Steganografi i billeder baseret på en model af human perception

Skrevet af:
Christian Hammer og
Lars Rosenberg Randleff

Vejledere:
Klaus Hansen og
Jens Damgaard Andersen

Datalogisk Institut
Københavns Universitet
september 2000

Forord

Dette speciale er skrevet af Christian Hammer og Lars Rosenberg Randleff i tidsrummet august 1997 til september 2000. Specialet omhandler steganografi, der er en betegnelse for det at camouflere information; mere specifikt omhandler specialet camouflering af information i digitale billeder.

Vi har valgt at skrive om steganografi, som i nogen henseender kan sidestilles med kryptografi, da emnet datasikkerhed har vores fælles interesse. Vi ønskede begge et speciale baseret på anvendelse af specifikke metoder fremfor en teoretiske gennemgang af matematiske definitioner, og derfor blev et forsøg med "zero-knowledge proofs"[1] vraget til fordel for steganografien. Af hensyn til bl.a. dokumentationen og tilgængeligheden blev det steganografi i billeder fremfor i f.eks. film, der blev målet for specialet. Steganografi i billeder betragter vi i høj grad som et felt med naturlig anvendelse, da bl.a. Internets udbredelse gør, at det er ligetil at distribuere billeder verden over i løbet af ganske kort tid. Da en kopi af et digitalt billede ofte ikke vil have en kvalitetsmæssig forringelse, vil distribution af billeder nemt kunne fører til splid i et spørgsmål om bl.a. ophavsret. Her kan steganografien finde en anvendelse ved bl.a. camouflering af digitale vandmærker i billederne, så ophavsretten kan bestemmes.

Dette speciale beskæftiger sig med steganografi baseret på menneskesynet, således at mængden af data, der kan camoufleres i et billede, er afhængig af hvordan billedet bliver behandlet i det menneskelige synsapparat. Det er kun lykkedes os at finde forholdsvist lidt datalogisk materiale om denne indgangsvinkel til emnet, og en stor del af den litteratur vi har læst, har derfor beskæftiget sig med de psykofysiske fænomener, der gør, at ikke alle dele af et billede behandles ligeligt af synet. Som en følge heraf har vi fundet det nødvendigt at have et antal forsøgspersoner til at se på billeder, som er ændret under hensyntagen til de føromtalte synsfænomener, for at finde ud af, om ændringer i billederne var observerbare eller ej. Vi vil her gerne bringe en tak til de forsøgspersoner, der deltog i forsøget. Desuden skal der lyde en tak til nuværende og tidligere kollegaer fra Brüel & Kjær A/S og Torsana Diabetes Diagnostics A/S, der har bistået med idéer og forslag i skriveprocessen. En speciel tak skal lyde til øjenlæge Nicolai Larsen, Øjenambulatoriet på Amtssygehuset i Herlev, der har bistået med materiale og idéer til afsnittet om menneskesynet.

Billedet på forsiden er et såkaldt fiksérbillede. I billedet er gemt en slotsherreinde, og da hendes tilstedeværelse kan være svær at se, er der tale om camoufleret information. Slotsherreinden er gemt yderst til venstre i billedet, hvor hun står med hovedet nedad. Billedet er taget fra [Lanners 75].

Christian Hammer og Lars Rosenberg Randleff
Datalogisk Institut ved Københavns Universitet, september 2000.

---

[1] Zero-knowledge proofs er, ganske kort, metoder til, at én part kan sandsynliggøre over for andre parter, at vedkommende er i besiddelse af en given information, uden at informationen, eller dele heraf, videregives.

Indhold

Forord i

Figuroversigt v

Tabeloversigt viii

1. Indledning 1
   1.1 Opbygning af denne rapport 1
   1.2 Indhold af cd-rom 2

2. Introduktion til steganografi 3
   2.1 En historisk tilgang 3
   2.2 Steganografi og kryptografi 5
   2.3 Steganografi og informationsteknologi 6
   2.4 Vandmærker 8
      2.4.1 Det spatielle domæne 11
      2.4.2 Frekvensdomænet 12

3. Introduktion til digitale billeder 15
   3.1 Billedmodeller 15
      3.1.1 Pixelbaserede billedmodeller 15
      3.1.2 Vektorbaserede billedmodeller 16
   3.2 Farvemodeller 16
      3.2.1 Beskrivelse af farver 16
      3.2.2 CIE-modellen 18
      3.2.3 RGB-modellen 20
      3.2.4 CMY-modellen 21
      3.2.5 HSV-/HSB-modellen 21
      3.2.6 HSI-modellen 22
      3.2.7 YIQ-modellen 24
      3.2.8 Sammenligning af farvemodeller 25
   3.3 Filformater 26

4. Perceptuel tolerance og steganografisk kapacitet 30
   4.1 Begreber 30
   4.2 Anvendelse 30
   4.3 Afgrænsning 31
   4.4 Den perceptuelle tolerance 32
      4.4.1 Toleranceområder 32
      4.4.2 Tolerancebilleder 36
   4.5 Den steganografiske kapacitet 39
      4.5.1 Sammenligning af metoder 40
   4.6 Kombinationsmetoden 40
      4.6.1 Indkodning 42
      4.6.2 Afkodning 46
      4.6.3 Kombinationsmetoden til vandmærkning 48

5. Den menneskelige synsproces 52
   5.1 Model af human perception 52
   5.2 Synets fysiologi 54
      5.2.1 Øjets behandling af synsindtryk 54
      5.2.2 Hjernens behandling af synsindtryk 57

5.3 Lysfænomener ... 59
5.3.1 Simultankontrast ... 60
5.3.2 Assimilering ... 61
5.3.3 Kantkontrast ... 62
5.3.4 Spatiel frekvens ... 64
5.3.5 Synsbedrag ... 65
5.4 Farvefænomener ... 66
5.4.1 Spatiel frekvens ... 67
5.4.2 Kanter ... 67
5.4.3 Farveforskel ... 68
5.4.4 Kontrastfarver ... 69

6. Bestemmelse af tolerance og kapacitet ... 71
6.1 Tolerancetildeling ... 71
6.1.1 Anvendt metode til tolerancetildeling ... 72
6.1.2 Anvendte filformater ... 74
6.1.3 Anvendte farvemodeller ... 74
6.2 Just noticeable distortion ... 74
6.3 Kantdetektion ... 78
6.4 Homogenitet ... 79
6.5 Assimilering ... 81
6.6 Bestemmelse af kapacitet ... 84

7. Personforsøg ... 89
7.1 Formål ... 89
7.2 Metode ... 89
7.3 Anvendte parametre ... 91
7.4 Anvendte billeder ... 93
7.5 Opsamlede data ... 95
7.6 Databehandling ... 96
7.6.1 Fejlkilder ... 96
7.6.2 Resultaternes validitet ... 97
7.6.3 Bestemmelse af parameterværdier ... 101
7.7 Resultater ... 103
7.7.1 Just noticeable distortion ... 103
7.7.2 Kantpixels ... 104
7.7.3 Homogenitet i value ... 105
7.7.4 Homogenitet i hue ... 107
7.7.5 Assimilering – vægtet ... 109
7.7.6 Assimilering – middelværdi ... 110
7.8 Konklusion på personforsøg ... 111

8. Behandling af spread spectrum ... 114
8.1 Metode ... 114
8.2 Optimering ... 116
8.3 Resultater ... 117

9. Konklusion ... 119

10. Efterskrift ... 121

Litteratur ... 123
Steganografi og vandmærkning ... 123
Billedbehandling ... 125
Fysiologi og psykofysik ... 125

Andet 126
Bilag A: Arbejdsbeskrivelse 128
Bilag B: Terminologiliste 130
Bilag C: Anvendelser af steganografi 133
Bilag D: Dokumentation af personforsøg 135
Beskrivelse af forsøgsopstilling 135
Instruktion til forsøgspersoner 136
Samtykkeerklæring 137
Bilag E: Programbeskrivelse – StegoShop 138
Bilag F: Programbeskrivelse – StegoViewer 140
'Spøgelsesvalg' 141
Bilag G: Programbeskrivelse – Spread Spectrum 143
Bilag E: Opbygning af cd-rom 144
Bilag F: Steganografi på WWW 146
Publikationer 146
Henvisninger 146
Billeder 146
Kommercielle 146
Diverse 146

Figuroversigt

Figur 1: TV-billede med synligt vandmærke. Taget fra TV2s hjemmeside (www.tv2.dk).....9
Figur 2: Vandmærke i spatielt domæne. Forskellen på de to øverste billeder er vist i det nederste billede.....12
Figur 3: Camouflering i frekvensdomænet foregår ved ændring af koefficienter i frekvensrepræsentationen.....13
Figur 4: Eksempler på pixelbaserede billeder. Her et binært billede, og et billede med mange farver.....15
Figur 5: Eksempler på vektorbaserede figurer.....16
Figur 6: Ostwalds farvemodel, der i [Foley et al, 96] også omtales som "kunstnermodellen".....17
Figur 7: Luminous-effekten. Graferne er taget fra [Foley et al, 96].....18
Figur 8: CIE-diagrammet. (a) Keglen med synlige farver. (b) Projektionen af trekanten fra (a) på XY-planet. Fra [Foley et al, 96].....19
Figur 9: RGB-kuben, der indeholder de farver, som beskrives ved RGB-modellen.....21
Figur 10: HSV-modellen. (a) En farve beskrives ved sine H, S og V-værdier. (b) HSV-keglen som en sub-kube af RGB-kuben.....22
Figur 11: HSI-dobbeltkegle. En farve er beskrevet ved vinklen fra Rød (H), afstanden fra centrum (S) og lodret placering (I).....23
Figur 12: Sammenligning af farvemodeller. Samme billede er gengivet i forskellige farvemodeller.....26
Figur 13: Den perceptuelle tolerance i to dimensioner.....32
Figur 14: Intervalbredde. (a) En kapacitet på 0 bit. (b) En kapacitet på 1 bit.....33
Figur 15: Irregulære og ikke-sammenhængende toleranceområder.....34
Figur 16: Farvede og gråtonede tolerancebilleder.....39
Figur 17: Udnyttelse af kapaciteten forskellige steder i et billede, ved camouflering i én pixel ad gangen.....42
Figur 18: Camouflering med kombinationsmetoden.....43
Figur 19: Model af visuel human perception. Perceptionen er opdelt i reception, bearbejdelse og bevidsthed.....52
Figur 20: Skematisk oversigt over synets vej til visuel cortex. Fra [Laursen 76].....53
Figur 21: Øjets anatomi (venstre øje).....55
Figur 22: Fordeling af tappe i nethinden (højre øje). Fra [Zrenner et al 90].....55
Figur 23: Nethindens opbygning. Lyset udefra kommer fra venstre i figuren, og går således igennem en del lag i nethinden, før det når receptorlaget.....56

Figur 24: Sammensætning af modtageområde for en gangliecelle. Fra [Zrenner et al 90] 56
Figur 25: Øjenbevægelser. De stiplede linier viser drift, og de fuldt optrukne viser mikrosaccader. Fra [Laursen 76] 57
Figur 26: Tre neurale systemer til opfattelse af synet. Fra [Zrenner et al 90] 58
Figur 27: Øjets fikseringer i løbet af et minuts betragtning af et billede. Fra [Moses&Hart 87]. ... 59
Figur 28: Simultankontrast. De fire inderste kasser har alle samme gråtone. 61
Figur 29: Assimilering. De tynde linier kan virke mørkere i venstre side end i højre side. 62
Figur 30: Mach-bånd. Den ubrudte line vise intensitetsvariation, og den stiplede linie viser den opfattede variation. 63
Figur 31: Chevruels effekt. Den ubrudte line vise intensitetsvariation, og den stiplede linie viser den opfattede variation. 64
Figur 32: En forsøgspersons kontrastfølsomhed som funktion af den spatielle frekvens. Tilrettet fra [Laursen 76]. 65
Figur 33: Kanizsas trekant. Det kan se ud som, der ligger en hvid trekant ovenpå tre udfyldte cirkler. 65
Figur 34: Hermann-gitre. I (b) er hjørneområderne 10% grå; det er de ikke i (a). 66
Figur 35: Billeder med ændrede pixels i henholdsvis (a) varieret område og (b) ikke-varieret område. 69
Figur 36: Kontrastfarve. Det brune område har samme intensitet i hele område, men opfattes som mørkest til højre. 70
Figur 37: Anvendt metode til camouflering og detektering af data. 71
Figur 38: N_4 og N_8-nabomatricerne. 72
Figur 39: JPEG-billeder, der er gemt med forskellige komprimeringsfaktorer. 77
Figur 40: Matricerepræsentation af kantdetektorer. (a) Prewitt og (b) Sobel. 79
Figur 41: Assimileringsmatricer. (a) Middelværdi og (b) vægtet. 82
Figur 42: Assimileringsbilleder. Billederne til højre benytter middelværdien, mens billederne til venstre benytter den vægtede værdi. 83
Figur 43: Skærmbillede fra programmet StegoViewer, der har været anvendt under personforsøgene. 90
Figur 44: De grundbilleder, der blev anvendt ved forsøgene. 93
Figur 45: Søjlediagrammer til sammenligning af forsøgspersoner. 99
Figur 46: Sammenhæng mellem antal rigtigt trufne valg og brugt tid. 100
Figur 47: Indsætning af vandmærke med spread spectrum. 116

Figur 48: Resultat af optimering af spread spectrum. (a) viser camouflering med fast α (0,1); i (b) er anvendt variabel α. 117

Tabeloversigt

Tabel 1: Kort beskrivelse af nogle grafiske filformater....29
Tabel 2: Start- og slutværdier for toleranceintervaller af to pixels....45
Tabel 3: Beregnede værdier for to pixels efter camouflering af 18 bit....46
Tabel 4: *I* og Δi for to pixels. Disse værdier skal benyttes i forbindelse med afkodning af camoufleret information....47
Tabel 5: En pixels initialværdier før tolerancetildeling....73
Tabel 6: Værdier efter tolerancetildeling....73
Tabel 7: Oversigt over kapaciteter....88
Tabel 8: Parameterværdierne, der blev undersøgt ved forsøgene....95
Tabel 9: JND – oprindelige data....104
Tabel 10: JND – korrigerede data....104
Tabel 11: Kantpixels – oprindelige data....105
Tabel 12: Kantpixels – korrigerede data....105
Tabel 13: Homogenitet i value, min-max-billeder – oprindelige data....106
Tabel 14: Homogenitet i value, arbitrærbilleder – oprindelige data....106
Tabel 15: Homogenitet i value, min-max-billeder – korrigerede data....107
Tabel 16: Homogenitet i value, arbitrærbilleder – korrigerede data....107
Tabel 17: Homogenitet i hue, min-max-billeder – oprindelige data....108
Tabel 18: Homogenitet i hue, arbitrærbilleder – oprindelige data....108
Tabel 19: Homogenitet i hue, min-max-billeder – korrigerede data....109
Tabel 20: Homogenitet i hue, arbitrærbilleder – korrigerede data....109
Tabel 21: Assimilering, vægtet – oprindelige data....110
Tabel 22: Assimilering, vægtet – korrigerede data....110
Tabel 23: Assimilering, middelværdi – oprindelige data....111
Tabel 24: Assimilering, middelværdi – korrigerede data....111

1. Indledning

Dette speciale beskæftiger sig med steganografi i digitale billeder, definition af begreberne perceptuel tolerance, tolerancebilleder og steganografisk kapacitet, samt bestemmelse af disse for en række digitale billeder på baggrund af en simpel model af menneskesynet.

Vi har forsøgt at holde fagtermer på dansk, hvor vi ikke har fundet disse unaturlige. For en del anvendte fagtermer har vi dog ikke været i stand til at finde danske ækvivalenter, hvorfor de engelske termer er forklaret og anvendt. I 'Bilag B: Terminologiliste' forklares en del af de anvendte termer, både danske og udenlandske, ganske kort.

1.1 Opbygning af denne rapport

Da begrebet steganografi for os har været forholdsvis nyt og ukendt, gives i afsnit 2 en kort introduktion til emnet. Her beskrives, hvad steganografi er, hvordan den adskiller sig fra kryptografien, og i hvilke sammenhænge den optræder. I 'Bilag C: Anvendelser af steganografi' er desuden bragt en række dagligdags eksempler på brug af steganografi.

Afsnit 3 giver en introduktion til digitale billeder. Afsnittet fortæller om, hvordan indholdet af digitale billeder kan beskrives, hvilke farvemodeller der kan anvendes, samt nogle af de filformater, som digitale billeder benytter.

I afsnit 4 defineres begreberne perceptuel tolerance og steganografiske kapacitet. Her beskrives bl.a. hvordan begreberne bestemmes og anvendes. Desuden angives metoder til, ved brug af disse begreber, at ind- og afkode information i billeder.

Da definitionen af den perceptuelle tolerance bygger på sammensætningen af det menneskelige syn, bringer afsnit 5 en meget forsimplet beskrivelse af dette. Heri beskrives fænomener ved synet, der kan benyttes ved camouflering af information. Bl.a. beskrives de fænomener, der i afsnit 6 danner baggrund for bestemmelse af den perceptuelle tolerance.

I afsnit 6 beskrives, hvordan bestemmelse af den perceptuelle tolerance for en række prøvebilleder har fundet sted. Her beskrives nogle af de filtre. der relaterer til en del af de i afsnit 5 nævnte synsfænomener, samt hvordan disse områder kan udvide et billedes perceptuelle tolerance.

For at bestemme nogle af de grænser menneskesynet i realiteten sætter for ændringerne i et billede, har vi valgt at lade en række forsøgspersoner se på billeder for at afgøre, om et givent billede er ændret observerbart eller ej. De billeder, der undersøges, er alle ændret efter metoderne beskrevet i afsnit 6. En beskrivelse af disse personforsøg, samt resultaterne heraf, kan findes i afsnit 7.

Da et formål med at bestemme et givent billedes perceptuelle tolerance kan være at benytte dette mål til optimering af bl.a. vandmærkningsalgoritmer, er også dette forsøgt. I afsnit 8 beskrives et forsøg på optimering af indsættelse af vandmærker med spread spectrum-metoden, som den bl.a. er beskrevet i [Cox et al 95].

1.2 Indhold af cd-rom

En del af de figurer og billeder, der er anvendt i denne rapport, kan være svære at gengive på papir, og er derfor vedlagt specialet på en cd-rom. Cd-rom'en indeholder, udover de viste billeder, også en række billeder, der har været brugt i forbindelse med personforsøgene. Disse billeder omfatter både originale billeder, samt billeder, der er ændret i større eller mindre grad fra originalerne

I arbejdet med dette speciale har vi skrevet en række programmer, der ligeledes er medtaget på cd-rom'en. Disse programmer er dels fremstillet i Java (Java 2 SDK, Standard Edition, v 1.2.2), hvor kildekoden er medbragt, dels i Object Pascal (Delphi 2.0) og (Visual) C++, hvor både kildetekst og eksekverbar kode (Windows 95/98/NT) er medbragt på cd-rom'en. Javas SDK (til Windows) er ligeledes at finde på cd-rom'en.

Resultaterne fra de afholdte personforsøg er også på cd-rom'en. Dette gælder også de udfyldte samtykkeerklæringer, der er blevet indscannet.

Endelig indeholder cd-rom'en denne rapport i flere formater. Desuden ligger 'Bilag F: Steganografi på WWW' som HTML-dokument for sig, så alle her nævnte URL'er kan tilgås direkte fra en web-browser.

2. Introduktion til steganografi

Dette afsnit giver en introduktion til begrebet steganografi. At steganografi ikke er et nyt begreb illustreres med en kort historisk tilgang i afsnit 2.1. Da steganografi og kryptografi begge er begreber, der dækker over det at gøre det svært at læse data for andre end dem de er tiltænkt, vil afsnit 2.2 vise forskelle på disse begreber. Endelig behandles brugen af steganografi i elektroniske medier (afsnit 2.3), og i vandmærkning af elektroniske medier (afsnit 2.4).

2.1 En historisk tilgang

Selvom begrebet steganografi er relativt nyt inden for datalogien, så er der på ingen måde tale om et nyt begreb. Ordet kommer fra de græske ord *stegano*, der betyder "dækket", og *grafos*, der betyder "at skrive", og i engelsksproget litteratur omtales steganografi ofte som "hidden information" eller "data hiding". En dansk forklaring af begrebet steganografi kunne derfor være "skjulte data". Da vi dog mener, at der er væsentlig forskel på, om en mængde data er ikke-observerbar (dvs. usynlig i billeder), eller blot skjult i et medium, hvor særlig opmærksomhed vil afsløre de skjulte data, har vi valgt at benytte begrebet "camoufleret data" til at dække over begge muligheder. Dette gælder uanset, om data er camoufleret i billeder, film, lyd eller andre medier.

I [Kahn 96] gives en række historiske eksempler på anvendelsen af steganografi. Som eksempler på den usynlige steganografi fortælles bl.a. om, hvordan man i det antikke Kina benyttede kurerer til at overbringe meddelelser mellem rigmænd og hærførerer. Hvis en kurer ikke kunne memorere en meddelelse, kunne den nedskrives på tynd silke eller papir, og rulles ind i en lille vokskugle, en *la wan*. Denne vokskugle blev så skjult på, eller i, kureren, der efter en naturlig ventetid, kunne fremvise meddelelsen hos modtageren. Såfremt det at bære en la wan ikke var forbundet med et sådant ubehag, at det direkte afslørede kureren, måtte meddelelsen siges at være usynlig.

Fra det antikke Grækenland kendes, gennem Herodots fortællinger (og videregivelse i [Kahn 96]), til anden brug af usynlig information. Her fortælles om Harpagus, der ønskede at hævne sig på sin slægtning, kongen af Medes, der havde narret Harpagus til at spise sin egen søn. Harpagus ville sende en besked til Cyrus, Persiens konge, der også havde mistet et barn på grund af kongen af Medes. Beskeden gik på, at Harpagus var villig til at hjælpe Cyrus med et angreb. Under kurerens rejse var meddelelsen gemt i en død hare, og kureren var udklædt som jæger. Da ingen vagter på vejen syntes at fatte mistanke til en jæger med sit nedlagte bytte, kunne kureren nå frem til Cyrus, der hurtigt indledte et angreb. Dette blev kongen af Medes' endeligt.

Herodot fortæller ligeledes om et angreb på perserne: Histiaeus ville sende en hemmelig meddelelse fra det persiske hof til sin svigersøn, tyrannen Aristagoras fra Miltus. Dette gjorde han ved at barbere en slave skaldet og tatovere meddelelsen på slavens isse. Da slavens hår var vokset ud igen blev, denne sendt til Aristagoras, der igen barberede ham, og derved fik den hidtil usynlige meddelelse.

Til forskel fra den usynlige steganografi, der er anvendt i ovenstående eksempler, kan skjult steganografi defineres som camouflerede data, der kan observeres, hvis blot observatøren er tilstrækkelig opmærksom på tilstedeværelsen. Om skjult steganografi fortæller [Kahn 96] bl.a. om Aeneas metode, der i sin simpelhed går ud på, at der i en avis eller lignende markeres bogstaver, ord eller sætninger, der i sin samlede helhed udgør en meddelelse. Markering kan f.eks. foregå ved streger eller huller i papiret. Hvis markering derimod foregår med "usynlig blæk", der skal behandles med varme eller kemikalier før markeringen kan ses, hører metoden til under den usynlige steganografi.

[Kahn 96] fortæller også om Thritemius, der i 1499 hævdede at have haft drømme om mange forskellige kodesystemer, og metoder til anvendelse af disse. Blandt disse nævnes en steganografisk metode, hvor de bogstaver, der udgør den camouflerede meddelelse, er gemt som hvert andet bogstav i hvert andet ord af en tilsyneladende nonsens-meddelelse. Denne form for steganografi kaldes *nul-chiffer*[2], da data er gemt i en nonsens-kontekst (med nul-værdi), hvilket igen tydeliggør, at der er tale om skjult, og ikke usynlig, steganografi.

Den ikke-datalogiske steganografi har dog også været anvendt i nyere tid. Under Anden Verdenskrig var brugen af steganografi hyppigt anvendt til hemmelig kommunikation, både blandt spioner, der bl.a. brugte skjulte meddelelser i almindelige breve for at kunne fortælle om krigsskibe og troppevandringer, og blandt menige soldater, der prøvede at fortælle familien hjemme, hvor de var udstationeret, selvom denne form for oplysninger officielt blev betragtet som statshemmeligheder. Metoderne var mangfoldige, og det var nødvendigt for alle parter i krigen at nedsætte censorkorps til gennemgang af udenlandsk korrespondance, for at finde de skjulte meddelelser. Blandt den skjulte (og altså ikke usynlige) steganografi kan nævnes brugen af *mikrodots*, der er fotografier på størrelse med et punktum. I et almindeligt brev kunne en del af punktummerne erstattes af mikrodots, der bl.a. kunne være mikro-billeder af kortudsnit eller tekniske tegninger. Selvom det har været svært at skelne mikrodots fra almindelige punktummer, så havde de den

[2] Efter engelsk: *null-cipher* [Kahn, s. 520]

egenskab, at den film der blev brugt, reflekterede lyset kraftigere end almindeligt blæk. Det var derfor muligt for censorkorpsene, om end besværligt, at finde breve med de meddelsomme punktummer.

Camouflering af data i almindelig tekst kan ske på mange forskellige måder: variationer i linieafstanden, bogstaver med forskellige (men næsten ens) skrifttyper, bogstaver i forskellige størrelser er bare nogle af de mange måder hvorpå almindelig tekst kan camouflere hemmelige data. Har man mistanke til, at en tekst indeholder data skjult på denne måde, er disse dog forholdsvis nemme at fjerne, idet teksten simpelthen kan skrives påny. I følge [Anderson&Petitcolas 98] havde Margarat Thatcher, Storbritanniens Premierminister i 1980'erne, så store problemer med oplysninger, der blev lækket til pressen fra Ministerkabinettet, at hun havde sit tekstbehandlingsanlæg programmeret således, at mellemrum mellem ord i et dokument angav, hvilken minister dokumentet var sendt til. Ved sammenligning mellem de kopier pressen var i besiddelse af, og de oprindelige dokumenter, kunne man så afsløre den minister, der skulle være skyld i "lækagen".

I 'Bilag C: Anvendelser af steganografi' er givet endnu en række eksempler på steganografiens anvendelser. Disse eksempler er alle taget fra dagligdags situationer, og de beskriver fænomener, som for de fleste vil synes kendte, dog uden at de tidligere har været forbundet med et begreb som steganografi.

2.2 Steganografi og kryptografi

Selvom både steganografi og kryptografi er begreber, der omhandler det at forhindre at fortrolige oplysninger kommer i de forkerte hænder, så er der konceptuel forskel på de to begreber. Ved begge begreber søges en kommunikation mellem to parter sikret således, at en ondsindet tredjepart ikke kan opsnappe, eller ændre, den kommunikation der foregår. Ved kryptografi søges kommunikation sikret ved en nøgle, som den modtagne part skal være i besiddelse af, for at kunne læse den hemmelige meddelelse. Algoritmen til krypteringen kan derfor være offentligt kendt, uden at dette kompromitterer sikkerheden ved kommunikationen. Selve sikkerheden hænger derfor sammen med nøglen, og den part der er i besiddelse af denne nøgle, hvadenten det er den tænkte modtager, eller en ondsindet tredjepart, kan dekryptere meddelelsen. Antages algoritmen til camoufleringen derfor kendt, på samme måde som med kryptering, kan sikkerheden med steganografi være reduceret til et spørgsmål om, hvorvidt et givent billede indeholder camoufleret information eller ej. Hvor en tredjepart ved kryptografi således udmærket kan være klar over, at en opsnappet meddelelse er krypteret, hvorefter en dekryptering kan gå i gang, så kan tredjepar-

ten ved steganografi ikke nødvendigvis vide sig sikker på, hvorvidt en meddelelse indeholder camoufleret information eller ej.

Kombineres kryptografien med steganografien kan der imidlertid opnås en stærkere sikring af de data, der ønskes udvekslet mellem to parter. Sendes f.eks. et stort antal billeder mellem to kommunikerende parter, hvoraf ét billede camouflerer en krypteret meddelelse, besværliggøres angrebet for tredjemand. Et billede, der ikke indeholder camoufleret information, vil, hvis det bliver udsat for en afkodningsalgoritme, formentlig give (nonsens-)information, der til forveksling kunne ligne en krypteret meddelelse. Hvis angriberen derfor har mistanke til, at der sendes en meddelelse i ét af et antal billeder, ville dette kræve et angreb på hvert billede, med efterfølgende dekryptering. Resultatet ville være, at det bliver mere besværligt, omend måske ikke umuligt, for angriberen at få fat i meddelelsen.

2.3 Steganografi og informationsteknologi

Med brugen af informationsteknologi er det oplagt, at data kan camoufleres digitalt i andre data. Dette kan være i elektroniske medier som lyd, film og billeder, men også i andre former for filer, som f.eks. tekstbehandlingsdokumenter og eksekverbare filer. I tekstbehandling er f.eks. begrebet "vis koder" i Word Perfect et forholdsvist kendt begreb blandt erfarne WP-brugere, og altså et eksempel på skjult (men som regel ikke usynlig) information.

Plombering (eng. *tamper proofing*) af et medium er en sikkerhedsmæssig anvendelse af steganografi i elektroniske medier. Ved plombering gemmes en bitfølge (plomben) i mediet, og hvis den bitfølge, der kan udledes af billedet på et senere tidspunkt, ikke stemmer overens med den camouflerede bitfølge, er dette et bevis på, at mediet er blevet ændret efter plomberingen. Dette kan f.eks. benyttes i digitale billeder, hvor en plombe (et vandmærke) gentages over hele billedet. Hvis en behandling af billedet viser, at en del af billedet ikke indeholder plomben, så er det sandsynligt, at denne del af billedet er blevet ændret.

Eksempel: Den naive camouflering

Et digital billede består af et antal pixels. Hver pixel har en værdi, der angiver, hvilken intensitet/farve/nuance pixelen har. Denne værdi er angivet i et antal bit, der bestemmer, hvor mange forskellige intensiteter/farver/nuancer en pixel kan have. Et typisk antal bit kan være 32, hvor tre byte angiver, hvor stor en mængde af rød, grøn og blå, der indgår i pixelen. Den fjerde byte angiver en α-værdi, der er et mål for den enkelte pixels gennemsigtighed. En naiv metode til camouflering af information i et billede går ud på, at den mindst betydende bit (LSB, *least significant bit*)

for hver pixel erstattes med en bit fra de data, der skal camoufleres. Ved camouflering med denne metode, er det maksimale antal af camouflerede bit givet ved antallet af pixel i billedet. En lille udbygning af denne metode går på at udskifte LSB for hver farve-/α-byte i en pixel, hvilket vil give mulighed for camouflering af fire bit pr. pixel.

Den naive metode er ikke robust over for ændringer af billedet, og vil ikke være brugbar ved vandmærkning (se afsnit 2.4). Ved at tilføje ustruktureret støj over hele billedet vil LSB ændres tilfældigt, og den camouflerede information vil dermed forsvinde.

❑

Digitale billeder kan indeholde tilføjede data, der camoufleres i billedet vha. billedets øvrige indhold. Et eksempel på dette kan ses på forsiden af denne rapport, hvor en person er placeret i billedet, uden at personen træder frem. Denne form for tilføjelse kan også ses i manipulation af pressefotos, hvor personer eller genstande placeres i omgivelser, for at give et billede, der ikke ellers kunne tages.

