hjørner. Den andre metoden er å benytte et ruteark som monteres på skannerflaten. Brukeren legger så arket slik at det flukter med rutenettet. Den tredje måten å plassere arket på er å legge arket på skannerflaten uten noen spesielle hjelpemidler annet enn øynene og ha kanten som sikte. Det er store forskjeller på hvor rett man legger arket på skannerplaten ut fra måten som benyttes. Det må også undersøkes om det, og i tilfelle i hvilken grad, skalering forekommer ved utskrift og innskanning.

Det er foretatt et forsøk på hvor mye rotasjon og skalering som oppstår. Det er vanskelig å finne ut hvor mye av dette som oppstår i skriveren. Det er prøvd å finne ved hjelp av ruteark og geometri, men det viste seg å være vanskelig å komme frem til et nøyaktig svar. Av denne grunn samles resultatet for rotasjon og skalering for skriver og skanner, Tabell 1.

**Tabell 1: Rotasjon og skalering hos forskjellig utstyr.**

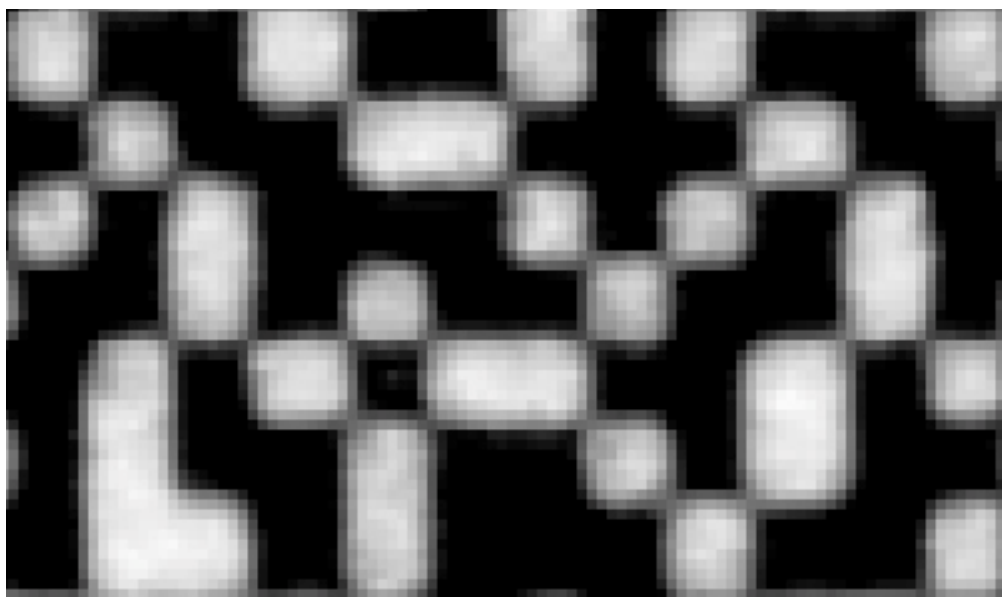
Printer	Skanner	Plassering	Rotasjon °	Skalering %
Epson CX5200	Epson CX5200	Hjørne	< 0,10	< 0,05
Epson CX5200	Epson CX5200	Fritt	0,10 – 0,15	< 0,05
Epson CX5200	Hp Scanjet 5200C	Hjørne	< 0,10	< 0,10
Epson CX5200	Hp Scanjet 5200C	Fritt	0,10 – 0,20	< 0,10
HP5000	Epson E. 1640 XL	Hjørne	0,10 – 0,13	-0,10 – 0,20
HP5000	Epson E. 1640 XL	Ruteark	0,05 – 0,10	-0,10 – 0,20
HP5000	Epson E. 1640 XL	Fritt	< 0,40	-0,10 – 0,20
Canon BJC-6100	HP Scanjet 5200C	Hjørne	< 0,10	< 0,10
Canon BJC-6100	HP Scanjet 5200C	Fritt	0,10 – 0,30	< 0,10
Epson Photo 900	HP Officejet 1170c	Hjørne	< 0,10	< 0,10
Epson Photo 900	HP Scanjet 5500c	Hjørne	< 0,10	< 0,10
HP Officejet 1170c	HP Officejet 1170c	Hjørne	< 0,10	< 0,10
HP Officejet 1170c	HP Scanjet 5500c	Hjørne	< 0,10	< 0,10

I tabellen over resultatet for rotasjon og skalering, Tabell 1, er det brukt forkortet navn på utstyret som er benyttet under forsøket. Korrekt navn og en nærmere beskrivelse av utstyret, og hvilke drivere som er installert finnes i eget kapittel bakerst i rapporten.

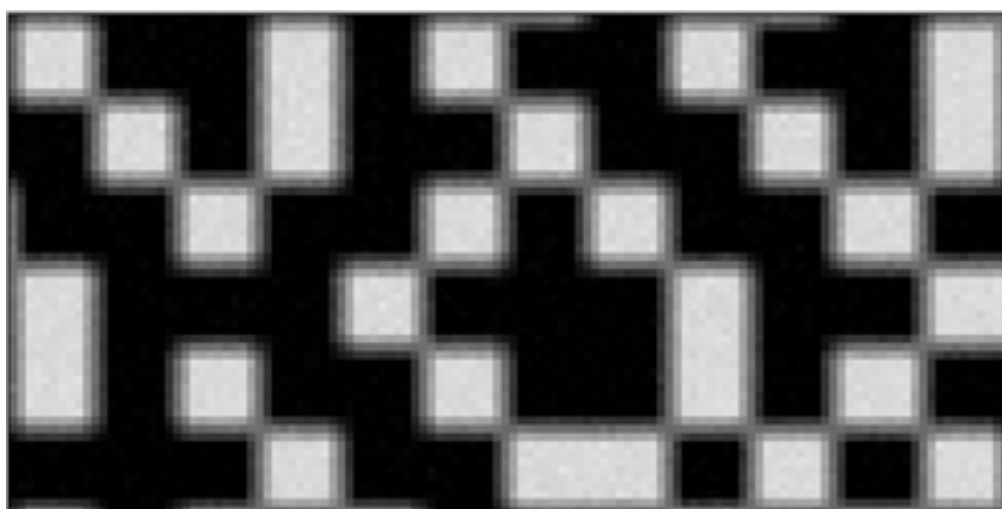
## **4.2. Støy og endring i intensitet**

I tillegg til rotering og skalering vil det skje endringer i bildet grunnet oppløsning og kvalitet på printer og skanner. En viss "dot gain" vil skje når bildet skrives ut avhengig av skriveren og papiret som benyttes. Dette fremkommer som en endring i intensiteten i bildet. Figur 7 viser et forstørret utsnitt av en reell innskanning i lav oppløsning. Denne ble skrevet ut i veldig lav oppløsning slik at en god visualisering av fenomenet kunne vises. Denne endringen må simuleres ved forsøkene som skal foretas for å få simuleringen så reell som mulig. Figur 8 viser en simulering av endringene i intensitet der målet har vært å komme nær den reelle innskanningen.





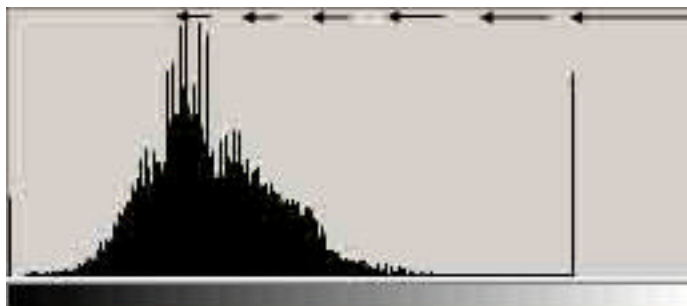
**Figur 7: Forstørret utsnitt av reell innskanning.**



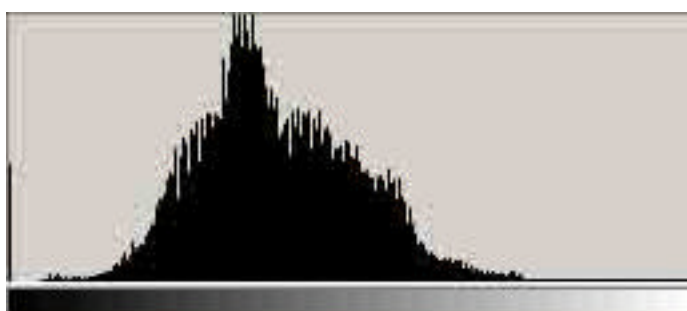
**Figur 8: Forstørret utsnitt av simulert skannet bilde.**

For å komme frem til det simulerte innskannede bildet, Figur 8, er det foretatt noen inngrep på bildet ved hjelp av Matlabfunksjoner. Det ble testet litt forskjellige filter, intensitetsjustering og støy. Etter noen tester kom en frem til et resultat som er rimelig likt det innskannede bildet. Først kjøres det et "average blur filter" på 4x4 elementer, Likning(1). Deretter endres intensiteten slik at svart forblir svart, mens hvitt settes til 85 % av helt hvitt. På Figur 9, som kun er ment som visualisering av fenomenet og som ikke har noe å gjøre med Figur 7 eller Figur 8, kan man se at intensiteten er noe "trekt" sammen, slik pilene markerer, i forhold til Figur 10 som viser histogrammet før en intensitetsjustering. Til slutt legges det til noe Gaussian støy med middelveirdi lik 0 og med 0,0002 som standardavvik.

$$\frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (1)$$



**Figur 9: Histogrammet til et bilde som er intensitetsjustert.**



**Figur 10: Histogrammet til et bilde før intensitetsjustering**

Som en ser av Figur 7 og Figur 8 har en kommet frem til en simulering som er ganske nær den reelle innskanningen, men det er ikke tatt hensyn til at den reelle innskanningen det her sammenliknes med er foretatt i veldig lav oppløsning.

## 5. Abstraksjon av bildet

Her kan man komme frem til at histogrammet er fornuftig...

Histogrammet forteller om bildets oppbygning og intensitet [3]. Det gir informasjon om kontrast og fordelingen av gråtoner i bildet. Histogrammet viser intensitetsfordeling for bildet i sin helhet og sier ingenting om hvordan gråtonene fordeler seg utover i bildet.

Histogrammet beskriver bildets intensitetsoppbygning som et aksediagram der x-aksen viser mulige gråtoner, og y-aksen viser hvor mye gråtonen er representert i bildet. Når en bruker 8 bit for hver gråtone har en 256 forskjellige gråtoner tilgjengelig. Histogrammet kan vise enten eksakt antall forekomster av hver gråtone eller andel av hva hver gråtone dekker av bildets totale antall punkter.

Som man kan se av Figur 11 skjer det en klar endring av intensiteten i bildet ved utskrift/innskanningsprosessen. Vannmerket skal bestå av informasjon som forteller noe om bildet slik at det kan korrigeres. Denne informasjonen kan være en abstraksjon av bildet, og da vil histogrammet være en av kandidatene som kan benyttes for korrigerende etter innskanning [2]. Hvis vi klarer å få histogrammet til det reproduerte bildet til å likne originalbildets vil det si det samme som at vi har klart å minke avstanden i intensiteten mellom de to bildene.

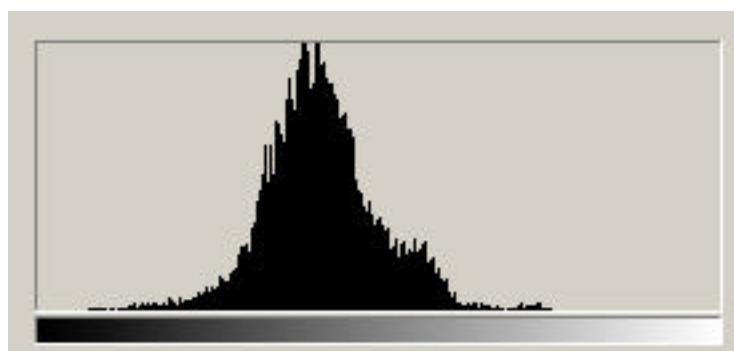
To vidt forskjellige bilder kan ha histogrammer som ligner på hverandre og av den grunn kan man ikke si på generelt grunnlag at selv om histogrammene er like vil bildene være like. I denne oppgaven vet vi at bildene som skal tilnærmes hverandre i utgangspunktet er det samme bilde og på den måten vil tilnærming av histogrammet til originalen endre intensitetsfordelingen til kopien slik at den vil komme nærmere originalbildets intensitetsfordeling.

Det vil kunne oppstå uønskede endringer hvis en prøver å endre intensiteten til et bilde med et histogram som ikke er det korrekte. Dette kan komme av lokal støy. Denne støyen kan komme fra skitt og fremmedlegemer på skannerflaten, papiret kan ha feil, eller det har skjedd feil ved utskrift slik at det har kommet feil mengder fargestoff på papiret. Da vil histogrammet som legges inn som vannmerke ha feil utforming, og kopien som prøves rettes opp kan bli prøvd rettet opp med feil grunnlag og resultatet blir uønsket. Ved første iterasjon vil ikke dette være et problem. Histogrammet som er hentet ut fra vannmerket vil være det korrekte ved første iterasjon fordi støy ennå ikke kan ha forekommet siden dette er histogrammet til det digitale originalbildet. Ved neste iterasjon reproduksjon er utgangspunktet det reproduerte bildet og der støy kan forekomme.



**Figur 11: Kvaliteten reduseres for hver kopi(print/skan). Utført på Epson cx5200.**

En kan ut fra histogrammene til originalbildet, Figur 12, og kopien, Figur 13, som har gått gjennom utskrift/innskanningsprosessen én gang, se at bildet endrer karakteristikk. Det har totalt sett blitt mørkere samtidig som det har skjedd en kontraststrekking.



**Figur 12: Histogram til originalbildet.**



**Figur 13: Histogram til bildet etter innskanning. Utført på Epson cx5200.**

Ved en reproduksjon vil en grunn til at intensiteten endres være at det oppstår en viss "dot gain" [4]. Dette vil si at et punkt som skrives ut vil dekke et større område og dermed vil bildet bli mørkere.

Histogrammet som legges inn i vannmerket skal ha én funksjon, være retningsgivende til hvordan vi ønsker at histogrammet til resultatbildet skal være. Det finnes flere typer eller grader av informasjon som kan legges inn i vannmerket for å

fortelle dette. En åpenbar mulighet er å legge inn hele histogrammet til originalen, enten i form av antall forekomster av en gråtone eller prosentandel av gråtoner i forhold til totalt antall punkter. Det første vil fort gi histogrammer som krever veldig mye informasjon for å gjengi. Den neste vil også kreve relativt mye informasjon som må kodes inn i vannmerket hvis en ønsker et relativt nøyaktig histogram. Et typisk gråtonebilde, Figur 14, har veldig store variasjoner i antall forekomster av hver gråtone, Figur 15. Hvis en velger å representere histogram som hele prosentandeler vil dette føre til at mange innslag vil bli satt til 0. Hvis en ønsker nøyaktighet ved å bruke prosentandel må en benytte to eller tre desimaler. Dette vil også gi relativt mye data som må kodes inn.



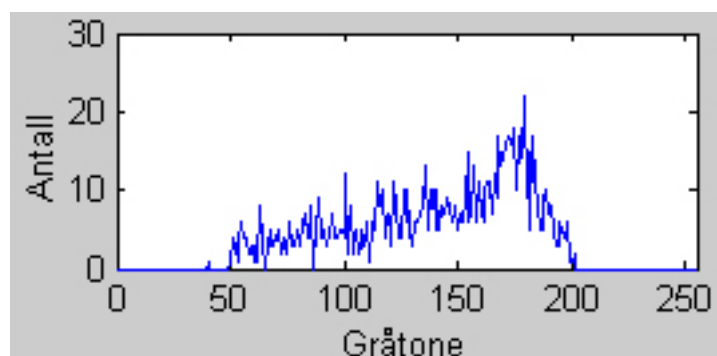
**Figur 14: Lena, et typisk gråtonebilde.**



**Figur 15: Histogrammet til "Lena".**

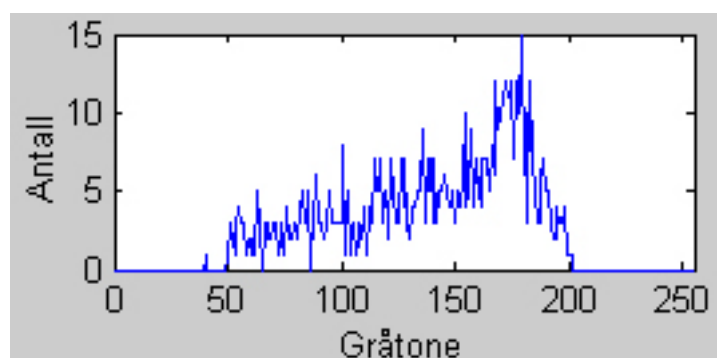
For å endre oppbyggingen av et bilde slik at dets histogram likner et annet bildets trenger man ikke informasjon om hele dette andre bilde. Det vil holde med informasjon som viser tendensen i histogrammet slik at formen på det er likt. Ved å ikke legge inn all informasjon som trengs for å gjenskape hele histogrammet vil en kunne spare plass som vannmerket vil oppta. Her er det to åpenbare muligheter; la histogrammet ha færre bins enn det er gråtoner ved å samle flere gråtoner i blokker eller, å legge inn lavere tall i histogrammet. Begge disse måtene vil gi informasjon om tendensen til hvordan intensiteten i bildet er. Sistnevnte metode er den som benyttes i denne oppgaven.