MPEG-4 standarden indeholder en hel del metoder, der kan reducere størrelsen af digitale filmsekvenser. En af disse metoder beskriver ansigtsbevægelser ved et antal parametre, således at en filmsekvens, der viser et animeret ansigt, ikke skal indeholde de animerede billeder fra sekvensen, men kun de ændringer der sker i ansigtsparametrene. Dette giver, at data for længere filmsekvenser bliver reduceret betydeligt, hvilket bl.a. giver et mindre behov for datarate på en datanetforbindelse. I [Hartung et al 98] beskrives en metode, hvorpå små ændringer i disse parametre kan benyttes til camouflering af data i en MPEG-4 ansigtsanimation.

Eksempel: Camouflering i medicinske billeder

I den medicinske billedbehandling bliver flere og flere billeder optaget direkte digitalt. Dette sker ofte selvom de analoge billeder har en langt bedre spatiel opløsning, og større dynamikområde, da overgangen til digitale billeder gør arbejdsgangen omkring billederne lettere. Dette er med til at sikre, at der kan undersøges flere patienter på en given tid. Brugen af digitale billeder er dog ikke uden problemer. For f.eks. at regulere diabetikeres medicinering og diæt kan der med fastsatte intervaller tages billeder af diabetikernes øjenbaggrund ("fundusbilleder"). Hvis en diabetiker ikke overholder sit fotointerval, kan det betyde, at det ikke er nemt at regulere diabetikerens medicin og diæt, og det kan i sidste ende resultere i, at diabetikeren mister synet. Man kan forestille sig følgende situation: En diabetiker overholder ikke sit fotointerval. Diabetikerens sidste fotoundersøgelse viste ikke tegn på stærkt synstruende diabetisk retinopati, og da diabetikeren

endelig mister synet, vælger hun at anlægge sag mod den klinik, der undersøgte hende sidst. For at "forbedre" sin sag, lader hun de sidste billeder, som hun har fået udleveret efter undersøgelsen, behandle på en sådan måde, at der på billederne fremkommer flere synstruende læsioner, end der rent faktisk var til stede ved undersøgelsen. Ved retssagen kan klinikken fremviser billeder magen til dem diabetikeren har, dog med færre læsioner. Udfaldet af retssagen er nu blevet til et spørgsmål om, hvem der kan bevise at være i besiddelse af de oprindelige billeder. Hvis man fra klinikkens side har lagt en plombering i alle billeder, vil en digital processering af diabetikerens version af billederne afsløre, at der er blevet ændret ved billederne, og diabetikeren vil formentlig stå til at tabe sagen.

❑

Lydfiler kan også, som digitale billeder, benyttes til steganografi. Rockmusik har f.eks. et så komplekst lydbillede, at det ofte er muligt at spille svage lyde, enten i et frekvensområde for sig, eller overlappende selve musikken, således, at disse lyde ikke er umiddelbart hørlige. En del metoder til camouflering af data i musik beskæftiger sig med signalanalyse af musikken, på samme måde som en del af de metoder, der står for camoufleringen i billeder. I [Kahn 96] fortælles dog også om andre metoder, der ikke kræver særlige matematiske kundskaber. Bl.a. fortælles om, hvordan J.S. Bach gemte sit efternavn i enkelte musikstykker, med tonerne H for B, og H-is for H.

Eksempel: Camouflering i rockmusik

Et eksempel på skjulte budskaber i rockmusik kendes fra en amerikansk retssag, der fandt sted i slutningen af 1980'erne: I 1985 blev det britiske band Judas Priest[3] anklaget for at være skyld i to amerikanske teenageres selvmord. Bandet skulle have skjult beskeden "just do it" på et nummer, og dermed have opfordret teenagerne til selvmord. I Nevada i 1990 blev bandet dog frikendt for ugerningen.

❑

2.4 Vandmærker

En særlig anvendelse af steganografi er brugen af vandmærker. Traditionelt er vandmærker i brevpapir og pengesedler en prægning af papiret, der skal sikre, at det er svært at lave nøjagtige kopier, og at forsøg på det kan afsløres. Dette gøres for at garantere, at den person eller organisation, der har ret til at sætte vandmærket, også har skrevet på brevpapiret eller fremstillet penge-

[3] Se bl.a. `www.parascope.com/articles/0497/sublim5.htm`

sedlen. Denne brug af vandmærker kan overføres til elektroniske medier, hvor vandmærker kan indlejres i både lyd, billeder og film.

En anvendelse af vandmærker i forbindelse med tv-transmissioner kan ses i Figur 1, hvor den danske tv-station TV2s logo er indsat i øverste venstre hjørne af skærmbilledet. Hvis udsendelsen, hvor billedet er taget fra, senere genudsendes på en anden tv-kanal, er det altså muligt at se, at udsendelsen oprindeligt blev udsendt af TV2.

Figur 1: TV-billede med synligt vandmærke. Taget fra TV2s hjemmeside (www.tv2.dk).

Figuren er et eksempel på et *synligt* vandmærke. Det er dog langt fra i alle tilfælde, at et synligt, eller observerbart, vandmærke er ønskeligt. Hvis vandmærket skal angive, hvem der har ophavsretten til et medium, hvem mediet er tiltænkt, eller som en plombering af mediet, er det ikke nødvendigvis ønskeligt, at vandmærket kan observeres, og måske derved fjernes. Mange kræfter er derfor blevet brugt på camouflering af ikke-observerbare vandmærker i f.eks. digitale billeder. For at kunne anvendes i denne forbindelse, bør vandmærker opfylde en række krav (jf. bl.a. [Cox&Miller 97]). I [Voyatzis&Pitas 98] gives en liste af krav, der gælder for vandmærkning af billeder; stort set uden ændringer vil kravene også kunne gælde for film og lyd:

1. Vandmærker skal ikke være observerbare. Efter vandmærkning af et billede, skal billedet således ikke være synligt forringet. En forringelse er i sig selv ikke ønskeligt, og forringelsen kan desuden give et fingerpeg om, at billedet er underkastet vandmærkning, hvilket igen vil have betydning for en eventuel angriber.
2. Vandmærker bør være robuste over for almindelige ændringer af billedet. Dette skiller vandmærker fra steganografi i øvrigt: Det vil ofte være muligt at gemme ganske store mængder data i et billede, som så vil kunne blive ramt af simple ændringer, såsom skalering af billedet eller digital/analog- og analog/digital-konvertering. Et vandmærke bør også dække hele billedet således, at det ikke er muligt at tage et udsnit, uden at vandmærket helt eller delvist kan udtrækkes herfra.

3. Det skal være muligt at tilføje et antal vandmærker til samme billede. Hvis alle behandlinger af et billede (f.eks. kopiering) indsætter et vandmærke i billedet, vil det være muligt at bestemme billedets historik. Det bør således kun være ejeren af billedets ophavsret, der kan fremvise det oprindelige billede (uden vandmærker).
4. Ved en sammenligning af billeder mærket med samme vandmærke, bør det ikke være muligt at udlede vandmærket ved hjælp af statistiske metoder. Det bør heller ikke være muligt at fjerne vandmærket ved f.eks. at tage gennemsnitlige pixelværdier i et antal vandmærkede billeder.
5. Det bør være muligt at genskabe et camoufleret vandmærke, uden kendskab til det oprindelige billede. Dette krav skyldes, at det kan være besværligt at finde det oprindelige billede i eksempelvis et større billedarkiv, for at kunne identificere ophavsretindehaveren. Dog skal den algoritme, hvormed vandmærket er camoufleret, være kendt, da det ellers besværliggør udtrækningen af vandmærket.
6. Vandmærker skal kunne modstå en række ondsindede angreb. Dette betyder, at det ikke skal være ligetil at fjerne eller ødelægge et vandmærke i et billede, uden ødelæggelse af billedet til følge, med mindre man har viden om vandmærkets tilblivelse og placering.

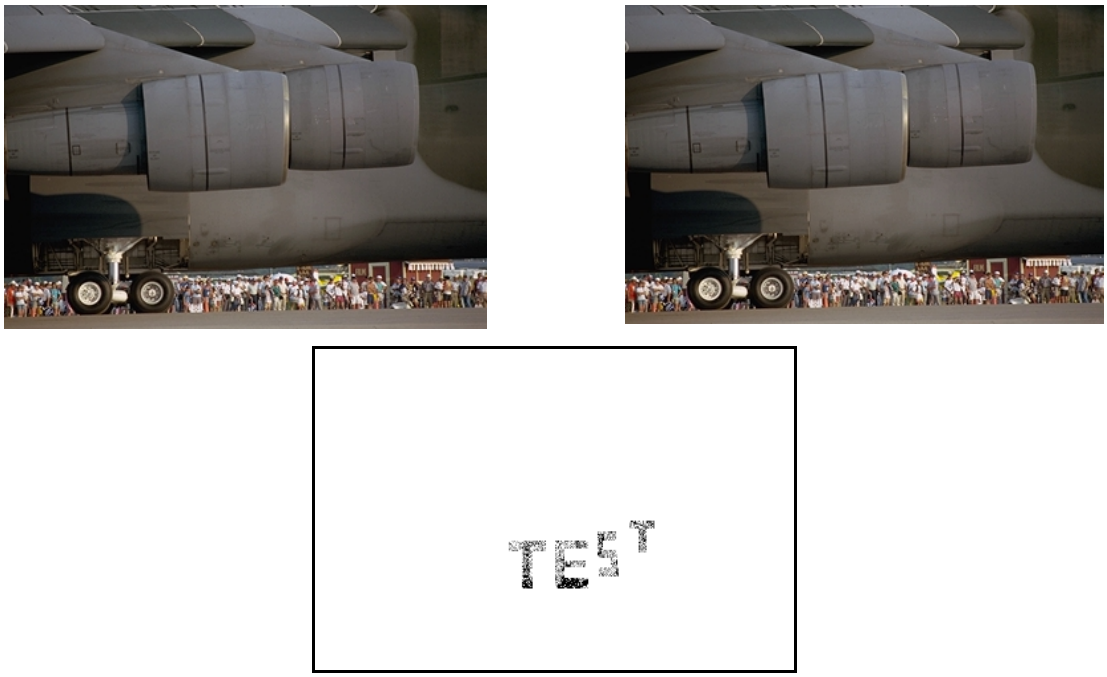
Der er i litteraturen beskrevet mange metoder til at camouflere vandmærker i f.eks. digitale billeder, og langt fra alle metoder opfylder de ovenstående krav. [Cox&Miller 97] og [Swanson et al 98] giver samlet en bred oversigt over beskrevne metoder til vandmærkning af lyd, billeder og film. De tidligste metoder camouflerede vandmærker i områder af billedet med en lav *perceptuel vigtighed*, dvs. områder, som ikke var vigtige for menneskesynets opfattelse af billedet. Dette skyldes, at det er muligt at ændre mere på billedet i disse områder, end i perceptuelt vigtige områder, før ændringen kan ses. Desværre bevirker dette også, at det er lettere at fjerne indsatte vandmærker, enten med vilje eller som resultat af f.eks. billedkomprimering, hvor en algoritme genkender de dele af et billede, som ikke er perceptuelt vigtige som dele der kan undværes eller ændres. Derfor beskæftiger nyere vandmærkningsmetoder sig med indkodning af vandmærker i de perceptuelt vigtige dele af billedet. Dette er mere kompliceret, da ændringer i disse dele af billedet skal ske, så de ikke ødelægger billedet. Til gengæld er det også væsentligt sværere at fjerne disse vandmærker igen, da en fjernelse af vandmærket, uden et nøjagtigt kendskab til, hvordan det er indsat, oftere vil resultere i en synlig forringelse af billedet.

Vandmærkningsmetoder kan opdeles i grupper, afhængigt af hvilke forudsætninger, der er krævet for at ind- og afkode data i et billede. I [Kutter&Petitcolas 99] opdeles metoderne i følgende fire kategorier:

1. Privat vandmærkning. Her er det oprindelige billede nødvendigt, før vandmærkerne kan findes. En afkodningsalgoritme vil her kunne give de oprindeligt camouflerede data, eller blot kunne afgøre, hvorvidt et påstået stegobillede indeholder et camoufleret vandmærke.
2. Semi-privat vandmærkning. Her benyttes det oprindelige billede ikke, og afkodningsalgoritmer kan kun afgøre, hvorvidt et billede indeholder et vandmærke. Sammen med de private vandmærkningsalgoritmer vil de semi-private vandmærkningsalgoritmer fortrinsvis finde anvendelse i forbindelse med afgørelse i strid om ophavsrettigheder, hvor det kan afgøres, om et givent billede indeholder netop det vandmærke, som ophavsretsindehaveren har placeret i billedet.
3. Offentlig vandmærkning. Denne type vandmærkningsalgoritmer kræver hverken kendskab til det oprindelige billede eller til vandmærket. Givet det oprindelige billede og kendskab til vandmærkningsalgoritmen er det muligt at udtrække et vandmærke.
4. Asymmetrisk vandmærkning. Denne form for vandmærkning er steganografiens pendant til asymmetriske kryptering. Ved krypteringen, der også er kendt som offentlig nøgle-algoritmer, kan alle læse en given besked, men kun en kan skrive den. Ved asymmetrisk vandmærkning kan alle, med kendskab til den anvendte algoritme, komme til at læse de camouflerede data, mens kun den part, der oprindeligt har camoufleret data, kan komme til at fjerne dem igen.

2.4.1 Det spatielle domæne

Spatielle vandmærker er vandmærker, der ændrer på områder af billedet. Et eksempel herpå er metoder, hvor områder af et billede kopieres til andre områder, med en lignende tekstur. Således skulle det ikke være muligt at se en direkte forskel på det oprindelige billede, og det billede, hvor der er blevet byttet om på områderne. En digital sammenligning af de to billeder vil dog vise de områder, hvor der er forskelle i billederne, og dermed afsløre de camouflerede data. Et eksempel på dette ses i nedenstående figur. I det andet billede er tekstur blevet kopieret rundt i billedet, uden at dette skulle give en synlig forskel i forhold til det første billede, der er det oprindelige. Det tredie billede viser forskellen mellem de to første billeder. Jo mørkere dette billede er, desto større er forskellen mellem de første to billeder på det pågældende område.



Figur 2: Vandmærke i spatielt domæne. Forskellen på de to øverste billeder er vist i det nederste billede.

Det har vist sig, at de usynlige vandmærker, der er baseret på det spatielle domæne ikke er særligt robuste. For et vandmærke placeret i det spatielle domæne gælder nemlig, at vandmærket skal placeres, så det ikke ødelægger observatørens indtryk af billedet. Derfor placeres vandmærket i områder af billedet, som ikke er "vigtige" for observatørens opfattelse af billedet. Da disse områder har denne karakteristik, er de også forholdsvis lette at ændre igen, så vandmærket forsvinder, uden at dette ødelægger billedet for observatørens. Camouflering af data i frekvensdomænet, hvor camoufleringen foregår i "vigtige" områder af billedet, har derimod vist sig at være væsentligt mere robuste, da vandmærkerne her ikke kan fjernes uden synlig ødelæggelse af billedet til følge.

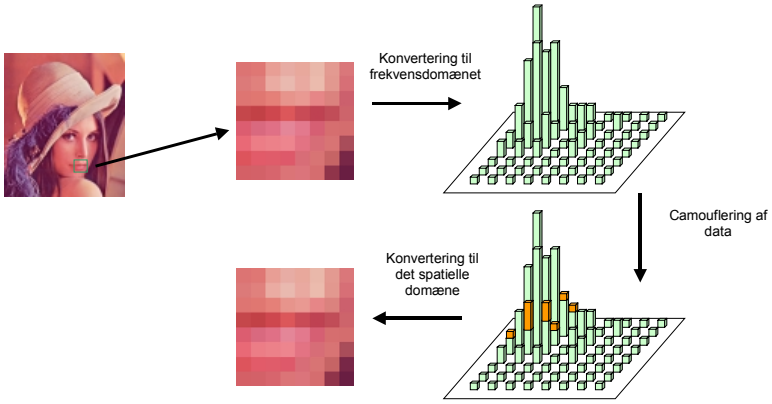
Vandmærkning i det spatielle domæne har dog også visse fordele idet vandmærkning her ofte er betydeligt mindre beregningskrævende end vandmærkning i frekvensdomænet. Derfor kan vandmærkning af store billedmængder, der skal foregå hurtigt, og hvor robusthed ikke er vandmærkningens primære formål, med fordel foregå i det spatielle domæne.

2.4.2 Frekvensdomænet

Hvor camoufleringsmetoder, der opererer i et billedes spatielle domæne ændrer direkte på udvalgte pixels værdier, så ændrer metoder, der virker i frekvensdomænet, pixelværdierne på en mere indirekte måde. Her konverteres pixelværdier i et område af billedet til en frekvensrepræsentation af området, hvorefter der opereres på frekvenskoefficienterne. Frekvensdomænet har

en stor anvendelse i f.eks. JPEG-komprimering (se bl.a. [Wallace 91] og [Pennebaker&Mitchell 93]), hvor områder på 8×8 pixels ved brug af den diskrete cosinustransformation (*Discrete Cosine Transform – DCT*) konverteres til en basis med 64 koefficienter. Efter en diskretisering, hvor mange af disse koefficienter får meget små værdier (tæt på nul), komprimeres koefficientværdierne således, at de mindste værdier falder bort. I [Pennebaker&Mitchell 93] begrundes brugen af denne fremgangsmåde med, at flere forsøg har vist, at metoden i større eller mindre grad stemmer overens med de funktioner menneskesynet benytter.

Vandmærkning i frekvensdomænet foregår ofte efter *spread spectrum* metoden, der oprindeligt er anvendt i kommunikation (se bl.a. [Cox et al 95]), hvor et signal, beregnet på en smal båndbredde, sendes over en meget bredere båndbredde, så energien af signalet i hvert enkelt delbånd bliver næsten "usynlig". Metoden til vandmærkning går, kort fortalt, ud på at camouflere data i et billede, ved at konvertere hele billedet, eller dele deraf, til frekvensdomænet, ændre på nærmere bestemte koefficienter, så ændringen er så lidt synlig som mulig, og herefter konvertere den ændrede frekvensrepræsentation af billedet tilbage. Figur 3 viser denne proces.



Figur 3: Camouflering i frekvensdomænet foregår ved ændring af koefficienter i frekvensrepræsentationen.

I ovenstående figur ses et billedudsnit på 8×8 pixels repræsenteret ved 64 reelle koefficienter. Metoder, der benytter sig af reelle koefficienter, baseres på frekvensrepræsentationer som DCT, Walsh, og Hadamard (se [Gonzales&Woods 93]). Dette gælder f.eks. metoden, der beskrives i [Cox et al 95]. Blandt artikler, der beskriver metoder, som benytter komplekse koefficienter, kan nævnes [Ó Ruanaidh et al 96], hvor man benytter *den diskrete fourier-transformation* (DFT) og camouflerer i fasen af de komplekse koefficienter. Menneskesynet er tilsyneladende er mere sensitiv over for ændringer her end i koefficienternes magnitude.

At ændringer af koefficienter i frekvensdomænet i en vis udstrækning kan baseres på menneskesynets opfattelse af et billede har flere fordele. For det første er det muligt at ændre ganske meget i et billede ud fra en model af menneskesynet, således at vandmærkningen kan ske med så lille synlig ændring af et billede som muligt. En anden fordel ved denne form for vandmærkning er, at den meget udbredte JPEG-komprimering benytter samme model. Dette gør, at man kan fremstille vandmærker, der i en vis udstrækning er i stand til at "overleve" i et billede, selvom billedet JPEG-komprimeres.

3. Introduktion til digitale billeder

Dette afsnit giver en introduktion til beskrivelse af digitale billeder. Beskrivelsen deles op i billedmodeller, farvemodeller og filformater, hvor billedmodellerne beskriver indholdet af et digitalt billede, farvemodellerne beskriver repræsentationer af farver, og filformater beskriver, hvordan de digitale billeder opbevares.

3.1 Billedmodeller

For at kunne behandle billeder digitalt, er det væsentligt at afgøre, hvordan billederne skal repræsenteres. De billeder, der skal repræsenteres digitalt, dækker mange forskellige modaliteter, lige fra "naturlige" billeder, f.eks. digitale feriefotos, over konstruerede landskaber og logoer, til CAD/CAM-tegninger. Billedets repræsentation bestemmes til dels af billedmodaliteten, og dels af den sammenhæng billedet skal anvendes i.

I [Murray&vanRyper 96] beskrives en række forskellige grafiske filformater (og en del af disse beskrives også i afsnit 3.3). Disse filformater beskriver alle billeder efter enten *pixelbaserede billedmodeller*, *vektorbaserede billedmodeller* eller *metamodeller*, hvor sidstnævnte beskriver billeder ved an blanding af de første to. Herunder beskrives kort hvad pixelbaserede og vektorbaserede billedmodeller er.

3.1.1 Pixelbaserede billedmodeller

Et pixelbaseret digitalt billede P er et rektangulært område, bestående af s søjler, indeholdende hver r pixels (*picture elements* = "*pixels*").

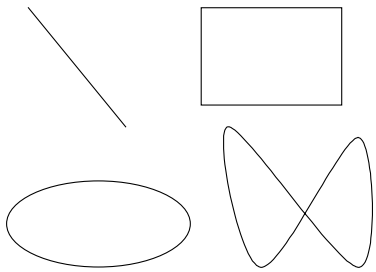
Figur 4: Eksempler på pixelbaserede billeder. Her et binært billede, og et billede med mange farver.

Hver pixel har en farve eller en gråtone, der beskrives ved en farvemodel. Uanset hvilken farvemodel der benyttes beskriver den et billedpunkt som en vektor $\underline{p}$ af værdier. Fortolkningen af elementerne i $\underline{p}$ er bestemt af, hvilken farvemodel, der benyttes. De mest almindelige farvemo-

deller er alle tredimensionale, men har hver deres fortolkning af de tre værdier (se afsnit 3.2). Elementerne i $\underline{p}$ kan desuden beskrives ved et varierende antal bit, og forskellige elementer i samme $\underline{p}$ kan udmærket beskrives med forskellige bitdybder.

3.1.2 Vektorbaserede billedmodeller

Ved vektorbaserede formater beskrives indholdet af et billede enten ved angivelse af simple vektorer (linier) eller ved brug af forholdsvis simple matematiske beskrivelser af kurver. I forhold til pixelbaserede billeder er det en forholdsvis uoverkommelig opgave at beskrive et naturligt billede i et vektorbaseret format. Formatet har dog også flere fordele i forhold til de pixelbaserede formater. For det første er formatet beregnet til konstruerede figurer, hvor enkle linier og kurver sammen danner selve billedet. For det andet er vektorbaserede billeder beregnet til at blive skaleret.



Figur 5: Eksempler på vektorbaserede figurer.

Der er stor anvendelse af vektorbaserede billedmodeller i CAD-/CAM-billeder, der ofte beskriver emner beskyttet af ophavsret. Det er derfor ikke uden betydning, at kunne camouflere f.eks. vandmærker i disse modeller.

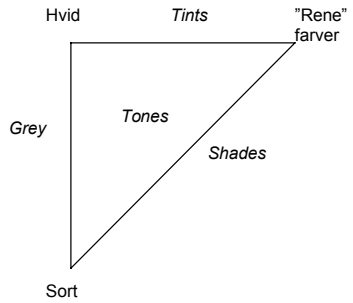
3.2 Farvemodeller

For at kunne fremstille digitale farvebilleder, er det nødvendigt at kunne give en beskrivelse af de involverede farver. Dette afsnit giver derfor først en kort fysisk beskrivelse af farver, og dernæst beskrivelser af en lille del af de farvemodeller, der findes til beskrivelse af farver.

3.2.1 Beskrivelse af farver

Synligt lys er elektromagnetisk stråling med bølgelængde i intervallet fra ca. 400 nm til ca. 700 nm. Når man opfatter en farve, skyldes opfattelsen den sammensætning af bølger med forskellige bølgelængder, der er i det lys, som falder ind i øjet. Farven kan altså beskrives ved den *spektrale energifordeling* $P(\lambda)$, dvs. fordelingen af lys med bølgelængder indenfor det synlige område. Når en farve skal beskrives, og farver skal sammenlignes med hinanden, er det dog sjældent at dette gø-

res med spektrale energifordelinger. Normalt vil man forsøge at beskrive farven ved et sæt af parametre, der er mere intuitive at forstå. I *Ostwalds farvemodel* (der i [Foley et al, 96] også omtales som *kunstnermodellen*) beskrives farver som en blanding af "rene" farver samt sort og hvid. Farver, der er opstået ved en blanding af en ren farve og hvid omtales som *tints*, en blanding mellem en ren farve og sort er en *shade*, og en blanding mellem hvid og sort er en *grey*. Blandinger mellem rene farver, hvid og sort kaldes *tones*, og en tone beskrives ved blandingsforholdet mellem indholdet.

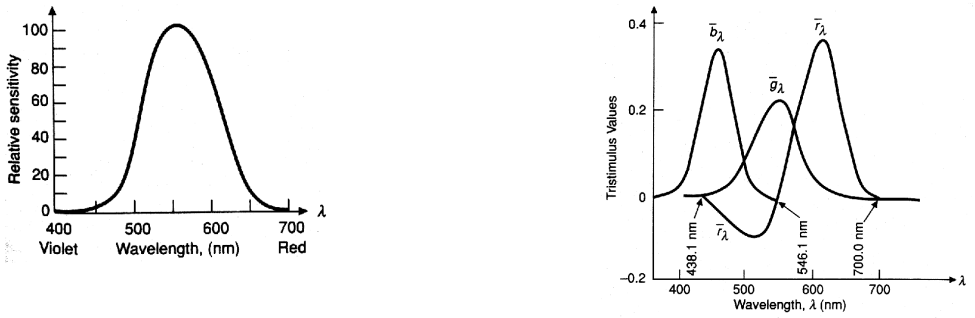


Figur 6: Ostwalds farvemodel, der i [Foley et al, 96] også omtales som "kunstnermodellen".

En anden udbredt beskrivelse af farver kendes fra *Munsell-systemet* (jf. [Foley et al, 96]). I Munsell-systemet beskrives en farve ved dens *hue* eller *chroma*, der er den nærmeste 'rene' farve og dens *saturation*, der er et mål for, hvor mættet farven er af den 'rene' farve. Desuden bestemmes farven ved dens lyskomponent, der er en akromatisk størrelse. To forskellige farver kan have samme lyskomponent, mens forskellige lyskomponenter kan beskrive samme grundfarve (f.eks. marineblå og himmelblå). I Munsell-systemet beskrives lyskomponenten ofte som *lightness* eller *brightness*. Her defineres lightness som størrelsen af den opfattede intensitet af lyset fra et reflekterende objekt, mens brightness hentyder til lyset fra et selvlysende objekt (f.eks. en computerskærm).

Parametrene i Munsell-systemet er alle subjektive, og det kan derfor være svært at beskrive og gendanne farver ud fra en Munsell-beskrivelse. De anvendte parametre har dog en fysisk pendant, der kan benyttes til at beskrive en farve med objektive parametre. En farves hue kan derfor bestemmes ved farvens *dominerende bølgelængde* (λ_{Dom}, se afsnit 3.2.2), saturation kan beskrives ved *mætningsrenheden* (*excitation purity*) og både lightness og brightness kan beskrives ved *luminansen*, der angiver den mængde lys, som forlader en genstand. Mange forskellige spektralenergifordelinger vil give samme beskrivelse af en farve, hvis de beskrives med det fysiske tripel (λ_{Dom}, mætningsrenhed, luminans). Dette betyder, at forskellige fordelinger vil opfattes ens. Fordelinger, der opfattes ens, kaldes *metamere*.

I Figur 7(a) ses *luminous-effekt* funktionen for det menneskelige øje. Funktionen beskriver øjets respons til lys med konstant luminans og varierende λ_{Dom}. Som det fremgår af figuren, er øjet mest følsomt over for farver der ligger i det gul-grønne område omkring 550 nm.



(a) (b)

Figur 7: Luminous-effekten. Graferne er taget fra [Foley et al, 96]

Rødt, grønt og blåt lys kan blandes til mange af de farver, der kan ses, og tidlige teorier om menneskesynet gik på, at farveopfattelsen var baseret på opfattelsen af netop disse tre farver, der betegne *primærfarverne*. I Figur 7(b) ses, hvordan rødt, grønt og blåt lys skal sammensættes for at opnå funktionen i Figur 7(a), og som det fremgår af figuren, kan nogle farver kun fremkomme ved negativ vægtning af enkelte af de primære farver. Dette betyder ikke, at rød, grøn og blå ikke kan blandes til andre farver, men blot at ikke alle farver kan vises ved denne sammensætning. En computerskærm, der netop sammensætter farverne rød, grøn og blå, kan gengive mange farve men altså ikke alle de farver, som er synlige for mennesker.

3.2.2 CIE-modellen

For at kunne beskrive farver ved tre grundfarver, uden at skulle benytte negativ vægtning af enkelte farver, beskrev *Commision Internationale de l'Eclairage* (CIE) i 1931 en farvemodel, der kan beskrive alle synlige farver ved en positiv vægtning af tre grundfarver, døbt **X**, **Y** og **Z**. De tilsvarende vægtningsfunktioner, der tilsammen kan beskrive grafen i Figur 7(a), får således navnene $\bar{x}_\lambda$, $\bar{y}_\lambda$ og $\bar{z}_\lambda$. For at bestemme, hvor stor en mængde af **X**, **Y** og **Z** der skal bruges til at beskrive en farve med den spektrale energifordeling $P(\lambda)$ benyttes:

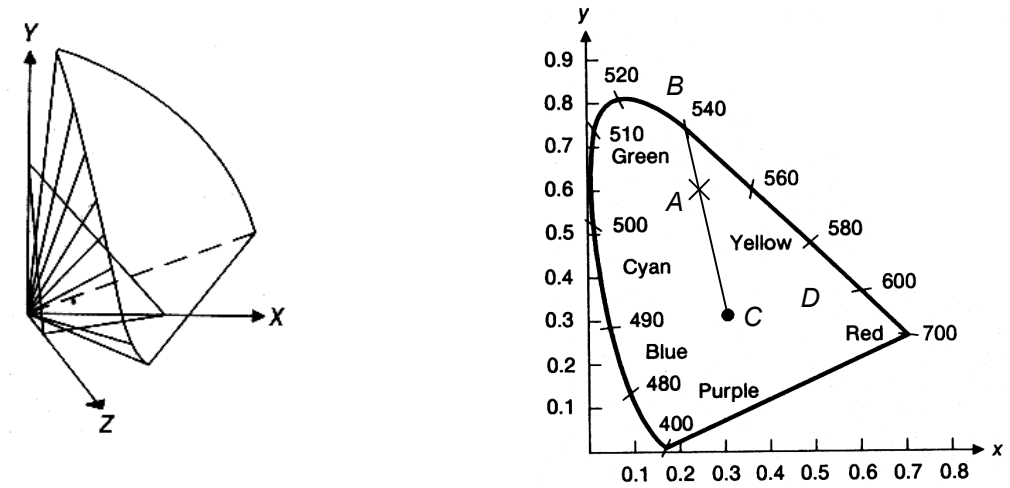
$$X = k \int P(\lambda)\bar{x}_\lambda \, d\lambda$$

$$Y = k \int P(\lambda)\bar{y}_\lambda \, d\lambda$$

$$Z = k \int P(\lambda)\bar{z}_\lambda \, d\lambda$$

For et selvlysende objekt, som f.eks. billedrøret i en computerskærm, sættes k til 680 $\frac{\text{lumens}}{\text{watt}}$.