For å holde størrelsen på vannerket relativt lite ble det bestemt at en ikke ønsket å overføre mer informasjon enn nødvendig. I utgangspunktet vil et histogram samle all informasjon om intensitet i bildet, Figur 16. Ved å konvertere eller normalisere det slik at høyeste antall forekomster av en gråtone settes til en gitt verdi, vil en sikre at antall forekomster kan beskrives med et gitt antall bit og allikevel vil histogrammet beholde formen sin, Figur 17. Hvis en benytter 4 bit for å fortelle om antall forekomster av hver gråtone betyr dette at reelt antall forekomster må normaliseres til området 0-15. Likning (2) foretar denne normaliseringen der  $h_r$  er det opprinnelige reelle antallet og  $h_n$  er den normaliserte av  $h_r$ .



Figur 16: Reelt antall forekomster av gråtoner, bilde med 32x32 piksler.

$$h_n = h_r \left( \frac{15}{\max(h_r)} \right) \quad (2)$$



Figur 17: Normalisering av antall forekomster. Antall forekomster er normalisert til område 0 - 15, bilde med 32x32 piksler.

Når en benytter 4 bit for hver gråtone og en har med 256 gråtoner, trenger man 1024 bit for å kode all informasjon om histogrammet. Dette blir abstraksjonen for bilde, og det er denne informasjonen som vannerket skal bestå av.

Egil Berntsen

---

Av denne grunn vil mye av oppgaven være å overføre histogrammet og tilegne dette til resultatbildet, og som en ser ut fra oversikten over oppgaver som skal løses, Figur 20 litt senere, er mange av disse oppgavene knyttet til histogrammet.

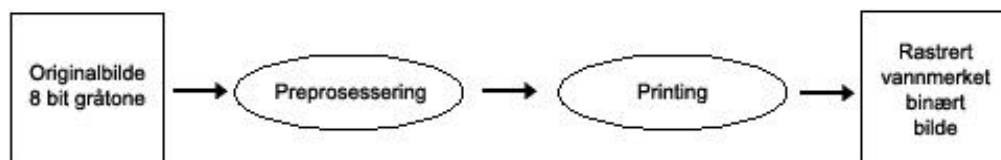
## 6. Utarbeidelse av testmetode

Metoden beskrevet i denne oppgaven baserer seg i all hovedsak på en tidligere metode [1]. Denne er generell i forhold til hva slags informasjon som skal kodes inn. Det er foretatt endringer for å tilnærme metoden den reelle verden. Metoden som benyttes som grunnlag [1] tar ikke hensyn til at det ved en utskrift/innskanningsprosess vil skje endringer i geometri og intensitet og mulig støy i bildet. Dette resulterer i at en ikke vet nøyaktig posisjon og fargetone på punktene.

I denne oppgaven benyttes både halvtonerastrering og vannerke.

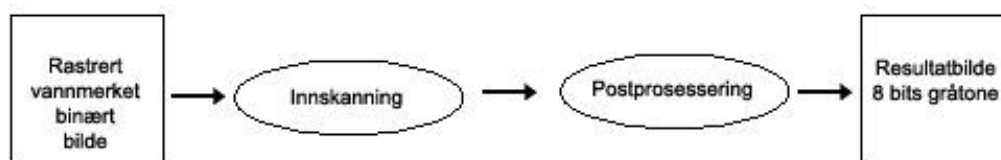
Halvtonerastrering benyttes da mange skrivere kun kan skrive ut binære bilder.

Samtidig vil det være lettere å finne igjen punktene i et rastret bilde etter en innskanning. Punktene er da i utgangspunktet binære og vil være lettere å skille fra hverandre.



**Figur 18:** Bildet må gå igjennom prosessering før det kan skrives ut.

Utgangspunktet er et 72dpi 8 bits gråtonebilde som ønskes skrevet ut på papir. Før bildet kan skrives ut må det klargjøres hvis en ønsker å benytte beskrevet metode, Figur 18. I stedet for at bildet skal gjengis i gråtoner skal det nå gjengis binært, det vil si med bare svarte eller hvite punkter. Det er gjerne programvaren til skriveren som bestemmer metoden dette utføres på. Nå skal vi overta denne jobben og på den måten bestemmer vi hvordan denne rastringen utføres, og dermed har vi også full kontroll over hvordan mønsteret som skal benyttes skal se ut. Vet vi hvordan mønsteret ser ut kan vi også lete etter det samme mønsteret når bildet skannes inn igjen. At vi kan bestemme mønster som benyttes kan vi dermed også si at et bestemt mønster har en bestemt mening. Det er dette som benyttes for å overføre informasjon i bildet på et substrat.

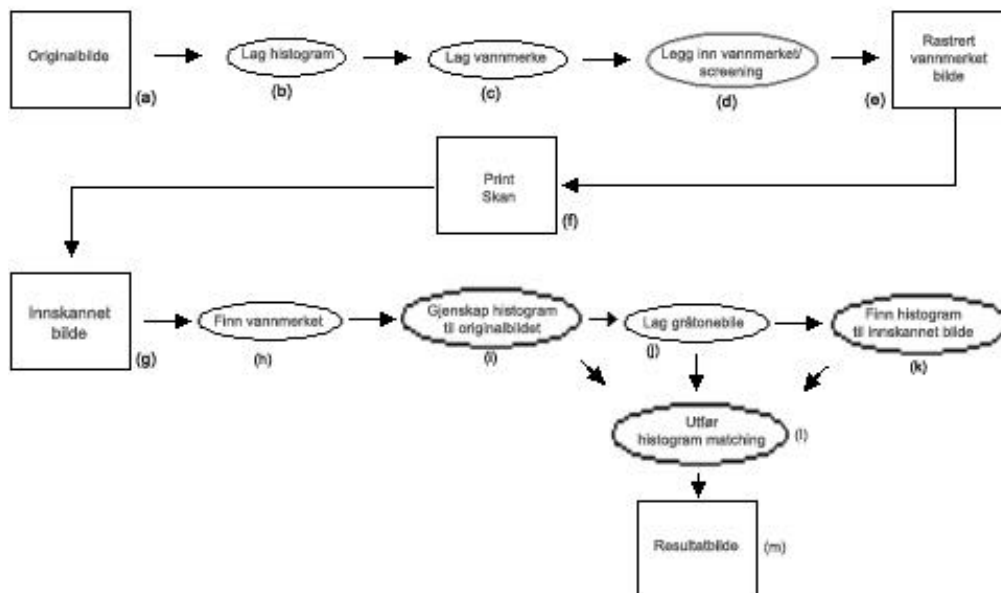


**Figur 19:** Postprosessering. Etter innskanning må bildet gjennomgå en prosessering for å tilnærme seg originalbildet.



Etter innskanning skal bildet gjennomgå en ny prosessering for å finne igjen vannmerket og produsere et resulterende bilde, Figur 19.

### 6.1. Metodebeskrivelse

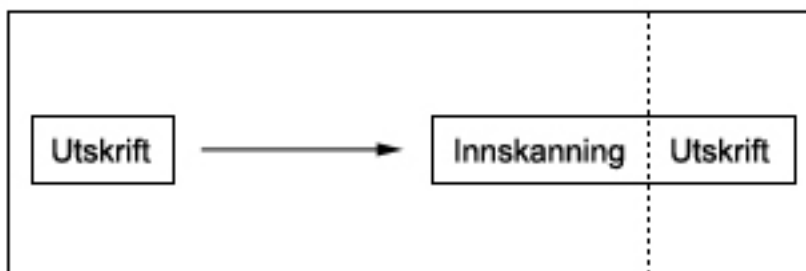


**Figur 20: Arbeidsflyt fra originalbilde til resultatbilde via utskrift og innskanning.**

Fra en har et gråtonebilde som en ønsker å skrive ut, til innskanning og resultatbilde, er det en del steg man må ta i løpet av denne prosessen, Figur 20. Før utskrift må bildet gjennomgå en prosess som klargjør bildet for utskrift og benyttelse av metode ved reproduksjon. Det er i denne metoden fokusert mye på histogrammet til originalbildet. Som det har kommet er histogrammet abstraksjonen som skal overføres med vannmerket. Med originalbildet(a) som utgangspunkt finner en dets histogram(b). Histogrammet er grunnlaget for opprettelse av vannmerket(c). Vannmerket legges inn når bildet gjennomgår halvtoneastrering(d). Bildet er nå halvtoneastrert(e) og innehar informasjon om originalbildets histogram.

Det vannmerkede bildet er skrevet ut i en artikkel eller liknende og noen ønsker å ta en kopi av det(f). Det innskannede bildet(g) må nå gjennomgå en prosess for at det skal tilnærme seg originalbildet. Først finner en vannmerket som er tillagt bildet(h). Ut fra vannmerket gjenskapes histogrammet til originalbildet(i). Videre opprettes et gråtonebilde ut fra det innskannede bildet ved å finne middelverdien av delområder og la dette være verdien i et punkt i det gråtonebildet som nå opprettes(j). Deretter opprettes histogrammet til dette midlertidige nye gråtonebilde(k). Ved bruk av histogrammet til originalbildet som er gjenskapt fra vannmerket, det midlertidige

gråtonebildet og dets histogram, kan en nå opprette et nytt gråtonebilde som vil ha et histogram som er spesifisert til å ligne histogrammet til originalbildet(l og m).

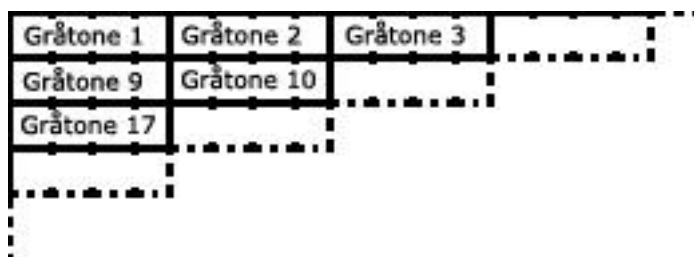


Figur 21: Metoden tar for seg prosess frem til første innskanning

Metoden som utarbeides tar for seg den delen av prosessen der en har det digitale originalbildet og frem til det er digitalt igjen inkludert prosesseringen etter innskanning for å gjøre det klart for ny utskrift, Figur 21. Det ses altså ikke de senere reproduksjonene av denne innskanningen osv.

## 6.2. Opprettelse av vannmerket

I oppbygningen av vannmerket som skal kodes inn i bildet er det histogrammet til originalbildet som brukes som utgangspunktet. Histogrammet har en størrelse på 1024 bit. Denne informasjonen formes slik at den danner en todimensjonal matrise. En matrise på 32 x 32 elementer vil ha plass til informasjonen.



Figur 22: Oppbygning av vannmerket. Histogrammet legges inn fra venstre mot høyre, fra topp til bunn.

Dataene om histogrammet som skal legges i vannmerket, kodes sekvensielt fra svart(0) til hvit(255) ved å starte i øvre venstre hjørne og gå mot høyre og ned, Figur 22. På denne måten vil det være informasjon om åtte gråtoner i hver rad i vannmerket. Ved å velge å legge inn informasjonen sekvensielt på denne måten trenger man ikke vite mer enn akkurat det når vannmerket skal gjenfinnes etter innskanning.

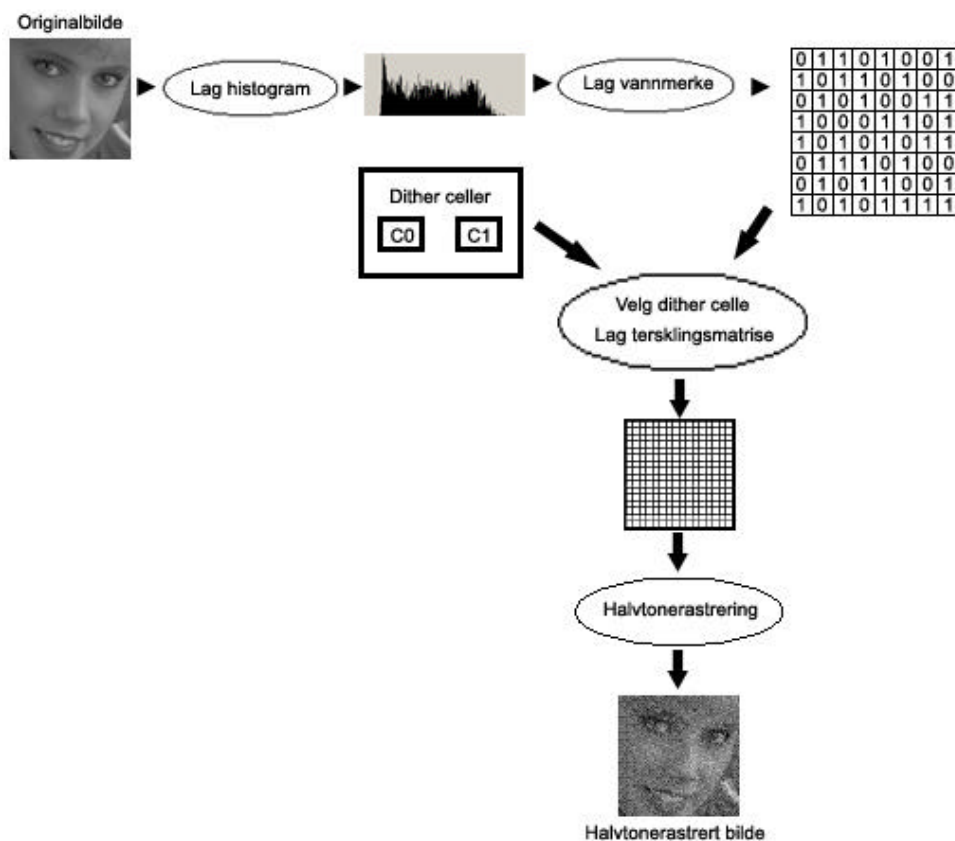
### **6.3. Halvtonerastrering og vannmerking**

Skriveren står vanligvis ansvarlig for rastreringen. Nå skal foreslått metode ta over denne delen slik at man har kontroll på hva som gjøres og benytte det for å kunne legge inn informasjon i bildet.

Ved å legge inn et slags mønster etter bestemte regler vil det være mulig å lete etter dette mønsteret etter innskanning og på den måten finne igjen hva som ble lagt til

Originalbildet som skal klargjøres for utskrift her har en oppløsning på 72dpi. Dette er høy nok oppløsning når det benyttes 256 gråtoner(8 bit) og vil gi et bilde som ser ut til å ha kontinuerlige gråtoner. Ved rastrering gjøres bildet binært, altså representert med svarte og hvite punkter. 72dpi for rastrerte bilder blir for lavt og rastreringen vil synes godt, også på litt avstand. Av denne grunn økes oppløsningen på bildet som første steg i denne preprosesseringen av bildet før utskrift. I denne oppgaven benyttes en oppløsning på åtte ganger oppløsning på originalbildet. Dette gir en oppløsning på 576dpi. Dette vil være høy nok oppløsning til at det vil føles som et kontinuerlige tonet bilde.

Hel-Or [1] legger grunnlaget for metoden som skal utarbeides og den er generell i forhold til hvilke data som skal legges inn i vannmerket. Her skal informasjon om histogrammet til originalbildet legges ved. Figur 23 viser en oversikt over gangen i hvordan metoden er tenkt benyttet.



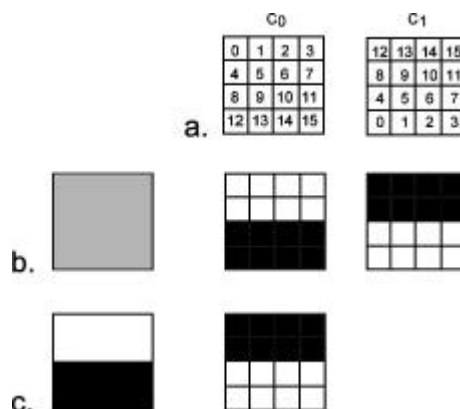
**Figur 23: Foreslått metode.**

Når et bilde skal gjøres binært benyttes dither celler som "legges over" bildet som skal gjøres binært.

Ved implementering av vannmerkingsteknikken må utforming av dither celler bestemmes ut fra flere hensyn og kriterier [1].

- Dither cellene bør produsere forskjellig halvtonemønster for alle mulige gråtoner. Jo større differanse det er mellom mønstrene som produseres jo lettere er å bestemme hvilken dither celle som er benyttet. For gråtoner i ytterpunktene av gråskalaen vil forskjellen mellom mønstrene som produseres bli mindre og mindre jo mer ut til ytterpunktene en kommer. Uansett dither celle vil et hvitt felt resultere i et felt der det ikke avsettes fargestoff.
- Dither cellene bør være visuelle kontinuerlige og ikke lage falske konturer eller kanter.
- Dither cellene bør ha korresponderende terskelverdier. Vannmerkingsteknikken baserer seg på at lokal middelvei i det binære

bildet skal være lik middelvei i det originale gråtonebildet. Hvis et område i originalen er uniformt grå vil den beskrevne metoden for å bestemme benyttet dither celle fungere fint. Men områder er sjeldent helt uniforme og da kan det bli vanskelig å bestemme dither celle. Av denne grunn bør terskelverdier i settet av dither celler ikke være for langt unna hverandre, plasseringen av en terskelverdi i en dither celle bør plasseres i nærheten også i andre dither celle. Figur 24 viser et tilfelle der feil dither celle blir valgt ved dekoding av vannmerket. Et uniformt felt med middelvei 8 (normalisert til område 0-15) (b, venstre) er tersklet med dither celle  $C_0$  (b, midten) og  $C_1$ , (b, høyre). Et felt med middelvei 8 (normalisert), men ikke uniformt, er vannmerket med dither celle  $C_0$  (c, venstre) og skal dekodes. Her vil dither celle  $C_0$  være den mest like, og feil dither celle blir dermed valgt.



Figur 24: Valg av feil dither celle ved dekoding [1].

Dither cellene som Hel-Or [1] benyttet blir brukt her med en endring. Dither cellene  $C_0$  og  $C_1$  er i utgangspunktet normalisert til område 0-63. Dette er endret slik at de nå benytter området 0-255, Tabell 2 og

Tabell 3.

Tabell 2: Dither celle 0,  $C_0$ .

206	109	24	186	239	49	219	28
65	255	198	93	0	170	97	134
8	138	45	198	125	61	247	190
223	105	215	69	231	154	20	85
40	174	16	166	32	113	210	158
243	81	117	251	89	178	53	130
4	194	146	36	202	12	235	101
142	57	227	73	121	150	77	182

Utgangspunkt er matrise fra [1], men endret til å terskle gråtoner med verdier fra 0 til 255.

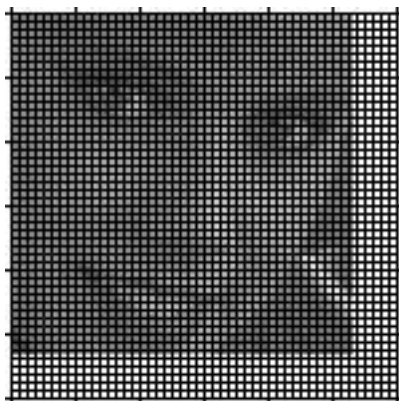
**Tabell 3: Dither celle 1,  $C_1$ .**

49	146	231	69	16	206	36	227
190	0	57	162	255	85	158	121
247	117	210	57	130	194	8	65
32	150	40	186	24	101	235	170
215	81	239	89	223	142	45	97
12	174	138	4	166	77	202	125
251	61	109	219	53	243	20	154
113	198	28	182	134	105	178	73

**Utgangspunkt er matrise fra [1], men endret til å terskle gråtoner med verdier fra 0 til 255.**

Vannmerket tillegges bildet som en del av rasteringsprosessen. Størrelsen på vannmerket er gitt ut fra hvor mye binær data som skal tillegges bildet. Det skal kodes inn 1024 bit og dette tilsvarer en todimensjonal matrise på 32 x 32 elementer.

Vannmerkets størrelse er bestemt til 32x32. For hver av disse bit velges en dither celle på 8x8 elementer. Dette vil da si at tersklingsmatrisen vil få en størrelse som vil dekke 256x256 punkter i et bilde. Samtidig har vi økt oppløsningen på bildet med åtte ganger original oppløsning. Dette betyr at et bilde må minst ha samme størrelse som vannmerket for å kunne benytte denne vannmerkingsmetoden, da er det plass til ett vannmerke. Dette betyr også at så mange hele ganger bildets størrelse er vannmerkets størrelse, samtidig blir så mange vannmerker en kan få lagt til bildet, Figur 25.

**Figur 25: Antall vannmerker som kan legges inn**

Tersklingsceller legges altså ved siden av hverandre til de dekker bildet som skal rasteres. Når bildet skal rasteres benyttes likning(3) for å bestemme verdien for hvert av punktene i det binære bildet som skal opprettes. For ett og ett punkt i det originale gråtonebildet hentes verdien  $f(i,j)$  og sjekker om denne er lavere enn

verdien på tilsvarende plass i terskelmatrisen,  $T(i,j)$ . Er den mindre settes punktet  $y(i,j)$  i det rastrede bildet til 0, ellers settes den til 1. Dette gjøres for alle punkter i originalbildet og resultatet blir et rastret bilde der informasjonen er lagt til bildet som et mønster.

$$y(i, j) = \begin{cases} 0, & f(i, j) < T(i, j) \\ 1, & f(i, j) \geq T(i, j) \end{cases} \quad (3)$$

Bildet er nå rastret og dermed binært, så når det skrives ut foretas ingen videre prosessering slik det ville bli gjort for et gråtonebilde.

#### **6.4. Innskanning og dekoding**

Det vil som nevnt oppstå endringer i bildet ved en utskrift/innskanningsprosess. Metoden som benyttes for å finne igjen vannmerket etter en innskanning må ta høyde for disse endringene.

De geometriske endringene som skjer ved utskrift/innskanning prosess vil være rotasjon og skalering. Dette betyr at punktenes plassering i forhold til hverandre vil være endret, og det vil derfor for eksempel ikke holde med å se bare på neste punkt til høyre i bildet når det er neste punkt til høyre i bildet som ønskes. Punktene må ses på sammen med flere andre punkt, en blokk med punkter. Det binære bildet lages ved hjelp av kjente dither celler, og på den måten kan man søke seg frem i bildet for å finne plassering som likner på disse dither cellene.

Graden av endringer av denne typen er avgjørende for metodevalg. Jo større disse endringene kan være, desto mer må det søkes etter de riktige punktene.

I tillegg til de geometriske endringene som går på plassering, eller lokalisering, av punkter vil det forekomme endringer av selve punktets verdi. Skitt og fremmedlegemer på skannerflaten vil resultere i støy. Videre kan begrensningene hos skriver og skanner endre intensiteten betraktelig slik at resultatet kan havne lagt fra det originalbildets verdire.

Bildet ble gjort binært før utskrift men vil ikke være det etter. Med dette menes det at bildet ikke kun vil bestå av helt svarte og hvite punkter selv om skriveren bare skriver ut med svart fargestoff. Dette skyldes av en viss "dot gain" som forekommer på skriveren. Fargestoffet holder seg ikke bare på destinasjonen på papiret som var ment, men vil flyte noe utover og dekke området rundt til en viss grad.

Ved innskanning må det bestemmes om det skal benyttes binær- eller gråtoneregistrering. Hvis det bestemmes at det skal skannes inn med registrering av gråtoner betyr det at selv om bildet i utgangspunktet var binært vil det etter

innskanning være et gråtonebilde og en kan ikke bare lete etter svarte og hvite punkter. Ved binær innskanning vil resultatbildet bestå av svarte og hvite punkter.

Grunnet endringer i intensitet ved utskrift og innskanning og problemer med oppløsning, blir bildet til dels svært mye mørkere. Ved binær innskanning terskles de registrerte verdiene til enten svart eller hvit ut fra en gitt terskelverdi bestemt av programvaren til skanneren. Dette kan resultere i et bilde med altfor mange svarte punkter hvis en benytter binær innskanning.

Grunnet støy kan ikke en registrert verdi alene ses på som korrekt. I et punkt med kun liten endring i intensitet kan en avgjøre om det skulle vært hvitt eller svart. Ved støy kan denne verdien være helt feil og en kan dermed ikke avgjøre hva utgangspunktet er.

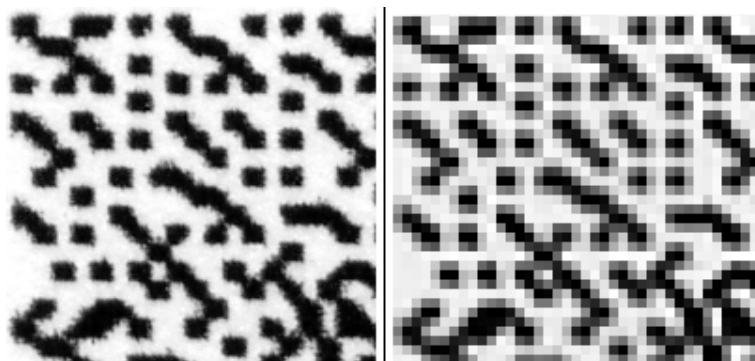
Ved utarbeidelse av metode må det tas hensyn til de forandringene som skjer ved en utskrift/innskanning prosess som er beskrevet.

Når bildet skal skannes inn må man passe på å gjøre dette med en passende oppløsning. Nyquist samplingsteorem, likning(4), forteller at en funksjon med maksimal frekvens  $f_{\max}$  må samples med minimum  $2f_{\max}$  for å finne all informasjon. Her vil funksjonen være bildet som skal skannes og dens frekvens er bildets oppløsning. Dermed må en minst foreta innskanning med dobbel oppløsning som bildet.

$$f_s = 2 \cdot f_{\max} \quad (4)$$

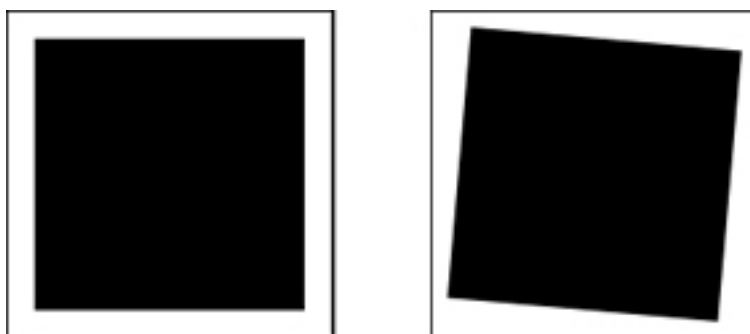
Nyquist's samplingsteorem beskriver den laveste oppløsning som må benyttes. I metoden som er utarbeidet er det valgt en mye høyere oppløsning enn dette minstekravet. Det benyttes en oppløsning på innskanning ti ganger høyere enn bildets oppløsning. Dette er valgt for å få en god avbildning av punktene slik at de skal være lettere å finne igjen ved dekoding etter innskanning. Jo lavere oppløsning som benyttes jo mindre nøyaktig blir gjengivelsen av punktet. Figur 26 viser resultatet av en reell innskanning med oppløsning på 144dpi og 720dpi utført på et rastret bilde med 72dpi.



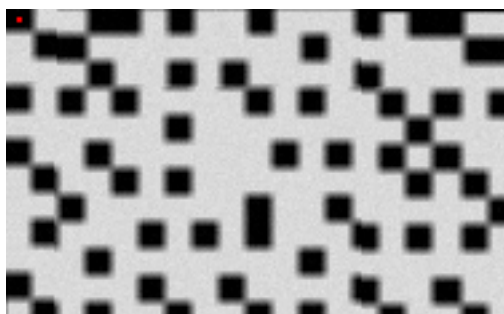


**Figur 26: Innskanning på to og ti ganger bildets oppløsning**

Hvor selve bildet starter vil det være mulig med en algoritme å finne ut. Metoden i denne oppgaven tar ikke for seg denne delen. Av denne grunn vil det være påkrevd å gi posisjon for hvor metoden skal starte letingen etter vannerket i vertikal og horisontal avstand fra øvre venstre punkt i selve bilde. Dette punktet er øverste venstre punkt i det som var originalbildet. Dette vil ikke tilsvare øverste venstre punkt i det innskannede bildet grunnet rotasjonen som forekommer, Figur 27, og den økte oppløsningen, Figur 28.



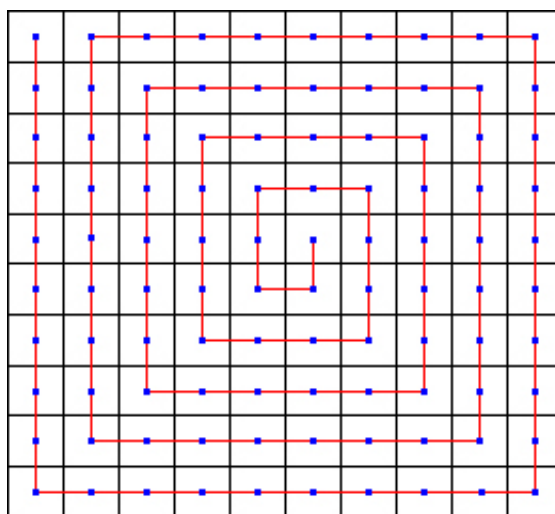
**Figur 27: Startposisjon for å lete etter vannerket. Ved rotasjon av bildet endres startposisjon når vannerket skal gjenfinnes. Metoden som benyttes trenger denne posisjonen.**



**Figur 28: Startpunkt må settes for å finne vannerket**

Ved innskanning vil bildet som sagt endre seg i blant annet geometrien. Derfor må man søke etter vannmerket. Brukeren må sette startpunkt for hvor metoden skal begynne søket. Derfra vil metoden begynne et sirkulerende søk etter hvor det er mest sannsynlig vannmerket befinner seg. Siden bildet er skannet inn med en oppløsning som er ti ganger høyere enn oppløsning på bildet som skannes inn, vil hvert punkt opptre som en blokk på ti ganger ti punkter. Av denne grunn vil startpunkt for metoden være i midten av den blokken som er oppe i venstre hjørne, Figur 29. Så tas et kvadrat som dekker 70x70 punkter av det innskannede bildet. Et slikt kvadrat vil da optimalt sett kunne ha sine fire hjørner i midten av hver sin blokk på 10 x 10 punkter som egentlig er et punkt, og sammen vil disse fire punkt være hjørnepunktene i den opprinnelige 8x8 blokken som ble benyttet for å legge inn ett vannmerkebit.

Øvre venstre hjørnet plasseres over punktet som er satt som startpunkt for søk. Deretter registreres de verdiene som befinner seg i det som optimalt skal være midtpunktet av de blokkene på 10 x 10 som representerer ett punkt i originalbildet. Det samme gjøres mange ganger ved å la kvadratet som dekker 70 x 70 punkter av bildet sirkulere. Figur 29 viser posisjonene for øvre venstre hjørne av denne blokken mens den foretar sitt sirkulerende søk.



**Figur 29: Sirkulerer for å finne riktig plassering.**

For hvert stopp i sirkuleringen tas middelvei av alle punkter i det som kan være en 8 x 8 blokk fra det bildet som skannes, og kjører en simulering med begge dither cellene på et uniformt område med denne middelveien. Så sjekkes differansen mellom resultatet fra disse to simuleringene. Jo større differanse det er jo større er sjansen for at det er nettopp her blokken vi leter etter er. Hvis en større differanse ikke er registrert fra før lagres denne plasseringen som den hittil mest sannsynlige

korrekte. Dette gjøres helt til alle plasseringer er testet. Den plasseringen som hadde størst differanse mellom simuleringen vil nå settes som stedet der blokken mest sannsynlig ligger. Fra denne plasseringen hentes så verdien fra de punktene som er den opprinnelige 8 x 8 blokken og lagres. Den opprinnelige 8 x 8 blokken ble multiplisert opp med 8 som det første som ble gjort før utskrift for å øke oppløsning, og nå tas middelveidi av denne og settes som et punkt i resultatbildet.

Når så neste blokk skal søkes etter, Figur 30, flyttes blokken med 70x70 punkter tilsvarende åtte punkter i det bildet som ble vannmerket. Bildet ble skannet inn med ti ganger så høy oppløsning, og derfor flyttes blokken 80 punkter for å prøve å finne neste plassering. Metoden jobber seg mot høyre og nedover. Når den treffer en kant slutter den å lete i den retningen. Fordi det oppstår rotering og skalering vil ikke disse startpunktene for hver blokk være på linje fordi denne blokken på 70 x 70 punkter flytter seg etter denne endringen. Hvis for eksempel bildet heller ned mot høyre vil 70x70 blokken også bevege seg ned mot høyre.



**Figur 30: Plassering av startpunkt for hver blokk. En del av det vannmerkede bildet forstørret. Viser plassering der det neste blokk for vannmerkebit mest sannsynlig befinner seg.**

Et bilde bør være stort nok til å inneha flere vannmerker. Da utføres metoden likt for hver av vannmerkene. Det er ikke alltid metoden klarer å finne riktig vannmerke og ved å legge inn flere vannmerker kan en ta en flertallsavgjørelse som bestemmer hvilket vannmerkebit som skal registreres som det riktige.

Etter at bildet er skannet inn skal en gjenskape histogrammet til originalbildet. Denne informasjonen ligger i vannmerket, og ved å benytte den omvendte av prosedyren

som benyttes når en lager vannmerket, vil en kunne gjenskape originalbildets histogram.

Etter innskanning lages først et midlertidig gråtonebilde kun med utgangspunkt i det innskannede bildet, og tar dermed i første omgang ikke hensyn til vannmerket som er lagt inn. Gråtonebildet opprettes med en oppløsning på 72dpi. Dette gjøres ved å ta ett og ett felt av bildet og finne gjennomsnittsverdien og la dette være gråtonen i det midlertidige gjenskapte bildet. Dette gjøres samtidig som man søker etter vannmerket. Da går man gjennom hele bildet og når man finner plassen der vannmerket befinner seg hentes verdier ut og et nytt bilde bygges opp. Dette bildet vil da være helt rett siden det som nevnt bygges opp helt fra bunnen av som et nytt bilde.

Dette bildet vil typisk ha et endret histogram i forhold til originalbildet, Figur 37 og Figur 40. For å endre histogrammet til det reproduserte bildet utføres histogramspesifikasjon [3]. Histogramspesifikasjon benytter originalbildets histogram som grunnlag for å endre gråtonebildet som ble lagd ut fra det innskannede bildet, slik at dette får et histogram som er tilnærmevis likt originalens i form.

## **6.5. Bruk av metoden**

Bruken av metoden kunne vært så enkel at man bare sendte med info om hvilket bilde som skal vannmerkes og hva resultatbildet skal hete når det skrives til fil. Men metoden er langt fra ferdig og det er stegene i metoden som må gjøres som er interessante i denne oppgaven her. Av denne grunn er hovedskriptet som skal benyttes stort sett en opplisting av funksjonskall. Metoden er utarbeidet med Matlab® fra The MathWorks Inc. Dette er benyttet fordi nesten alt som skal foretas i metoden er arbeid med matriser, og Matlab® er spesielt beregnet for å jobbe med dette. I dette kapitlet følger et oppsett på hvordan dette gjøres. Det benyttes en del funksjonskall og disse blir forklart på en overordnet måte underveis. Dette gjøres slik fordi det her er ment kun å komme med en oppskrift på hvordan benytte metoden. I vedlegg til rapporten ligger all kode til de forskjellige funksjonene som er utarbeidet. I disse filene er det nærmere beskrevet hva som gjøres. Det benyttes også en del ferdig funksjonalitet som ligger innebygd i Matlab®. Denne blir ikke utførlig forklart men det henvises til dokumentasjonen for Matlab® for å se nærmere på funksjonene.