Da alle synlige farver således kan beskrives ved en entydig sammensætning af **X**, **Y**, og **Z**, er det nærliggende at betegne en farve ved dens beliggenhed i **XYZ**-rummet. I Figur 8(a) ses dette rum, samt den kegle der definerer de synlige farver. Sammensætninger af **X**, **Y** og **Z**, der kommer uden for denne kegle, er altså ikke synlige for mennesker.



(a) (b)

Figur 8: CIE-diagrammet. (a) Keglen med synlige farver. (b) Projektionen af trekanten fra (a) på XY-planet. Fra [Foley et al, 96].

Lad (X, Y, Z) være de vægte på **X**, **Y** og **Z** der skal til for at beskrive en farve **C**. Så defineres *kromaticitetsværdierne*, der altså ikke er afhængige af farvernes lyskomponent, som:

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$

Da $x + y + z = 1$ må x, y og z være på planet $(X + Y + Z = 1)$. Ved projektion af dette plan ned på (X,Y)-planet fremkommer CIE-diagrammet, der ses i Figur 8(b), hvor punktet med navnet *illuminant C* er markeret. Illuminant *C* er valgt som det punkt, der angiver sollys' farve.

Punktet *B* i figuren angiver en "ren" farve med λ_{Dom} = 540 nm. Farven *A* har ligeledes λ_{Dom} =540 nm, da linien gennem *C* og *A* skærer periferien af det synlige område ved *B*. *A* er dog ikke en "ren" farve da *mætningsrenheden* for *A* ikke er 100%. Mætningsrenheden bestemmes ved:

$$Mætningsrenhed = \frac{|AC|}{|AB|} \cdot 100\%$$

Med farverne *B* og *C* er det muligt at blande enhver farve på liniestykket *BC*, og alle farver på dette linestykke har λ_{Dom} = 540 nm. Med farverne *B*, *C* og *D* er det således muligt at få alle farver i *BCD*-trekanten.

Da farverne rød, grøn og blå (r, g, b) også danner en trekant i CIE-diagrammet, er det muligt at danne alle farver inden for denne trekant, ved en blanding af rød, grøn og blå. Placering af punkterne r, g og b i CIE-diagrammet afhænger af det apparat, der skal fremvise farverne, og selvom de alle tre placeres som 'rene' farver, vil der stadig være farver i diagrammet, der ikke kan nås ved en kombination af r, g og b. Dette er altså endnu et bevis på, at det ikke er muligt at fremstille alle synlige farver ved en positiv kombination af de tre primærfarver.

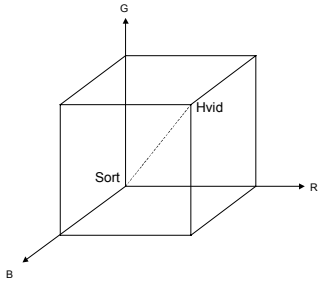
3.2.3 RGB-modellen

På trods af, at rød, grøn og blå ikke kan kombineres til fremvisning af alle synlige farver, er RGB-modellen (for *R*ed, *G*reen og *B*lue) en meget anvendt farvemodel. Desuden er RGB-modellen en basismodel for en lang række andre farvemodeller, hvor beregning af farveværdier til og fra RGB-modellen giver et fælles grundlag.

Beskrives vægtningen af R, G og B med $\underline{p} = (r, g, b)$, hvor $r, g, b \in [0;1]$, vil en dæmpet gul være givet ved $\underline{p} = (0,½,½)$, sort har $\underline{p} = (0,0,0)$ mens hvid har $\underline{p} = (1,1,1)$.

I Figur 9 ses den kube, der indeholder de farver, som kan sammensættes af primærfarverne. Diagonalen fra (0,0,0) til (1,1,1) indeholder de grå nuancer fra sort til hvid. Ved konvertering fra et RGB-billede til et gråtonebillede kan en farve projiceres ind på denne diagonal, og den resulte-

rende grå nuance anvendes. Her benyttes altså en afbildning fra et tredimensionalt rum til et endimensionalt rum, og den resulterende $\underline{p}$-vektor er altså endimensional, $\underline{p} = \frac{r+g+b}{3}$.



Figur 9: RGB-kuben, der indeholder de farver, som beskrives ved RGB-modellen.

I de tre hjørner, der er modsat de *additive* primærfarver (R, G, og B), findes de *subtraktive* primærfarver cyan, magenta og gul (se afsnit 3.2.4). Cyan har $\underline{p} = (0,1,1)$, magenta har $\underline{p} = (1,0,1)$ og gul har $\underline{p} = (1,0,0)$. At R, G og B betegnes som additive beror på, at farver dannes ved at lægge rødt grønt og blåt lys sammen. Samme effekt kan opnås ved at trække cyan, magenta og gult lys fra hvidt lys, hvorfor disse farver betegnes som subtraktive.

3.2.4 CMY-modellen

CMY-modellen (C står for *Cyan*, M for *Magenta*, og Y står for *Yellow*, og ikke for Intensit*y* som i YIQ-modellen, se afsnit 3.2.7) er RGBs komplementære model. Den benyttes i forbindelse med farvetryk, hvor modellen ofte er suppleret med sort trykfarve, da denne ellers skulle fremstå ved en blanding mellem C, M og Y. I denne sammensætning kaldes modellen for CMYK (*C*yan, *M*agenta, *Y*ellow, Blac*k*)[4]. Omregning fra RGB-værdier til CMY-værdier er givet ved:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

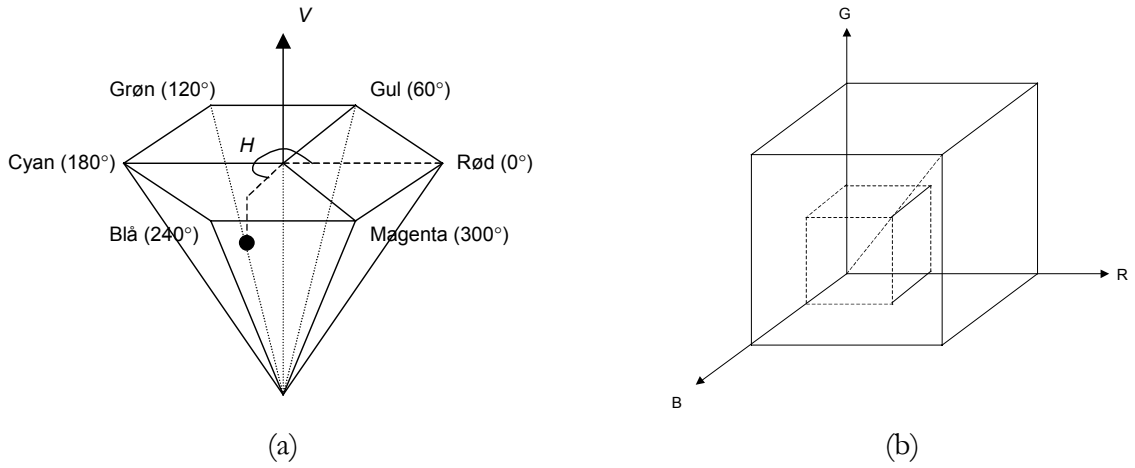
3.2.5 HSV-/HSB-modellen

Hvor både RGB- og CMY-modellen er designet til at beskrive farver som de bliver frembragt, så er HSV-/HSB-modellen designet til at beskrive farver som menneskesynet gør det. HSV står for

[4] At *K*, og ikke *B*, vælges som repræsentant for *Black* skyldes, at *B* kunne skabe forvirret sammenligning med *Blue* i RGB-modellen.

*H*ue, *S*aturation og *V*alue, og modellen er nogle gange beskrevet som HSB, hvor B står for *B*rightness.

Hvor farver i RGB- og CMY modellen blev beskrevet ud fra deres koordinater i en farvekube, så beskrives farverne i HSB ud fra deres placering i en keglestub (se Figur 10(a)).



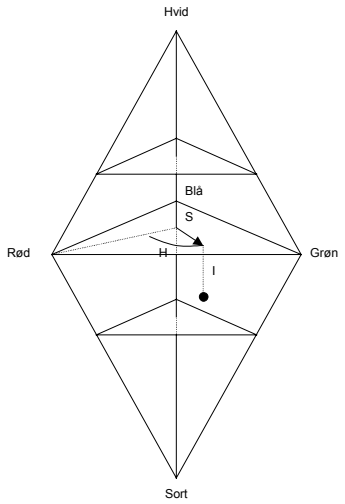
Figur 10: HSV-modellen. (a) En farve beskrives ved sine H, S og V-værdier. (b) HSV-keglen som en sub-kube af RGB-kuben.

En farves hue er bestemt som vinklen mellem linien fra keglens centrum til farven, og linien fra centrum til rød. F.eks. har grøn således et hue-tal på 120°. Farvens mætning er sat til 0% i keglens centrum og 100% på keglens sider. Siderne er sat som den begrænsning RGB-modellen har af CIE-diagrammet, og dette betyder, at en *mætning* på 100% for det meste har en mætningsrenhed på mindre end 100%. Farvernes brightness eller value er sat til 0 i keglens spids og til 1 i på keglens topflade. Selvom modellen ligner den menneskelige beskrivelse af farver, er den blot en anden måde at beskrive farverne i RGB-/CMY-kuben på. Hvis man således ser på keglen direkte ovenfra, er billedet det samme som hvis man så på RGB-kuben fra det lyse hjørne, sigtende mod origo (se Figur 10(b)). Værdien af V er således den normerede afstand til origo, og alle farver med en given V-værdi vil ligge på overfladen af en sub-kube af RGB-kuben. Dette gør, at farver med samme V-værdi ikke alle vil have samme perceptuelle brightness. F.eks. vil hvid, rød, grøn og blå alle have en V-værdi på 1, selvom hvid i alle tilfælde vil have den største perceptuelle brightness.

3.2.6 HSI-modellen

Selvom HSB-/HSV-modellen angiver parametre, der efterligner menneskesynet, så gør inkonsistensen ved V-/B-værdien, at modellen kan være svær at benytte til bestemmelse af de faktorer ved et digitalt billede, der påvirker menneskesynet. Det kan derfor være en fordel af benytte HSI-

modellen istedet. HSI står for *H*ue, *S*aturation og *I*ntensity, og farverne er her beskrevet ved deres placering i en dobbeltkegle (se Figur 11)



Figur 11: HSI-dobbeltkegle. En farve er beskrevet ved vinklen fra Rød (H), afstanden fra centrum (S) og lodret placering (I).

Som i HSB-/HSV-modellen er hue angivet som vinklen mellem linien gennem centrum og rød, og linien gennem centrum og farven. Saturation er ligeledes angivet som forholdet mellem afstanden fra centrum til farven og afstanden fra centrum til kanten af dobbeltkeglen. Intensiteten er angivet som farvens lodrette placering, således at farver tæt på sort har en intensitet tæt på 0, mens farver tæt på hvid har en intensitet tæt på 1. Farver med ens intensitetsværdier opfattes således også som havende den samme brightness, hvilket gør, at denne model er tilpasset menneskesynet bedre end HSB-/HSV-modellen.

I [Gonzales&Woods 93] gives en detaljeret gennemgang af konversionerne fra RGB-værdier til HSI-værdier og fra HSI-værdier til RGB-værdier. Omregning fra RGB-værdier til HSI-værdier er givet ved:

$$H = \arccos\left(\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right)$$

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right]$$

$$I = \tfrac{1}{3}(R+G+B)$$

Hvis $B/I > G/I$ sættes:

$$H = 360° - H\,.$$

Omregning fra HSI-værdier til RGB-værdier er afhængig af H-værdien. For $0° < H \leq 120°$ er omregningerne givet ved:

$$
\begin{aligned}
B &= \tfrac{1}{3}(1 - S) \\
R &= \tfrac{1}{3}\left(1 + \frac{S \cos H}{\cos(60° - H)}\right) \\
G &= 1 - (+b)
\end{aligned}
$$

For $120° < H \leq 240°$ er omregningerne givet ved:

$$
\begin{aligned}
H &= H - 120° \\
R &= \tfrac{1}{3}(1 - S) \\
G &= \tfrac{1}{3}\left(1 + \frac{S \cos H}{\cos(60° - H)}\right) \\
B &= 1 - (R + G)
\end{aligned}
$$

For $240° < H \leq 360°$ er omregningerne givet ved:

$$
\begin{aligned}
H &= H - 240° \\
G &= \tfrac{1}{3}(1 - S) \\
B &= \tfrac{1}{3}\left(1 + \frac{S \cos H}{\cos(60° - H)}\right) \\
R &= 1 - (G + B)
\end{aligned}
$$

3.2.7 YIQ-modellen

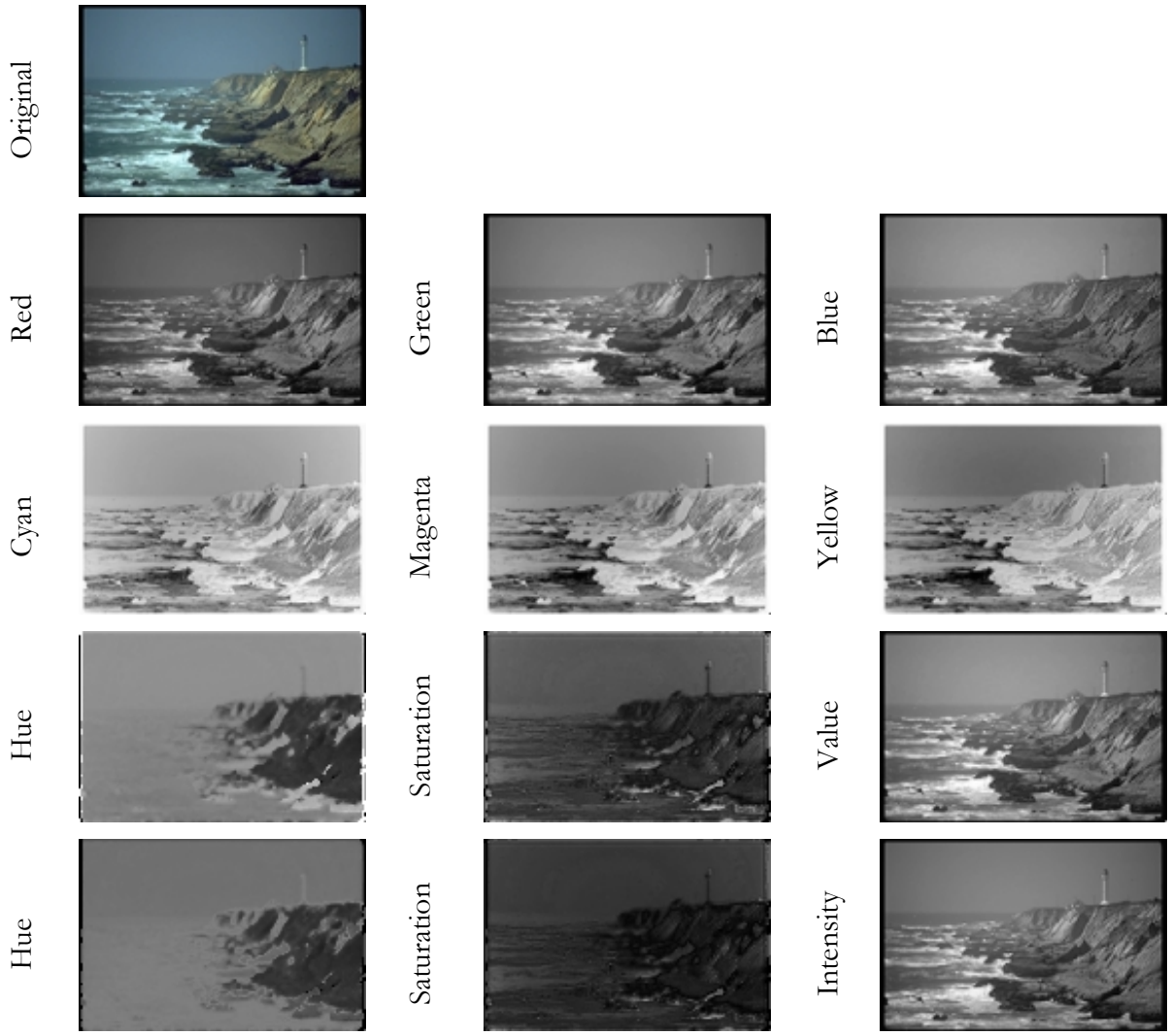
HSI-modellen efterligner menneskesynet bedre end HSB-/HSV-modellen. Lysintensiteten er dog baseret på en ligelig fordeling af farverne rød, grøn og blå, hvilket ikke svarer særlig godt overens med menneskesynets opfattelse af farver (mere herom i afsnit 5). En bedre repræsentation af lysintensiteten benyttes af YIQ-modellen, der bl.a. benyttes ved transmission af TV-signaler. Y-delen af repræsentationen omhandler en beregnet intensitet ud fra RGB-værdierne. Denne intensitet bruges bl.a. i sort/hvide-fjernsyn, hvor det er den eneste værdi, der anvendes. I- og Q-komponenterne indeholder farveinformationen i et billedpunkt. Da Y er den vigtigste kompo-

nent repræsenteres den med flest bit, hvilket efterligner måden mennesket opfatter lys på. Omregning fra RGB-værdier til YIQ-værdier foregår ved:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ 0{,}596 & -0{,}275 & -0{,}321 \\ 0{,}212 & -0{,}523 & 0{,}311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

3.2.8 Sammenligning af farvemodeller

I Figur 12 ses resultatet af et originalt billede, der er opsplittet i de beskrevne farvemodellers komponenter.



Figur 12: Sammenligning af farvemodeller. Samme billede er gengivet i forskellige farvemodeller.

3.3 Filformater

Farvemodellerne beskrevet i forrige afsnit er vigtige for den repræsentation et digitalt billede vil have ved behandling i en computer. Til lagring af billederne, enten til brug for senere behandling, eller til udveksling mellem computere, benyttes forskellige filformater, der kan være lige så vigtige for billedernes steganografiske karakter, som den interne repræsentation (bl.a. farvemodellerne) er. Der er flere grunde til at overveje, hvilke filformater, der kan benyttes ved steganografisk behandling af digitale billeder:

1. Nogle steganografiske metoder kræver, at alle pixelværdier i et stegobillede bliver gemt nøjagtig som de er, hvis den camouflerede information skal kunne afkodes igen. Derfor er tabsgivende filformater, der i forbindelse med komprimering af data ændrer på pixelværdier i billedet, ikke nødvendigvis velegnede til steganografi. Vandmærkning af billeder skal helst være robust over for konversion mellem forskellige filformater, også de tabsgivende, og bl.a. derfor er mange vandmærkningsmetoder optimeret til at kunne "overleve" en JPEG-komprimering.
2. Et billede er kun velegnet som medie for camouflerede data, hvis en udenforstående part kan håndtere billedet, uden at der fattes mistanke til billedets indhold af camouflerede data. Dette kræver bl.a., at billedet gemmes i et almindeligt kendt format, som ikke vil tiltrække sig opmærksomhed.
3. I opbygningen af en steganografisk metode er det praktisk hvis billedindholdet er relativt nemt tilgængeligt. Nogle filformater er i denne sammenhæng bedre egnet end andre, da billeddata i disse formater kan være nemmere at håndtere.

I Tabel 1 beskrives en række gængse billedformater, som bør overvejes i forbindelse med arbejdet med en steganografisk metode. Tabellen er ikke tænkt som en udtømmende liste over filformater, men kun som en liste af filformater, der kan overvejes. En nærmere beskrivelse af DICOM kan findes i [DICOM 93], mens en beskrivelse af de resterende filformater kan findes i [Murray&vanRyper 96]. I tabellen nævnes en række kompriomeringsalgoritmer. En teknisk beskrivelse af disse algoritmer kan også findes i [Murray&vanRyper 96].

Navn:	Forkortelse:	Beskrivelse:
Graphics Interchange Format	GIF	Når et billede gemmes i GIF-format, bliver der genereret en farvetabel indeholdende 256 farver. Da et billedes ofte beskrives ved 3×8 bit, altså 16.777.216 farver, er GIF ikke nødvendigvis et bevarende filformat. Hvis camoufleringen dog resulterer i pixelværdier, der ligger inden for GIF-formatets 256 farver, vil camoufleringen kunne "overleve" en konvertering til GIF.
Joint Photographic Experts Group File Interchange Format	JPEG	JPEG-filformatet benytter JPEG-komprimering, hvilket ofte kan resulterer i mindre ændringer af pixelværdier til fordel for mindre filstørrelser. Da JPEG-filer ofte er væsentlig mindre end filer med ikke-komprimerede billeder, har de vundet stort indpas på WWW. Bl.a. derfor er mange vandmærkningsalgoritmer optimeret til at kunne overleve en JPEG-komprimering.
Microsoft Windows Bitmap	BMP	BMP er det grundlæggende filformat til brug under Microsoft Windows™. Formatet findes i flere versioner, der er tilpasset forskellige versioner af Windows, og det bruges bl.a. til lagring af grafiske komponenter i brugergrænsefladen. Formatet er meget anvendt under Windows, hvor de fleste grafik-, dtp- og tekstbehandlingsprogrammer kan håndtere formatet. Formatet benytter 4-bit run-length encoding (RLE4) til komprimering, og da dette ofte ikke er særligt effektivt, benyttes formatet sjældent til naturlige billeder.
Tag Image File Format	TIFF	TIFF er et meget alsidigt bitmap-format, der bl.a. ofte bruges ved lagring af indscannede billeder. De fleste grafik-, dtp- og tekstbehandlingsprogrammer kan behandle en delmængde af billeder, der er gemt i TIFF-formatet, men da formatet er så bredt, at det f.eks. både dækker komprimering af data (med f.eks. RLE, LZW eller JPEG), og da data er gemt som enten binær eller ASCII, er der ofte dele af formatet, som ikke er dækket af en implementation.
Portable Bitmap Utillities	PBM, PGM, PPM, PNM	Formatet benyttes til binære billeder (Portable Bitmap, PBM), gråtonebilleder (Portable Greymap, PGM) og 3×8 bit RGB-billeder (Portable Pixmap, PPM). Endelig kan filer i PNM-

Navn:	Forkortelse:	Beskrivelse:
		formatet (Portable ANymap) indeholde de tre andre formater. Formatet er meget simpelt; det indeholder en lille ASCII-header, der fortæller om det anvendte format, antallet af farver og billedets størrelse. Derefter kommer billeddata, der enten kan være ASCII-tekst eller binært repræsenteret. Da formatet er så simpelt, kan det i forbindelse med steganografi med fordel benyttes til udvikling og afprøvning af algoritmer, Formatet er imidlertid ikke meget udbredt, og en endelig implementering af en algoritme bør derfor ikke anvende PBM-formatet.
Truevision Targa Image File	TGA	TGA-formatet er udviklet til understøttelse af billedbehandling på specifikke grafikkort fra Truevision. Dette til trods er formatet blevet en vidt udbredt standard, der også kan benyttes af ikke-Truevision grafikkort. I sin oprindelige udformning minder TGA-formatet om PBM-formatet, hvor en simpel header angiver formatet af de (typisk rå) data, der følger. TGA-formatet giver mulighed for en simpel RLE-kompression, men da denne ofte ikke anvendes har TGA-filer en tendens til at blive meget store. Formatet er altså, som PBM, mest egnet til udvikling af algoritmer, og mindre egnet til den egentlige implementation.
	HIPS	HIPS-formatet har en ringe udbredelse indenfor den kommercielle billedbehandling, men anvendes ofte indenfor udvikling af algoritmer til billedbehandling. Som både PBM og TGA indeholder HIPS en beskrivende header efterfulgt af billeddata. Til forskel fra de simplere formater, har HIPS dog muligheden for at behandle billeddata der er andet end heltallige værdier. Dette gør anvendeligheden af HIPS som intern repræsentation i en algoritme til et stærkt værktøj. Til udveksling af stego-billeder gør HIPS' ringe udbredelse dog, at formatet ikke er egnet.

Navn:	Forkortelse:	Beskrivelse:
	DICOM	DICOM-formatet nævnes her som et kuriosum. Inden for den medicinske verden vinder DICOM-formatet frem som de facto standarden, og flere og flere applikationer til behandling af medicinske billeder skal kunne behandle DICOM-billeder. Ved camouflering af data i medicinske billeder (som nævnt i 'Eksempel: Camouflering i medicinske billeder'), er det derfor en nødvendighed at undersøge, hvordan dette kan gøres med billeder, der er gemt i DICOM-formatet.

Tabel 1: Kort beskrivelse af nogle grafiske filformater.

4. Perceptuel tolerance og steganografisk kapacitet

I dette afsnit introduceres begreberne *perceptuel tolerance* og *steganografisk kapacitet*. Den perceptuelle tolerance er et mål for, hvor meget et givent billede kan ændres, før det opfattes som ændret, og den steganografiske kapacitet er et mål for, hvor meget data der kan camoufleres i billedet, under forudsætningen af, at der ikke er perceptuel forskel på det oprindelige billede og billedet der camouflerer data. Endelig sluttes dette afsnit af med en gennemgang af en metode, for hvilken den steganografiske kapacitet er let at bestemme.

4.1 Begreber

Det billede, der skal camoufleres data i, og som ofte vil blive refereret til som det originale billede, kaldes også for *værtsbilledet*, og det billede, der perceptuelt ligner værtsbilledet, men som indeholder camouflerede data kaldes *stegobilledet* (jf. [Pfitzmann]). Vi skelner mellem information og data på følgende måde: *data* er en mængde af bit, der kan camoufleres i et billede, og disse bit kan så repræsentere en given *information*, f.eks. om billedets ophavsret, eller hvem billedet er tilsendt.

Den perceptuelle tolerance i et billede er en angivelse af, hvor meget billedet kan ændres, dels i hver af billedets pixels, og dels billedet samlet, uden at billedet dermed er perceptuelt ændret. En grafisk repræsentation af fordelingen af perceptuel tolerance i et billede kan opnås med et *tolerancebillede*, der gengiver tolerancen for områder i billedet.

Billedets evne til at camouflere data betegner vi *den steganografiske kapacitet*[5]. Kapaciteten måles i antallet af bit der kan camoufleres i et givent billede, og den afhænger af den metode der vælges til camoufleringen.

4.2 Anvendelse

At kunne gemme data i et billede kan f.eks. bruges i forbindelse med indlejrede oplysninger om selve billedet, evt. som ASCII-tekst, der forklarer motiv og indehaver af ophavsret. Metoden kan også anvendes ved vandmærkning/plombering, hvor den perceptuelle tolerance angiver grænserne for vandmærkets/plombens ændring i billedet, den steganografiske kapacitet angiver, hvor meget vandmærket/plomben kan fylde, og tolerancebilledet angiver, hvor vandmærket/plomben kan anbringes.

---

[5] I [Hansen&Andersen 98] gives andre definitioner på steganografisk kapacitet.

JPEG og andre billedkomprimeringsalgoritmer beskæftiger sig med irreversibel datakompression, hvor et billede, der først komprimeres, ikke nødvendigvis vil dekomprimeres til nøjagtigt det samme billede. Den perceptuelle tolerance kan her benyttes til vurdering af en komprimeringsalgoritmes "venlighed" over for et billede. Hvis forskellen på et billede før og efter komprimering med efterfølgende dekomprimering er større end det originale billedes tolerance, har algoritmen ændret på synligt på billedet; algoritmen har ikke været "venlig". At kunne bestemme, hvornår en komprimeringsalgoritme er "venlig" over for et billede kan bl.a. se sin anvendelse ved større billeddatabaser, hvor nogle billeder kan komprimeres betydeligt mere end andre, uden at den stærkere komprimering giver en forringelse af billedet. Ved en automatisk indstilling af f.eks. de kvantiseringstabeller JPEG-algoritmen benytter til komprimeringen, vil man kunne opnå en mere optimal komprimering, hvor billedernes kvalitet samtidig er bevaret.

4.3 Afgrænsning

Det skulle umiddelbart være muligt at beregne mål for den perceptuelle tolerance og den steganografiske kapacitet for alle former for digitale medier. Vi har i dette speciale valgt at afgrænse os til digitale farvebilleder af flere årsager:

1. Det er relativt nemt for os at komme i besiddelse af elektroniske billeder, som vi kan arbejde med. Der findes et meget stort antal billeder på Internet, og vi kan fremstille vore egne, enten ved indskanning eller ved konstruktion i et tegneprogram.
2. At der findes store mængder af billeder frit tilgængeligt på Internet gør også, at behovet for bedre metoder til vandmærkning af digitale billeder er stigende. Vi mener, at en bestemmelse af et billedes perceptuelle tolerance kan få betydning for, hvor godt billedet kan vandmærkes.
3. Billeder er nemme for os at bruge som dokumentation. I en rapport som denne er billeder nemmere at behandle, og intuitive at forstå, end f.eks. afbildning af lyde ville være.
4. At tage skridtet fra enkeltbilleder til film er ikke så stort. Her ville de metoder, der kan benyttes for billeder, sandsynligvis kunne videreføres, og en del flere metoder ville formentlig komme til. F.eks. kan man ikke nå at fokusere på alle detaljer i de billeder, der sammenlagt udgør filmen, hvorfor nogle af disse billeder formentlig kan være ændret ganske betydeligt, uden at dette vil ændre oplevelsen af filmen
5. Der findes en række billedformater, som kan læses og behandles på en lang række forskellige systemer (Windows, UNIX, Mac m.m.). Derfor kan resultaterne af de forsøg vi udfører ses og viderebehandles stort set uafhængigt af behandlerens system.

En yderligere afgrænsning kunne være, at vi kun beskæftigede os med gråtonebilleder. Vi har dog valgt at behandle farvebilleder, da en del af de fænomener ved menneskesynet, som vi finder interessante, netop omhandler farvesynet. Vi har desuden valgt at beskæftige os med forholdsvis små og simple digitale billeder, da de som regel er nemme at arbejde med, og uden videre kan illustrere de emner vi beskæftiger os med.

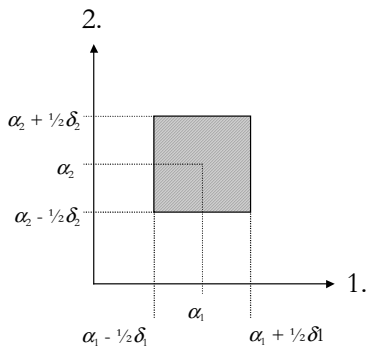
4.4 Den perceptuelle tolerance

Den perceptuelle tolerance i en pixel afhænger dels af pixelens placering i billedet (se afsnit 5.3) og af den farvemodel, der er benyttet i billedet (se afsnit 3.2). En pixel $\underline{p}$ kan opfattes som en pixel-vektor med dimension af det antal værdier, pixelen repræsenteres ved. For RGB-modellen, hvor hver pixel repræsenterer tre værdier, er $\underline{p} = (p_r, p_g, p_b)$, hvor p_r, p_g og p_b betegnes som pixelkomponenter. En pixel kan have en tolerance for hver pixelkomponent i $\underline{p}$.