Funksjonskall gjøres med filnavn og parametere. De utarbeidede funksjonene har fått navn som gjør at det skal være lett å forstå hva de gjør. Videre må filen som lagrer funksjonen ha samme navn som funksjonen med .m som filletternavn. Dette gjør at det vil være lett å se hvilken fil som benyttes ved funksjonskall.

Det første man må gjøre er å lese inn bildet man ønsker å vannmerke, Likning(5).

$$I\_org = imread('barbara.tif'); \quad (5)$$

Dither cellene som benyttes ved rastring og dekoding må opprettes. For opprettelse av disse sendes med tall som forteller om en skal opprette dither celle  $c0$  eller  $c1$ , Likning(6) og (7)

$$c0 = dither(0); \quad (6)$$

$$c1 = dither(1); \quad (7)$$

For å få lagd vannmerket må først histogrammet til originalbildet opprettes, Likning(8). Histogrammet er det som ble bestemt skulle være abstraksjonen til bildet. Bildet som man skal ha histogrammet til sendes som parameter til funksjonen.

$$h = lag\_histogram(I\_org); \quad (8)$$

Når vannmerket skal opprettes sendes histogrammet med som parameter, Likning(9). I retur kommer vannmerket i form av en matrise på  $32 \times 32$  elementer.

$$vm = lag\_vannmerke(h); \quad (9)$$

Nå er vannmerket opprettet og skal tillegges bildet. Men som forklart i utarbeidelsen av metoden økes oppløsning på bildet før dette gjøres siden 72dpi blir for lavt for et binært bilde, Likning(10). Det benyttes bilinear interpolasjon slik at de interpolerte punktene ikke bare tar nabopunktets verdi, men blir en vektet middelverdi av de nærmeste  $2 \times 2$  punktene.

$$I\_r = imresize(I\_org, 8, 'bilinear'); \quad (10)$$

Nå kan bildet vannmerkes, Likning(11). Her må bildet som skal vannmerkes og vannmerke sendes med som parametere. Det må også med de dither cellene som skal benyttes ved vannmerkingen.

$$I\_vm = vannmerk(I\_r, vm, c0, c1); \quad (11)$$

Nå er det vannmerkede bildet opprettet som en matrise. Denne må da skrives til bildefil, Likning(12). Her må det sendes med hvilken matrise som det skal lages et bilde av, hvilket navn denne skal ha og hvilken oppløsning det skal være på bildet. Her må det passes på å sette oppløsning lik det som var utgangspunktet og multiplisere det med samme faktor som ble gjort før vannmerket ble lagt til.

$$imwrite(I\_vm, 'vm_barbara.tif', 'Resolution', (72 * 8)); \quad (12)$$

Nå er det opprettet et binært bilde "vm\_barbara.tif" som er vannmerket med histogrammet til originalbildet. Bildet er nå klargjort for en papirutskrift.

Bildet skrives så ut og skannes inn igjen, og dette skal simuleres.

Som det kommer frem i utarbeidelsen av metoden skal bildet skannes inn med ti ganger oppløsningen på det binære bildet som ble skrevet ut. Samtidig vil det skje en liten skalering i denne prosessen. Begge disse to oppgavene simuleres sammen ved å øke størrelsen på bildet. Det går her ut fra at det skjer en skalering av ved innskanning på 0,1 %, eller 0,01 ganger opprinnelig størrelse. Dette legges sammen med simuleringen av den økte oppløsningen, Likning(13).

$$I\_scan = imresize(I\_vm, (10 + 0,01)); \quad (13)$$

Videre må det simuleres den endringen som skjer ved en utskrift/innskanningsprosess. Først skal det kjøres et "average blur filter" med størrelse 4x4 punkter. Det opprettes, Likning(14) og sendes sammen med bildet, til funksjonen som kjører filteret på bildet, Likning(15).

$$f\_avg = ones(4,4)/16; \quad (14)$$

$$I\_filter = imfilter(I\_scan, f\_avg); \quad (15)$$

Etter at bildet er filtrert skal intensiteten bil bildet endres. Dette gjøres ved at intensitetskurven til bildet justeres slik at det gjøres mørkere på den måten at hvitt settes til 85 % av helt hvitt, mens svart forblir svart, Likning(16).

$$I\_intens = imadjust(I\_filter, [0.0 \ 1.0], [0.0 \ 0.85]); \quad (16)$$

Det siste som gjøres av endringer i intensitet i bildet er å legge til støy. Det benyttes Gaussian støy med middelværdi lik 0 og med et standardavvik på 0,0002, Likning(17).

$$I\_noise = imnoise(I\_intens, 'gaussian', 0, 0.0002); \quad (17)$$

Noe av geometrien ble endret når skalering ble lagt til tidligere, men fortsatt gjenstår roteringen. Dette er siste punkt før simuleringen av det innskannede bildet er ferdig. Her må bildet som skal roteres og hvor mange grader mot klokka det skal roteres, sendes med funksjonen, Likning(18). Her roteres det 0,1 grad med klokka.

$$I\_rotert = imrotate(I\_noise, -0.10); \quad (18)$$

Nå er simuleringen klar og nå kan man begynne på å finne igjen vannmerket som ble tillagt bildet før utskrift. Her må det simulerte bildet, dither cellene og hvor metoden skal starte å lete etter vannmerket sendes med funksjonen. Tilbake får man det gjenskapte vannmerket og bildet som er bygd opp av middelværdier fra delområder av det simulerte innskannede bildet, Likning(19)

$$[gvm, I\_res] = finn\_vm(I\_rotert, c0, c1, v0, h0); \quad (19)$$

Når har man det gjenskapte vannmerket. Fra dette gjenskapes histogrammet til originalbildet, Likning(20). Her må det gjenskapte vannmerket sendes med funksjonen.

$$h\_g = gjenskap\_histogram(gvm); \quad (20)$$

Så kan histogramspesifikasjonen utføres på bildet som ble bygget opp av middelverdier av det innskannede bildet, Likning(21). Her må bildet og det gjenskapte histogrammet sendes med funksjonen.

$$I = histeq(I\_res, h\_g); \quad (21)$$

Nå er det bildet korrigert etter histogrammet til originalbildet og skrives ut til fil, Likning(22). Her må matrisen som innehar verdiene i bildet og navnet på hva fila skal hete sendes med funksjonen. Når fileternavn er kjent oppretter Matlab® et riktig type bilde.

$$imwrite(I, ' Resultatbilde.tif' ); \quad (22)$$

Nå har man det digitale korrigert bildet skrevet til fil og man er ferdig med prosesseringen av bildet.

Metoden er benyttet på et bilde med 32x32 punkter. Vannmerket har 32x32 elementer. Bildet multipliseres opp med 8 før vannmerket legges inn, og vannmerket benytter 8x8 punkter for å legge kode inne ett bit. Dermed er det plass til ett vannmerke. Her er det jukset litt. Metoden som foreligger legger ikke inn informasjonen i vannmerket på en slik måte at mønster unngås. Histogrammet kodes sekvensielt inn i vannmerket og da vil det oppstå mønster. Derfor er det her lagt inn et tilfeldig generert vannmerke ved rastreingen. Bildet som benyttes her er testet med positivt resultat på om det kan holde på et vannmerke og at det lar seg gjenskape. Videre er det benyttet 0,1 % skalering og 0,1 °, noe som man vet er under det metoden håndterer av geometriske endringer. Derfor vil bildene her representere godt hvordan metoden er tenkt brukt, og hvordan bildet endres underveis i prosessen. Man kan ikke se så godt endringene i bildet på grunn av oppløsningen. Derfor vises også et forstørret utsnitt av hvert enkelt bilde for å prøve å få vist det bedre. De to første og de to siste bildene har en mye lavere oppløsning. Derfor kan ikke utsnittet sammenliknes med de fire bildene i midten som har en mye høyere oppløsning.

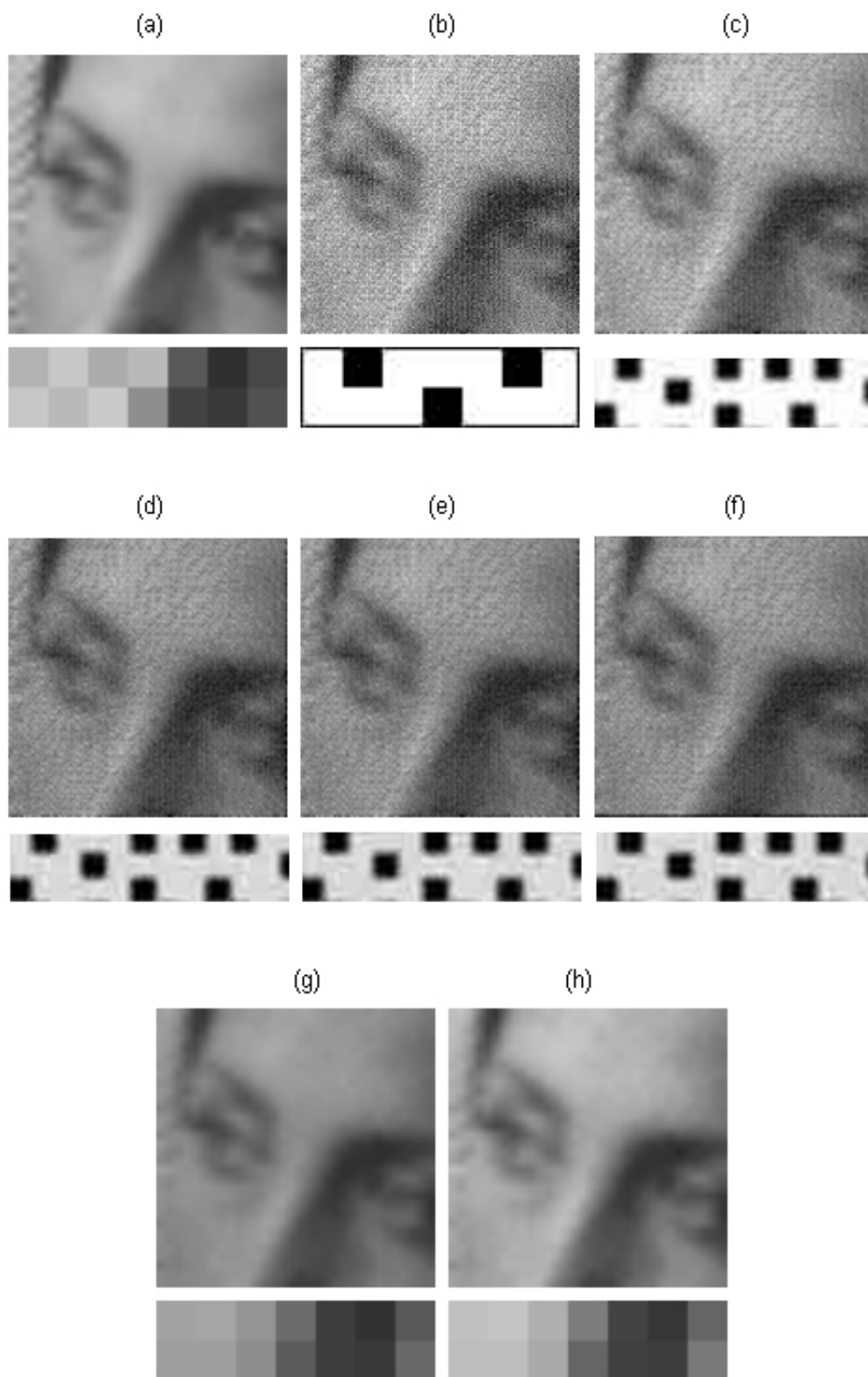
**Error! Reference source not found.** viser bildet etter hvert som metoden endrer bildet. Originalbildet (a) er et 8-bits gråtonebilde på 72dpi. Så legges vannmerket inn (b). Før vannmerket ble lagt ble oppløsning økt til 8 ganger så høy, altså 576. Det er dette bildet som skal skrives ut. Så starter simuleringen med å legge over er "average blur filter" på 4x4 elementer (c). Videre justeres intensiteten (d), det legges

Egil Berntsen

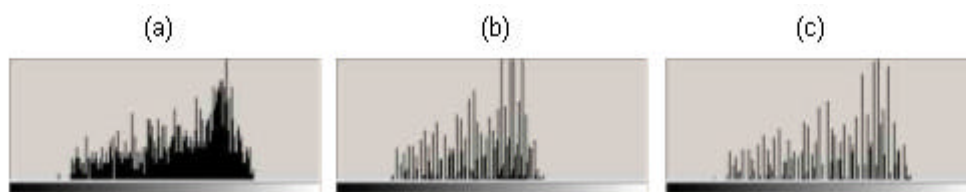
---

til støy (e) og roteres (f). Så leter metoden opp vanmerket og lager et midlertidig gråtonebilde (g). Deretter utføres histogramspesifikasjon på det midlertidige bildet med histogrammet som er gjenskapt ut fra vanmerket (h).





Figur 31: Gjennomgang av metode med visning av endring i bildet.



**Figur 32: Histogrammene til original-, midlertidig- og resultatbildet**

Når en ser på histogrammene for originalbildet Figur 32 (a), det midlertidige gjenskapt bilde (b) og resultatbilde etter histogramspesifikasjon (c) ser man tydelig at bildet som har gjennomgått histogramspesifikasjonen ligger nærmere originalbildet enn det midlertidige gjenskapte bildet. Dette er ikke veldig overraskende siden det er spesifisert hvordan histogrammet skal se ut, og da bør det ikke fjerne seg fra ønsket resultat.

## 7. Forsøk

Forsøkene som er gjort kan deles opp i tre deler. Først blir det testet hvor mye skalering og rotasjon den foreslåtte metoden håndterer før den feiler. Så kommer et spesifikt eksempel på et bilde for å se hvor mye bildet tilnærmes originalbildet med bruk av metoden. Til slutt er metoden benyttet på et sett av bilder for å se hvilke typer bilder metoden fungerer på og når den feiler.

### 7.1. Rotasjon og skalering

Det er foretatt forsøk på hvor godt metoden takler skalering og rotasjon. Det ble benyttet et bilde på 32x32 punkter og et vannmerke på 8x8 elementer., Figur 33. Siden originalbildet multipliseres med 8 før halvtonerastrering og vannmerket legges inn, vil det være plass til 16 vannmerker.



Figur 33: Bilde og vannmerke som benyttes i forsøk.

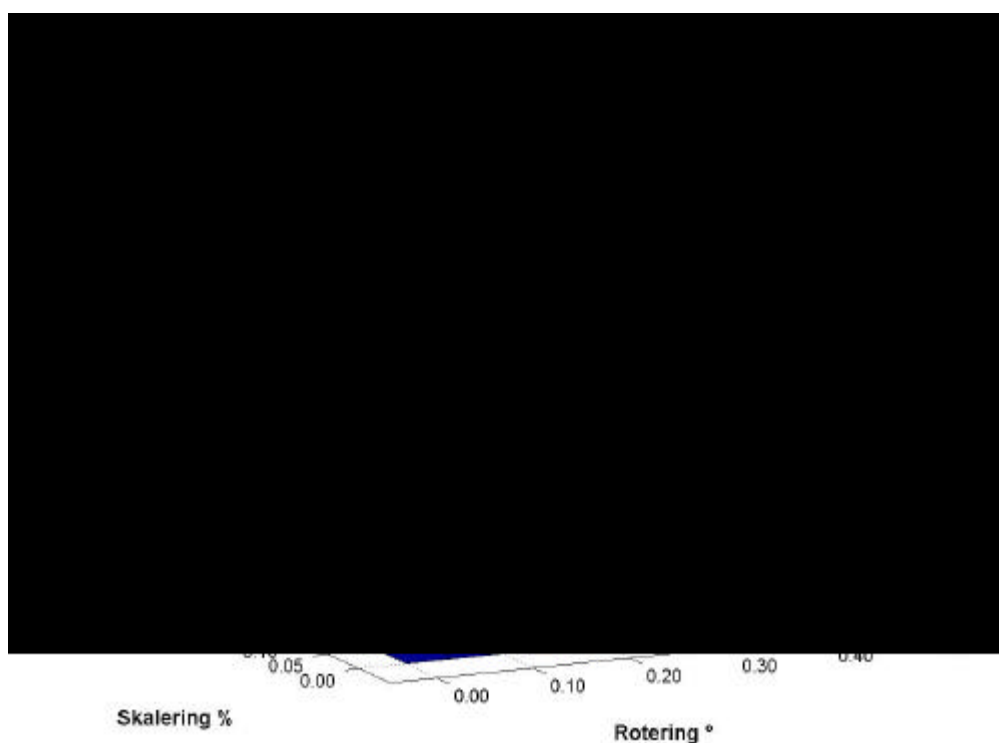
Grunnen til at det ikke benyttes et reelt vannmerke som innehar histogrammet slik det er tenkt i oppgaven, er et ønske om å kunne legge inn et antall vannmerker for å finne grensen der bare noen bit i vannmerket blir gjenfunnet, men allikevel nok til at hele vannmerket kan gjenskapes. Det viste seg at Matlab R13 har visse begrensninger som gjorde at et forsøk med 16 reelle vannmerker på 32x32 punkter ikke ville la seg gjennomføre. Begrensningene ligger i at matrisene som ville ble lagd ble for store til å håndtere. Dette fordi det må benyttes datatypen "double precision" og vekslingsfilen ble for liten. Disse begrensningene vil være rettet opp i R14 som settes opp med større vekslingsfil og med datatypen "singel precision". R14 vil være tilgjengelig høsten 2004.

Testresultatene viser om metoden takler rotering og skalering som testes ut. Disse resultatene må ses opp mot reell skalering og rotering som ble resultatet av de innledende forsøkene.

Det er benyttet en skalering fra 0% til 0,35 % og en rotering fra 0° til 0,5°. Dette er gjort med utgangspunkt i de innledende forsøkene der normal skalering og rotasjon ble funnet for utstyr som benyttes. Det ble foretatt ti forsøk for hvert sett av skalering og rotasjon. Vannmerket består av 64 bit og det ble registrert hvor mange av disse som ble feil registrert når vannmerket skulle bli gjenfunnet. Gjennomsnittet av disse ti forsøkene for hvert sett av skalering og rotasjon er gjengitt i Tabell 4. En grafisk fremstilling av det samme resultatet vises i Figur 34.

Tabell 4: Resultater fra forsøk med rotering og skalering

		Rotering											
		0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50	
Skalering	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,40	3,80	6,20	17,30	
	0,05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,50	7,70	9,80	
	0,10	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,60	2,50	18,90	11,00	
	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,80	0,40	9,00	34,90	
	0,20	0,00	0,00	0,00	0,00	0,00	0,00	0,10	0,00	9,20	25,80	25,90	
	0,25	0,00	0,00	0,00	0,00	0,10	3,00	0,80	13,80	11,50	28,20	18,10	
	0,30	0,00	0,00	0,00	0,00	0,00	1,40	1,50	3,60	3,80	17,20	25,00	
	0,35	0,00	0,00	0,00	0,00	0,00	4,40	12,20	13,30	18,80	23,40	24,50	



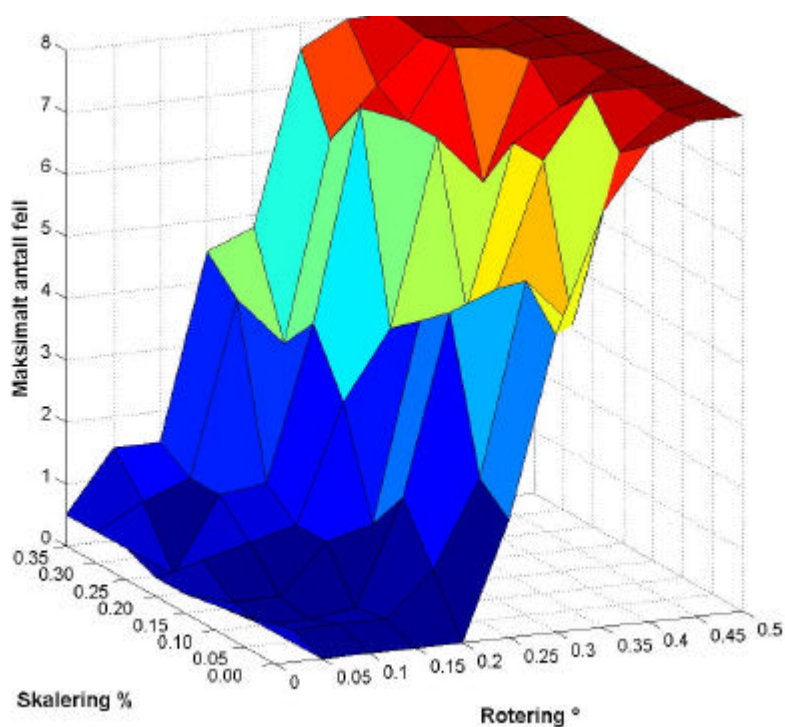
Figur 34: Antall feil bit i gjenskapt vannerke som funksjon av skalering og rotering.

Videre ble det under forsøket sett på antallet bit som benyttes ved flertallsavgjørelsen når det skal bestemmes om det var lagt inn en "1" eller "0" for hver plassering i vannerket. Det ble registrert antallet der differansen mellom de to alternativene var minst. Det ble lagt inn 16 vannerker og eksempelvis kan det for en plassering i vannerket bli gjenfunnet tre 1'ere og tretten 0'ere. Ut fra en flertallsavgjørelse sier vi at det er mest sannsynlig at på denne plassen i vannerket skal det være en "0". De forskjellige plasseringene i vannerket vil ha forskjellig antall som det skal tas en flertallsavgjørelse av. Tabell 5 gjengir snittet av de ti forsøkene som ble gjort for det dårligste resultatet i vannerket for hver gang. Som

tidligere fremkommer resultatverdiene som en funksjon av rotering og skalering. Et resultat på 0,5 betyr at det nesten alltid ble gjenfunnet alle korrekte vannmerkebit hver gang for alle de 16 vannmerkene som var lagt inn. Et resultat på 8,0 betyr at det er helt tilfeldig hva slags vannmerkebit som ble registrert som antatt korrekt. Figur 35 en grafisk fremstilling av det samme tallmaterialet.

**Tabell 5: Differanse flertallsavgjørelse.**

		Rotering											
		0,00	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50	
Skalering	0,00	0,50	0,00	0,00	0,00	0,00	1,90	4,80	6,70	7,70	8,00	8,00	
	0,05	0,50	0,20	0,20	0,20	0,90	2,30	5,40	4,60	7,30	7,90	8,00	
	0,10	0,40	0,00	0,00	0,10	1,80	4,70	5,00	7,00	8,00	8,00	8,00	
	0,15	0,30	0,10	0,10	0,90	1,10	4,30	4,40	7,00	7,50	8,00	8,00	
	0,20	0,30	0,50	0,70	0,90	2,80	3,90	6,90	6,10	8,00	8,00	8,00	
	0,25	0,50	0,00	1,20	1,30	3,80	7,20	6,90	8,00	7,90	8,00	8,00	
	0,30	0,50	0,90	1,20	4,00	3,20	6,40	7,20	7,90	7,80	8,00	8,00	
	0,35	0,50	1,50	1,50	4,50	4,80	7,60	8,00	8,00	8,00	8,00	8,00	



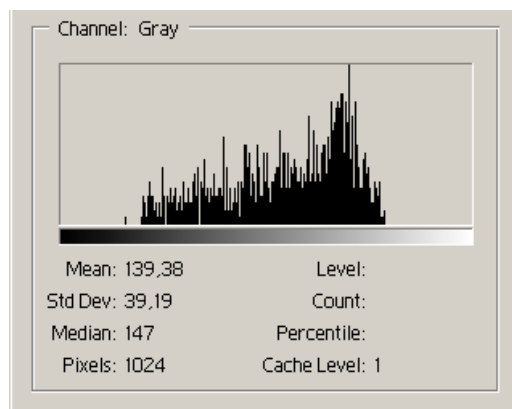
**Figur 35: Maksimalt antall feil ved flertallsavgjørelsen.**

Forsøksresultatene i sin helhet finnes som vedlegg til rapporten.

## **7.2. Forsøk og resultat foretatt på ett bilde**

Her ble det testet på et bilde ved å legge inn ett reelt vannmerke. Ved dette forsøket er rotasjon og skalering lavere enn grensen for hva metoden takler slik at en er sikker på at korrekt vannmerke blir gjenfunnet. Dette ble gjort for å sikre at histogrammet som gjenskapes ut fra vannmerket blir korrekt og kan brukes til histogramspesifikasjon på gråtonebildet som opprettes ut fra det innskannede bildet.

Utgangspunktet er et bilde på 32x32 piksler, Figur 36, med histogram som vist i Figur 37. Figur 38 viser en forstørrelse av dette bildet slik at det er lette å se intensiteten i bildet.

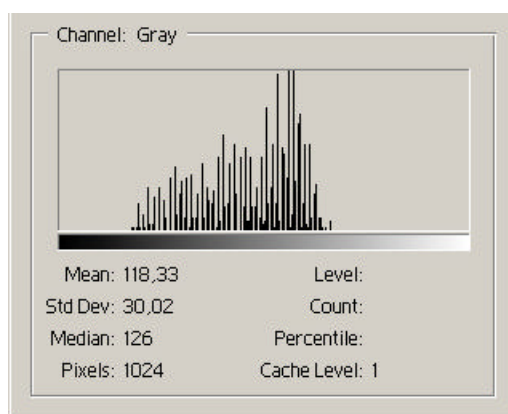
**Figur 36: "Barbara" 32x32 piksler.****Figur 37: Histogram til "Barbara" 32x32 punkter.****Figur 38: Forstørrelse av "Barbara" 32x32 punkter.**

Etter en simulering av en utskrift/innskanningsprosess er det først opprettet et gråtonebilde kun ut fra de registrerte verdier ved innskanning ved å ta middelerverdier av delområder og la dette være verdien i et punkt i bildet, Figur 39. Histogrammet til dette bildet, Figur 40, viser at bildet nå ikke har samme intensitet som originalbildet. Originalen har en middelerverdi på 138, men det gjenskapte bildet har en middelerverdi på 118. Dette kan ikke sammenliknes direkte siden dette gjenskapte bildet er bygget

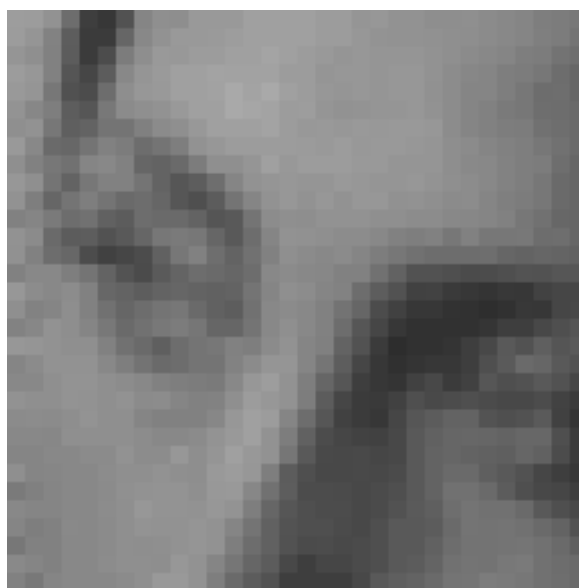
opp ved hjelp av middelverdier av delområder av det innskannede bildet. En forstørrelse av samme bilde ses i Figur 41.



**Figur 39: Gjenskapt bilde.**



**Figur 40: Histogram til gjenskapt bilde.**



**Figur 41: Forstørrelse av gjenskapt bilde.**

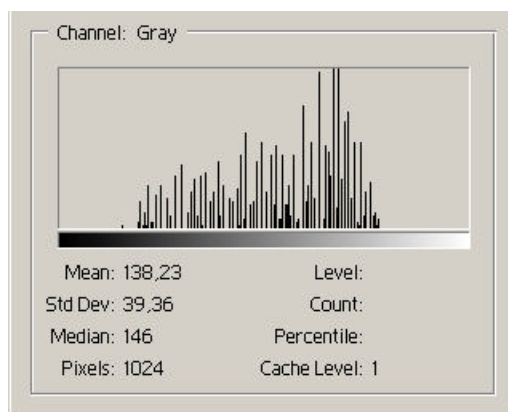
Ut fra det innskannede bildet er vannmerket gjenfunnet og histogram til originalbildet gjenskapt. Dette histogrammet er benyttet til histogramspesifikasjon på gjenskapte bildet, Figur 39, og resultatet er et nytt bilde, Figur 42 der histogrammet, Figur 43, nærmer seg histogrammet for originalbildet. Bredden på histogrammet og middelverdi har nå blitt nesten identisk, men verdiene samler seg noe i grupper slik



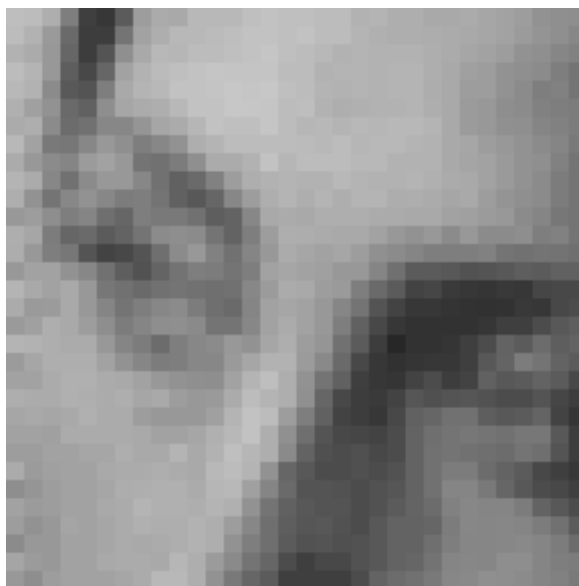
at det ikke er verdier registrert i alle bins i histogrammet. Figur 44 viser en forstørrelse av det histogramspesifiserte resultatbildet.



**Figur 42: Gjenskapt bilde med histogramspesifikasjon.**



**Figur 43: Histogram til gjenskapt bilde med histogram matching.**



**Figur 44: Forstørrelse av gjenskapt bilde med histogram matching.**

Som en kan se både visuelt ved å se på bildet, og ved å studere histogrammene, ligger resultatbildet nærmere originalen hva gjelder intensitet etter histogramspesifikasjonen.

Videre kunne det være interessant å sammenlikne resultatbildet etter en innskanning mot resultatbildet oppnådd med bruk av beskrevet metode. Simuleringen har gjennomgått alle stegene beskrevet i kapittel 6, men som en ser er den ikke nær nok realistisk, Figur 45, og man kan derfor ikke sammenlikne dette direkte.



**Figur 45: Forstørrede bilder, vanlig reproduksjon vs testmetode. Originalbildet(øvre venstre) kopiert på vanlig måte(øvre høyre). Simulert utskrift(nedre høyre) er prosessert med metode(nedre venstre).**

### **7.3. Begrensning**

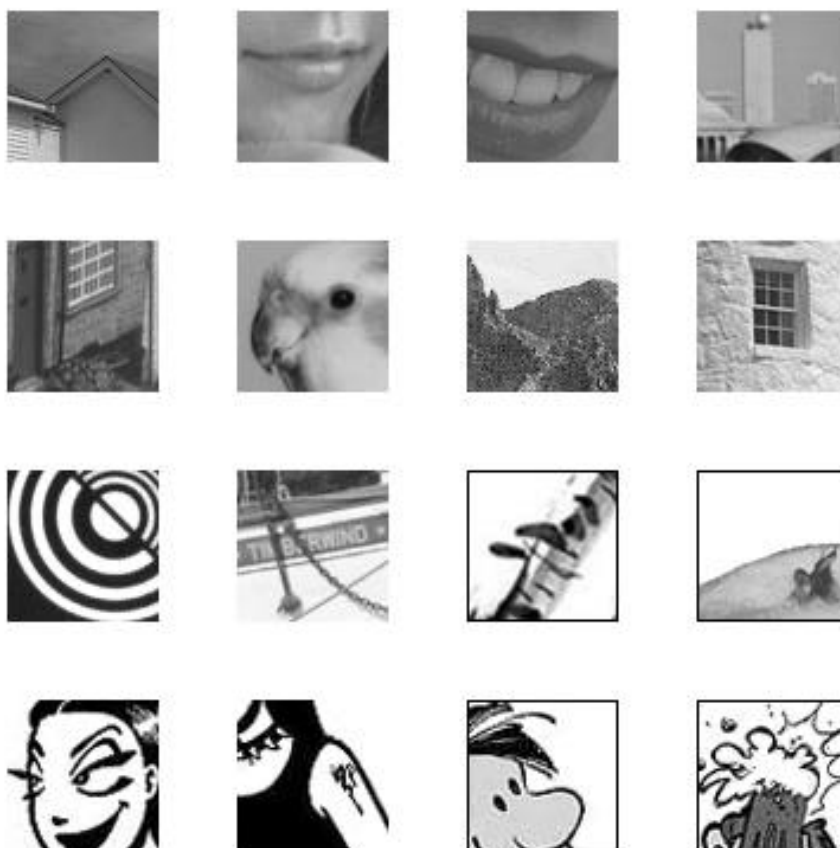
Uniforme gråtonede felter i bildet nær enden av gråtoneskalaen vil ikke la seg halvtonerastrere. Dette fordi alle, eller de fleste, punkter vil falle på samme side av terskelverdi i dither cellen. Et helt hvitt felt forblir hvitt og svart forblir svart, og dermed har man ikke fått lagt inn noe informasjon. Dette betyr at bildets oppbygning betyr mye for om vannmerket lar seg legge inn i bildet og bli gjenskapt. Siden en legger inn mange vannmerker vil plasseringen av disse feltene også bestemme resultatet. Hvis en legger inn mange vannmerker, og feltene i bildet som ikke lar seg halvtonerastrere ikke er plassert likt relativt til vannmerkeplassering, kan allikevel vannmerket gjenskapes siden det tas en flertallsavgjørelse på hvilket vannmerkebit som mest sannsynlig er det korrekte. Av denne grunn er det ikke alltid like

innlysende hvilke bilder som kan vannerkes med positivt utfall, og hvilke som vil feile.

I forsøkene for å se hvilke bilder der metoden fungerer og der den feiler er det benyttet bilder på 64x64 punkter og et tilfeldig generert vannerke på 32x32 elementer. Dermed kan det legges inn 4 vannerker. Videre er det benyttet en rotasjon og skalering basert på hva som er vanlig på utstyr som benyttes. Dette ble testet i de innledende forsøkene og det viste seg rotasjon stort sett lå på under 0,1 grad hvis brukeren legger arket inn i hjørnet på skannerplaten. Videre viste forsøkene at skaleringen for det meste lå godt under 0,1 %. Av denne grunn benyttes disse verdiene i forsøkene for å finne ut hva slags bilder som metoden fungerer på og hvilke på hvilke bilder den feiler. Det må tas hensyn til at det bare legges inn 4 vannerker. Ved å legge inn flere vannerker vil det være større mulighet for positivt resultat på flere bilder. Figur 46 viser et sett med bilder der vannerket ble korrekt gjenskapt. Figur 47 viser et sett bilder der metoden feilet og vannerket ble ikke korrekt gjenskapt.