Den perceptuelle tolerance for en pixelkomponent betegner det område som komponenten kan antage værdier i, uden at en værdiændring vil resultere i et synligt ændret billede. Dette område kan være et interval omkring komponentens værdi i værtsbilledet, hvilket betyder, at camouflering i billedet kan ske ved at ændre komponentens værdi lidt op eller ned, så længe værdien ikke overskrider intervallets grænser. Der er dog ingen grund til at begrænse den perceptuelle tolerance til et interval omkring den oprindelige værdi.

4.4.1 Toleranceområder

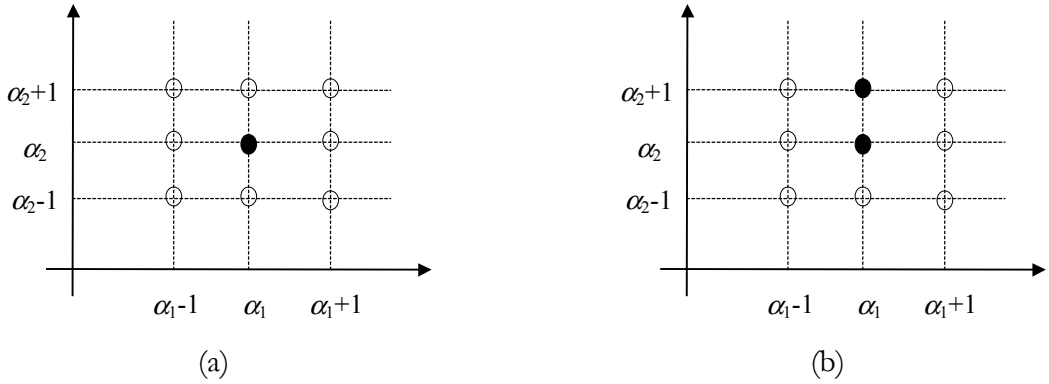
I den simpleste fortolkning har hver pixelkomponent p_i i pixelen $\underline{p}$ en perceptuel tolerance på $p_i \pm ½\delta$, altså med en tolerancebredde på δ. Denne tolerance er vist geometrisk i Figur 13, der for illustrationens skyld viser en pixel i en todimensional farvemodel.



Figur 13: Den perceptuelle tolerance i to dimensioner

I Figur 13 ses det tydeligt, at δ_1 og δ_2 udspænder et areal omkring centerpunktet (α_1, α_2). I den generelle fortolkning, hvor en pixel kan være repræsenteret ved mere end to værdier, er der tale om rumfanget af det udspændte legeme. I afsnit 4.6 beskrives en metode til camouflering af data, der benyttes sig af de kombinationer af værdier der ligger inden for toleranceintervallerne. I denne forbindelse vil den steganografiske kapacitet (se afsnit 4.5) være lig med rumfanget udspændt af δ_1 og δ_2.

Ovenstående beskrivelse passer på pixelkomponenter, der kan antager alle reelle værdier indenfor et givent interval. Som oftest vil pixelværdier dog kun kunne antage heltallige værdier i deres repræsentation, hvorfor intervallerne skal tage højde for dette. I det tilfælde, hvor både α_1 og α_2 i Figur 13 kun kan antage deres oprindelige værdier, dvs. at enhver anden tildeling af værdier til disse pixelkomponenter vil overskride den perceptuelle tolerance (markeret med åbne cirkler), vil intervaller med længden nul give, at δ_1 og δ_2 udspænder et areal på nul, som det kan ses i figuren. Da arealet kan være et mål for den steganografiske kapacitet, og dermed angiver, hvor mange bit der kan camoufleres i et område, er dette plausibelt.



Figur 14: Intervalbredde. (a) En kapacitet på 0 bit. (b) En kapacitet på 1 bit.

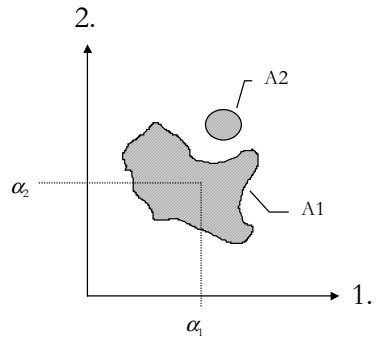
I Figur 14(b) ses dog et eksempel, der vil stride i mod denne tildeling af intervallængder. Her er den første pixelkomponent igen fastlagt til α_1 mens den anden komponent kan veksle mellem α_2 og α_2+1, uden at dette vil give nogen perceptuel ændring. Sættes tolerancebredden for den første komponent til nul vil dette igen give et resulterende nul-areal. I dette tilfælde vil det dog være muligt at indkode en enkelt bit, idet der er to muligheder for at tildele pixelkomponenter værdier i den viste pixel. Produktet af tolerancebredderne bør derfor være antallet af mulige kombinationer af pixelkomponentværdier, uden at den perceptulle tolerance overskrides. Intervalbredden for heltallige værdier for pixelkomponenterne med symmetriske toleranceintervaller er altså givet ved:

$$
\begin{aligned}
I &= 1 + (\alpha + \tfrac{1}{2}\delta) - (\alpha - \tfrac{1}{2}\delta) \\
&= 1 + \delta
\end{aligned}
$$

Der er dog ingen grund til at antage, at en pixel kun kan have værdier indenfor et så symmetrisk område, som den foregående diskussion har langt op til. En pixelkomponent kan have en skæv fordeling af tolerancen, og vil derfor have to δ-værdier: δa og δb. Herved bliver en enkelt pixel-komponents samlede toleranceinterval givet ved:

$$
\begin{aligned}
I &= 1 + (\alpha + \delta b) - (\alpha - \delta a) \\
&= 1 + \delta b + \delta a
\end{aligned}
$$

Selv med den ovenstående udvidelse beskrives kun rektangulære toleranceafbildninger, som den der ses i Figur 13. Der er imidlertid intet der forhindrer, at afbildningen kan beskrives ved en vilkårlig funktion. Funktionen, der beskriver afbildningen, behøver hverken at være monoton eller kontinuert. Som det ses i Figur 15, er området A2 ikke sammenhængende med området A1. Dette kunne give mening i et billedområde, hvor der er tekstur (af f.eks. sort, hvid og gul). Her må et punkt kun ændres i forhold til de værdier der indgår i teksturen; hvis punktet bliver rødt kan det skille sig ud, og dermed være observerbart.



Figur 15: Irregulære og ikke-sammenhængende toleranceområder.

Typisk vil de enkelte pixelværdier kun kunne varieres afhængigt af hinanden. Det er derfor ikke nok at sige, at en ændring i en pixel kun er "lovlig" (dvs. at ændringen ikke kan ses), hvis enhver ændring indenfor toleranceintervallet for hver enkelt pixelkomponent er "lovlig". F.eks. kan det tænkes, at alle pixelkomponenter må ændres udover den øvre ende af intervallet, hvilket for RGB-modellen vil resultere i en lysere pixel, hvis blot alle omkringliggende pixels også blev sat lysere, så en enkelt pixel ikke længere vil skille sig ud. En anden mulighed kan være, at pixels må ændres meget i farven, blot der ikke ændres i intensiteten af betydning. Er pixelen beskrevet ved

HSI-modellen (eller tilsvarende) kan ændringerne her begrænser sig til H-komponenten. Er beskrivelsen i stedet sket i henhold til RGB-modellen, er der tale om en koordineret ændring af R-, G- og B-komponenterne. Der kan derfor være tale om yderligere begrænsninger på tildelingen af nye værdier til pixelkomponenter end dem der sættes af toleranceintervallet.

Eksempel: Fastholdt intensitet

I en pixel $\underline{p}$ beskrevet efter RGB-modellen med komponenterne $(R, G, B) = (0\,,\, 0{,}5\,,\, 0{,}5)$ er farven efter camouflering bestemt til ikke at være vigtig. Imidlertid er det vigtigt, at pixelens samlede intensitet ikke ændres med mere end 10%. Hvis alle komponenter får et toleranceinterval, der strækker sig 10% på hver side af originalværdien, vil dette være overholdt. Dette udelukker dog en del kombinationer af pixelkomponenter, da den resulterende farve ikke er vigtig. For hver enkelt kombination kan det derfor undersøges, om kombinationen overholder de begrænsninger, der sættes for den ændrede pixel.

Oprindelig kombination: $(R, G, B) = (0\,,\, 0{,}5\,,\, 0{,}5)$

Oprindelig intensitet: $\dfrac{R+G+B}{3} = \dfrac{0+0{,}5+0{,}5}{3} = 0{,}33$

Intensitetsinterval: $[0{,}33 - 10\%;\ 0{,}33 + 10\%] = [0{,}30;\ 0{,}37]$

Oprindelig hue-værdi: $\arccos\left(\dfrac{\frac{1}{2}[(0-0{,}5)+(0-0{,}5)]}{\sqrt{(0-0{,}5)^2+(0-0{,}5)(0{,}5-0{,}5)}}\right) = 3{,}14$

Kombination 1: $(R, G, B) = (0{,}1\,,\, 0{,}4\,,\, 0{,}6)$

Intensitet: $\dfrac{0{,}1+0{,}6+0{,}6}{3} = 0{,}43$

Hue-værdi: $2\pi - \arccos\left(\dfrac{\frac{1}{2}[(0{,}1-0{,}4)+(0{,}1-0{,}6)]}{\sqrt{(0{,}1-0{,}4)^2+(0{,}1-0{,}6)(0{,}4-0{,}6)}}\right) = 3{,}55$

Kombination 2: $(R, G, B) = (0{,}9\,,\, 0{,}1\,,\, 0)$

Intensitet: $\dfrac{0{,}9+0{,}1+0}{3} = 0{,}33$

Hue-værdi: $\arccos\left(\dfrac{\frac{1}{2}[(0{,}9-0{,}1)+(0{,}9-0)]}{\sqrt{(0{,}9-0{,}1)^2+(0{,}9-0)(0{,}1-0)}}\right) = 0{,}10$

Af de to kombinationer vil den første resultere i en farve, der med hue-værdi (ifølge HSI-modellen, her målt i radianer) ligger tættest på den oprindelige farve. Den resulterende farve vil dog være mere end 10% lysere end den oprindelige, hvorfor denne kombination ikke kan bruges. Den anden kombination resulterer i en farve, der er en hel del mere rød end den oprindelige farve. Til gengæld har den samme intensitet som den oprindelige farve, hvorfor kombination 2 er en lovlig erstatning for den oprindelige kombination.

❑

Ovenstående eksempel viser, at det ikke altid er muligt at sætte et toleranceinterval, der tillader alle de kombinationsmuligheder, som ikke resulterer i synlige ændringer af billedet. I eksempel kunne intervallerne ganske vist være blevet sat, hvis pixels var baseret på HSI-modellen i stedet for RGB-modellen. At benytte en anden repræsentation af pixels vil dog ikke løse alle problemer. F.eks. kan der blive tilladt større variationer i den blå pixelkomponent end i henholdsvis den røde og den grønne, hvilket ville være i overensstemmelse med synets behandling af farver. Denne forskel i variationer kunne så kræve, at pixels blev repræsenteret efter RGB-modellen.

4.4.2 Tolerancebilleder

For at få et overblik over, hvor store ændringer der kan gøres i et værtsbillede, samt hvor i billedet der kan ændres mest, har vi valgt at definere og fremstille *tolerancebilleder*.

Lad $\underline{\underline{T}}$ være den $m \times n$-matrix, der repræsenterer tolerancebilledet, hvor m er billedhøjden, og n er billedbredden, og lad matrix-elementerne $t_{i,j}$ repræsentere toleranceintervallerne for de enkelte pixels.

$$\underline{\underline{T}} = \begin{bmatrix} t_{11} & \Lambda & t_{1n} \\ \mathrm{M} & & \mathrm{M} \\ t_{m1} & \Lambda & t_{mn} \end{bmatrix}$$

Da $\underline{\underline{T}}$ skal repræsentere tolerancerne i et billede med en x-dimensionel farvemodel (hvor x er et heltal, typisk 3), er det nærliggende, at lade $t_{i,j}$ være vektorer af x dimensioner. For en tredimensionel farvemodel som RGB kan $t_{i,j}$ have udseendet

$$t_{i,j} = \left(I_r, I_g, I_b\right),$$

hvor I_k er antallet af tilladte værdier den k'te komponent kan antage, som defineret i afsnit 4.4.1. For RGB-billeder vil udseendet af $\underline{\underline{T}}$ kunne fortælle, både i hvilke områder af værtsbilledet der kan ændres mest, samt i hvilke af de tre farvekanaler de største ændringer kan foretages. Jo lysere et område er, desto mere kan værtsbilledet ændres, og hvis et område af $\underline{\underline{T}}$ kan ændres mere i en farvekanal end i de andre to, vil den farvekanal, der kan ændres mest i, være den fremtrædende i området. Benyttes i stedet en model baseret på menneskesynet, såsom HSI eller HSV, er den umiddelbare fortolkning af billederne mindre gennemskuelig. Da hue i begge de nævnte modeller har 0 ved rød vil røde områder i et billede umiddelbart tyde på en lav tolerance i hue, mens grønne områder har en større tolerance, og blå områder har en endnu større. Områder med den største tolerance ville således også være røde, og den umiddelbare fortolkning af tolerancebillederne ville således kræve et kendskab til de røde farver, der ligger med en hue omkring 0.

Skulle interessen kun gælde de områder af værtsbilledet, der kan ændres mest, kan der ses bort fra farveinformationen af $\underline{\underline{T}}$. Dette kan gøres ved at definere $\underline{\underline{T}}^g$, der er en $m \times n$-matrix, hvor hvert matrixelement $t^g_{i,j}$ indeholder en gråtoneværdi. For RGB-modellen har vi valgt at lade værdien af $t^g_{i,j}$ være bestemt ved:

$$t^g_{i,j} = \left\lceil \frac{I_r + I_g + I_b}{3} \right\rceil,$$

der svarer til en intensitetsværdi efter HSI-modellen.

Da I-værdierne i både $\underline{\underline{T}}$ og $\underline{\underline{T}}^g$ ofte ikke er særligt høje, set i forhold til værdierne i et tilsvarende gråtone billede (en typisk værdi for $t^g_{i,j}$ vil være i størrelsesordenen 3-5 i et 8-bit gråtonebillede, med 256 gråtoneværdier), er forskellene i tolerance ofte ikke særligt tydelige, og tolerancebilledet vil derfor se ensartet sort ud. En simpel måde at forstærke forskelle mellem dele af billedet med lav kapacitet og dele med høj kapacitet, er at multiplicere alle $t_{i,j}$-værdier med en given faktor, og så afrunde de værdier, der falder uden for gråtoneintervallet, til intervallets grænser. Vi definerer derfor tolerancebillederne $\underline{\underline{T}}^{50}$ og $\underline{\underline{T}}^{g50}$, hvor alle pixelkomponenter i et 8-bit gråtonebillede, $t^{50}_{i,j}$ og $t^{g50}_{i,j}$, har værdierne:

$$t_{i,j}^{50} = \max\left(255,\ 50 \cdot t_{i,j}\right)$$

og

$$t_{i,j}^{g50} = \max\left(255,\ 50 \cdot t_{i,j}^{g}\right)$$

En anden metode, der i højere grad fremhæver forskellene, er brugen af 2-potenser. For *I*-værdierne i et 8-bit gråtonebillede og i et 3×8-bit RGB-farvebillede, defineres derfor tolerance-billederne $\underline{\underline{T}}^{pot}$ og $\underline{\underline{T}}^{gpot}$, hvor alle pixelkomponenter, $t_{i,j}^{pot}$ og $t_{i,j}^{gpot}$, har værdierne:

$$t_{i,j}^{pot} = \min\left(255,\ 2^{t_{i,j}} - 1\right)$$

og

$$t_{i,j}^{gpot} = \min\left(255,\ 2^{t_{i,j}^{g}} - 1\right)$$

I Figur 16 ses otte billeder, der skal illustrere brugen af tolerancebilleder[6], og forskelle mellem dem. De første to billeder danner grundlaget for tolerancebillederne, idet det første billede indeholder de mindste værdier af toleranceintervallerne, og det andet billede indeholder de største værdier. Herefter vises de seks former for tolerancebilleder, der er beskrevet herover.

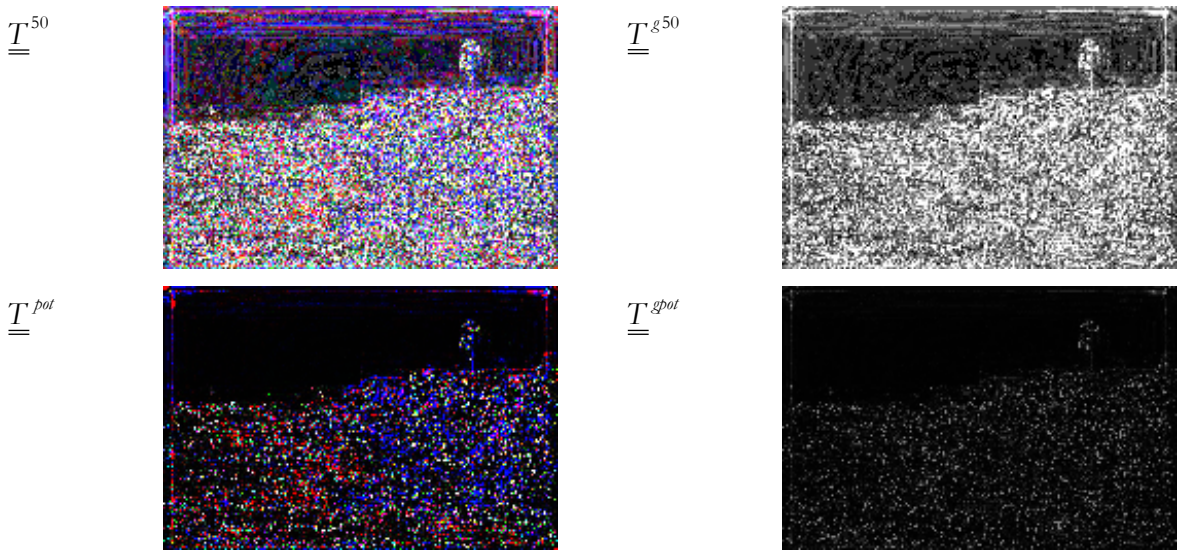
Min.

Max.

$\underline{\underline{T}}$

$\underline{\underline{T}}^{g}$

[6] Billederne er dannet ud fra et originalt billede, og samme billede udsat for 10%-JPEG-komprimering.



Figur 16: Farvede og gråtonede tolerancebilleder.

4.5 Den steganografiske kapacitet

Den steganografiske kapacitet er, som omtalt tidligere, en angivelse af den maksimale mængde data, der kan gemmes i et billede, uden at billedets perceptuelle tolerance er overskreden. Denne kapacitet er i høj grad afhængig af, hvordan data skal camoufleres i billedet. For nogle metoder vil hele billedets steganografiske kapacitet være på ganske få bit, mens kapaciteten for andre metoder vil være ganske stor. For vandmærkningsalgoritmer, hvor vandmærkets robusthed ofte er vigtigere end vandmærkets størrelse, er den steganografiske kapacitet generelt mindre end den vil være for steganografiske metoder, der ikke behøver at tage hensyn til robustheden af de camouflerede data. I [Hansen&Andersen 99] anslås vandmærkningsalgoritmernes generelle kapacitet på 30-160 bit pr. billede. Dette vil ofte være i underkanten, hvis det camouflerede vandmærke skal benyttes til f.eks. autentifikation.

For metoder, der benytte sig af statistik på det hele billede, kan det være relevant at måle kapaciteten i bit pr. billede. Disse metoder vil ofte være særdeles robuste over for de fleste angreb, men til gengæld ikke særligt gode til større mængder data. For metoder, der benytte et mindre udsnit af billedet til camouflering, kan det typisk være relevant at måle kapaciteten i bit pr. blok. I metoden beskrevet i [Ó Ruanaidh et al 95] kan metodens kapacitet måles i antallet af bit pr. 8×8 blok, mens kapacitet i [Hansen&Andersen 99] kan måles i bit pr. 11×11 blok. Metoder der benytter sig af ændringer i enkelt-pixels, kan måle kapacitet i bit pr. pixel. Den metode, der beskrives i afsnit 4.6 har en kapacitet der er mange gange højere end f.eks. metoderne nævnt i [Ó Ruanaidh et al

95], målt i bit pr. pixel. Til gengæld er den her beskrevne metode slet ikke robust over for nogen form for angreb på billedet, som beskrevet i afsnit 4.6.3.

4.5.1 Sammenligning af metoder

Da der efterhånden er mange steganografiske metoder til camouflering af data i digitale billeder, og herunder en hel del, der beskæftiger sig med vandmærkning, kan man savne en måde at sammenligne de forskellige metoder på. Dette er dog ikke en triviel opgave, da metoder er optimeret til vidt forskellige forhold. Et mål til sammenligning kan være kapaciteten af metoderne, hvadenten den er målt i bit pr. pixel, bit pr. blok eller bit pr. billede. Et andet mål kan være, hvor meget anvendelsen af en given metode på et billede ændrer billedet perceptuelt. Dette mål kan nemt stride mod det første mål, idet en metode, der ikke ændrer meget på et billede, måske heller ikke har en særlig stor kapacitet, og vice versa. For metoder til vandmærkning kommer der yderligere et mål om robusthed ind i en sammenligning. Metoder som kombinationsmetoden (se afsnit 4.6), har f.eks. en forholdsvis stor kapacitet, og som en følge af definitionen kun en ganske lille indflydelse på perceptionen af billedet. Som en metode til vandmærkning er metoden imidlertid ikke anvendelig, hvilket beskrives i afsnit 4.6.3. Spørgsmålet om robusthed i forbindelse med deciderede vandmærkningsmetoder er heller ikke her helt simpelt. En definition på robusthed kunne være, at det skulle være muligt at genfinde det camouflerede vandmærke i stego-billedet efter et eller flere angreb på billedet, hvor et angreb bl.a. kan være en rotation af billedet eller en JPEG-kompression. [Kutter&Petitcolas 99] nævner en række mulige angreb på et billede, som et vandmærke camoufleret med en robust metode skulle kunne overleve. Et andet spørgsmål går på, hvordan det afgøres, om et vandmærke kan genskabes efter et angreb.

Som ovenstående skulle afsløre, er der en lang række spørgsmål i forbindelse med en generel sammenligning af metoder, der skal besvares, før metoder kan sammenlignes. Selvom der, så vidt vi ved, endnu ikke eksisterer nogle standard for sammenligningen, så bliver der gjort mange tiltag i mod det. I [Kutter&Petitcolas 99] samles bl.a. en række billeder, som kan udgøre et standardsæt i forbindelse med måling af kapacitet, og generel sammenligning, og denne samling er endnu det tætteste man kommer på en fælles standard.

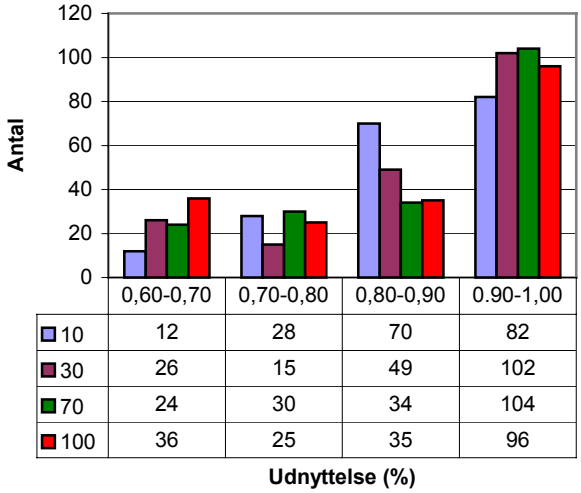
4.6 Kombinationsmetoden

For at bestemme billedets steganografiske kapacitet for denne metode, bestemmes først, for hver pixel, de tolerance-intervaller som værdierne i hver enkelt pixelkomponent kan ligge i. For RGB-modellen er der altså tale om tre intervaller; et for rød, et for grøn og et for blå. Betegnes antallet af værdier i et interval med I_i, $i \in \{r, g, b\}$, bestemmes kapaciteten for en enkelt pixel som

$K = I_r \cdot I_g \cdot I_b$. Dette betyder, at der er K måder at kombinere de værdier for r, g og b, der ligger inden for toleranceintervallerne. Antallet af bit, der kan gemmes i en enkelt pixel, er så givet ved $k = \log_2 K$, der kun sjældent vil være et heltal. Rundes k ned til et heltal, er dette altså det antal bit, der kan gemmes i en pixel. At k ikke rundes op skyldes, at man derved kan opnå kombinationer med værdier, der ligger uden for den perceptuelle tolerance. Kombineres de N pixel, der findes i et billede, er den samlede kapacitet i billedet givet ved $K_{tot} = K_1 \cdot K_2 \cdot \mathbf{K} \cdot K_N$, hvor K_i angiver K for den i'te pixel. $k_{tot} = \log_2 K_{tot}$ angiver således det antal bit, der kan camoufleres i det samlede billede.

Den ovenfor bestemte k-værdi, k_{tot}, er dog kun en teoretisk øvre grænse. At camouflere information på k_{tot} bit vil ofte være en langsommelig proces, da k_{tot} let kan overskride det antal bit en processor arbejder med. Derfor kan det være nødvendigt at begrænse camoufleringen til et mindre antal pixels ad gangen. Ved camouflering af information i én pixel, med en 24 bit-repræsentation af RGB-værdien, skal processoren kunne behandle tal af 24 bit, da dette er den teoretiske øvre grænse for kapaciteten i én pixel, der kan ændres vilkårligt. Normalt vil den mængde bit, der kan camoufleres i én pixel være noget mindre, og eftersom logaritmen af K kun sjældent vil være et heltal, benyttes den steganografiske kapacitet i en enkelt pixel sjældent fuldt ud. Udnyttelsen afhænger derfor af, hvor stor en forskel der er på k_{tot} og nedrundede værdi af k_{tot}.

I Figur 17(b) ses en graf for udnyttelsen af den steganografiske kapacitet ved camouflering i én pixel ad gangen. Grafen er baseret på de tolerancebilleder, der er vist i Figur 16, og den viser den procentuelle udnyttelse ved camouflering i enkeltpixels placeret i forskellige scanlinier af billedet i Figur 17(a). Af billedets 128 linier er udnyttelsen beregnet for linierne 10, 30, 70 og 100.



	0,60-0,70	0,70-0,80	0,80-0,90	0.90-1,00
10	12	28	70	82
30	26	15	49	102
70	24	30	34	104
100	36	25	35	96

(a) (b)

Figur 17: Udnyttelse af kapaciteten forskellige steder i et billede, ved camouflering i én pixel ad gangen.

Figuren viser tydeligt, at camouflering i en enkelt pixel ad gangen i mange tilfælde ville udnytte pixelens steganografiske kapacitet næsten fuldt ud. Det er her værd at bemærke, at en pixel, hvor værdierne af pixelkomponenterne ikke kan varieres, uden at de kommer uden for toleranceintervallerne, vil have en samlet kapacitet på 0 bit. Når der derfor ikke camoufleres data i denne pixel, vil udnyttelsen for pixelen være på 100%. De interessante værdier fra Figur 17(b) må derfor være de værdier, hvor udnyttelsen på 100% ikke skyldes en kapacitet på 0 bit. For de 384 pixels i Figur 17(b), hvor udnyttelsen ligger mellem 90% og 100%, har 30 pixels en udnyttelse på 100% grundet en kapacitet på 0 bit.

I de næste underafsnit beskrives metoder til ind- og afkodning af data i et billede. Metoderne benytter en bestemmelse af heltallige toleranceintervaller og den steganografiske kapacitet, der er beskrevet i herover. Desuden vises, hvorfor denne metode til camouflering af data i et billede ikke er særlig velegnet til vandmærkning, da metoden er særdeles sårbar over for selv små ændringer af stegobilledet. Da metoden baserer sig på kombinationer af pixelværdier inden for toleranceintervallerne, har vi valgt at kalde metoden for *kombinationsmetoden*.

4.6.1 Indkodning

For at indkode data i et antal pixels, skal disse pixels perceptuelle tolerance først bestemmes for alle pixelkomponenter (beskrives i afsnit 6), hvilket resulterer i en *Slutværdi* og en *Startværdi* af intervallet. Herefter bestemmes antallet af værdier, *I*, som den enkelte pixelkomponent kan antage:

$$I = Slutværdi - Startværdi + 1$$

Antallet af kombinationsmuligheder, K_{tot}, for værdier indenfor den perceptuelle tolerance for alle n pixelkomponenter i de N involverede pixels bestemmes (som beskrevet i afsnit 4.6).

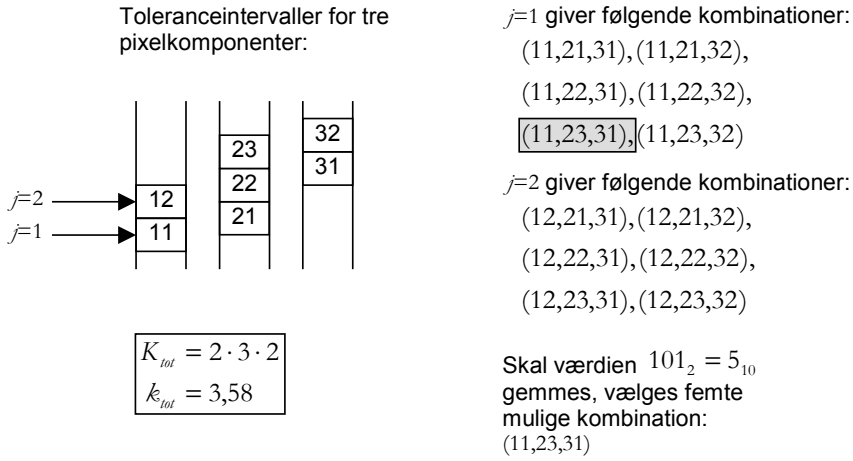
$$K_{tot} = \prod_{i=1}^{N} \prod_{j=1}^{n} I_{i,j}$$

Antallet af bit, k_{tot}, der kan indkodes i de involverede pixels bestemmes herefter som:

$$k_{tot} = \log_2 K_{tot}$$

Endelig beregnes de værdier som de enkelte pixelkomponenter skal antage efter indkodningen af data. For at gøre dette skal kombinationerne nummereres (se Figur 18). Ved nummereringen har den første kombination en værdi i bunden af alle intervaller, den næste har værdier i bunden af alle intervaller, pånær i et interval, hvor værdien lige over bunden antages, osv.

Hvis antallet af kombinationer nummereres nedefra, så vil den j'te værdi i det første interval dække $j/I_{1.\ interval}$ af alle kombinationerne.



Figur 18: Camouflering med kombinationsmetoden.

Til bestemmelse af ændringen af hver pixelkomponent, herunder benævnt Δi, defineres en række størrelser, der beregnes for hver enkelt pixelkomponent i forbindelse med camoufleringen.

f er forholdet mellem antallet af kombinationer, som den enkelte pixelkomponent skal dække, og bredden af toleranceintervallet *I*. For første pixelkomponent er f_{j-1} lig K_{tot}.,.

$$f_j = \frac{f_{j-1}}{I_j}$$

Lad *r* være den rest af kombinationer, der skal dækkes af den næste pixelkomponent. Fra starten er r_{j-1} lig K_{tot}, da ingen kombinationer endnu er dækket:

$$r_j = r_{j-1} - \Delta i_j \cdot f_j$$

Δi er den ændring, der skal lægges til starten af intervallet for en pixelkomponent, når denne skal camouflere data:

$$\Delta i_j = \left\lfloor \frac{r_{j-1}}{f_j} \right\rfloor,$$
$$Start \leq Start + \Delta i_j \leq Slut$$

For at udligne et eventuelt mønster i de bit, der ønskes indkodet, kan disse kombineres med en tilfældigt genereret bitfølge. Hvis de bit, der indkodes f.eks. indeholder lange sekvenser af 1'er, vil det resulterende billede i et område komme til at indeholde de maksimale værdier for alle pixelkomponenter i de involverede pixels. Selvom dette ikke vil give perceptuelle forandringer i billedet, kan en angriber ved behandling af billedet få mistanke til mønstret i de indkodede bit. Hvis angriberen derimod ikke kender indkodningsalgoritmen, men har mulighed for at diktere de data, der skal indkodes, kan dette også give en anvisning af, hvordan algoritmen fungerer. Ved XOR af de data, der skal indkodes med den tilfældige bitfølge, vil dette blive undgået.

Eksempel: Indkodning

I dette eksempel indkodes et antal tilfældigt valgte bit i to pixel. Tabel 2 viser start- og slutværdierne for de perceptuelle toleranceintervaller for alle pixelkomponenter i de to pixels.

	Pixel 1 ($\underline{p}_1$):		**Pixel 2 ($\underline{p}_2$):**	
	Start:	**Slut:**	**Start:**	**Slut:**
Rød	110	113	108	112
Grøn	50	55	51	54
Blå	100	125	98	120

Tabel 2: Start- og slutværdier for toleranceintervaller af to pixels

Først bestemmes K_{tot} og k_{tot} for de to pixels:

$$K(\underline{p}_1) = (113 - 110 + 1) \cdot (55 - 50 + 1) \cdot (125 - 100 + 1) = 624$$
$$K(\underline{p}_2) = (112 - 108 + 1) \cdot (54 - 51 + 1) \cdot (120 - 98 + 1) = 460$$

$$K_{tot} = 624 \cdot 460 = 287040$$
$$k_{tot} = \log_2(287040) = 18{,}13$$

Der kan altså camoufleres en bitfølge på 18 bit i de to pixels. Denne bitfølge kan se således ud:

$b = 01\ 1011\ 0010\ 1100\ 1010_2 = 111306_{10}$

For at bestemme de værdier de to pixels skal have efter camoufleringen, vælges den 111306'te kombination af pixelværdier.

For den første komponent foretages følgende beregning:

$$f = \frac{287040}{4} = 71760$$
$$\Delta i = \left\lfloor \frac{111306}{71760} \right\rfloor = 1$$
$$r = 111306 - 1 \cdot 71760 = 39546$$

Forholdet f og resten r går derefter videre til næste komponent:

$$f = \frac{71760}{6} = 11960$$

$$\Delta i = \left\lfloor \frac{39546}{11960} \right\rfloor = 3$$

$$r = 39546 - 3 \cdot 11960 = 3666$$

Fortsættes beregningerne for de resterende fire komponenter i de to pixels, fås værdierne i Tabel 3.

	Pixel 1 ($\underline{p}_1$):			**Pixel 2 ($\underline{p}_2$):**		
	Start:	**Δi:**	**Værdi:**	**Start:**	**Δi:**	**Værdi:**
Rød	110	1	111	108	4	112
Grøn	50	3	53	51	3	54
Blå	100	7	107	98	9	107

Tabel 3: Beregnede værdier for to pixels efter camouflering af 18 bit.

I ovenstående tabel, er tallene i 'Værdi'-kolonnerne de nye værdier for rød, grøn og blå i de to pixels.

❑

4.6.2 Afkodning

For at afkode informationen gemt i et stegobillede, er kendskab til toleranceintervallernes størrelse, deres startværdi, den oprindelige pixelkomponentværdi, og værdien efter indkodning nødvendig. Selvom indkodningen har resulteret i et stegobillede, der ikke perceptuelt skulle skille sig fra værtsbilledet, og bestemmelsen af toleranceværdierne netop er afhængig af den visuelle opfattelse af billedet, er det langtfra sikkert, at værtsbillede og stegobillede vil give samme toleranceintervaller ved en analyse. Intervallerne vil sandsynligvis være stort set overlappende, med forskellige start- og slutværdier, men dette er ikke nok til, at de oprindelige toleranceværdier kan undværes ved afkodningen. Da en forskel på 1 i bare en enkelt pixelkomponent kan ændre antallet af kombinationer betydeligt, er det derfor strengt nødvendigt, at de oprindelige toleranceintervaller er kendte. Selvom det kun er nødvendigt at kende toleranceintervallerne, og ikke hele det oprindelige billede, er det ikke nødvendigvis en fordel at have disse gemt separeret fra billedet. Ud fra det oprindelige billede, der har en værdi for hver pixelkomponent, skal der beregnes to værdier, nemlig start- og slutværdierne, for hvert toleranceinterval. Da disse jo kan beregnes fra det oprindelige billede, er spørgsmålet om, hvorvidt det er fordel at genbruge toleranceintervallerne blevet til et spørgsmål om lagerplads (til de ekstra data) kontra beregningstid (til genberegning af tolerance-

intervaller). At nøjes med at gemme længden af toleranceintervallerne løser ikke problemet, da det er essentielt, hvor i intervallet, den nye pixelkomponentværdi er placeret. Dette forudsætter naturligvis, at der ikke er tale om komprimering af de gemte værdier, da andre forhold, så som fast intervalbredde m.m. kan spille ind.

Ud fra værtsbilledet, eller et sæt af intervalstart- og intervalslutværdier, bestemmes k_{tot} og I som i det tidligere eksempel, og med stegobilledet bestemmes Δi. For at bestemme, hvilken kombination (og dermed, hvilken bitfølge b) der er gemt i stegobilledet, skal alle pixelkomponenter først nummereres fra 1 til nN, hvor pixelkomponenter fra samme pixel har efterfølgende numre. Derefter benyttes nedenstående:

$$b = \sum_{j=1}^{nN} \Delta i_j \prod_{l=j+1}^{nN} I_l$$

Hvis b i ovenstående kan beskrives med mindre end k_{tot} bit, forlænges b med et passende antal 0-bit på venstre side, så værdien af b ikke ændres.

Eksempel: Afkodning

For de to pixels, der er brugt i det tidligere eksempel, bestemmes I og Δi. Værdierne ses i nedenstående tabel.

	Pixel 1 ($\underline{p}_1$):		**Pixel 2 ($\underline{p}_2$):**	
	***I*:**	**Δi :**	***I*:**	**Δi :**
Rød	4	1	5	4
Grøn	6	3	4	3
Blå	26	7	23	9

Tabel 4: I og Δi for to pixels. Disse værdier skal benyttes i forbindelse med afkodning af camoufleret information.

Med tallene fra Tabel 4 beregnes nummeret på den kombination af værdier, der er brugt:

$$
\begin{aligned}
b = {} & 1 \cdot 6 \cdot 26 \cdot 5 \cdot 4 \cdot 23 + \\
& 3 \cdot 26 \cdot 5 \cdot 4 \cdot 23 + \\
& 7 \cdot 5 \cdot 4 \cdot 23 + \\
& 4 \cdot 4 \cdot 23 + \\
& 3 \cdot 23 + \\
& 9 \\
= {} & 111306_{10} \\
= {} & 1\ 1011\ 0010\ 1100\ 1010_2
\end{aligned}
$$

Da k_{tot} for ovenstående pixels er beregnet til 18 bit, og b kan skrives med 17 bit, må der mangle 18-17=1 bit. Den fulde bitfølge kommer altså til at se ud som:

b = 01 1011 0010 1100 1010

Denne bitfølge er præcis den samme, som den der blev indkodet i det forrige eksempel.

❑

4.6.3 Kombinationsmetoden til vandmærkning

Den her angivne metode, til ind- og afkodning af bit i en kombination af mulige pixelværdier, tjener det steganografiske formål, at camouflere data i et billede, udmærket. Den er dog af flere årsager ikke egnet som vandmærkningsmetode. Til illustration af dette skitseres her, i en række eksempler, hvordan flere parter kan kommunikere med et billede som medium, og hvordan et vandmærke genereret med den omtalte metode vil kunne ødelægges. I eksemplerne benyttes navne på kommunikerende parter taget fra den kryptografiske litteratur. Disse navne er:

Alice Den ene af to kommunikerende parter. Kan være den part, der har ophavsretten til billedet, og som ønsker at sende det til Bob.

Bob Den anden af de to kommunikerende parter. Kan være den part, der skal modtage billedet.

Willy (eller *Warden*). Den part, der overbringer stegobilledet fra Alice til Bob, uden nødvendigvis at være vidende om, at billedet indeholder camouflerede data.

Mallory Den part, som er interesseret i de data, der kan være camoufleret i et billede fra Alice til Bob, og gerne vil have data uden Alice og Bobs vidende, eller som blot ønsker at ødelægge de givne data. Kan evt. samarbejde med Willy.

Metoden er meget ustabil over for ændringer af stegobilledet. Enhver ændring af værdierne for en pixelkomponent vil resultere i en ændring af den valgte kombination i forhold til de lovlige toleranceintervaller. Er der tale om blot en ændring af en enkelt pixelkomponent i det pixelsæt, der benyttes til indkodningen (f.eks. rød i $\underline{p}_1$), vil dette kunne give en ændring i alle de udledte bit for det pixelsæt.

Eksempel: Metoden er ikke robust

Alice camouflerer en meddelelse til Bob i et billede, og gemmer billedet i et format, der bevarer alle pixels værdier uændrede. Alice beder Willy sende billedet til Bob, og da Willy ønsker at gøre dette, uden at bruge for lang tid på dataoverførslen til Bob, komprimerer han billedet med en tabsgivende algoritme. Da Bob modtager billedet, er han ikke i stand til at udtrække den meddelelse Alice har gemt.

❑

Selvom metoden ikke er særlig robust over for ændringer i billedet, er der dog enkelte tilfælde, hvor et vandmærke kan overleve ændringer. Et af disse tilfælde er *cropping*, hvor dele af et billede skæres væk, uden at de resterende pixelværdier ændres.

Eksempel: Robust over for cropping

Alice gemmer et vandmærke, i form af en kort bitfølge, i et billede. For at sikre vandmærket mod Mallorys forsøg på at fjerne det, gemmer Alice det én gang for hver linie i billedet. Da Mallory får fat i billedet, fjerner hun toppen og bunden af billedet, så Bob ikke kan genfinde vandmærket. Det har Bob dog ingen problemer med, da han endelig får billedet.

❑

Ved kendskab til det oprindelige billede, er det muligt at afkode den bitfølge, der er gemt i et billede, også for andre end dem bitfølgen var beregnet for. Dette betyder dog ikke, at metoden ikke kan benyttes til at camouflere hemmelig information.

Eksempel: Camouflering af krypterede data

Alice ønsker at sende en hemmelig meddelelse til Bob, uden at Mallory skal få fat i meddelelsen. For at skjule over for Mallory, at Alice sender meddelelsen til Bob, kan Alice vælge at benytte et billede, som Bob i forvejen har, til at camouflere informationen i. De data Alice vil sende til Bob

krypteres med en tilstrækkelig sikker krypteringsalgoritme (f.eks. DES[7] eller RSA[8]), og den krypterede bitfølge (kryptogrammet) camoufleres herefter i et stegobillede, der sendes til Bob. Hvis Mallory skulle opsnappe billedet, og få mistanke om, at det indeholder hemmelig information, er det desuden nødvendigt for Mallory at have det oprindelige billede (eller toleranceintervallerne) for at kunne finde den krypterede bitfølge, der så herefter skal dekrypteres. Camouflering i et billede vil altså ikke hindre Mallorys forsøg på at opsnappe beskeden fra Alice til Bob; det vil blot forsinke og besværliggøre processen. Mallory kan dog let ødelægge beskeden fra Alice, så Bob heller ikke kan læse den: ved tilføjelse af støj til billedet, vil det camouflerede kryptogram blive ændret. Ved en afkodning af de camouflerede data, vil det forandrede kryptogram formentlig stadig ligne et kryptogram, men Bob vil ikke kun dekryptere det.

❑

Det er muligt iterativt at beregne toleranceintervaller for et billede, gemme data i billedet, udregne intervaller for det nye billede, osv. Ved afkodning i forhold til det oprindelige billede, vil det dog kun være muligt at udtrække ét vandmærke, der så vil være en kombination af alle beregnede intervaller og indkodninger. For at afkode data i det *n*'te stegobillede, kræves kendskab til det *n*-1'te stegobillede (eller dette billedes toleranceintervaller). Metoden kan derfor ikke sikre, at den der udgiver sig for at være indehaver af ophavsretten til et givent billede også er det. Dette illustreres af følgende eksempel:

Eksempel: Falsk vandmærke

Alice er den oprindelige ejermand af billede *P*, og Mallory ønsker at gøre krav på ophavsretten. For at forhindre denne situation, vælger Alice at indsætte et vandmærke med kombinationsmetoden i *P*, og derved få billedet P', som derefter frigives. Mallory får fat i billedet P', der er frit tilgængeligt, og bestemmer herefter toleranceintervallerne for dette, der i større eller mindre grad vil være overlappende med *P*s. Mallory konstruerer herefter billedet $\widetilde{P}$ ved at vælge tilfældige pixelkomponentværdier indenfor toleranceintervallerne for P'. I billedet $\widetilde{P}$ camouflerer Mallory herefter sit eget vandmærke, hvilket giver billedet $\widetilde{P}'$.

Da alle ændringer af fra *P* til $\widetilde{P}'$ formentlig er holdt i nærheden af (men ikke nødvendigvis indenfor) *P*s perceptuelle tolerancer, skulle det ikke umiddelbart være muligt at skelne P' fra $\widetilde{P}'$, bort-

[7] *Data Encryption Standard*, se f.eks [Schneier 96].

[8] Forkortelse for algoritmens opfindere: *Rivest*, *Shamir* og *Adleman*. Se f.eks. [Schneier 96].

set måske fra små detaljer, der ikke nødvendigvis kan afgøre, hvilket billede, der er en vandmærket kopi af det originale.

Ved en strid om ophavsretten, kan både Alice og Mallory fremvise et billede med vandmærke (P' og $\widetilde{P}'$), og begge kan ligeledes udlede et validt vandmærke (fra henholdsvis (P og P') og ($\widetilde{P}$ og $\widetilde{P}'$)). Metoden kan altså ikke med sikkerhed fastslå, hvem der er i besiddelse af det originale billede.

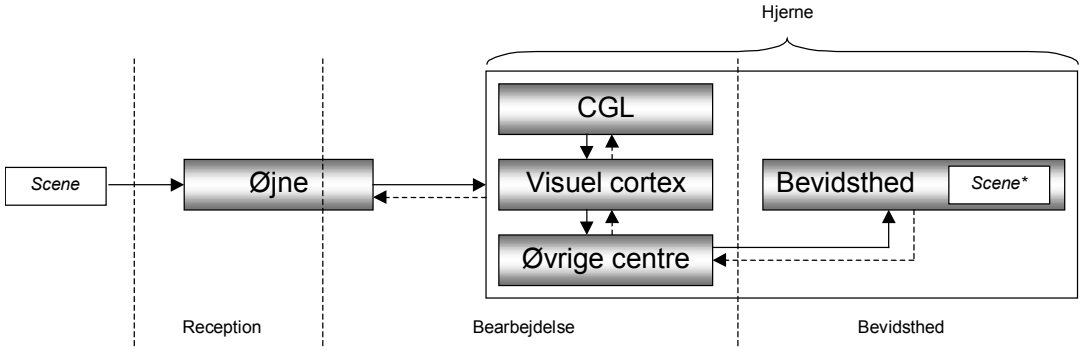
❑

5. Den menneskelige synsproces

Det menneskelige syn (*Human Visual System*, herefter *HVS*) er en forholdsvis kompliceret proces, der stadig kun delvist er forstået i detaljer. Synsindtrykket kan siges at begynde foran øjet, hvor en scene betragtes. Denne scene kan bestå af f.eks. belyste objekter, lyspletter eller billeder på en computerskærm. Scenen betragtes og opfattes af en menneskelig observatør. Herunder beskrives derfor en simpel model af den humane perception, med hovedvægten lagt på HVS. Beskrivelsen af HVS opdeles i en beskrivelse af det der sker i øjet, og det der sker i hjernen. Afsnittet beskriver også en del af de fænomener ved HVS, der gør det muligt at ændre ved den observerede scene, uden at der er en forskel i den opfattede scene. Fænomenerne beskrives som henholdsvis lys- og farvefænomener. Der beskrives de detaljer, der er nødvendige for en bestemmelse af de dele af billeder, som kan ændres i større eller mindre grad, uden at ændringen kan opfattes. Desuden beskrives, hvordan det med forholdsvis simple algoritmer er muligt at bestemme de områder af et billede, der kan være tale om.

5.1 Model af human perception

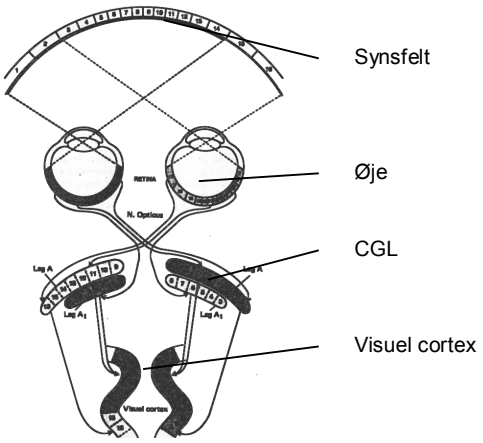
I nedenstående figur ses en simpel model af visuel human perception. Figuren demonstrerer, at en scene, der vises for en menneskelig observatør, gennemgår tre faser hos observatøren: reception, bearbejdelse og bevidsthed.



Figur 19: Model af visuel human perception. Perceptionen er opdelt i reception, bearbejdelse og bevidsthed.

Fra scenen, der altså kan være et billede på en computerskærm, kastes der lys ind i observatørens øjne. Her bliver lyset opfanget i øjnenes fotoreceptorer, der videresender elektriske impulser afhængig af lysets farve og intensitet. Disse impulser bearbejdes derefter, dels i øjnene og dels i dele af hjernen, og ender til sidst i observatørens bevidsthed. I figuren er den scene, der vises for observatøren benævnt *Scene*, mens den scene, som observatøren er bevidst om, er benævnt *Scene**. Dette er gjort for at vise, at der vil være forskel på det lys der kastes ind i øjet, og den scene ob-

servatøren bliver bevidst om. *Scene** vil vi også benævne *den perceptuelle signifikans*[9]. Forskellen mellem *Scene* og *Scene** skyldes den bearbejdelse, der sker af lysindfaldet fra scenen, og som allerede starter når lyset kommer ind i øjet. Den fortsætter i *corpus geniculatum laterale* (herefter CGL, se afsnit 5.2), derefter i *den visuelle cortex* (vist i Figur 20), og dernæst i andre relevante centre i hjernen. At der i modellen er angivet en forbindelse fra cortex tilbage til øjnene beror på, at øjnene stiller skarpt afhængigt af synsindtrykkets behandling i cortex, hvilket sker før observatøren bliver bevidst om det sete.



Figur 20: Skematisk oversigt over synets vej til visuel cortex. Fra [Laursen 76]

Forskellen på *Scene* og *Scene** er ganske afgørende i mange sammenhænge. F.eks. bliver man som erfaren bilist ikke bevidst om alle de faresignaler der reageres på i trafikken; man opfatter måske nok en fare, men bliver aldrig, eller først senere, bevidst om den. I forbindelse med steganografi er forskellen også afgørende, da den i nogle tilfælde betyder, at en ændring på *Scene* ikke medfører en ændring på *Scene**. Hvis *Scene** (S^*) opsættes som en funktion af *Scene* (S):

$$S^* = B(S),$$

hvor B er en funktion, der betegner den samlede bearbejdelse af S, så går steganografi baseret på human perception altså ud på at fremstille en ændret scene $\hat{S}$ så opfattelsen af S og $\hat{S}$ er helt eller næsten helt den samme:

[9] I følge en mailingliste om vandmærkning, gøres der forsøg med andre definitioner af den perceptuelle signifikans for billeder. Her bruges bl.a. signifikante koefficienter ved DCT-transformationer af billederne som en bestemmelse af billedets signifikans. Kan man ikke se forskel på ændringer i en given koefficient, er der ikke ændret på koefficientens perceptuelle signifikans.

$$S^* = B(S) \cong B(\hat{S})$$

I nogle tilfælde vil det være muligt, at kunne ændre et billede (en scene), så det kan opfattes som ændret, uden at observatøren bliver bevidst om dette (jf. den tidligere beskrevne trafiksituation). Om den anvendte steganografi i dette tilfælde er skjult eller blot usynlig, vil så være en definitionssag.

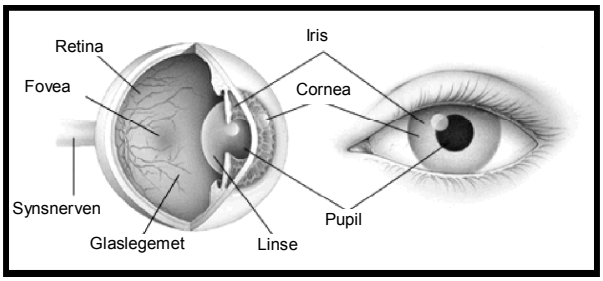
5.2 Synets fysiologi

I dette afsnit beskrives fysiologien i øjet, samt i de første synsbehandlende centre i hjernen. For en mere udførlig beskrivelse henvise til bl.a. [Laursen 76], [Moses&Hart 87] og [Zrenner et al 90].

Når en genstand observeres, er der et stort antal faktorer, der spiller ind på hvad der ses. En del af disse er fysiske faktorer i og omkring den genstand der ses på, mens andre faktorer skyldes HVS selv. For at en genstand kan ses, skal der komme lys fra genstanden til øjet. Genstanden må altså enten selv lyse, eller den skal reflektere en belysning. Lyset fra genstanden kan have en enkelt bølgelængde (monokromatisk), eller det kan bestå af en sammensætning af lysintensiteter ved forskellige bølgelængder, som det f.eks. gælder for solskin. Lysets indfaldsvinkel til genstanden, samt lyset karakteristik (f.eks. indirekte eller direkte) vil også spille ind på synsindtrykket. Ligeledes vil genstandens beskaffenhed, dens farve, overflade og refleksion give et bidrag. Genstandens baggrund og omgivelser spiller også ind, ligesom afstanden til genstanden, og mediet mellem genstanden og beskueren alt sammen vil bidrage til den endelige synsoplevelse.

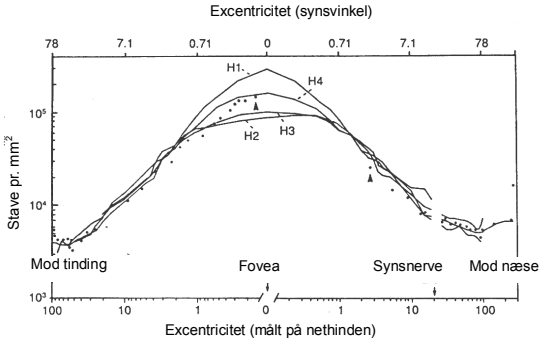
5.2.1 Øjets behandling af synsindtryk

Første del af synsopfattelsen sker i starten af øjet, hvor lyset først skal passere *pupillen* hvis størrelse afhænger af mængden af lys. Herefter passerer lyset øjets *linse* hvis krumning akkomoderer efter afstanden til den genstand der skal observeres, så billedet der projiceres ind på øjets *nethinde* (*retina*) bliver så skarpt som muligt. På grund af linsen er billeddannelsen skarp i afstande af 10-20 cm og opefter.



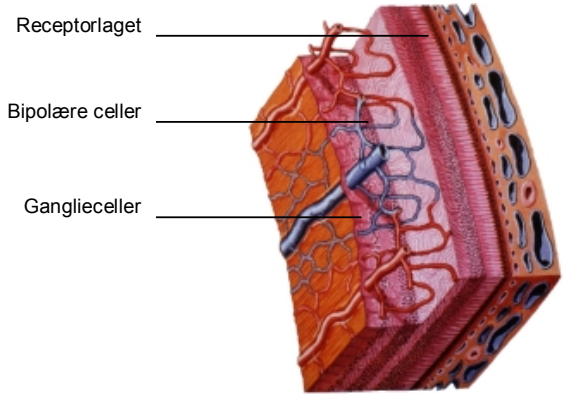
Figur 21: Øjets anatomi (venstre øje)

Lyset opfanges i nethinden vha. *fotoreceptorer*, som opdeles i *stave* og *tappe*. Stavene opfanger lysforskelle, mens tappene er orienteret mod opfattelse af farver. Receptorlaget består af 100 millioner stavceller og fem millioner tapceller, og hverken stave eller tappe er ligeligt fordelt i nethinden; fordelingen af tappe fremgår af Figur 22. I den centrale del af nethinden (*fovea* eller "*den gule plet*") er der ingen stave, og denne del af øjet opfatter således kun farver ("dagssyn"). Fovea er det område i øjet hvor farvesynet er bedst, og det er også her det skarpeste syn forekommer. Opfattelsen af den genstand der observeres, er derfor afhængigt af den afstand der er mellem fovea og genstandens billede på nethinden: Billeder dannet på fovea er skarpe, og det er f.eks. her at billedet dannes når tekst skal læses.



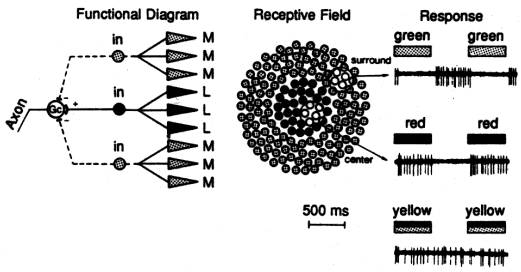
Figur 22: Fordeling af tappe i nethinden (højre øje). Fra [Zrenner et al 90] .

Siden midten af 1800-tallet har forskere formodet, at der fandtes tre forskellige typer af tappe og i [Zrenner et al 90] beskrives en række af de forsøg, der har fundet og klassificeret de tre tap-typer. Tappene giver et signal (*output*) videre i HVS, og de har forskellige output afhængig af type og lysets bølgelængde. Derfor har de fået navne efter den del af bølgespektret, der giver det største output. Der findes således S-, M- og L-tappe, der er følsomme overfor korte (*s*hort), mellem (*m*edium) og lange (*l*ong) bølgelængder. Disse tappe er også fejlagtigt blevet kaldt rød-, grøn- og blåtappe, da tidlige forsøg har vist, at synsindtrykket er afhængigt af disse tre primærfarver.



Figur 23: Nethindens opbygning. Lyset udefra kommer fra venstre i figuren, og går således igennem en del lag i nethinden, før det når receptorlaget.

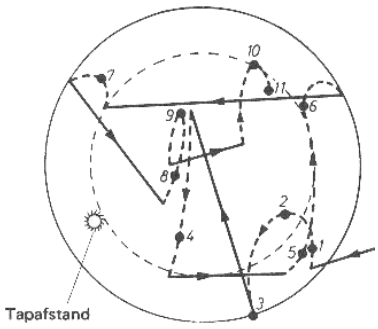
Synsindtrykket afhænger af en mere kompleks sammensætning af signalerne fra S-, L- og M-tappene. Denne sammensætning foregår i *gangliecellerne*, der i nethinden ligger foran fotoreceptorerne. Der er igen forskellige typer af ganglieceller, der kombinerer receptorerne på forskellige måder. Gangliecellernes modtager signaler fra tappe i et cirkulært område, og dette område er delt i en central og en perifer zone (se Figur 24). Den mest hyppige gangliecelle, der findes i øjet, er L^+-M^-cellen (21% af gangliecellerne), der er afhængig af output fra koncentriske cirkler med L- og M-tappe. I centrum af cirklerne er placeret et antal L-tappe, der bidrager positivt til gangliecellens videre signal (L^+), og rundt om disse er en ring med M-tappe, der bidrager negativt til signalet (M^-). I nedenstående figur ses resultatet af at belyse tappene til en given L^+-M^-celle med henholdsvist grønt, rødt og gult lys. Ved grønt lys udligner bidragende fra L- og M- tappene hinanden således, at der ikke kommer et output ved selve belysningen, men kun når der er en pause i det grønne lys. Ved belysning med rødt lys, er der tale om den omvendte effekt: her opstår signal ved belysningen, og ikke ved pauserne. Ved belysning med gult lys, der stimulerer både L- og M-tappe, ses der ingen forskel i signalet ved lys og pause, da signalerne fra L- og M-tappene her netop udligner hinanden fuldstændigt. Nogle ganglieceller svarer på hvidt lys, andre på blåt (og hæmmes af gult), og atter andre på grønt (og hæmmes af rødt).



Figur 24: Sammensætning af modtageområde for en gangliecelle. Fra [Zrenner et al 90]

Input til fotoreceptorerne behandles og reduceres af op til 10 lag af forskellige celler i nethinden. De ca. 100 millioner fotoreceptorers signal reduceres således til output-signal fra ca. 1 million celler. Denne reduktion har mange forskellige formål, som f.eks. at forstærke indtrykket af linier.

Nogle ganglieceller er ikke i stand til at reagere på konstant lys, og reagerer således kun på ændringer i det indfaldne lys. Dette kompenserer øjnene for, ved at bevæge sig, så billedet af et punkt der fikseres på, hele tiden bevæges rundt i den centrale del af fovea (diameter på 0,3 mm eller ca. 20 bueminutter). Dette gøres ved tre forskellige former for øjenbevægelser: *Mikrosaccader*, der forekommer 1-3 gange i sekundet, og som gennemsnitligt flytter det projicerede billede seks bueminutter, *drift*, der ca. sker en gang hvert sekund, og en *fin tremor*, der sker 30-100 gange per sekund, og som flytter billedet ca. to bueminutter. I Figur 25 vises henholdsvis mikrosaccader og drift.

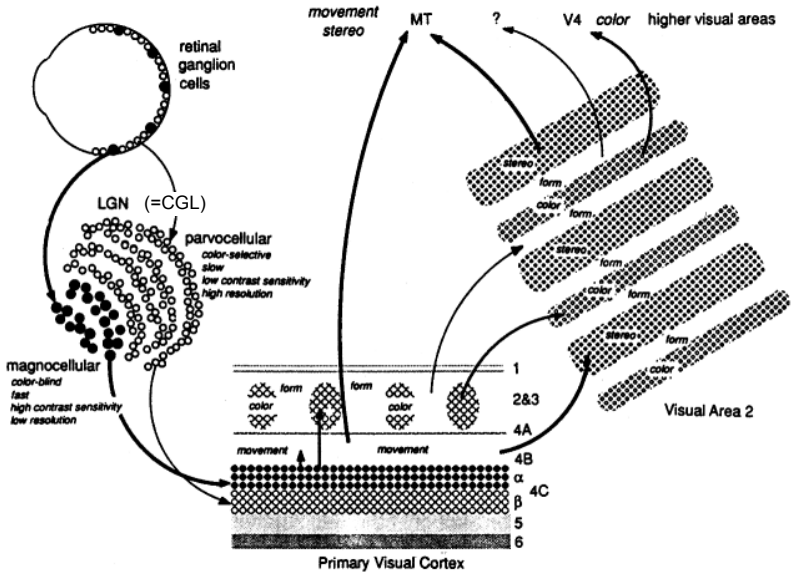


Figur 25: Øjenbevægelser. De stiplede linier viser drift, og de fuldt optrukne viser mikrosaccader. Fra [Laursen 76]

Stationære billeder fikseres normalt mindre end et halvt sekund. Hvis billedet er interessant, fikseres hurtigere, og saccaderne er hyppigere. Saccadiske bevægelser mellem fikseringer er hurtige (op til 500° pr. sekund) og ballistiske (ændrer ikke hastighed og retning). Under de saccadiske bevægelser kan hjernen ikke nå at opfatte synsindtrykkene. At man ikke ser et udvisket billede skyldes, at synsopfattelsen under bevægelsen hæmmes aktivt i øjet, hvilket hjernen så kompenserer for.