**Figur 46: Testbilder der vannerket ble korrekt gjenskapt.**



Figur 47: Testbilder der vanmerket ikke ble korrekt gjenskap.

## 8. Diskusjon

Grunnlaget for oppgaven var å se på om man kunne benytte vannmerking som et hjelpemiddel ved reproduksjon av gråtonebilder. Eksisterende metoder og forskningsteori skulle kombineres med nye elementer for å prøve å benytte dette til en alternativ metode som vil gi en sanntidsprofil som ikke vil være utsatt for problemene som oppstår fordi utstyret endrer karakteristika. Det har blitt sett på om det går an å overføre informasjon som legges inn i bildet til å korrigere for feil som oppstår ved utskrift og innskanning. En vanlig skriver kan ikke skrive ut gråtoner, men benytter såkalt halvtonerastrering. Dette betyr at gråtoner simuleres ved å benytte en kombinasjon av svarte og hvite punkter slik at det på litt avstand vil oppfattes som en reell gråtone for den som ser på bildet.

Ved å bestemme hvordan oppsettet eller mønsteret av disse punktene gjøres kan vi lete etter det samme mønsteret etter en innskanning. Ved å la dette mønsteret opptre etter bestemte regler betyr det at vi samtidig kan legge inn informasjon i bildet. Vi kan dermed etter innskanning av bildet lese ut igjen informasjonen når vi vet mønsteret vi skal lete etter.

I denne oppgaven er det prøvd å simulere til en viss grad en reell verden. Problemet med dette er at når man jobber med bilder som har vært digitale gjennom hele prosessen har ikke oppløsning på bildet noen begrensende effekt. Det spiller ingen rolle om man for eksempel jobber med et bilde på 72dpi eller 7200dpi. Når man beveger seg over i det analoge domenet har oppløsning veldig mye å si, og er veldig begrensende. Både det tekniske utstyret og papiret som benyttes har klare begrensninger på hva som kan og ikke kan håndteres. Når noe skrives ut vil en viss "dot gain" forekomme, og ved høy oppløsning vil dette fenomenet påføre altfor mye fargestoff på papiret slik at punkter ikke kan skilles fra hverandre. Her vil papirkvalitet ha mye å si. Papir med glatt overflate vil ha en mindre "dot gain" enn et porøst papir fordi fargestoffet ikke trekkes like mye utover.

Utstyr som skriver og skanner er i stadig utvikling og blir bedre og bedre, og det skjer i rasende fart. Så en kunne tenke seg at oppløsning ikke vil være et problem om noen år. Men det er som nevnt ikke bare utstyret som har begrensninger, papiret er like viktig. Det er ikke så nøye om skriveren kan skrive ut med en utrolig høy oppløsning om papiret ikke klarer å gjengi det. Det er kanskje lite trolig at papiret skal få de egenskapene som trengs for å kunne benytte høy oppløsning.

Histogrammet har hatt en sentral plass i denne oppgaven. Det er benyttet som en abstraksjon av bildet man ønsker å tilnærme seg. Det finnes sikkert andre typer abstraksjoner av bildet som kunne vært benyttet. Metoden som foreligger er utarbeidet kun for å vise at det finnes muligheter, og er på ingen måte en fasit på

hvordan det skal gjøres. Kanskje vil et bedre alternativ være å legge inn histogrammet sammen med annen relevant informasjon i vannmerket. Kanskje er ikke histogrammet så viktig at det bør være med i det hele tatt. Metoden slik den nå foreligger, er tenkt på å kunne benyttes uansett informasjon så lenge denne informasjonen gjøres binær før vannmerket opprettes. All informasjon kan gjøres binær, eneste krav er at den som benytter metoden vet hvordan denne informasjonen ble kodet binært slik at man kan dekode den korrekt.

Størrelsen på vannmerket i denne oppgaven er gjort så liten som mulig slik at det er plass til så mange som mulig. På denne måten vil en flertallsavgjørelse av hvilket vannmerkebit som var lagt inn bli så sikker som mulig. Hvis metoden ble mer intelligent i søking etter riktig vannmerkebit vil det ikke være så viktig å legge inn så mange eksemplarer av vannmerket. Da vil det også være mulig å legge inn mye mer informasjon i vannmerket om det skulle vise seg å være nødvendig eller ønskelig.

Det ble bestemt å benytte 4 bit for å beskrive andel av en gråtone i et histogram, og denne informasjonen skulle legges inn i vannmerket. Den gråtonen med høyest andel vil da representeres som 1111 binært. Alle andre gråtoner vil starte med minst en 0'er. Når disse tallene stilles opp i en matrise vil de danne et mønster. Slike mønster i vannmerket vil være synlig i det vannmerkede bildet i form av rette vertikale og horisontale streker og uniforme felter. For å unngå slike mønster må vannmerket være så randomisert som mulig.

Det finnes minst to måter for å redusere mønster i vannmerket. En vil være å ikke legge inn de binære tallene sekvensielt, men heller å benytte en pseudo-random metode. Denne må da også være kjent når vannmerket skal gjenskapes etter en innskanning. Den andre vil være å benytte flere enn 4 bit for å representere tall opp til 15. På denne måten kan tall representeres på flere måter og dette må da være kjent ved dekoding.

I denne oppgaven er det ikke tatt hensyn til at det vil oppstå mønster. Ved forsøk er det et randomisert vannmerke som legges til, og dette vil da ikke føre til nevnte problem med mønster.

Før metoden ble satt sammen ble det testet hvilke endringer som oppstår i et bilde når det skrives ut og skannes inn igjen. Det oppstår endringer i geometrien i form av rotering og skalering. Videre endres intensiteten over hele bildet, og det oppstår tilfeldig lokal støy. Ved utarbeidelse av metode ble svarene fra disse innledende testene brukt som grunnlag for å prøve å sikre at metoden ville håndtere disse.

Så lenge brukeren legger bildet inn i hjørnet på skannerflaten vil metoden slik den foreligger takle den lille rotasjonen og skaleringen som skjer. Samtidig vil metoden, når den jobber seg gjennom det simulerte innskannede bildet, korrigere for rotasjon og skalering slik at bildet blir rettet opp.

Resultatene av disse forsøkene det vises til må tas med en klype salt. Som nevnt er ikke simuleringen optimal og da kan ikke resultatene fra forsøket heller ses på som reelle.

Halvtonerastrering utføres på de fleste skrivere siden de ikke kan gjengi reelle gråtoner. Denne rastreringen gjøres etter algoritmer som gjerne bare er kjent for produsent av skriveren. Siden denne rastreringen gjøres uansett kan man like godt ta over denne jobben og dermed ha kontroll på hvordan denne gjøres. Forskjellen her blir at metoden som benyttes, og som ligger skjult i programvaren til skriveren, med all sannsynlighet er av bedre kvalitet enn de algoritmene som ligger åpent for alle i form av forskningsarbeider. Teoretisk sett kan de kjente og åpne algoritmene på sikt bli like bra som de skjermede. Det vil kunne si at rastreringen vil kunne skje med like god kvalitet, bare nå med innlagt informasjon som kan leses ut igjen etter innskanning. En forskjell vil alltid være der. Algoritmer der en ikke skal legge inn informasjon kan konsentrere seg om å få rastreringen så lite synlig som mulig. Når en skal legge inn informasjon må en ta like mye hensyn til dette, og dette vil være et hinder for å nå like langt.

Hvis det gikk an å øke oppløsningen mye, og allikevel være mulig å finne igjen informasjonen som er lagt inn i bildet, vil en metode som slik som i denne oppgaven gjøre det mulig å overføre informasjon i bildet uten at det kan oppfattes av det menneskelige øyet. For hvis dette gikk kunne man overføre eksakt informasjon om store deler av bildet, ikke bare en abstraksjon slik det gjøres i foreslått metode.

Ved å benytte en slik preprosessering av bildet etter innskanning vil en ikke bare ha mulighet til å korrigere de endringer som skjer med intensitet og kontrast, men også de geometriske endringene. Her i denne oppgaven ble rotasjon og skalering som oppsto under skriving og innskanning korrigert slik at bildet fikk tilbake sin opprinnelige geometriske form.

I metoden som benyttes i denne oppgaven blir det brukt mye høye oppløsninger. Dette gjelder blant annet ved innskanning. Her benyttes det en oppløsning ti ganger høyere enn oppløsning på bildet som skannes inn. Dette er nok for høyt i forhold til hva det kunne vært. Hovedmålet med høy oppløsning var å få en god avbildning av punktene, og det gjorde det også, i hvert fall i simuleringen. Men det krever mye av maskinvaren som benyttes, og er an grunnene til at metoden ikke kunne benyttes på bilder med litt størrelse. Ved å forbedre metoden kan det være mulig å sette ned oppløsning ved innskanning og på den måten spare ressurser.

## **9. Konklusjon og videre arbeid**

### **9.1. Konklusjon**

Arbeidet som er gjort viser at det bør finnes muligheter for å benytte en slags sanntidsprofil for å forbedre kvaliteten ved en utskrift- og innskanningsprosess. Metoden som er satt sammen av andres arbeider, modifisert og tillagt nye elementer, er langt fra optimal men det gir en pekepinn på at det bør finnes muligheter for å få til en metode som oppnår en sanntidsprofil for korrigerende av et innskannet bilde.

Det største hinderet i denne oppgaven er begrensningene rundt oppløsning. Dette kan bedres i nær fremtid hvis en ser på den rasende utviklingen ellers når det gjelder det tekniske utstyret som benyttes ved utskrift og innskanning enten en snakker om vanlig skriver og skanner eller kopimaskin. Om dette også kan gjelde papirkvalitet er vanskelig å forutse, men utviklingen på dette området går nok noe tregere. Papiret som benyttes ved utskrift i dag er mye av det samme som ble benyttet for noen år siden, mens utstyret er kraftig forbedret.

I denne oppgaven er det tenkt brukt så høy oppløsning på det rastrede bildet at man ikke ser forskjell på dette og et bilde som benytter reelle gråtoner eller har kontinuerlige toneoverganger. Bilder som i dag er skrevet ut på printere som skriver binære bilder kan man se oppløsningen hvis man ser nøye etter. Derfor vil det ikke være et krav om at oppløsning må være så høy ved utarbeidelse av en metode som legger inn informasjon i bildet. Av denne grunn benytter metoden som er foreslått for høy oppløsning. Ved å benytte lavere oppløsning vil man komme nærmere den reelle situasjonen, og det vil være lettere å utvikle en metode som kan benyttes på bilder som har gjennomgått utskrift og innskanning.

### **9.2. Videre arbeid**

Arbeidet som er gjort i denne oppgaven har ikke som hensikt å komme med en fullstendig fungerende metode, men heller se på om det finnes muligheter som kan føre frem til en slik metode. Det som er gjort kan ses på som et grunnlag for å gå videre med utvikling av en metode. Derfor gjenstår det arbeid på alle deler av foreslått metode, og få satt delene sammen slik at de danner en helhet.

Det har vært mye forskning rundt hvordan dither celler som benyttes til rastreing bør bygges opp, og området er ikke ferdig forsket på.

I denne oppgaven har histogrammet vært sentralt. Det er ikke sikkert det er riktig å fokusere på dette, det kan være andre kjennetegn ved et bilde som kan være langt viktigere fokusområde. Metoden for vannmerking er generell i forhold til hva slags informasjon som legges inn som vannmerke, derfor kan metoden fortsatt være et



utgangspunkt for videre arbeid selv om fokus på hva slags kjennetegn som er viktig eller annen informasjon ønskes å legge inn i vannmerket.

Selve oppbygningen av vannmerket må det også jobbes videre med. I denne oppgaven er det brukt randomisert informasjon i vannmerket for å unngå mønster ved rastring. Det som må gjøres er å benytte de reelle data som skal legges inn i vannmerket, og randomisere disse på en måte. Det minst to måter å gjøre dette på. Vannmerket kan bygges opp på en pseudo-random måte ved hjelp av en randomgenerator. Nøkkelen må da være kjent for alle som vil benytte metoden ved dekoding av vannmerket etter innskanning. En annen måte er å benytte flere bit for hver gråtone i histogrammet enn det som er et minimum. På denne måten kan flere kombinasjoner av bitmønstre bety det samme, og så kan man velge en av kombinasjonene som betyr det samme på en randomisert måte. Kombinasjonene som betyr det samme må være kjent ved dekoding av vannmerke. Dette kan for eksempel være bygd opp som en oppslagstabell. En annen mulighet er å ha et grunnmønster av bits som betyr en spesiell verdi, og så ha permutasjoner av denne generert med en bestemt nøkkel. Videre vil det være mulig å foreta en komprimering av informasjonen som skal tilordnes vannmerket. Komprimeringsalgoritme må da være kjent ved dekoding.

I denne oppgaven har det vært et tema å gjøre vannmerket så lite som mulig slik at en kan få lagt inn så mange eksemplarer som mulig av det. Det er ikke sikkert dette er så viktig hvis metoden blir mer sikker på å finne igjen korrekt vannmerke. Det kan også være ønskelig å få lagt inn så mye som mulig informasjon inn i vannmerket for å kunne gjenskape originalen på best mulig måte.

Metoden slik den foreligger nå er svært lite intelligent med tanke på å søke etter hvor vannmerket er plassert. Det vil antakelig ikke være vanskelig å finne en mer intelligent måte, og dermed vil metoden med større sikkerhet finne igjen korrekt vannmerke og på den måten trenger en ikke legge inn så mange eksemplarer av vannmerket. Det vil da være mulighet for å legge inn mer informasjon vannmerket om dette skulle være ønskelig. Videre vil det ikke være så store krav til at brukeren legger arket i skanneren rimelig rett.

Simuleringen av en reell utskrift og innskanning er ikke realistisk nok. Det vil være nødvendig å komme frem til en bedre simulering for å fortsette forsøk.

## Utstyr som er benyttet

Følgende utstyr og drivere er benyttet.

Skanner: Epson Expression 1640 XL  
Model: G650C  
Serie nr.: CK20002504  
Driver:  
Produksjonsår: 1998

Hewlett Packard Scanjet 5200C  
Model: HP C7190A  
Serie nr.:SG9CK171M4  
Driver: Version 2.02

HP Scanjet 5500c  
Model: C9920A  
Serie nr: CN28VS80XP  
Drivere: usbscan.sys, hpgtbb.dll, hpgwiamd.dll,  
hpsjvset.dll, hpqgends.tmp

Printer: Hewlett Packard LaserJet 5000N  
Model: C4111A  
Serie nr.: NLT1037726  
Driver: HP LaserJet 5000 Series PCL 5e

Canon BJC-6100  
Driver: Canon BJ Printer Driver Ver. 1.50GB

Epson Stylus Photo 900  
Modell: B211A  
Serie nr: E3PE018637  
Driver: Epson Stylus Photo 900  
Version: 5.40.0.0[05.02.2003]

Multifunksjon: Epson Stylus CX5222  
Model: C141A  
Serie nr.: EP6E057685  
Innkjøpt høst 2003  
Printerdriver: 09/09/2002, 5.41.00.00  
Skannerdriver:06/21/2002, 5.7.0.0

HP Officejet Pro 1170c  
Model type(nr): C5364A  
Serienr: SGB86AG8GM  
Printerdrivere: Dot4.sys, Dot4Prt.sys  
Skannerdrivere: Dot4scan.sys, hpo5300a.aio,  
hpo5400a.aio, hpo5500a.aio,  
hpojwia.dll

I de innledende forsøkene er alt utstyr benyttet for å finne typiske egenskaper ved utstyr. Ved simulering er skanneren Epson Expression 1640 XL og skriveren HP 5000N benyttet.

## Referanser

- [1] H.Z. Hel-Or, Watermarking and copyright labeling of printed images, *Journal of electronic Imaging*, Vol. 10, Issue 3, 2001, s. 794-803
- [2] Z. Fan, S. Wang, og H. Cheng. Retaining Color Fidelity in Multiple Generation Reproduction Using Digital Watermarks, *SPIE*, vol. 5008, 2003, s. 349-355
- [3] R. Crane, *A simplified approach to Image Processing*, kap. 2, Prentice Hall PTR, New Jersey, USA, 1997
- [4] P. Green, *Understanding Digital Color*, second edition, kap. 9, GATFPress, Sewickley, USA, 1999
- [5] K. Johansson, P. Lundberg og R. Rydberg, *Grafisk kokebok, guiden til grafisk produksjon*, kap. 9, Fälth & Hässler, Värnamo, Sverige, 1999
- [6] R. Ulichney, *Digital Halftoning*, The MIT Press, Cambridge, MA, USA, 1987
- [7] M. S. Fu, O. C. Au, Data Hiding for Halftone Images, *SPIE*, vol. 3971, 2000, s. 228-236
- [8] J. F. Delaigle, C. De Vleeschouwer, B. Macq, Watermarking algorithm based on a human visual model, *Signal Processing*, Vol. 66, Issue 3, 1998, s. 319-335
- [9] J. Liu, S Chen, Fast two-layer image watermarking without referring to the original image and watermark, *Journal of Computational Letters* Vol. 19, Issue 14, 2001, s. 1083-1097
- [10] D. Wu, W. Tsai, Embedding of any type of data in images based on a human visual model and multiple-based number conversion, *Pattern Recognition Letters*, Vol. 20, Issue 14, 1999, s. 1511-1517
- [11] X. Kong, Y. Liu, H. Liu, D. Yang, Object watermarks for digital images and video, *Image and Vision Computing*, Vol. 22, Issue 2, 2004, s. 583-595
- [12] S. Pereira, S. Voloshynskiy, T. Pun, Optimal transform domain watermark embedding via linear programming, *Signal Processing*, Vol. 81, Issue 6, 2001, s. 1251-1260
- [13] B. Hannigan, A. Reed, B. Bradley, Digital watermarking using improved visual system model, *SPIE* Vol. 4314, 2001, s. 468-474
- [14] N. Johnson, Z. Duric, S. Jajodia, *Information Hiding, Steganography and Watermarking – Attacks and Countermeasures*, kap. 2, Kluwer Academic Publishers, Boston, USA, 2003

## Figurliste

Figur 1: Høy oppløsning på binært bilde vil se ut som et gråtonebilde. ....	11
Figur 2: Arkitektur for foreslått metode [2]. ....	14
Figur 3: Skjematisk oversikt over vannmerkingsteknikk [1]. ....	16
Figur 4: Tersklingsmatrisen oppbygning [1]. Matrisen er satt sammen av alle dither cellene som er valgt ut fra vannmerket som skal legges inn i bildet [1]. ....	17
Figur 5: Opprettelse av tersklingsmatrise [1]. Vannmerket med 9 bit stilles opp som en todimensjonal matrise der svart piksel representerer "0" og hvit representerer "1". To dither celler benyttes for terskling der tersklingsverdiene er normalisert til område 0 – 3. For hvert bit i vannmerket velges korresponderende dither celle, C0 for "0" og C1 for "1". De valgte dither cellene stilles opp som en større matrise(d), dette er tersklingsmatrisen som skal benyttes på bildet som skal terskles. ....	18
Figur 6: Eksempel på vannmerking av et bilde [1]. Originalbildet som skal vannmerkes(a). Vannmerket på 64 bit(b) er oppstilt som en 8x8 matrise. For hvert bit velges korresponderende dither celle(c) og sammen danner de tersklingsmatrisen(d). Resultatbildet etter rastring(e). For sammenlikning skal (f) vise samme bilde rastret uten vannmerke. Hvilken metode som da er benyttet kommer ikke frem i artikkelen. ....	19
Figur 7: Forstørret utsnitt av reell innskanning. ....	24
Figur 8: Forstørret utsnitt av simulert skannet bilde. ....	24
Figur 9: Histogrammet til et bilde som er intensitetsjustert. ....	25
Figur 10: Histogrammet til et bilde før intensitetsjustering. ....	25
Figur 11: Kvaliteten reduseres for hver kopi(print/skan). Utført på Epson cx5200. ...	27
Figur 12: Histogram til originalbildet. ....	27
Figur 13: Histogram til bildet etter innskanning. Utført på Epson cx5200. ....	27
Figur 14: Lena, et typisk gråtonebilde. ....	28
Figur 15: Histogrammet til "Lena". ....	28
Figur 16: Reelt antall forekomster av gråtoner, bilde med 32x32 piksler. ....	29
Figur 17: Normalisering av antall forekomster. Antall forekomster er normalisert til område 0 - 15, bilde med 32x32 piksler. ....	29
Figur 18: Bildet må gå igjennom prosessering før det kan skrives ut. ....	31
Figur 19: Postprosessering. Etter innskanning må bildet gjennomgå en prosessering for å tilnærme seg originalbildet. ....	31
Figur 20: Arbeidsflyt fra originalbilde til resultatbilde via utskrift og innskanning. ....	32
Figur 21: Metoden tar for seg prosess frem til første innskanning. ....	33
Figur 22: Oppbygning av vannmerket. Histogrammet legges inn fra venstre mot høyre, fra topp til bunn. ....	33
Figur 23: Foreslått metode. ....	35
Figur 24: Valg av feil dither celle ved dekoding [1]. ....	36
Figur 25: Antall vannmerker som kan legges inn. ....	37
Figur 26: Innskanning på to og ti ganger bildets oppløsning. ....	40
Figur 27: Startposisjon for å lete etter vannmerket. Ved rotering av bildet endres startposisjon når vannmerket skal gjenfinnes. Metoden som benyttes trenger denne posisjonen. ....	40
Figur 28: Startpunkt må settes for å finne vannmerket. ....	40
Figur 29: Sirkulerer for å finne riktig plassering. ....	41
Figur 30: Plassering av startpunkt for hver blokk. En del av det vannmerkede bildet forstørret. Viser plassering der det neste blokk for vannmerkebit mest sannsynlig befinner seg. ....	42
Figur 31: Gjennomgang av metode med visning av endring i bildet... <b>Error! Bookmark not defined.</b>	
Figur 32: Histogrammene til original-, midlertidig- og resultatbildet. ....	49
Figur 33: Bilde og vannmerke som benyttes i forsøk. ....	50

Figur 34: Antall feil bit i gjenskapt vannermerke som funksjon av skalering og rotering. ....	51
Figur 35: Maksimalt antall feil ved flertallsavgjørelsen. ....	52
Figur 36: "Barbara" 32x32 piksler. ....	54
Figur 37: Histogram til "Barbara" 32x32 punkter. ....	54
Figur 38: Forstørrelse av "Barbara" 32x32 punkter. ....	54
Figur 39: Gjenskapt bilde. ....	55
Figur 40: Histogram til gjenskapt bilde. ....	55
Figur 41: Forstørrelse av gjenskapt bilde. ....	55
Figur 42: Gjenskapt bilde med histogramspesifikasjon. ....	56
Figur 43: Histogram til gjenskapt bilde med histogram matching. ....	56
Figur 44: Forstørrelse av gjenskapt bilde med histogram matching. ....	56
Figur 45: Forstørrede bilder, vanlig reproduksjon vs testmetode. Originalbildet(øvre venstre) kopiert på vanlig måte(øvre høyre). Simulert utskrift(nedre høyre) er prosessert med metode(nedre venstre). ....	57
Figur 46: Testbilder der vannermerket ble korrekt gjenskapt. ....	58
Figur 47: Testbilder der vannermerket ikke ble korrekt gjenskapt. ....	59

## Tabelliste

Tabell 1: Rotasjon og skalering hos forskjellig utstyr. ....	23
Tabell 2: Dither celle 0, $C_0$ . ....	36
Tabell 3: Dither celle 1, $C_1$ . ....	37
Tabell 4: Resultater fra forsøk med rotering og skalering ....	51
Tabell 5: Differanse flertallsavgjørelse. ....	52
Tabell 6: Fullstendige resultater fra test med skalering og rotering. ....	69

## Vedlegg – Resultater

Det ble foretatt forsøk av hvor mye skalering og rotering metoden håndterte før den feilet. Det ble lagt inn 16 vanmerker i bildet. Fullstendige resultater fra forsøket finnes i Tabell 6.

Forklaring til Tabell 6:

- SK: Skalering i %  
 RO: Rotering i grader  
 v: Antall feil gjenskapte bit i vanmerke  
 m: Dårligste resultat av antall av 16 mulige like bit som ble gjenskapt for én plassering i vanmerket..

I slutten på hver rad vises gjennomsnittet for de ti forsøkene som ble foretatt for hver kombinasjon av skalering og rotering. Det er disse data som er gjengitt i rapportens hoveddel.

**Tabell 6: Fullstendige resultater fra test med skalering og rotering**

SK	RO	Feil		Feil		Feil		Feil		Feil		Feil		Feil		Feil		Feil		Snitt			
		v	m	v	m	v	m	v	m	v	m	v	m	v	m	v	m	v	m	v	m		
0,00	0,00	0	15	0	16	0	16	0	16	0	14	0	15	0	16	0	16	0	16	0	15	0,0	15,5
0,00	0,05	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0,0	16,0
0,00	0,10	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0,0	16,0
0,00	0,15	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0,0	16,0
0,00	0,20	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0,0	16,0
0,00	0,25	0	14	0	15	0	14	0	15	0	15	0	15	0	13	0	13	0	13	0	14	0,0	14,1
0,00	0,30	0	11	0	11	0	11	0	11	0	12	0	11	0	11	0	11	0	11	0	12	0,0	11,2
0,00	0,35	0	10	0	11	2	8	1	8	0	10	0	9	0	10	0	9	1	8	0	10	0,4	9,3
0,00	0,40	0	9	4	8	6	8	8	8	0	8	1	8	13	8	0	9	0	9	6	8	3,8	8,3
0,00	0,45	5	8	14	8	3	8	2	8	13	8	2	8	5	8	1	8	3	8	14	8	6,2	8,0
0,00	0,50	16	8	25	8	9	8	15	8	15	8	19	8	14	8	20	8	23	8	17	8	17,3	8,0
0,05	0,00	0	15	0	15	0	16	0	15	0	15	0	16	0	16	0	16	0	15	0	16	0,0	15,5
0,05	0,05	0	16	0	16	0	16	0	16	0	15	0	16	0	16	0	16	0	15	0	16	0,0	15,8
0,05	0,10	0	16	0	15	0	16	0	16	0	16	0	16	0	16	0	16	0	15	0	16	0,0	15,8
0,05	0,15	0	16	0	16	0	16	0	16	0	15	0	16	0	16	0	15	0	16	0	16	0,0	15,8
0,05	0,20	0	15	0	16	0	15	0	15	0	15	0	15	0	15	0	15	0	15	0	15	0,0	15,1
0,05	0,25	0	13	0	13	0	13	0	14	0	14	0	14	0	13	0	14	0	14	0	15	0,0	13,7
0,05	0,30	0	12	0	10	0	11	0	9	0	11	0	10	0	10	0	12	0	11	0	10	0,0	10,6
0,05	0,35	0	11	0	12	0	12	0	12	0	13	0	11	0	12	0	11	0	11	0	9	0,0	11,4
0,05	0,40	0	9	1	8	3	8	5	8	0	11	0	9	0	9	0	9	3	8	3	8	1,5	8,7
0,05	0,45	5	8	3	8	17	8	13	8	0	9	4	8	11	8	1	8	2	8	21	8	7,7	8,1
0,05	0,50	4	8	5	8	1	8	11	8	10	8	18	8	11	8	8	8	12	8	18	8	9,8	8,0
0,10	0,00	0	16	0	15	0	16	0	16	0	15	0	16	0	16	0	16	0	15	0	15	0,0	15,6
0,10	0,05	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0,0	16,0
0,10	0,10	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0,0	16,0
0,10	0,15	0	16	0	16	0	16	0	16	0	16	0	16	0	15	0	16	0	16	0	16	0,0	15,9
0,10	0,20	0	14	0	14	0	15	0	14	0	15	0	14	0	14	0	15	0	14	0	13	0,0	14,2
0,10	0,25	0	13	0	11	0	13	0	11	0	13	0	10	0	11	0	9	0	9	0	13	0,0	11,3
0,10	0,30	0	11	0	11	0	13	0	11	0	10	0	12	0	11	0	11	0	10	0	10	0,0	11,0
0,10	0,35	0	9	4	8	0	9	0	9	1	8	1	8	0	11	0	9	0	10	0	9	0,6	9,0

BRUK AV VANNERMERKE VED REPRODUKSJON AV GRÅTONEBILDER

Egil Berntsen

0,10	0,40	3	8	0	8	1	8	1	8	5	8	5	8	7	8	1	8	1	8	1	8	2,5	8,0
0,10	0,45	5	8	25	8	11	8	34	8	20	8	25	8	2	8	12	8	26	8	29	8	18,9	8,0
0,10	0,50	16	8	9	8	9	8	9	8	16	8	9	8	11	8	3	8	7	8	21	8	11,0	8,0
0,15	0,00	0	15	0	15	0	16	0	16	0	15	0	16	0	16	0	16	0	16	0	16	0,0	15,7
0,15	0,05	0	16	0	16	0	16	0	16	0	16	0	16	0	15	0	16	0	16	0	16	0,0	15,9
0,15	0,10	0	16	0	16	0	16	0	16	0	16	0	15	0	16	0	16	0	16	0	16	0,0	15,9
0,15	0,15	0	15	0	15	0	15	0	16	0	15	0	15	0	14	0	16	0	15	0	15	0,0	15,1
0,15	0,20	0	15	0	15	0	14	0	16	0	15	0	15	0	14	0	16	0	15	0	14	0,0	14,9
0,15	0,25	0	10	0	13	0	11	0	12	0	13	0	11	0	12	0	12	0	12	0	11	0,0	11,7
0,15	0,30	0	12	0	11	0	11	0	12	0	13	0	12	0	11	0	11	0	12	0	11	0,0	11,6
0,15	0,35	3	8	0	10	0	10	0	10	3	8	2	8	0	9	0	10	0	8	0	9	0,8	9,0
0,15	0,40	0	9	1	8	0	9	0	8	1	8	0	9	0	8	0	9	2	8	0	9	0,4	8,5
0,15	0,45	6	8	7	8	17	8	14	8	9	8	5	8	6	8	13	8	4	8	9	8	9,0	8,0
0,15	0,50	35	8	45	8	41	8	20	8	51	8	38	8	16	8	51	8	31	8	21	8	34,9	8,0
0,20	0,00	0	15	0	16	0	16	0	16	0	16	0	16	0	16	0	15	0	16	0	15	0,0	15,7
0,20	0,05	0	15	0	16	0	16	0	15	0	15	0	16	0	15	0	16	0	16	0	15	0,0	15,5
0,20	0,10	0	15	0	15	0	15	0	16	0	16	0	14	0	15	0	16	0	16	0	15	0,0	15,3
0,20	0,15	0	15	0	15	0	15	0	15	0	15	0	16	0	16	0	16	0	14	0	14	0,0	15,1
0,20	0,20	0	13	0	13	0	12	0	13	0	13	0	14	0	14	0	14	0	13	0	13	0,0	13,2
0,20	0,25	0	12	0	12	0	13	0	13	0	11	0	11	0	13	0	13	0	11	0	12	0,0	12,1
0,20	0,30	0	9	0	10	0	9	0	9	0	9	0	10	0	9	1	8	0	10	0	8	0,1	9,1
0,20	0,35	0	11	0	10	0	10	0	9	0	9	0	9	0	10	0	10	0	10	0	11	0,0	9,9
0,20	0,40	2	8	4	8	2	8	20	8	3	8	7	8	14	8	7	8	10	8	23	8	9,2	8,0
0,20	0,45	42	8	27	8	18	8	29	8	19	8	22	8	35	8	18	8	35	8	13	8	25,8	8,0
0,20	0,50	41	8	26	8	31	8	16	8	27	8	16	8	34	8	15	8	28	8	25	8	25,9	8,0
0,25	0,00	0	16	0	16	0	16	0	15	0	15	0	16	0	15	0	16	0	15	0	15	0,0	15,5
0,25	0,05	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0	16	0,0	16,0
0,25	0,10	0	15	0	15	0	16	0	14	0	14	0	14	0	15	0	16	0	15	0	14	0,0	14,8
0,25	0,15	0	15	0	14	0	15	0	14	0	15	0	14	0	15	0	16	0	15	0	14	0,0	14,7
0,25	0,20	0	14	0	14	0	13	0	12	0	12	0	12	0	10	1	8	0	13	0	14	0,1	12,2
0,25	0,25	0	9	0	9	18	8	3	8	0	9	0	9	9	8	0	9	0	10	0	9	3,0	8,8
0,25	0,30	2	8	0	9	0	9	0	10	0	9	0	9	0	10	0	9	0	10	6	8	0,8	9,1
0,25	0,35	21	8	20	8	7	8	20	8	10	8	10	8	6	8	15	8	23	8	6	8	13,8	8,0
0,25	0,40	5	8	7	8	0	9	26	8	9	8	26	8	15	8	17	8	8	8	2	8	11,5	8,1
0,25	0,45	38	8	26	8	41	8	14	8	22	8	23	8	44	8	25	8	23	8	26	8	28,2	8,0
0,25	0,50	20	8	23	8	25	8	24	8	12	8	7	8	26	8	24	8	10	8	10	8	18,1	8,0
0,30	0,00	0	16	0	15	0	15	0	15	0	16	0	16	0	16	0	15	0	15	0	16	0,0	15,5
0,30	0,05	0	15	0	15	0	16	0	15	0	15	0	15	0	15	0	14	0	15	0	16	0,0	15,1
0,30	0,10	0	14	0	15	0	14	0	15	0	15	0	15	0	15	0	15	0	15	0	15	0,0	14,8
0,30	0,15	0	13	0	14	0	13	0	12	0	11	0	12	0	11	0	12	0	11	0	11	0,0	12,0
0,30	0,20	0	12	0	13	0	12	0	12	0	14	0	14	0	12	0	14	0	13	0	12	0,0	12,8
0,30	0,25	0	10	0	11	0	9	0	9	0	10	0	10	12	8	2	8	0	11	0	10	1,4	9,6
0,30	0,30	2	8	0	9	0	10	0	9	3	8	0	9	0	8	0	10	0	9	10	8	1,5	8,8
0,30	0,35	2	8	0	8	5	8	1	8	4	8	3	8	12	8	3	8	6	8	0	9	3,6	8,1
0,30	0,40	0	9	0	9	2	8	11	8	4	8	4	8	1	8	5	8	7	8	4	8	3,8	8,2
0,30	0,45	16	8	9	8	5	8	23	8	10	8	30	8	19	8	34	8	15	8	11	8	17,2	8,0
0,30	0,50	34	8	13	8	34	8	22	8	19	8	22	8	29	8	20	8	29	8	28	8	25,0	8,0
0,35	0,00	0	15	0	15	0	15	0	16	0	15	0	15	0	16	0	16	0	16	0	16	0,0	15,5
0,35	0,05	0	14	0	14	0	16	0	15	0	15	0	15	0	14	0	14	0	14	0	14	0,0	14,5

## BRUK AV VANMERKE VED REPRODUKSJON AV GRÅTONEBILDER

Egil Berntsen

0,35	0,10	0	13	0	15	0	15	0	14	0	13	0	16	0	15	0	14	0	15	0	15	0,0	14,5
0,35	0,15	0	10	0	12	0	13	0	11	0	11	0	13	0	12	0	11	0	10	0	12	0,0	11,5
0,35	0,20	0	11	0	12	0	11	0	12	0	11	0	10	0	12	0	11	0	11	0	11	0,0	11,2
0,35	0,25	0	10	4	8	13	8	3	8	8	8	5	8	0	10	4	8	1	8	6	8	4,4	8,4
0,35	0,30	9	8	22	8	13	8	3	8	23	8	6	8	7	8	17	8	13	8	9	8	12,2	8,0
0,35	0,35	8	8	4	8	9	8	19	8	8	8	21	8	13	8	23	8	14	8	14	8	13,3	8,0
0,35	0,40	8	8	23	8	24	8	11	8	28	8	15	8	18	8	23	8	18	8	20	8	18,8	8,0
0,35	0,45	24	8	11	8	30	8	9	8	29	8	29	8	11	8	19	8	27	8	45	8	23,4	8,0
0,35	0,50	18	8	30	8	13	8	22	8	31	8	27	8	23	8	32	8	21	8	28	8	24,5	8,0



## Vedlegg – Kode

Her følger kode for benyttelse av metoden. Først kommer hovedskriptet som beskriver rekkefølgen av funksjonskall som må gjøres, deretter følger hver enkelt fil.