5.2.2 Hjernens behandling af synsindtryk

I [Zrenner et al 90] angives det, at visuelle signaler fra øjet til bevidstheden gennemgår tre, mere eller mindre uafhængige, neurale systemer (se Figur 26). Det første system angår opfattelsen af former, det næste opfattelsen af farver og det sidste tager sig af bevægelse, placering og rumlig organisering. Disse tre systemer følger tre separate neurale stier, der alle begynder i selve øjet.



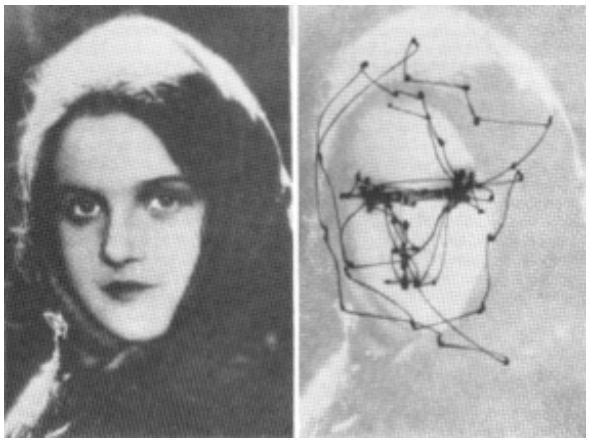
Figur 26: Tre neurale systemer til opfattelse af synet. Fra [Zrenner et al 90]

Gangliecellerne sender signaler til CGL. I dette område eksisterer to slags celler: de *magnocellulære* og de *parvocellulære*. Disse celletyper har vidt forskellige egenskaber og giver input til forskellige dele af HVS. De mangocellulære celler er farveblinde, og giver kun en lille opløsning. Til gengæld er de hurtige og følsomme over for en høj kontrast. De parvocellulære er langsomme celler, og der kan måles en vis forsinkelse fra intensiteten i øjet ændres til cellerne giver udslag. Desuden er de følsomme for lav kontrast og farver, og har en høj opløsning.

Fra CGL ledes et signal videre til *den primære visuelle cortex*, der befinder sig bagest i hjernen. Området består af seks delområder, der optager signaler fra de magnocellulære og parvocellulære områder, behandler dem, og sender dem videre i HVS. Synet behandles derefter samlet af forskellige centre i resten af hjernen, hvor sammenhængen mellem disse endnu ikke i alle tilfælde er kortlagt.

Efter HVS' første behandlinger af et billede, kommer billedopfattelsen, der er en særdeles subjektiv proces. For nogle vil et givent billede være forbundet med følelser, som også vil indgå i billedopfattelsen, mens samme billede for andre kan være følelsesmæssigt neutralt. At fremstille en automatisk analyse, der ud fra et billede kan analysere sig frem til, hvordan billedet vil blive opfattet hos en specifik beskuer, er derfor ikke en simpel opgave. Der er dog en del egenskaber ved dele af billeder, der gør at disse dele vil blive opfattet på bestemte måder. Dette kan f.eks. benyttes til at karakterisere dele af et billede, som er mere følsomme over for ændringer, set i relation til den menneskelige opfattelse af billedet, end andre dele er.

I nedenstående figur er vist, hvordan en observatør har set på et billede af et ansigt. I et minut har observatøren set på billedet, mens dennes øjenbevægelser er blevet registreret. Som det tydeligt fremgår af figuren, er der store dele af billedet, som ikke bliver fikseret på observatørens nethinde.



Figur 27: Øjets fikseringer i løbet af et minuts betragtning af et billede. Fra [Moses&Hart 87].

5.3 Lysfænomener

I dette afsnit behandles opfattelsen af lys som værende uafhængigt af lysets bølgelængde. Det betyder, at de lysfænomener der beskrives, kan opfattes som kun værende afhængige af lysets intensitet, og altså uafhængige af lysets farve. Dette stemmer, jf. [Fiorentini et al. 90] og [Zrenner et al. 90], overens med en del observationer der tyder på, at farve- og lysfænomener har separate neurale stier i HVS (jf. afsnit 5.2.2). Nogle af disse stier har karakteristika der betyder, at tidslige ændringer af de billeder der observeres, som det sker med film, ville give anledning til yderligere lys- og farvefænomener end de her beskrevne. Camouflering af data ud fra disse fænomener kunne derfor være en nærliggende mulighed, som vi dog har valgt ikke at beskæftige os yderligere med.

I behandlingen af lysfænomener er der en del begreber, som beskriver mængden af lys, der rammer øjet. Herunder følger en kort forklaring på de mest væsentlige af disse begreber. Da disse udtryk ikke alle har en naturlig oversættelse til dansk, benyttes for nogle den engelske betegnelse.

Luminans (lysstyrke)	Den mængde af lys, der udsendes fra et område eller en genstand. Dette er en fysisk målbar størrelse, også kaldet lysflux, som er uafhængig af HVS.
Reflektans	Et mål for, hvor meget af det lys, der falder på et objekt, som reflekteres fra objektet. Reflektansen er høj for et spejl og lav for et stykke sort pap. Denne størrelse er også uafhængig af HVS.
Brightness (lyshed)	Et subjektivt mål for, hvor lys en overflade opfattes. Dette hænger delvist sammen med luminansen, men er altså stærkt afhængig af HVS.
Darkness (dunkelhed)	Det omvendte af *brightness*. At et område med høj *darkness* ikke blot beskrives som et område med lav *brightness* skyldes, at disse begreber formentlig opfattes af forskellige dele af HVS.

Enhver belyst genstand, der er i stand til at reflektere lys, vil i mørke omgivelser blive opfattet som værende lys. Dette gælder uanset om genstanden er et stykke kridt eller et stykke trækul. Hvis både kridt og kul derimod er at finde i de mørke omgivelser samtidig, vil kridtet blive betegnet som lyst, og kullet som mørkt. Denne effekt, som også kaldes Gelb-effekten (1929), hænger sammen med genstandenes brightness, og den er et eksempel på et kontrastfænomen. I de følgende afsnit behandles andre kontrastfænomener.

Ved billedbehandling er det ligetil at bestemme lyskomponenten for et givent område, udfra en af de tidligere nævnte farvemodeller (afsnit 3.2), og det er derfor muligt at behandle en del af de her beskrevne lysfænomener. Der vil dog ligge en del begrænsninger i forhold til den digitale billedbehandling. F.eks. kan en skærm kun udsender lys med en stærkt begrænset intensitet. Ligeledes vil et billede på udskrift også kun kunne udsende lys afhængigt af det indfaldne lys, hvorfor intensiteten f.eks. ikke kan blive større end en vis brøkdel af intensiteten for det indfaldne lys. I det følgende ses der bort fra disse begrænsninger i billedbehandlingen, og lysintensiteten behandles som en størrelse, der er uafhængig af skærm/papir.

5.3.1 Simultankontrast

I omgivelser med dagslys eller normalt kunstigt lys ("læselys"), vil et objekt synes lyst eller mørkt, afhængigt af den mængde af lys, der kommer fra objektet, enten ved selvstråling eller ved reflektering. Et objekt med en moderat reflektans (dvs. hverken lyst eller mørkt) vil dog opfattes lysere eller mørkere, afhængig af objektets omgivelser. Hvis objektet således har lyse omgivelser, vil det

blive opfattet mørkere, og hvis det har mørke omgivelser, vil det blive opfattet lysere. Dette fænomen kaldes *simultankontrast*, og et eksempel på dette kan ses i Figur 28. I denne figur ses også et andet fænomen, nemlig *områdekontrast*[10] hvor kontrasteffekten ikke kun holder sig til grænseområdet mellem objektet og omgivelserne, men breder sig til hele objektets areal. Således er det ikke kun kanterne på den inderste kasse, der virker lysere eller mørkere afhængig af omgivelserne, men hele kassens område. (I kasserne til venstre i figuren kan det se ud som om, der er en lys kant om den inderste kasse. Hvis dette er tilfældet, kan det skyldes udskriftens kvalitet.)

Figur 28: Simultankontrast. De fire inderste kasser har alle samme gråtone.

En anvendelse af kombinationen af simultan- og områdekontrast til camouflering af data i et billede kunne være, at ændre intensiteten for et område, der grundet de to kontrastfænomener allerede ville blive opfattet som værende af en anden intensitet. Således kunne et billedområde, der er omgivet af en mørkere baggrund, og som derfor vil blive opfattet lysere end det rent faktisk er, kunne sættes lysere i det ændrede billede. Ved sammenligning med det oprindelige billede, vil den ændrede intensitet kunne detekteres, og ændringen kunne benyttes til at camouflere data i billedet. De to situationer, hvor området enten beholder sin intensitet (og gøres lysere af HVS) eller gøres lysere på billedet (og dermed giver en effekt, der minder om HVS'), gør at denne form for ændring kunne camouflere netop én bit. Hvis der beregnes flere mulige intensiteter, kunne ændringen camouflere flere bit.

5.3.2 Assimilering

I [Fiorentini et al. 90] beskrives et fænomen, der kan virke paradoksalt i forhold til simultankontrasten. Fænomenet gør, at øjet vil have en tendens til at mindske kontrasten i nogle områder. Dette kan f.eks. ses i Figur 29, hvor de grå striber kan opfattes mørkere i venstre del af billedet, da baggrunden her er sort, end i højre del, hvor baggrunden er lys. Dette fænomen kaldes *assimilering*, og det optræder fortrinsvis i forbindelse med mønstre, hvori der indgår tynde linier. Hvis illusionen i figuren ikke træder tydeligt frem, kan det skyldes, at linierne ikke er tynde nok. Hvis afstanden til figuren øges, skulle effekten derfor træde tydeligere frem.

[10] Fra engelsk: *area contrast* (jf. [Fiorentini et al 90]).

Figur 29: Assimilering. De tynde linier kan virke mørkere i venstre side end i højre side.

At assimilering kun forekommer, hvor der er tynde linier, betyder at områder, hvor dette fænomen kan detekteres i et billede, ved en transformation til frekvensdomænet vil være præget af relativt høje koefficienter for de høje frekvenser. Da det, der ses ved assimilering, er en tilnærmelse af baggrund og forgrund, er dette det samme som en mindskelse af koefficienter for de høje frekvenser. I digitale billeder kan dette opnås på flere måder, enten ved brug af spatielle lavpasfiltre (som medianfiltret) eller ved brug af filtrering direkte i frekvensrummet (som i JPEG-algoritmen).

5.3.3 Kantkontrast

Til forskel for de tidligere beskrevne kontrastfænomener, er der også fænomener, der kun kan observeres i nærheden af overgange mellem områder med relativ stor forskel i intensitet. Fænomenerne gør, at den brightness, der opleves, afviger væsentligt fra den aktuelle luminans. Disse fænomener, som vi vælger at betegne kantkontrastfænomener, er lokale, og effekten af fænomenerne aftager altså med afstanden til kanten.

Et eksempel på et kantkontrast-fænomen er *Mach-bånd*[11]. Mach-bånd er lyse eller mørke striber, der optræder ved bløde overgange mellem lyse og mørke områder (se Figur 30(a)). Striberne fremhæver overgangen mellem det lyse og det mørke område, som rigtige striber ville have gjort det, og de kan forklares med HVS' evne til at forstærke kanter. Fænomenet optræder kun, hvis overgangen mellem områderne hverken er for brat eller for lang. Dette svarer til, at hældningen ved overgangen i Figur 30(b) skal ligge inden for et bestemt interval, før fænomenet optræder. I figuren viser den ubrudte linie den egentlige intensitetsvariation, mens den stiplede linie angiver den opfattede variation.

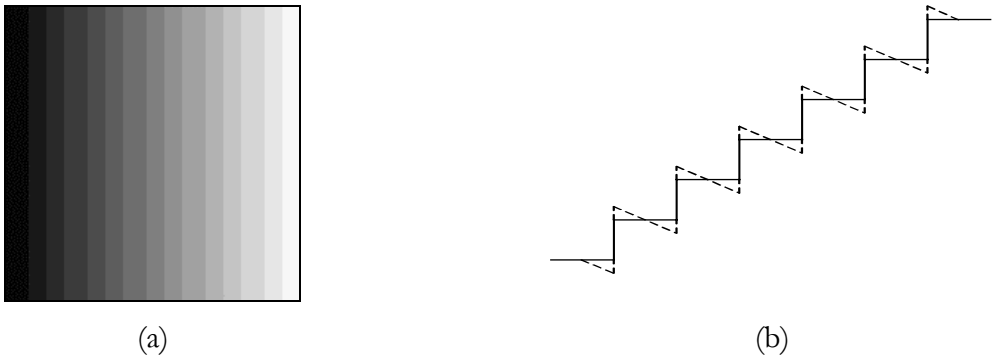
[11] Mach-bånd er først beskrevet af den østrigske fysiker og filosof Ernst Mach i 1865.

(a) (b)

Figur 30: Mach-bånd. Den ubrudte line vise intensitetsvariation, og den stiplede linie viser den opfattede variation.

Camouflering af data ved Mach-bånd kan, som det var tilfældet med simultankontrasten, gøres ved at ændre intensiteten af et mindre område til den intensitet (brightness) som opfattes af HSV. Ved Mach-bånd vil dette altså kræve, at de lyse og mørke områder kan segmenteres til en vis størrelse, og ved hjælp af en analyse af stykket mellem områderne kan det afgøres, om den aktuelle del af billedet kan benyttes til camouflering af data. Denne analyse ville kræve en undersøgelse af, om overgangen er monoton, forskellen i intensiteten af det lyse og det mørke område, samt om hældningen ved overgangen ligger inden for det tilladte interval. Endelig skal de smalle områder, hvor fænomenet skulle observeres, kunne bestemmes, og intensiteten i områderne kunne så ændres afhængigt af de data der skal gemmes.

Figur 28 viser, hvordan områdekontrast-effekten virker, hvis hele området grundet omgivelserne kan opfattes lysere eller mørkere. Hvis ikke hele området er omkranset af den gråtone, der skaber områdekontrasten, kan der optræde andre effekter. Et eksempel herpå er Chevruel-effekten (se Figur 31(a)), hvor et antal striber med forskellige intensiteter, er placeret ved siden af hinanden. Intensiteten varierer i en trappefunktion (se Figur 31(b)), men på grund af kantkontrasteffekten, der gør, at lyse områder synes endnu lysere, og mørke områder synes endnu mørkere, hvor disse støder op til hinanden, vil intensitetvariationen synes savtand-formet. Forskellen mellem dette fænomen og Mach-bånd ligger bl.a. i, at der her ikke er tale om meget tynde striber, der vil opfattes som havende en anden brightness end den luminansen angiver. Desuden vil fænomenet optræde, hvor der ikke er samme strenge krav til overgangen mellem lyse og mørke områder, som Mach-båndene stiller. Dette gør blot effekten endnu nemmere at bestemme, og den kan, som tidligere nævnt, benyttes til camouflering ved at justere intensiteten af et område efter, hvad HVS opfatter.



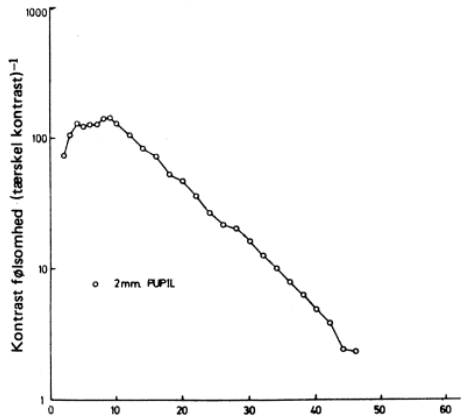
Figur 31: Chevruels effekt. Den ubrudte line vise intensitetsvariation, og den stiplede linie viser den opfattede variation.

Generelt har HVS, som tidligere nævnt, en vis evne til at forstærke opfattelsen af kanter. Dette kan i sig selv benyttes til camouflering, da detektion af kantpixels kan være en forholdsvis simpel operation (f.eks. ved brug af Prewitt- eller Sobel-filtre, se afsnit 6.3). De detekterede kantpixels kan formodentlig ændres en del, uden at dette kan ses, da HVS vil fremhæve kanten uanset om denne er ændret eller ej.

5.3.4 Spatiel frekvens

For at synet skal kunne skelne mellem to punkter, skal der være en vis minimal afstand mellem punkterne, da de ellers vil blive opfattet som samme punkt. Denne minimale afstand er afhængig af flere faktorer i synet, hvor den vigtigste er størrelsen af de enkelte fotoreceptorer. For at kunne skelne mellem to punkter, skal de altså projiceres ind på to forskellige fotoreceptorer, og hvis punkterne er "ens" kræves der ydermere mindst en fotoreceptor i mellem de to førnævnte, for at kunne skelne, at der er tale om to punkter, og ikke ét. Hvis øjet var fikseret i forhold til de to punkter, ville afstanden mellem to diskrete punkter, der skulle kunne opfattes af HVS, være direkte afhængig af afstanden mellem fotoreceptorer. Da øjet hele tiden bevæger sig i større eller mindre ryk (som beskrevet i afsnit 5.2.1) kan afstanden mellem punkterne dog være mindre end afstanden mellem fotoreceptorer umiddelbart angiver, uden at punkterne flyder sammen som ét.

I [Laursen 76] beskrives, hvordan HVS' evne til at opfatte forskelle i kontrast i et billede, hvor intensiteten beskrives ved en sinusfunktion, er aftagende når funktionens frekvens stige, som vist i Figur 32. Dette kan benyttes til camouflering i digitale billeder, der indeholder cykliske intensitetsvariationer, hvor frekvensen af variationerne gør, at der kan ændres i enkelte pixels intensitet, og dermed i kontrasten til de omkringliggende pixels, uden at dette kan ses.

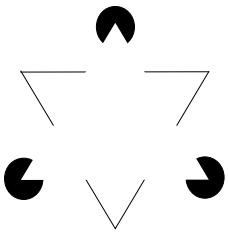


Figur 32: En forsøgspersons kontrastfølsomhed som funktion af den spatielle frekvens. Tilrettet fra [Laursen 76].

På en 17" computerskærm med en opløsning på 1024×768 pixels vil hver pixel have en sidelængde på ca. 0,3 mm. Med en afstand fra observatør til skærmen på 50 cm, og en betingelse om, at en periode for et cyklisk mønster kræver mindst to pixels, vil der højst kunne være ca. 13 perioder per grad. Som det ses af ovenstående figur giver denne opsætning en spatielle frekvens, hvor HVS er mest følsomt. At kunne udnytte den spatielle frekvens til camouflering på denne måde vil altså formentlig kræve en noget højere spatial frekvens, der enten kan opnås ved større afstand til skærmen eller ved en højere opløsning på skærmen. Som følge heraf har vi valgt ikke at se mere på de muligheder, camouflering ved brug af spatial frekvens giver.

5.3.5 Synsbedrag

Al camouflering ud fra de lysfænomener, der hidtil er blevet beskrevet, bygger på, at HVS bearbejder indtrykket af en scene, så den opfattes anderledes end den rent faktisk er. Der findes også mere avancerede former for synsbedrag, som kan benyttes til camouflering af data i et billede. Desværre ville det ofte kræve megen analyse af et billede for bare at kunne bestemme enkelte af de situationer, hvor der muligvis vil kunne opleves synsbedrag.



Figur 33: Kanizsas trekant. Det kan se ud som, der ligger en hvid trekant ovenpå tre udfyldte cirkler.

I Figur 33 ses en figur, der ofte benævnes *Kanizsas trekant*. På grund af hakkene i de tre sorte figurer, kan det her se ud som om der ligger en hvid trekant oven på tre sorte udfyldte cirkler og en anden trekant. At den hvide trekant kan skille sig ud betyder, at det kan se kan ud som den har en

anden intensitet end baggrunden. Ved dette fænomen har HVS altså en tendens til at sætte liniestykker ind i et billede (som afskærer den hvide trekant fra omgivelserne), der ikke er på figuren selv. Selvom det ikke nogen triviel opgave at definere algoritmer, der kan genkende denne form for synsbedrag, så kan en Hough-transformation (se bl.a. [Gonzales&Woods 93]), der kan finde og færdiggøre liniestykker, måske være brugbar. Hvis ovenstående fænomen optrådte i et naturligt billede, hvori der var mere støj end i Figur 33, kunne en ændring af de pixels, der af en Hough-transformation ville blive genkendt som manglende kantpixels, og som derved ville afmærke den hvide trekant, formentlig kunne blive brugt til camouflering af data, i henhold til den camouflering som kantpixels ellers kan benyttes til.

I Figur 34 ses endnu et eksempel på en figur, der i forbindelse med synsbedrag er forholdsvist kendt. Figuren viser to *Hermann-gitre*, der består af et antal sorte kvadrater på en hvid baggrund. Effekten består i, at mønstret får HVS til at opfatte gråtonede områder ved de sorte kvadraters hjørnepunkter (Figur 34(a)). Dette kan benyttes til camouflering (hvis placering af de grå områder kan bestemmes) ved at gøre de områder, der i forvejen opfattes som grå af HVS, til grå på billedet. Dette er gjort, forholdsvist tydeligt, i Figur 34(b), hvor hjørneområder er 10% sorte. Hvis de omvendte gitre (hvide kvadrater på sort baggrund) benyttes, fås lignende effekter.

(a) (b)

Figur 34: Hermann-gitre. I (b) er hjørneområderne 10% grå; det er de ikke i (a).

5.4 Farvefænomener

Herunder beskrives nogle af de fænomener ved farvesynet, der kan benyttes i forbindelse med camouflering af data. De fleste metoder til camouflering kræver en undersøgelse af det digitale billede, som der skal gemmes data i. Hvis det var muligt at undersøge, hvor på et billede en observatør vil fastholde blikket, kunne der sandsynligvis ændres endnu mere ved billedet, end de her nævnte metoder angiver. F.eks. vil hovedparten af farvegenkendelsen opstå i fovea i øjet, og de dele af et billede, der ikke afbilledes herpå, kan derfor have større farveændringer.

Som beskrevet i afsnit 5.2.1 er forholdet mellem de lysfølsomme stave og de farvefølsomme tappe i øjet ca. 20:1. Da lysændringer også opfattes før farveændring, pga. tidsforskellen i de magnocellulære og parvocellulære celler i CGL, er det nærliggende at antage, at hvis ændringen i et område kun omhandler farverne, og ikke ændrer intensiteten, vil denne ændring ikke kunne opfattes lige så let som hvis der også var sket en ændring i intensiteten.

5.4.1 Spatiel frekvens

I følge [Zrenner et al. 90] vil detaljer på et billede, der pga. afstanden til øjnene kommer indenfor en minimal synsvinkel, og derfor ikke kan opfattes af det parvocellulære system, blive assimileret til en fælles farve i området. Dette er en effekt, der er blevet brugt indenfor pointilismen, hvor kunstmalere har fremstillet billeder bestående af pletter i klare farve, som fra en given synsafstand virker som en blandingsfarve. Et andet eksempel på denne effekt ses i fjernsyn og computerskærme, hvor farverne grøn, blå og rød observeres i tilpas stor afstand til, at de opfattes som en blandingsfarve. Dette kan naturligvis benyttes til camouflering, hvor detaljer er så små, at de kan ændres, uden at HVS vil kunne se forskellen. For at kunne udnytte dette forhold kræves derfor en kombination af en høj opløsning og en given afstand til observatøren. Som med den spatielle frekvens for lysfænomener, er dette derfor ikke en mulighed, vi vil arbejde videre med.

5.4.2 Kanter

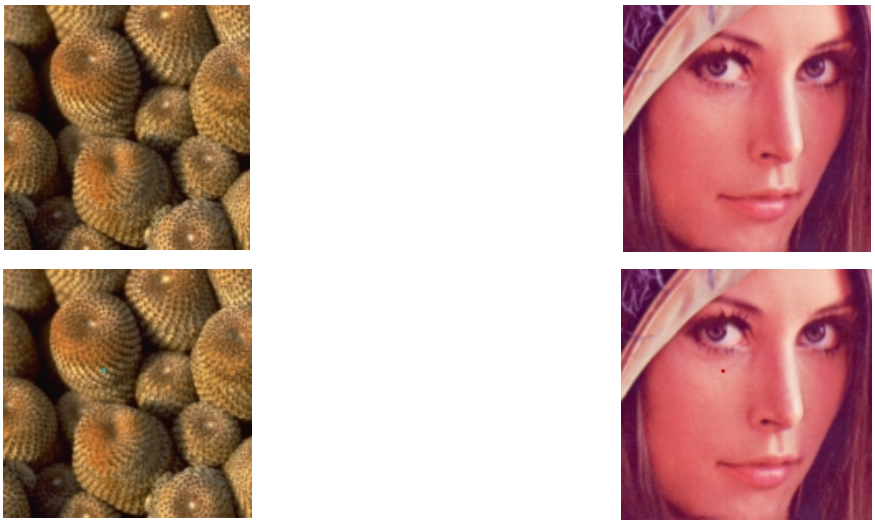
At kanter mellem farver er vigtige for farveopfattelsen ses bl.a. af følgende forsøg, der er beskrevet i [Gilchrist]: På Bell Laboratories benyttedes en forsøgsopstilling, hvor en skive af en given farve (f.eks. grøn) blev sat på en større skive af en anden farve (f.eks. rød). Skiverne blev sat til at vibrere således, at overgangen mellem skiverne fulgte de saccadiske bevægelser af øjnene på en forsøgsperson. Resultatet af dette blev, at forsøgspersonen ikke kunne se overgangen mellem de farvede områder, og da den yderste skives farve kunne opfattes som rød (da skivernes baggrund ikke vibrerede med) blev hele opstillingen, og dermed også den grønne skive, opfattet som rød.

I forbindelse med opfattelse af farver er HVS altså, som tilfældet var med opfattelsen af lys, stærkt afhængig af opfattelsen af kanter. Da forskellen i opfattelse af lys og farver i stor grad optræder i synsbearbejdelsen efter øjnene, er det nærliggende at antage, at lagene i nethinden også forstærker farvekanter. Som med intensitets-kantpixels, kan man altså formentlig ændre en del i farve-kantpixels, da der sker bearbejdelse af opfattelsen af disse allerede i øjnene.

5.4.3 Farveforskel

Som en naturlig følge af, at fotoreceptorerne i nethinden ikke har en jævn spektral fordeling mht. bølgelængder indenfor det synlige spektrum, kan menneskesynet ikke opfatte alle farver lige godt. Dette er et faktum, der bl.a. er blevet benyttet til YIQ-farvemodellen (se afsnit 3.2.7). Modellen benytter, som HVS, den største del af båndbredden til at overføre den intensitet en pixel har. Hvor en mere simpel HVS-baseret farvemodel som HSI (se afsnit 3.2.6) beregner intensiteten som en ligelig fordeling af farverne rød, grøn og blå, benytter YIQ, at øjet ikke opfatter blå ligeså godt som rød og grøn. Derfor beregnes intensiteten (Y) her med en mindre del af blå (11,4%) end med rød (29,9%) og grøn (58,7%). Ved en digital repræsentation af et billede er RGB-modellen ofte den grundliggende anvendte farvemodel, hvilket giver, at der benyttes lige mange bit til alle tre farvekomponenter. Det er derfor nærliggende, at der kan camoufleres mere information i den blå farvekomponent, da den kan ændres mere end grøn og rød kan, uden at HVS kan skelne en forskel. Dette er bl.a. benyttet til camouflering af data i [Kutter et al].

Et andet sted, hvor en pixels farve sandsynligvis kan ændres i større grad, uden at dette vil have en perceptuel ændring, er i områder, hvor der findes pixels af mange forskellige farver. HVS vil her have en tendens til at assimilere farverne til den samme farve, og en pixels forskel fra denne farve ses mindre tydeligt. Således vil en pixel i et hudområde på kort afstand skille sig meget ud, hvis den antog en rød farve, der var meget mørkere end omgivelserne, hvor en pixel, der ligger i et blomsterbed, set på lang afstand, sikkert kunne antage enhver farve, uden at dette ville bemærkes. Et eksempel på dette ses i Figur 35, hvor to billeder er blevet ændret ved at indsætte et område på 2×2 pixels med en anden farve end omgivelserne. I det første billede er blevet indsat et område på 2×2 pixels med en hue, der er ca. 180° fra de oprindelige pixels. Dette område er blevet indsat i et større område, hvor der er stor variation i pixels. I det andet billede er indsat 2×2 pixels med næsten samme hue som de oprindelige pixels, men med en ændring i intensiteten. Dette område er indsat i et større område, hvor der er meget lidt variation i pixels.

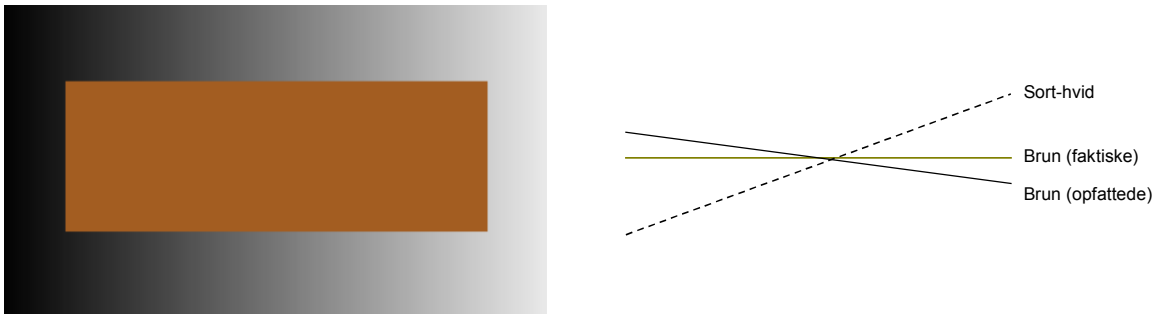


(a) (b)

Figur 35: Billeder med ændrede pixels i henholdsvis (a) varieret område og (b) ikke-varieret område.

5.4.4 Kontrastfarver

Hvis HVS observerer en genstand, som udsender eller reflekterer lys med en bølgelængde omkring 580 nm, og genstanden har en hvid baggrund, vil HVS opfatte genstanden som værende brun (se f.eks. [Zrenner et al. 90]). Hvis baggrundens intensitet nu ændres, vil opfattelsen af genstandens farve også ændres således, at alle farver fra gul over beige til mørk brun kan opfattes. Dette fænomen gør, at brun tilhører de såkaldte *kontrastfarver*, altså farver, der kun kan opfattes i kraft af den kontrast de optræder i. Figur 36(a) viser et brunt område omgivet af et område, der varierer fra sort til hvid. I Figur 36(b) ses variationen af intensiteten af den brune klods, af den opfattede intensitet (luminans) af den brune klods, samt af det sort-til-hvide område. Hvis denne figur måske ikke med stor tydelighed viser, at der skulle være forskel i opfattelsen af den brune farve fra den venstre del af figuren til den højre, kan det bl.a. skyldes, at figuren er relativt lille, og at forskellen i intensiteten fra den sorte del til den hvide del ikke er særlig stor, da begge dele er ses i samme belysning.



(a) (b)

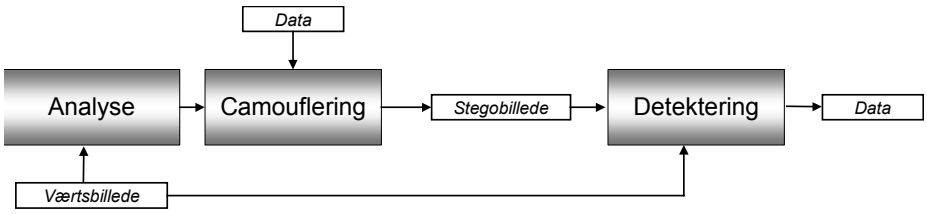
Figur 36: Kontrastfarve. Det brune område har samme intensitet i hele område, men opfattes som mørkest til højre.

At bestemme en kontrastfarve som brun er forholdsvis nemt at gøre. Den brune farve, der er anvendt i figuren, har RGB-sammensætningen (163, 93, 33) i intervallet [0, 255], og farver med en lignende sammensætning kan altså betragtes som brune

6. Bestemmelse af tolerance og kapacitet

I afsnit 5 beskrives en række fænomener, der af forskellige årsager gør dele af et billede egnet til camouflering af information. I dette afsnit beskrives, hvordan vi med digital billedbehandling kan finde frem til nogle af de føromtalte områder. Dette gøres ved hjælp af en række undersøgelser (filtre), hvis formål er at bestemme, hvor et billede kan ændres efter forudbestemte værdier. Formålet med dette er at kunne ændre et billede mest muligt, uden at ændringerne kan ses. Brugen af filtre adskiller sig her fra, hvad vi opfatter som den gængse brug. Et filter vil ofte være designet til at have en synlig effekt på et givent billede, og i nogle tilfælde er den største effekt at foretrække. I vort tilfælde gælder det altså om at finde den filtersammensætning, der ændrer billedet mest muligt, uden at ændringerne kan ses.

Figur 37 beskriver den metode vi har anvendt til camouflering af data i et billede. I analyse- og detekteringsfaserne underkastes værtsbilledet en række filtre, for bestemmelse af billedets perceptuelle tolerance i alle pixelkomponenter. Såfremt den samme filterbank, og indstillingen heraf, benyttes i både analysen og ved detekteringen, vil den perceptuelle tolerance være ens disse steder. Indtillingen af filtrene i metoderne er afhængig af resultaterne fra en række personforsøg (se afsnit 7), og ud fra disse har vi behandlet et antal billeder. I afsnit 6.6 vises de fundne kapacitetsmål for disse billeder.



Figur 37: Anvendt metode til camouflering og detektering af data.

6.1 Tolerancetildeling

For at bestemme den perceptuelle tolerance af et billede, har vi valgt at begrænse denne til et sammenhængende interval af værdier for hver pixelkomponent. Således vil vi ikke beskæftige os med tolerancer, hvor en pixelkomponent kan antage værdier i forskellige adskilte intervaller (f.eks. kun sort og hvid på en gråtoneskala). For at finde toleranceintervallerne for hver enkelt komponent i alle pixels, går vi ud fra den oprindelige værdi, og udvider derefter et interval omkring denne værdi, hvis undersøgelse af pixelens omgivelser viser, at disse udvidelser ikke overskrider den perceptuelle tolerance.

Ved hvert filter undersøges en pixel og dennes 8-naboer. At der i de fleste af filtrene kun undersøges de allernærmeste naboer i en 3×3-matrix (herefter refereret som N_8, se Figur 38) skyldes udelukkende et valg om simpelhed: For det første er det nemt at definere filtre på denne måde, og for det andet understøtter Java 2 direkte implementering af filtre vha. 3×3-matricer. Det er derfor tænkeligt, at de fundne toleranceintervaller kunne udvides yderligere ved et valg af større undersøgelsesområder. Ved en undersøgelse af de ni pixels, er det pixelen i midten (p_5) hvis toleranceværdier ændres.

	p_2	
p_4	p_5	p_6
	p_8	

(a)

p_1	p_2	p_3
p_4	p_5	p_6
p_7	p_8	p_9

(b)

Figur 38: N_4 og N_8-nabomatricerne.

6.1.1 Anvendt metode til tolerancetildeling

Lad P være mængden af billeder med R rækker og S søjler af pixel, hvor hver pixels værdi er beskrevet ved en given farvemodel (f.eks. HSV) og med et givent antal bit (f.eks. 8 bit pr. farve).

En pixels placering i et billede $V \in \mathsf{P}$ er beskrevet ved række og søjle. $p_V(r,s)$ er således en pixel fra billedet V, placeret i række r, $0 \le r < R$, og søjle s, $0 \le s < S$.

Lad desuden $X = \min(Y,Z)$, $X,Y,Z \in \mathsf{P}$, være den funktion, der for alle pixelkomponenter (f.eks. H, S og V) giver en pixel i X, $p_X(r,s)$, de mindste værdier fra tilsvarende pixels i Y og Z, $p_Y(r,s)$ og $p_Z(r,s)$.

Med ovenstående definitioner består en undersøgelse af følgende tre trin:

1) Ud fra det originale billede $P \in \mathsf{P}$ initieres pixelværdierne i billederne $\underline{P}, \overline{P} \in \mathsf{P}$ med $\underline{P} = \overline{P} = P$.

2) Givet det originale billede P og et filter $F \in \mathsf{F}$ (hvor $\mathsf{F} = \{F | F : \mathsf{P} \rightarrow \mathsf{P}\}$), genereres et billede $P' = F(P)$, $P' \in \mathsf{P}$, der ikke er perceptuelt forandret fra P. Det er her vigtigt, at P' ikke blot er perceptuelt uændret fra P; ændringen skal også være foretaget således, at alle billeder med mulige pixelkomponentværdier imellem P og P' heller ikke vil være perceptuelt ændret fra P.
3) Med P og P' opdateres pixelværdierne i billederne $\underline{P}$ og $\overline{P}$. Dette gøres ved sammenligning af værdierne for hver pixelkomponent i alle pixels i henholdsvis $P', \underline{P}$ og $\overline{P}$: $\underline{P} = \min(\underline{P}, P')$, og $\overline{P} = \max(\overline{P}, P')$.

Punkterne 2) og 3) gentages med følgende filtre: Just Noticeable Distortion, Kantdetektion, Homogenitet (i hue og value) og Assimilering (middelværdi og vægtet), der alle kan resultere i en udvidelse af intervallet. De seks filtre er forklaret i kommende afsnit.

Eksempel: Udvidelse af toleranceintervaller

I Tabel 5 og Tabel 6 ses de værdier, der behandles ved en eventuel udvidelse af toleranceintervallerne for pixelkomponenterne rød, grøn og blå i en given pixel. Tabel 5 viser de værdier pixelkomponenterne har på en given position i billederne $P, \underline{P}$ og $\overline{P}$, og Tabel 6 giver de værdier $P', \underline{P}$ og $\overline{P}$ vil have efter et gennemløb af ovenstående algoritme.

	Pixel i P:	**Pixel i $\underline{P}$:**	**Pixel i $\overline{P}$:**	***I*:**
Rød	78	78	78	1
Grøn	114	114	114	1
Blå	124	124	124	1

Tabel 5: En pixels initialværdier før tolerancetildeling

	Pixel i P:	**Pixel i $\underline{P}$:**	**Pixel i $\overline{P}$:**	***I*:**
Rød	75	75	78	4
Grøn	122	114	122	9
Blå	120	120	124	5

Tabel 6: Værdier efter tolerancetildeling

❑

6.1.2 Anvendte filformater

En væsentlig betingelse for succesfuld camouflering af data i et stegobillede er, at det ikke adskiller sig fra et "almindeligt" billede. Derfor har vi valgt, at stegobilledet, såvel som værtsbilledet, skal kunne gemmes i et af de gængse billedformater, der eksisterer til kommerciel brug. Der er store forskelle på, hvordan billedformater behandler billedets indhold, og da det er væsentligt for især afkodningen af camoufleret data, at der ikke er ændret i de pixelkomponentværdier, der er blevet bestemt under indkodningen, har vi måttet vælge, hvilke billedformater, der behandles. Filformatet for værtsbilledet kan vælges bredere end filformatet for stegobilledet, da værtsbilledet blot skal kunne læses (med samme resultat) ved camoufleringen og den efterfølgende detektering. Som filformat for værtsbilledet har vi derfor valgt et af følgende: TIFF, BMP, PNM eller JPEG. Filformatet for stegobilledet har vi valgt således, at det er muligt relativt simpelt at gemme billedet uden tab af pixelværdier. Til at gemme stegobilledet, har vi derfor valgt at benytte enten BMP eller PNM.

6.1.3 Anvendte farvemodeller

Som beskrevet i afsnit 3.2, er det ikke alle farvemodeller, der er lige velegnede til beskrivelse af synsfænomener. RGB er en hyppigt anvendt model, da et billede her beskrives ved de værdier, der benyttes af den hardware, billederne skal behandles på. Modellen ligger bl.a. til grund for BMP- og PPM-formaterne. Selvom farverne rød, grøn og blå udmærket kan opfattes af HVS, er RGB-modellen ikke særlig velegnet ved bestemmelse af en del af de synsfænomener vi ønsker at behandle. Vi har derfor valgt at konvertere RGB-værdierne i de billeder der behandles, til de tilsvarende HSV-værdier, da disse i højere grad repræsenterer de værdier vi mener, der kan ændres på. Umiddelbart giver HSI-modellen en mere intuitiv beskrivelse af farver, end HSV-modellen. Vi har dog valgt HSV-modellen, da den er nemmere at implementere. Efter at have behandlet HSV-værdierne for et billede, og fundet de største ændringer i denne repræsentation, konverteres ændringerne tilbage til RGB-værdier, da billedet vil være repræsenteret ved disse værdier i det valgte filformat.

6.2 Just noticeable distortion

Med et ønske om at kunne komprimere billeder med størst mulig kompression, uden at dette går ud over billedets perceptuelle kvalitet, har JPEG (Joint Pictures Expert Group) fremstillet en irreversibel komprimeringsmetode. Dette betyder, at billedet efter komprimering/dekomprimering kun er ændret, hvor dette skulle have minimal indflydelse på opfattelsen af billedet. Man opererer her med begrebet *just noticeable distortion*, der betyder, at ændringen i et givent billede kun må være netop synlig.

Herunder beskrives JPEG-komprimering for billeder, hvor hver pixel har netop én komponent, eksempelvis en gråtone. JPEG kan også benyttes på andet end gråtonebilleder, hvor metoden benyttes på lyskoefficienten (intensiteten) og to farvekomponenter for sig. Ved JPEG-komprimering behandles et område på 8×8-pixels ad gangen. Værdierne for disse 64 pixels skiftes til området omkring nul. Således vil et 8-bit område, der før havde værdier i intervallet 0 til 255, nu have værdier i intervallet -128 til 127. Disse 64 værdier kan beskrives ved en basis af 64 enheder, og i JPEG har man valgt at lade 64 cosinustransformationer danne denne basis. Dette gøres ved den diskrete cosinustransformation (DCT). Herved kan de 64 oprindelige pixelværdier altså beskrives ved 64 DCT-koefficienter. Det viser sig, at lave frekvenser har de største koefficienter, mens højere frekvenser ofte har koefficienter tæt ved nul.

At tilnærme koefficienterne for de høje frekvenser til nul svarer til at lade billedet gennemgå et lavpasfilter, eller at ændre ved hver pixels mindst betydende bit. Den bedste komprimering fås ved at opnå et stort antal 0'er blandt koefficienterne til de høje frekvenser. For at opnå dette bliver alle koefficienter kvantiseret udfra en kvantiseringstabel. Kvantiseringen foregå efter metoden:

$$CQ = \text{round}\left(\frac{C}{Q}\right)$$

Her er C er den oprindelige koefficient, og Q er værdien fra kvantiseringstabellen. Kvantiseringstabellen, der ikke er en fastsat del af JPEG-standarden, vil være opbygget således, at CQ-værdierne bliver så små som muligt, og med flest mulige 0'er imellem. Dette skal dog gøres udfra kravet om just noticeable distortion. Resten af JPEG-komprimeringen går ud på at komprimere repræsentationen af de kvantiserede koefficienter. Dette gøres ved at ordne koefficienterne, så man får flest mulige 0'er i træk, og derefter benyttes et entropimål og f.eks. Hoffman-kodning.

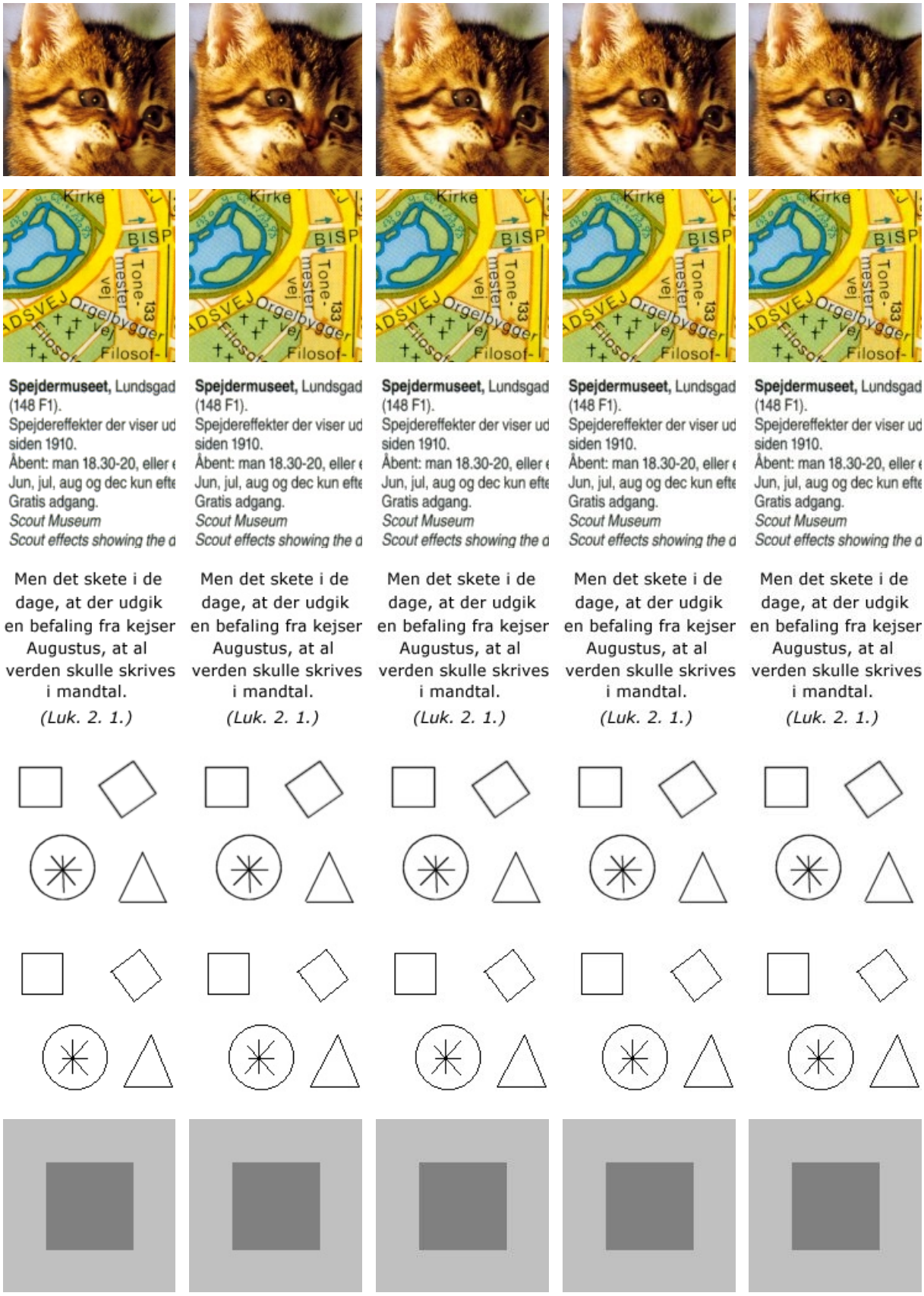
JPEG-gruppen har opsat anbefalinger om kvantiseringstabellens udseende, men den brugte tabel i en given situation, er op til den anvendte applikation. I stedet for at bruge tid på at implementere en JPEG-algoritme, har vi valgt at benytte den implementering af metoden, der findes i tegneprogrammet PaintShop Pro 6.02. Da vi her ikke kender kvantiseringstabellens udseende[12], har vi eksperimenteret os frem til, hvilket kvalitetsmål vi kunne lægge til grund for et JPEG-billede, der skal anvendes ved tolerancebestemmelsen. Til dette eksperiment har vi her valgt en række billeder

[12] En henvendelse til JASC, der står bag Paint Shop Pro 6.02 , gav ikke opklarende oplysninger.

med forskellige egenskaber, og har gemt disse billeder med forskellige komprimeringsfaktorer (i intervallet [5%, 25%]). Billederne, der kan ses i Figur 39, er følgende:

1. Kattebillede. Dette billede er naturligt, med bløde overgange mellem farver, og med mange farvenuancer.
2. Kortudsnit. Et udsnit af et kort, der er blevet indscannet. Her er færre farver, og kanterne er mere tydelige end i Kattebilledet.
3. Indscannet tekst. Her er endnu færre farver end i kortudsnittet, og opfattelsen af detaljerne (bogstaverne) er mere følsom over for fejl.
4. Konstrueret tekst. Dette billede er skabt i et tegneprogram, hvor en tekst er skrevet med antialiasing. Billedet indeholder mindre støj end det indscannede billede, og opfattelsen af det er derfor endnu mere følsom over for fejl.
5. Bløde kanter. Dette billede er konstrueret i et tegneprogram, og forestiller geometriske figurer. Der er anvendt antialiasing.
6. Hårde kanter. Dette billede er, som det tidligere, skabt med et tegneprogram. Her er ikke benyttet antialiasing, hvilket gør kanter mere takkede.
7. Mørk kasse på lys baggrund. Endnu et konstrueret billede. Kanterne på kassen er anbragt således i billedet, at de vil komme til at ligge midt i en 8×8-blok, hvorfor JPEG-forandringer bliver tydeligst.

Da JPEG er baseret på billeder med bløde overgange (*continues tone*), må der ganske naturligt komme den største forringelser af billeder, hvor der ikke er bløde overgange. Derfor må de sidste to billeder stille de største krav til komprimeringen, og dermed forlange den bedste billedkvalitet og den mindste komprimering.



Figur 39: JPEG-billeder, der er gemt med forskellige komprimeringsfaktorer.

Figur 39 viser, at billeder nemt kan konstrueres, så de ikke tåler særlig stor JPEG-komprimering, før forandringen i billedet er tydelig. I Kattebilledet er det dog næsten umuligt at se forskel på billederne med en komprimeringsfaktor på henholdsvis 5% og 25%. Derfor har vi i personforsøgene valgt at benytte JPEG-billeder, med noget højere komprimeringsfaktorer.

6.3 Kantdetektion

Da kanter er en vigtig del af HVS (se afsnit 5.4.2), er kantdetektion en vigtig del af bestemmelsen af den perceptuelle tolerance. Det er her vores teori, at det er muligt at ændre mere i pixels, der ligger på en kant i billedet, end det er i de omkringliggende pixels.

Vi ønsker kun at finde pixels, der også af HVS vil blive genkendt som kantpixels. Derfor har vi ikke behov for at implementere forholdsvis komplekse algoritmer, der kan detektere kanter i støjfyldte områder. I stedet kan vi nøjes med simple algoritmer, der i N_8 kan afgøre om p_5 er en kantpixel. For at bestemme pixels, som også HVS ville finde, og samtidig finde så mange som muligt, da mængden er afgørende for den endelige steganografiske kapacitet, har vi valgt at undersøge og implementere tre metoder. For alle tre metoder er en kantpixel defineret som en pixel, der ligge på grænsen mellem to områder, med forskellige intensiteter eller forskellige farver.

Den første metode går ud på at finde forskel på intensiteten af pixels i parrene (p_1,p_9), (p_2,p_8), (p_3,p_7) og (p_4,p_6), altså de pixel-par, der omgiver p_5. Hvis forskellen i et blot et af parrene overskride en given tærskelværdi, defineres p_5 til at være en kantpixel.

De næste to metoder, der stadig er meget simple, benytter de derivative Prewitt- eller Sobel-filtre, der ligeledes bruges på pixelintensiteterne. At filtrene er derivative betyder, at de begge er baseret på en approximation af gradienten i en pixel. Gradienten er givet ved:

$$\nabla \underline{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Gradienten taget i en given pixel angiver både hvilken retning den største ændring af intensiteten vil forekomme i, samt hvor stor en ændring der er tale om. Størrelsen af gradienten $\nabla f = \sqrt{G_x^2 + G_y^2}$ giver således et mål for intensitetsændringer i området omkring pixelen, og overstiger ∇f en given tærskelværdi, kan pixelen defineres til at være en kantpixel.

At beregne gradienten i en pixel, efter ovenstående definition, kan ikke umiddelbart lade sig gøre, bl.a. fordi gradienten kun kan defineres i et kontinuert område, hvilket i et billede kun kan være opfyldt med fuldstændig samme intensitet over hele billedet; i sådanne tilfælde vil gradienten ikke længere være interessant. Størrelsen af gradienten kan dog tilnærmes med $\nabla f = |G_x| + |G_y|$, og hvis G_x og G_y defineres som en vægtning af henholdsvis de rækker og de søjler, der omgiver den behandlede pixel, ser beregningerne lidt anderledes ud:

I Prewitt-metoden tildeles G_x og G_y værdierne:

$$G_x = (p_7 + p_8 + p_9) - (p_1 + p_2 + p_3)$$
$$G_y = (p_3 + p_6 + p_9) - (p_1 + p_4 + p_7)$$

I Sobel-metoden, til forskel for Prewitt-metoden, vægtes 4-naboer højere end hjørne-naboer, hvilket får G_x og G_y til at se således ud:

$$G_x = (p_7 + 2p_8 + p_9) - (p_1 + 2p_2 + p_3)$$
$$G_y = (p_3 + 2p_6 + p_9) - (p_1 + 2p_4 + p_7)$$

Den ulige vægtning gør naturligvis, at de to metoder ikke kan sammenlignes med samme tærskelværdi, for at bestemme, hvorvidt en given pixel er en kantpixel. I Figur 40 ses matricerepræsentationer (*templates*) af de to metoder: Prewitt-operatorerne ses i Figur 40(a) mens Sobel-operatorerne ses i Figur 40(b).

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

(a)

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

(b)

Figur 40: Matricerepræsentation af kantdetektorer. (a) Prewitt og (b) Sobel.

6.4 Homogenitet

Som det fremgår af afsnit 5.4.3, er der stor forskel på, hvor meget en pixel kan ændres, afhængigt af dens omgivelser. At bestemme om en pixel ligger på en kant er relativt simpelt (se afsnit 6.3),

men dette er ikke den eneste form for kontekstafhængighed i bestemmelse af den perceptuelle tolerance. F.eks. vil en pixel i ensartede omgivelser ikke kunne ændres så meget som en pixel i mere varierede omgivelser, uden at ændringen vil kunne ses.

I [Brink 96] omtales et mål for den kontekstafhængige entropi, målt på gråtonerne i et billede. Entropien, der her adskiller sig ganske lidt fra entropien i [Shannon 48], anvendes til bestemmelse af lokale tærskelværdier ved konvertering af et gråtonebillede til et binært billede. Inspireret af denne metode har vi valgt at bestemme et mål for, hvor ensartet et område er. Dette mål kalder vi *homogeniteten*, og den bestemmes for den pixel, der befinder sig i midten af området.

Ifølge afsnit 5.2.2 er HVS tilpasset til at behandle farver og lysintensiteter hver for sig. Vi har derfor valgt at definere to homogenitetsmål: *homogeniteten for intensitet*, som vi betegner H_i, og *homogeniteten for hue*, som betegnes H_h.

Homogeniteten for intensiteten bestemmes for en 3×3-matrix (N_8), hvor H_i er en egenskab, der tildeles pixelen i midten. Lad μ_{i,N_8} angive den gennemsnitlige intensitet for pixlene i N_8, og $g_{i,j}$ være intensiteten for den j'te pixel. Indekset i angiver her, at der er tale om homogeniteten for intensiteten. Variansen over intensiteterne i matricen er da givet ved:

$$\sigma_i^2 = \frac{1}{9} \sum_{j \in N_8} \left(g_{i,j} - \mu_{i,N_8} \right)^2$$

Jo større variansen i et område er, jo mindre bør homogeniteten i området være. Hvis g-værdierne ligger i intervallet [0; 1], ønskes homogeniteten bestemt, så den også ligger indenfor dette interval. Homogeniteten defineres derfor som:

$$H_i = 1 - 4\sigma_i^2$$

For homogeniteten for hue betegner f de normerede hue-værdier. Variansen og homogeniteten for hue bestemmes derfor ved:

$$\sigma_h^2 = \frac{1}{9} \sum_{j \in N_8} \left(f_{h,j} - \mu_{h,N_8} \right)^2$$

$$H_h = 1 - 4\sigma_h^2$$

Definitionen for H_h er ikke helt uproblematisk. Hvis alle hue-værdier ligger i området omkring rød, således, at nogle værdier har en høj hue-værdi (mellem f.eks. 350° og 360°) og andre har en lav hue værdi (f.eks. mellem 0° og 10°), så vil definitionen bestemme, at variansen i dette område er høj, hvorfor homogeniteten er lille. Der vil altså kunne ændres meget i disse pixels, selvom de burde have en stor homogenitet. Dette er dog et særtilfælde, hvorfor vi har valgt ikke at gøre mere ud af denne definition.

6.5 Assimilering

I afsnit 5.3.2 beskrives, hvordan områder kan opfattes lysere/mørkere, afhængig af omgivelserne. Dette kan bruges til at udvide toleranceintervallerne for en given pixel i retning af gennemsnittet for området. For gråtonebilleder kan dette simpelt anskues som en tilnærmelse af gråtoneværdien til den gennemsnitlige gråtoneværdi for området, hvor området kan være bestemt af en pixel og dens 8-naboer. Vi har valgt at se på to forskellige former for tildelinger, der begge vil være passende for eksempelvis gråtoner i et gråtonebillede. Begge tildelinger er afhængige af en given tærskelværdi *T*, hvor *T* sammenlignes med værdien $|\mu(N_8) - x|$.

I den første metode rykkes den ende af toleranceintervallet, *x*, der er nærmest gennemsnittet i området, $\mu(N_8)$, til gennemsnitsværdien:

$$|\mu(N_8) - x| < T \qquad x \leftarrow \mu(N_8)$$

Den anden metode rykker ikke *x* til $\mu(N_8)$, men vægter i stedet x og $\mu(N_8)$ligeligt, og tildeler denne værdi til *x*:

$$|\mu(N_8) - x| < T \qquad x \leftarrow \frac{\mu(N_8) + x}{2}$$

Begge metoder vil medføre en homogenisering af intensiteten i et område, altså det samme som øjet selv forsøger.

Den eneste forskel på ovenstående metoder ligger i vægtningen af værdien for den midterste pixel. Metoderne kan også beskrives ved to tildelingsmatricer (Figur 41 (a) og (b)), hvor den eneste forskel i matricerne består i vægtningen af den midterste pixel i forhold til de andre pixels.

$$x = \frac{1}{9} \cdot \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

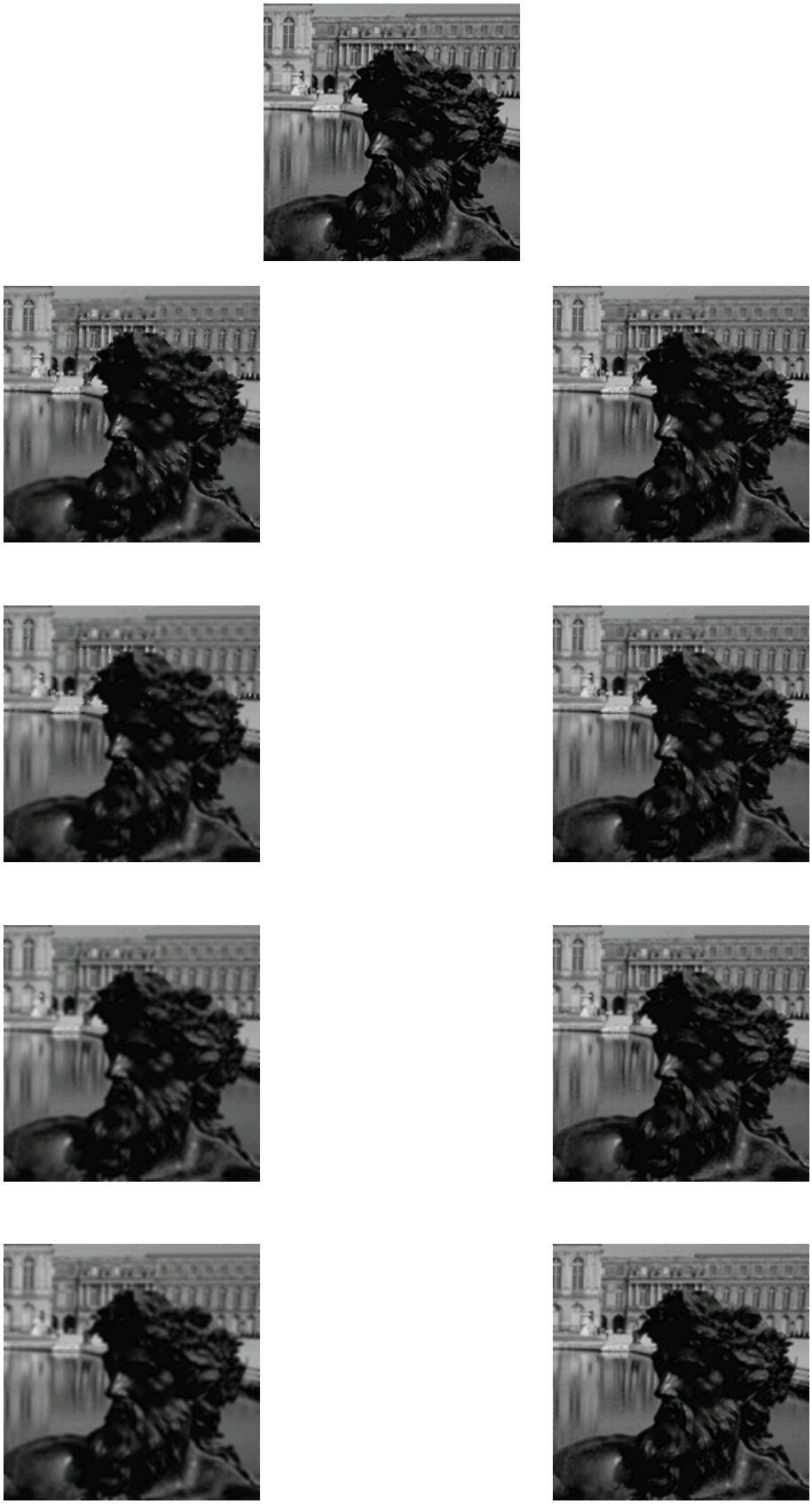
(a)

$$x = \frac{1}{18} \cdot \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 10 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

(b)

Figur 41: Assimileringsmatricer. (a) Middelværdi og (b) vægtet.