### **vannmerking.m**

```
%Lister opp rekkefølgen på det som skal kjøres
%Leser inn originalbildet som skal vannmerkes
    l_org=imread('barbara_32x32_72.tif');
%Oppretter dither-cells
    c0=dither(0);
    c1=dither(1);
%lager histogram
    h=lag_histogram(l_org);
%oppretter vannmerke
    %Hvis skal benytter histogram, 1 bit pr blokk:
    vm=lag_vannmerke(h);
    %ved forsøk kan det være interessant å benytte et random vannmerke
    %vm=randint(32,32,[0,1]);
%Resizer bildet før vannmerket legges til
    l_r=imresize(l_org, 8, 'bilinear');
%legge inn vannmerket
    l_vm=vannmerk(l_r, vm, c0, c1);
%Skriv det vannmerkede bildet
    imwrite(l_vm, 'wm_barbara.tif','Resolution', (72*8));
%Nå er det vannmerkede bildet skrevet ut,
%det neste bil å simulere en innskanning
%ganger opp bildet slik at 1 px blir 10px, og legger til skalering
    l_scan = imresize(l_vm, 10.010);
%lager et average-filter
    f_avg=ones(4,4)/16;
%Filtrerer bildet
    l_filter=imfilter(l_scan, f_avg);
%intensity, mapping
    l_intens = imadjust(l_filter, [0.0 1.0], [0 0.85]);
%legger til støy
    l_noise = imnoise(l_intens, 'gaussian', 0,0.0002);
%Rotering
    l_rotert=imrotate(l_noise, -0.10);
%setter startpunkt og finner vannmerket og midlertidig gråtonebilde
    v0=5;
    h0=10;
    [gvm, l_res]=finn_vm(l_rotert, c0, c1, v0, h0);
%Gjensker histogram til originalbilde ut fra vannmerket
    h_g=gjenskap_histogram(gvm);
%Utfører histogramspesifikasjon på det midlertidige bildet
    l=histeq(l_res, h_g);
    %Hvis man har benyttet random generert vannmerke for å fa best
vannmerke må man allikevel benytte korrekt
    %histogram ved histogramspesifikasjonen, kan da bruke det originale
    %l=histeq(l_res, h);
%Så kan man skrive resultatbilde til fil
    imwrite(l, 'Resultatbilde.tif');
```

Egil Berntsen

---

### lag\_histogram.m

```
function histogram=lag_histogram(I);
%Parameter inn:
% Gråtonebilde
%
%Returnerer:
% Histogram med 256 bins normalisert til 0-15

%leser inn dimensjonene til bildet
[m,n]=size(I);
%Oppretter histogram til bildet ved å lese inn verdiene på punktene og lagre de
histogram_antall=zeros(1,256);
for i=1:m
    for j=1:n
        histogram_antall(1,I(i,j))=histogram_antall(1,I(i,j))+1;
    end
end
%har nå et histogram med reellt antall, normaliserer til 0-15
histogram=histogram_antall*(15/max(histogram_antall));
```

Egil Berntsen

---

### lag\_vannmerke.m

```
function vm=lag_vannmerke(histogram);
%Parameter inn:
% Histogrammet som skal konverteres til et vannmerke
%Returnerer:
% Vannmerket

%konverterer til binære data
vm=dec2bin(histogram,4);
%dette er nå en 256x4 matrise med strenger

%lager en lang streng av alle bit
str="";
for i=1:256
    for j=1:4
        str=[str, vm(i, j)];
    end
end
%og setter de sammen til en 32x32 matrise
teller=1;
vm2=uint8(zeros(32, 32));
for i=1:32
    for j=1:32
        s=str(1, teller);%henter neste bit i strengen
        s=str2num(s);%konverterer til nummer
        vm2(i,j)=s;%legger det inn i vannmerket
        teller=teller+1;%skal hente neste bit i strengen
    end
end
end
vm=vm2;
```

Egil Berntsen

---

vannmerk.m

```
function lvm=vannmerk(l, vm, c0, c1);
%Parameter inn:
% Bildet som skal vannmerkes
% Vannmerket
% Dither celle 0, c0
% Dither celle 1, c1
%Returnerer:
% Det vannmerkede bildet

%leser inn dimensjonene til bildet
[m,n]=size(l);

bstr=8;
xmax=floor(m/bstr)-1;
ymax=floor(n/bstr)-1;
%Thresholdarray, dimensjon
T=zeros(m,n);

%bygger opp treshholdarray med riktig mønster
%Leser da originalbildets verdi og vannmerkebit
for x=0:xmax
    for y=0:ymax
        if(vm(1+mod(x,32),1+mod(y,32))==0)
            for i=1:bstr
                for j=1:bstr
                    T(x*bstr+i,y*bstr+j)=c0(i,j);
                end
            end
        else
            for i=1:bstr
                for j=1:bstr
                    T(x*bstr+i,y*bstr+j)=c1(i,j);
                end
            end
        end
    end
end

%bygger så opp resultatbildet ut fra tresholdarray og originalbildet
%
lvm=zeros(m,n);
for i=1:m
    for j=1:n
        if(l(i,j)<T(i,j))%hvis bildepiksel er mindre enn tresholdverdi settes piksel til "0"
            lvm(i,j)=0;
        else%ellers er det "1"
            lvm(i,j)=1;
        end
    end
end
end
```

Egil Berntsen

---

finn\_vm.m

```
function [scan_gvm, lres]=finn_vm(l_scan, c0, c1, v0, h0);
%Parameter inn:
% Det simulerte skannede bildet, l_scan
% Dither celle 0, c0
% Dither celle 1, c1
% Vertikal startposisjon for å lete etter vannmerke, v0
% Horisontal startposisjon for å lete etter vannmerke, h0
%
%Returnerer:
% Vannmerket
% Det midlertidige gråtonebildet

%skal finne igjen vannmerket, dette gjøres slik:
% 1. Finn blokk
% 2. Regn ut snittverdi på gråtone
% 3. Simulér halftoning med de to dither-cellene
% 4. Velg dither-celle som er nærmest det halftonedede originalbilde
% 5. Denne bestemmer vannmerket som er lagt inn i denne blokken

max_verdi=0.85;
%Bruker 2 matriser til å telle opp "0" og "1"
vm0=zeros(32, 32);%for "1"
vm1=zeros(32, 32);%for "0"
%størrelsen på bildet
[m,n]=size(l_scan);
%Totalt for en blokk
Btot=double(0);
%Middelverdi for en blokk
Bmid=0;
%Blokkstørrelse
bstr=8;
%Blokk i kvadrat
Bstr=bstr*bstr;
%Dither celler må normaliseres
c0_norm=c0/255;
c1_norm=c1/255;
%Det skal scannes inn med 10 ganger så høy oppløsning,
%det simuleres ved å skalere opp med 10.
scale=10;

%array som forteller rekkefølge på testing, synkronisering
sync_ned= [0 0 1 1 1 0 -1 -1 -1 0 1 2 2 2 2 2 1 0 -1 -2 -2 -2 -2 -2 -1 0 1 2 3 3 3 3 3 3
3 2 1 0 -1 -2 -3 -3 -3 -3 -3 -3 -3 -2 -1 0 1 2 3 4 4 4 4 4 4 4 4 4 3 2 1 0 -1 -2 -3 -4 -4 -4 -
4 -4 -4 -4 -4 -4 -3 -2 -1];

sync_bort= [0 1 1 0 -1 -1 -1 0 1 2 2 2 1 0 -1 -2 -2 -2 -2 -2 -1 0 1 2 2 3 3 3 3 2 1 0 -1 -
2 -3 -3 -3 -3 -3 -3 -2 -1 0 1 2 3 3 3 4 4 4 4 4 3 2 1 0 -1 -2 -3 -4 -4 -4 -4 -4 -4 -4 -4 -
-3 -2 -1 0 1 2 3 4 4 4 4];

%opprettet en del tellere for å holde styr på hvor søkingen skal skje,
%Hvor starte på neste blokk,
%hvor starte på neste rad
ned_vm_0=0;
bort_vm_0=0;
```

Egil Berntsen

---

```

forste=1;
bortover=0;
nedover=0;
vm_hor=0;
vm_ver=0;
start_neste_bort_hor=0;
start_neste_bort_ver=0;
start_neste_ned_hor=0;
start_neste_ned_ver=0;
lrespeker_bilde_ver=0;
lrespeker_bilde_hor=0;
lrespeker_vm_ver=1;
lrespeker_vm_hor=1;
for ned_bilde=v0:32*8*scale:32*8*scale%hele bildet vertikalt
  for bort_bilde=h0:32*8*scale:32*8*scale%hele bildet horisontalt
    %her starter å finne hvert av vannmerkene som er lagt inn
    %pekere inn i vannmerkearray
    vmpeker_ver=1;
    vmpeker_hor=1;
    beste_sync=1;
    bortover=start_neste_bort_hor;
    nedover=start_neste_bort_ver;
    for ned_vm=0:8*scale:8*scale*31%ett vannmerke vertikalt
      bortover=bort_vm_0;
      nedover=ned_vm_0;
      for bort_vm=0:8*scale:8*scale*31%ett vannmerke horisontalt
        bortover=bortover+sync_bort(beste_sync);
        nedover=nedover+sync_ned(beste_sync);

        %Her starter den sirkulerende synkroniseringen
        beste_sync=1;
        beste_differanse=0;
        for sync=1:81
          %sjekker at man ikke vil komme utenfor bildekant ved bruk av sync-
          tall
          if((ned_bilde+nedover+sync_ned(sync)>0) &&
            (bort_bilde+bortover+sync_bort(sync)>0))
            Btot=0;
            for i=0:scale:7*scale
              for j=0:scale:7*scale
                if(ned_bilde+ned_vm+nedover+i+sync_ned(sync)<=m &&
                  bort_bilde+bort_vm+bortover+j+sync_bort(sync)<=n)

                  Btot=Btot+double(l_scan(ned_bilde+ned_vm+nedover+i+sync_ned(sync),bort_bil
                    de+bort_vm+bortover+j+sync_bort(sync)));
                end
              end
            end
            %Middelverdi for blokken
            Bmid=double(Btot)/double(Bstr);%*255
            %simulerer tresholding
            sim0=zeros(bstr,bstr);
            sim1=zeros(bstr,bstr);
            for i=1:bstr
              for j=1:bstr
                if(Bmid>c0_norm(i,j))

```

Egil Berntsen

---

```

        sim0(i,j)=max_verdi;
    end
    if(Bmid>c1_norm(i,j))
        sim1(i,j)=max_verdi;
    end
end
end
end
%tellere som skal fortelle om det var c0 eller c1 som er lagt inn i
denne blokken
cc0=0;
cc1=0;
for i=0:7
    for j=0:7

        if(ned_bilde+ned_vm+nedover+i*scale+sync_ned(sync)<=m &&
bort_bilde+bort_vm+bortover+j*scale+sync_bort(sync)<=n)
            if(abs(sim0(i+1,j+1)-
double(l_scan(ned_bilde+ned_vm+nedover+i*scale+sync_ned(sync),bort_bilde+bort
_vm+bortover+j*scale+sync_bort(sync))))<0.05)
                cc0=cc0+1;
            end
            if(abs(sim1(i+1,j+1)-
double(l_scan(ned_bilde+ned_vm+nedover+i*scale+sync_ned(sync),bort_bilde+bort
_vm+bortover+j*scale+sync_bort(sync))))<0.05)
                cc1=cc1+1;
            end
        end
    end
end
end
if(abs(cc0-cc1)>abs(beste_differanse))
    %da vet man til slutt også om cc0 eller cc1 er størst
    %hvis beste_differanse>0 er cc0 størst og vm=0
    %"beste_sync" forteller om det er nødvendig med
synkronisering
    %dette er en minste endring fra opprinnelig lokalisering etter
forrige justering

    beste_differanse=cc0-cc1;
    beste_sync=sync;
end
end
end%end sync
%her er synkronisering gjennomført for denne blokken
%Legger inn punkt i midlertidig reuslatatbilde
for i=0:7
    for j=0:7
        Ires(Irespeker_bilde_ver+Irespeker_vm_ver+i,
Irespeker_bilde_hor+Irespeker_vm_hor+j)=l_scan(ned_bilde+ned_vm+nedover+syn
c_ned(beste_sync)+i*scale,
bort_bilde+bort_vm+bortover+sync_bort(beste_sync)+j*scale);
    end
end
%oppdaterer pekere
if((ned_bilde+ned_vm+nedover+sync_ned(beste_sync)>0) &&
(ned_bilde+ned_vm+nedover+sync_ned(beste_sync)<=m) &&
(bort_bilde+bort_vm+bortover+sync_bort(beste_sync)>0) &&
(bort_bilde+bort_vm+bortover+sync_bort(beste_sync)<=n))

```

Egil Berntsen

---

```

        if(l_scan(ned_bilde+ned_vm+nedover+sync_ned(beste_sync),
bort_bilde+bort_vm+bortover+sync_bort(beste_sync))>0.5)
            lrespeker_bilde_ver+lrespeker_vm_ver;
        else
            lrespeker_bilde_ver+lrespeker_vm_ver;
        end
    end
    %oppdaterer synkronisering
    if(bort_vm==0)
        ned_vm_0=sync_ned(beste_sync);
        bort_vm_0=sync_bort(beste_sync);
    end
    if(ned_vm==0 && bort_vm==8*scale*7)
        %skal begynne på ny blokk til høyre
        start_neste_bort_hor=bortover+sync_bort(beste_sync);
        start_neste_bort_ver=nedover+sync_ned(beste_sync);
    elseif(ned_vm==8*scale*7 && bort_vm==0)
        %ny blokk på neste rad
        start_neste_ned_hor=bortover+sync_bort(beste_sync);
        start_neste_ned_ver=nedover+sync_ned(beste_sync);
    end
    if(beste_differanse>0)
        %vannmerket i denne blokken er "0"

vm0(vmpeker_ver,vmpeker_hor)=vm0(vmpeker_ver,vmpeker_hor)+1;
    else
        %vannmerket i denne blokken er "1"

vm1(vmpeker_ver,vmpeker_hor)=vm1(vmpeker_ver,vmpeker_hor)+1;
    end
    vmpeker_hor=vmpeker_hor+1;
    lrespeker_vm_hor=lrespeker_vm_hor+8;
end%bort_vm
vmpeker_ver=vmpeker_ver+1;
vmpeker_hor=1;
lrespeker_vm_ver=lrespeker_vm_ver+8;
lrespeker_vm_hor=1;
end%ned_vm
lrespeker_bilde_hor=lrespeker_bilde_hor+64;
lrespeker_vm_ver=1;
end%bort_bilde
lrespeker_bilde_ver=lrespeker_bilde_ver+64;
lrespeker_bilde_hor=0;
end%ned_bilde
%%%%%%
%Nå er lres 256x256 punkter, skal ha den til 32x32-->delt på 8
%Benytter middelveri
lres2_ver=1;
lres2_hor=1;
tot=0;
mid=0;
for i=1:8:256
    for j=1:8:256
        for k=0:7
            for l=0:7
                tot=tot+double(lres(i+k, j+l));

```



Egil Berntsen

---

```
        end
    end
    lres2(lres2_ver,lres2_hor)=tot/64;
    lres2_hor=lres2_hor+1;
    tot=0;
end
lres2_ver=lres2_ver+1;
lres2_hor=1;
end
lres=uint8(round(lres2*255));

%her må det så finnes ut om det var flest "0" eller "1" på denne plassen i vannermerket
for i=1:32
    for j=1:32
        if(vm1(i,j)>vm0(i,j))
            scan_gvm(i,j)=1;
        else
            scan_gvm(i,j)=0;
        end
    end
end
end
```

Egil Berntsen

---

**gjenskap\_histogram.m**

```
function ghist=gjenskap_histogram(gvm);
%Parameter inn:
% Vannermerket som er funnet
%
%Returnerer:
% Histogram til originalbildet med 256 bins

%Først settes alle binære data sammen til en lang rekke
t=1;
for i=1:32
    for j=1:32
        rekke(1, t)=gvm(i,j);
        t=t+1;
    end
end

%Så plukkes det ut 4 og 4 som hver seg er et 4-bits binært tall,
%konverterer det til decimaltall og legger det inn i histogramarrayen som skal
returneres
t=1;
for i=1:4:1024
    s="";
    for j=0:3
        s=[s, num2str(rekke(1,i+j))];
    end
    tall=bin2dec(s);
    ghist(t)=tall;
    t=t+1;
end
```