I Figur 42 ses et antal gråtonebilleder, hvor enkelte pixels har fået nye gråtoneværdier, beregnet efter tildelingsmatricerne i Figur 41, afhængigt af forskellige tærskelværdier. Det øverste billede er det originale billede, billederne i venstre søjle benytter tildelingen fra Figur 41(a), mens billederne i højre søjle benytter tildelingen fra Figur 41(b). I de nederste fire rækker er tærskelværdierne 25, 50, 100 og 200, og billederne indeholder op til 256 gråtoneværdier.



Figur 42: Assimileringsbilleder. Billederne til højre benytter middelværdien, mens billederne til venstre benytter den vægtede værdi.

Som det ses af billederne i Figur 42, så vil tildelingsmatricen i Figur 41(b) resultere i den mindste forringelse af originalbilledet, selv ved høje tærskelværdier.

De her beskrevne assimileringsfiltre betegnes også som *gennemsnitsfiltre*, der er lavpasfiltre (se bl.a. [Gonzales&Woods 93]), idet de i frekvensrepræsentationen af et billede vil fjerne de høje frekvenser, og kun lade de lave frekvenser passere. At denne filtrering sættes afhængig af en tærskelværdi gør, at filtreringen kun vil forekomme nogle steder i billedet, og sættes tærskelværdien lavt nok, filtreres kun i de områder, der i øvrigt ikke er præget af høje frekvenser.

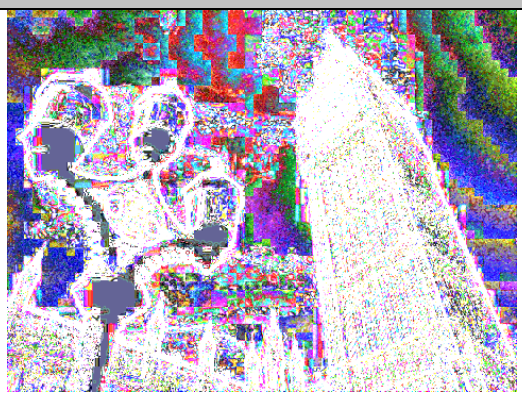
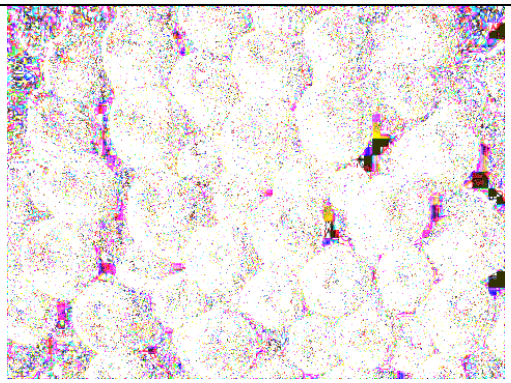
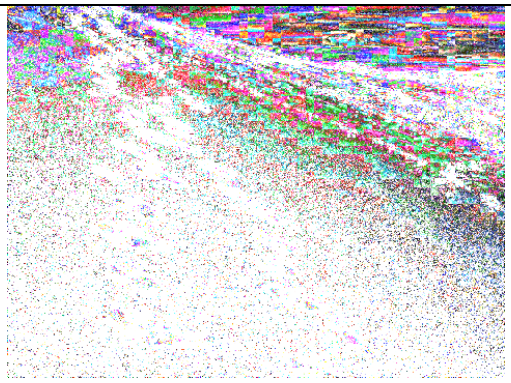
Et andet lavpasfilter, der i billedrestaurering konkurrerer med gennemsnitsfiltret, er medianfiltret. Medianfiltret er et ikke-lineært filter, der, som gennemsnitsfiltret, baserer sig på værdierne i en pixels omgivelser. I stedet for at tildele en pixel værdien af gennemsnittet i omgivelserne, tildeles i stedet median-værdien. Dette gør, at få værdier, der varierer meget fra gennemsnittet i området, ikke kommer til at spille ind på tildelingen af nye værdier. Dette kan have sin fordel i forbindelse med restaurering af støjfyldte billeder, hvor en pixel der af støj har fået ændret sin værdi betydeligt, ikke vil spille ind på det færdige resultat. At medianfiltret har en fordel over for gennemsnitsfiltret ved billedrestaurering er dog ikke ensbetydende med, at det er fordelagtigt at bruge i forbindelse med bestemmelse af den perceptuelle tolerance. I et tænkt eksempel, hvor alle pixel i et større område har samme værdi, og kun en pixel adskiller sig, vil medianfiltret kun resulterere i ændring af den pixel der adskiller sig, mens gennemsnitsfiltret vil resultere i ændringer for den ene pixel såvel som for dens 8-naboer. Da det i forbindelse med bestemmelse af den perceptuelle tolerance gælder om at bestemme den størst mulige ændring for alle pixels, er det altså i dette eksempel en fordel at benytte gennemsnitsfiltret. Dermed dog ikke sagt, at dette filter i alle tilfælde vil give den største ændring, men da det for os at se er det filter, der bedst efterligner øjets evne til at mindske kontrasten i et område med i forvejen lav kontrast, har vi valgt at bruge dette.

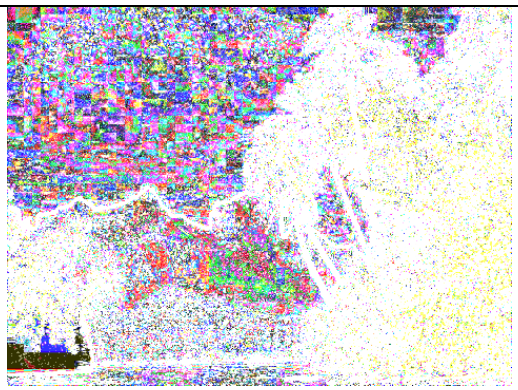
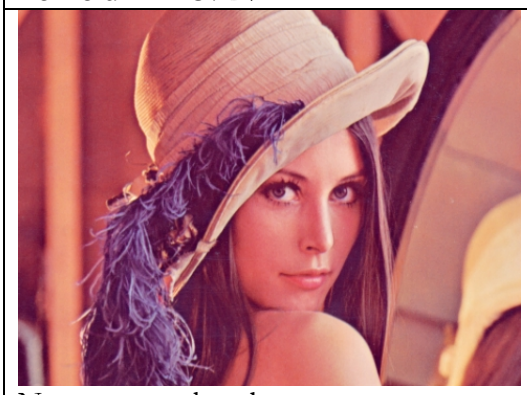
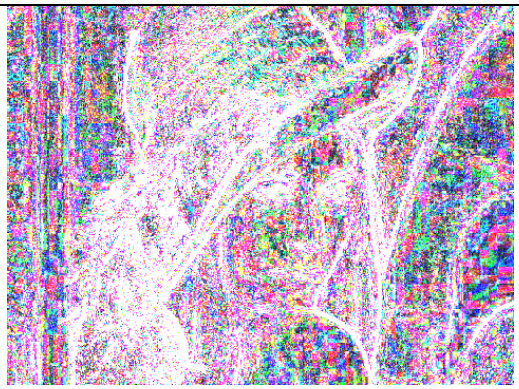
6.6 Bestemmelse af kapacitet

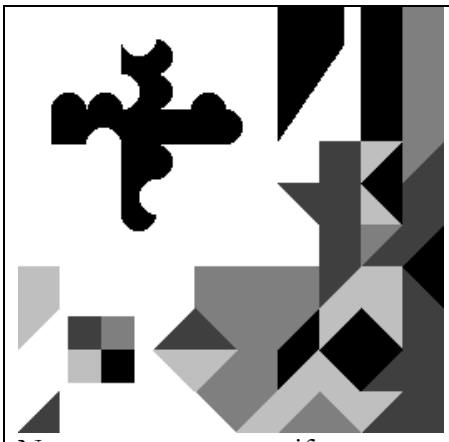
Vi har underkastet et udvalg af billeder ovenstående tildeling af toleranceintervaller, for at se, hvilken samlet kapacitet billederne hver især har. Til tildelingen af toleranceintervaller for billederne har vi benyttet metoden beskrevet i afsnit 6.1, hvor de enkelte filtre er blevet indstillet med parameterværdier i henhold til resultaterne af personforsøgene (se afsnit 7). Ved at benytte beregningerne fra afsnit 4.5, bestemmes det antal bit, der kan indkodes i billederne, og som camoufleret information er valgt teksten fra en ren tekstfil[13]. I Tabel 7 ses en oversigt over de udvalgte

[13] Vi har valgt at benytte "I have a dream"-talen af Martin Luther Kings, Jr. (www.bcpl.net/~dbroida/mlkihad.html).

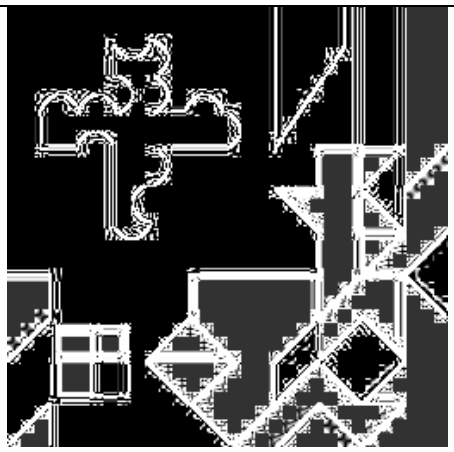
billeder, og deres $\underline{\underline{T}}^{50}$ tolerancebilleder. Desuden angives billederne navne, deres størrelser og steganografiske kapacitet, samt forholdet mellem kapaciteten og størrelsen. Både værtbilleder og stegobilleder kan findes på den medfølgende cd-rom.

Originalt billede	Tolerancebillede
 Navn: bigben.bmp Størrelse: 480×360×24 bit Kapacitet: 1.283.542 bit Forhold: 30.9%	 Navn: 1Tolerance.ppm
 Navn: cacti.bmp Størrelse: 480×360×24 bit Kapacitet: 1.765.294 bit Forhold: 42.6%	 Navn: 2Tolerance.ppm
 Navn: duneprints.bmp Størrelse: 480×360×24 bit Kapacitet: 1.562.723 bit Forhold: 37.7%	 Navn: 3Tolerance.ppm

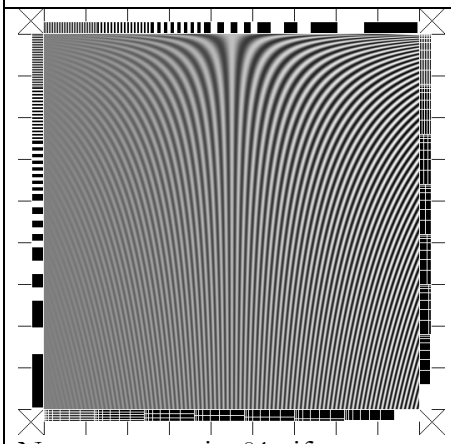
 Navn. versailles.bmp Størrelse: 480×360×24 bit Kapacitet: 1.464.593 bit Forhold: 35.3%	 Navn: 4Tolerance.ppm
 Navn: bora.bmp Størrelse: 480×360×24 bit Kapacitet: 1.543.589 bit Forhold: 37.2%	 Navn: 5Tolerance.ppm
 Navn: lena.bmp Størrelse: 480×360×24 bit Kapacitet: 1.382.729 bit Forhold: 33.3%	 Navn: 6Tolerance.ppm



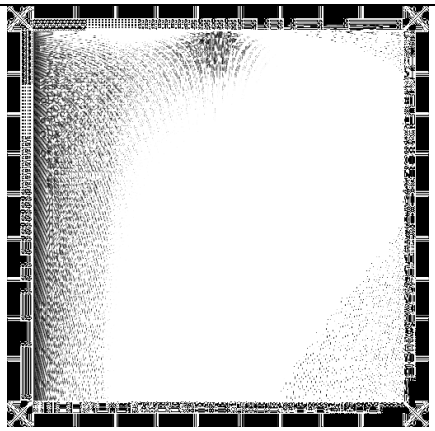
Navn: corner.gif
Størrelse: 256×256×24
Kapacitet: 209.484 bit
Forhold: 13.3%



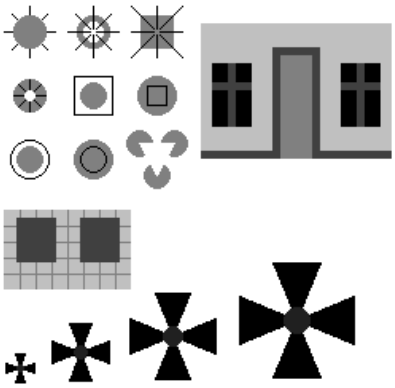
Navn: 7Tolerance.ppm



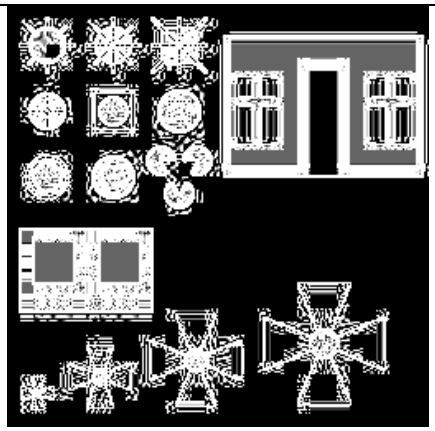
Navn: sine01.gif
Størrelse: 512×512×24
Kapacitet: 2.649.776 bit
Forhold: 42.7%



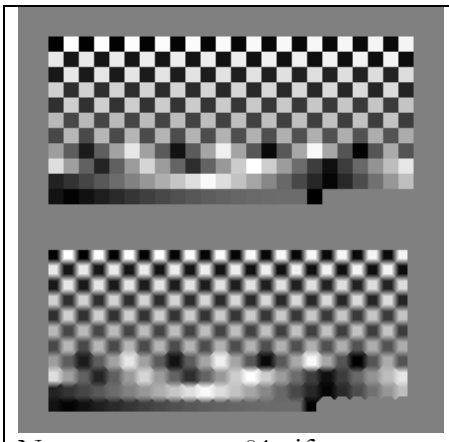
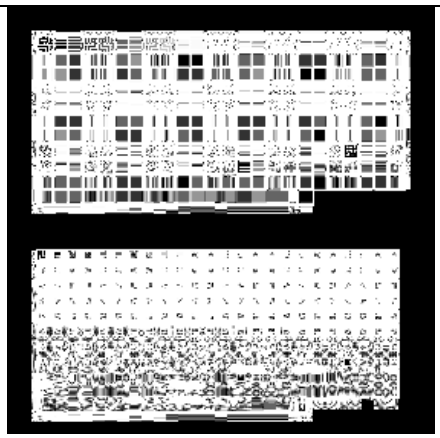
Navn: 8Tolerance.ppm



Navn: small.gif
Størrelse: 256×256×24
Kapacitet: 269.337 bit
Forhold: 17.2%



Navn: 9Tolerance.ppm

 Navn: step01.gif Størrelse: 280×280×24 Kapacitet: 541.326 bit Forhold: 28.8%	 Navn: 10Tolerance.ppm

Tabel 7: Oversigt over kapaciteter.

7. Personforsøg

I afsnit 6 defineres den perceptuelle tolerance for et givent billede, og i afsnit 6.1 beskrives, hvordan tolerancen beregnes ud fra forskellige filtre. De seks filtre, som vi har valgt at beskæftige os med, er alle afhængige af en eller flere parametre, som vil have indflydelse på, hvor stor et billedes tolerance beregnes til at være. At bestemme hvilke værdier disse parametre skal antage er ikke ganske trivielt. Jo større ændringer en parameter tillader i et billede, jo større vil billedets tolerance blive. Desværre betyder større ændringer ofte også, at en beskuer nemmere vil opfatte billedet som ændret. En bestemmelse af den perceptuelle tolerance kræver derfor, at de parametre, der bestemmer filtrenes ændringer af et billede, er tilpasset, så de tillader størst mulig ændring af et billede, uden at billedet er perceptuelt ændret.

7.1 Formål

Det primære formål med personforsøgene har været at bestemme en del af de parameterværdier, der indgår i de anvendte filtre til bestemmelse af et billedes perceptuelle tolerance. Disse parameterværdier er bestemt udfra et antal forsøgspersoners bedømmelser af billeder, der er ændret med de filtre, som er beskrevet i afsnit 6.

Digitale billeder kan ændres af andre filtre end de her beskrevne, og de anvendte filtre kan justeres med flere parametre end de der er undersøgt. Desuden er ændringernes synlighed ikke alene afhængig af parameterværdierne, men også af billedet selv. Af disse grunde er resultaterne af undersøgelsen kun at betragte som vejledende, og altså ikke udtømmende.

7.2 Metode

I det samlede forsøg har en række forsøgspersoner[14] individuelt været placeret ved en PC med skærm, hvor et program tog sig af fremvisning af billeder, samt opsamling af resultater. Inden et forsøg er blevet sat i gang, har hver forsøgsperson læst en instruktion, der fortæller, hvordan billederne ville blive vist, samt hvilken opgave forsøgspersonen havde i forsøget. Desuden har forsøgspersonen angivet en bedømmelse af sit eget syn på forsøgstidspunktet. Både instruktion og skema til bedømmelse er gengivet i 'Bilag D: Dokumentation af personforsøg'.

[14] Forsøgspersonerne er ikke udvalgt som et bredt demografisk udvalg af den danske befolkning. De er i stedet valgt ud fra kriteriet: "Vi kender dem, og de vil gerne stå til rådighed".

Under forsøgene har forsøgspersonerne benyttet programmet *StegoViewer*, hvis skærmbillede er gengivet i Figur 43. I programmet, der er udviklet specielt til disse forsøg, er blevet vist ni billeder ad gangen. Af disse ni billeder har de syv været helt ens, mens de sidste to muligvis har skilt sig ud. De to billeder, der skilte sig ud, har i mange tilfælde lignet de andre syv så meget, at det ikke har været muligt for forsøgspersonen at se forskel, og i enkelte tilfælde har alle ni billeder været helt ens. Forsøgspersonens opgave har været, at bestemme to billeder som han/hun mente skilte sig ud, markere dem med computermusen, og derefter klikke på 'Næste'-knappen. Hvis det kun har været muligt for forsøgspersonen at finde og markere ét billede i et sæt, har dette også været tilladt, ligesom det har været tilladt at gå videre uden at markere noget billede.

Figur 43: Skærmbillede fra programmet StegoViewer, der har været anvendt under personforsøgene.

Der er blevet vist 106 forskellige sæt af billeder for hver forsøgsperson. Rækkefølgen af sættene har været tilfældigt genereret for hver forsøgsperson, ligesom placeringen af de billeder, der skiller sig ud, har været tilfældigt valgt for hvert sæt. Sættenes rækkefølge er valgt således, for at mindske de tidslige effekter under forsøgene: I starten af et forsøg er det ikke givet, at en forsøgsperson vil være helt klar over, hvilke detaljer der skal ses efter. Mod slutningen af forsøget kan forsøgspersonen blive træt i øjnene, og dermed få svært ved at se forskel på billederne. At billederne inden for et sæt har været vist på tilfældige positioner er gjort for at mindske effekten af fejl ved skærmen. Således skulle skæv skærmudlysning, konvergensfejl og reflektioner få mindre indflydelse på de trufne valg.

Sættene i forsøget har været genereret ud fra seks grundbilleder (se 7.4). Inden forsøgene er gået i gang, har hver forsøgsperson set på et udpluk af de billeder, der har været anvendt i selve forsøget, for bl.a. at se alle de seks grundbilleder. I den anledning er forsøgspersonerne også blevet gjort opmærksomme på størrelsesordenen af detaljer, der kan være ændret i billederne, samt

hvordan de skulle anvende *StegoViewer*. Desuden er forsøgspersonerne blevet gjort opmærksomme på, at de ikke behøvede at bruge mere end ca. et minut på hvert sæt, og at de blot skulle gå videre til næste sæt, hvis de ikke havde observeret ændringerne inden for denne tid. Dette blev gjort for at begrænse den tid forsøgspersonerne skulle anvende på forsøgene.

Under forsøgene er forsøgspersonernes tidsforbrug løbende blevet registreret. Forsøgspersonerne er dog ikke blevet gjort opmærksomme på dette, da de ikke skulle føle sig presset til at gennemføre forsøget så hurtigt som muligt. Forsøgspersonerne har under forsøget frit kunne forsyne sig med drikkevare, holde pauser, mm.

7.3 Anvendte parametre

I dette afsnit beskrives nogle af de parametre ved de anvendte filtre, som kan indgå i en undersøgelse af den optimale parametersammensætning til bestemmelse af toleranceintervallet. Desuden beskrives de af disse parametre, som vi har valgt at undersøge nærmere.

Just noticeable distortion er, som tidligere beskrevet, en betingelse ved JPEG-komprimering, der angiver, hvor meget et billede må være ændret, før dette giver netop synlig forringelse af billedkvaliteten. I stedet for at implementerer en algoritme til JPEG-komprimering af billeder, har vi valgt at benytte Paint Shop Pro™ 6.02 til konvertering af billederne. I dette produkt er der mulighed for at vælge, hvilken kompressionsstyrke, i intervallet 1 til 99, et billede skal JPEG-komprimeres med, og det er denne parameter vi har valgt at undersøge.

I kantdetektions-filtret lader vi pixels, der af en kantdetektionsalgoritme bestemmes til at være kantpixels, blive ændret med en given værdi. Da et billedes steganografiske kapacitet er afhængig af, at så mange pixels som muligt bliver tildelt et toleranceinterval, kunne det her være nærliggende at undersøge en række forskellige kantdetektionsalgoritmer, for at undersøge, hvilke algoritmer, der giver flest kantpixels. Desuden kunne det være oplagt at undersøge resultatet af kantdetektionsalgoritmer i kombination med en Hough-transformation, der kan fuldende utydelige kanter, og dermed udpege yderligere kantpixels. Nogle kantdetektionsalgoritmer tildeler hver pixel en "kant-værdi", og denne værdi skal overgå en given tærskelværdi, før pixelen siges at være en kantpixel. Det er derfor her relevant at undersøge, hvilken tærskelværdi, der netop godkender de pixels som kantpixels, der kan ændres, uden at det kan ses. Endelig er det muligt at variere, hvor meget den enkelte kantpixels pixelkomponenter kan ændres, før dette kan ses, og på hvilke måder komponenterne skal ændres. Vi har her valgt kun at undersøge den sidste parameter, altså hvor meget der kan ændres i pixels, der af én kantdetektionsalgoritme, og med én fastsat tærskel-

værdi, blive genkendt som kantpixels. Dette har vi valgt, da forskellige algoritmer ikke ville give perceptuelle forskelle i ændringen af pixelkomponenter, men kun i antallet af valgte pixels. Ligeledes gør det for tolerancen i de fundne pixel ingen forskel om de er fundet med én tærskelværdi eller en anden. I kantdetektionsfiltret har vi valgt at benytte et Sobel-filter (beskrevet i afsnit 6.3) til at udvælge kantpixels, udfra værdien V af pixels, hvor V er fundet med HSV-farvemodellen (se afsnit 3.2.5). De pixels, der er bestemt som kantpixels, har fået deres V-værdi ændret med en given procentsats.

I forbindelse med homogenitet skal der undersøges paramterværdier ved både homogenitet for intensiteten, H_i, og homogeniteten for hue, H_h. Da vi har valgt at benytte HSV-modellen i personforsøgene, fremfor HSI-modellen, er det parameterværdier for homogeniteten for value (H_v) der skal undersøges i stedet for værdier for homogenitet for intensiteten. I beregningen af erstattes I-værdierne med de tilsvarende V-værdier. Som tilfældet var med kantpixels, så vil et homogenitet-filter tildele en pixel en værdi, i dette tilfælde en homogenitets-værdi, som pixelændringen vil være afhængig af. Denne ændring kan ske for alle pixels som funktion af homogenitetsværdien, eller den kan f.eks. være afhængig af både en tærskelværdi og value/hue, således, at kun pixels med en homogenitets-værdi under en given værdi ændres. Vi har valgt den sidste af disse metoder, og den giver derfor to parametre, der kan indgå i forsøget: Tærskelværdien, og den ændring der sker for pixels, som bestemmes af tærskelværdien. Disse to parametre medtages for både H_v og H_h. Da parameterværdierne ikke kan siges at være uafhængige, har vi valgt at lade parametrene indgå med 15 kombinationer, hvor fem forskellige tærskelvædier hver danner basis for tre forskellige pixelændringer. For H_v ændres V-værdien procentuelt, afhængig af tærskelværdien, mens det er H-værdien der ændres ved H_h.

Assimilerings-filtret, der svarer til en middelværdifiltrering, som er afhængig af forskellen mellem originalværdi og middelværdien, er kun afhængig af, om denne forskel er større eller mindre end en given tærskelværdi. Vi har derfor her valgt at lade tærskelværdien indgå i samlingen af parameterværdier, der skal undersøges. Da der er tale om to forskellige assimileringsfiltre, hvor det ene filter tildeler middelværdien, og det andet filter tildeler en værdi, der er en ligelig vægtning af den originale værdi og middelværdien, er der tale om to forskellige tærskelværdier, der skal undersøges. En anden oplagt parameter at undersøge, er vægtningen mellem original- og middelværdi ved det vægtede filter. Denne parameter har vi dog valgt at se bort fra, da vi ønsker at holde antallet af forsøgsbilleder nede.

7.4 Anvendte billeder

De billeder, der har været anvendt i forsøgene, er blevet fremstillet ud fra "naturlige" grundbilleder, så de har dækket de paramterværdier, der er beskrevet det forgående afsnit. Figur 44 viser de anvendte grundbilleder. Billederne 'bigben', 'bora', 'versailles', 'duneprints' og 'cacti' er alle hentet fra Kodaks hjemmeside ('www.kodak.com'), mens 'lena' er hentet fra 'www.lena.org'. Billederne er tilpasset en størrelse på 480×360 pixels, og pånær 'cacti' er alle billeder vist på hovedet. Dette er valgt for at gøre det lettere for forsøgspersonerne at koncentrere sig om detaljer i billederne, i stedet for billedernes motiver.

(a) bigben (b) lena (c) bora

(d) versailles (e) duneprints (f) cacti

Figur 44: De grundbilleder, der blev anvendt ved forsøgene.

Hver parameter, der indgår i forsøget, er blevet afprøvet med et sæt af forskellige parameterværdier, for at bestemme den parameterværdi, der er den bedst anvendelige til ændring af et billede, uden at ændringerne kan ses. Vi har valgt at afprøve hver parameterværdi i to typer billeder. Den ene type billeder, som vi har valgt at kalde *min-max-billeder*, er genereret sådan, at hver anden pixelsøjle har minimale værdier i alle pixelkomponenter, mens de resterende pixelsøjler har maksimale pixelværdier. For et ensfarvet område, hvor tolerancen for alle pixels har næsten samme ikke-nul værdi, vil søjlerne med maksimale pixelværdier få en højere intensitet (uanset om vi benytter HSV- eller RGB-billeder), og de andre søjler vil få en lavere intensitet, end i originalbilledet. Vi har valgt at generere disse billeder, da det formentlig er nemmere at se ændringer i et billede med denne effekt end f.eks. i et billede, hvor alle pixelkomponenter har deres minimale vær-

dier. I den anden type billeder, som vi har valgt at kalde *arbitrærbilleder*, har alle pixelkomponenter en arbitrær værdi, der ligger indenfor komponentens toleranceinterval. Denne type billeder svarer til stegobilleder, hvor f.eks. kombinationsmetoden, beskrevet i afsnit 4.6, er anvendt, og formålet med disse billeder er, at det skulle være sværere at se ændringer her end i min-max-billederne. For hvert filter er ét grundbillede blevet anvendt til alle min-max-billeder, og et andet grundbillede til alle arbitrærbillederne.

De parameterværdier, der er indgået i forsøgene, er udvalgt således, at vi selv, med viden om hvordan et billede er blevet ændret, netop kan skelne en forskel i ca. halvdelen af billederne. For at holde antallet af anvendte billeder på et niveau, der ikke ville give forsøgspersonerne mere end ca. 1½ times forsøg, har vi desværre været nødt til at sprede de forsøgte paramterværdier meget ud. Tabel 8 angiver hvilke parameterværdier hvert filter er blevet undersøgt med, samt hvilke af grundbillederne der har været anvendt til henholdsvis min-max-billeder og til arbitrærbilleder. For homogenitetsfiltrene angiver det første tal den procentvise ændring, og det andet tal angiver tærskelværdien.

Filter:	Parameterværdier:	Min-max-billede:	Arbitrærbillede:
Just noticeable distortion. Kompressionsstyrke i Paint Shop Pro™ 6.02	(35, 50, 65, 80, 95)	lena	duneprints
Kantpixels. Procentvise forandring i en kant-pixels value (bestemt efter HSV-modellen)	(5%, 10%, 15%, 20%, 25%)	bigben	versailles
Homogenitet i value. Kombination af procentuel ændring i value og tærskelværdi for H_v.	(50, 70, 90) × (70%, 75%, 80%, 85%, 90%)	duneprints	bora
Homogenitet i hue. Kombination af procentuel ændring i hue og tærskelværdi for H_h.	(50, 70, 90) × (70%, 75%, 80%, 85%, 90%)	cacti	bigben
Assimilering – middelværdi. Tærskelværdi for tildeling af mid-	10%, 20%, 30%, 40%, 50%	versailles	lena

delværdi i value.			
Assimilering – vægtet værdi. Tærskelværdi for tildeling af lige-vægtning mellem middelværdi og originalværdi af value.	10%, 20%, 30%, 40%, 50%	bora	cacti

Tabel 8: Parameterværdierne, der blev undersøgt ved forsøgene.

7.5 Opsamlede data

Under forsøgene blev data og resultater løbende opsamlet i en fil. I hver fil har StegoViewer opsamlet følgende data:

Navn på forsøgsperson: Navnet registreres for senere databehandling. Hvis flere forsøgspersoner havde samme fornavn blev de bedt om at angive både for- og efternavn. Dette gav hver forsøgsperson en unik identifikation.

Anvendte indstillinger: For hver forsøgsperson blev registreret, hvilke billeder, der indgik i forsøget, hvordan StegoViewer var indstillet med hensyn til billedstørrelse, m.m. Disse indstillinger har været ens for de afholdte forsøg.

Følgende resultater er registreret for hvert enkelt sæt af billeder:

Indeks: Hvilket billedsæt (originalt billede, ændret billede), der blev vist. Indeks refererer her til en liste af de ovenfor nævnte billeder.

Filer: Navne på de anvendte filer. Denne information er redundant, da den fremgår af indekseringen. Den er dog medtaget, da den letter forståelsen ved gennemgang af resultaterne.

Ændrede billeder: Koordinaterne på placering af de to billeder, der er ændrede. I forsøgene har billederne været placeret i tre rækker og tre kolonner, så koordinatværdierne har ligget mellem (1,1) og (3,3).

Valgte billeder: Koordinaterne på de billeder, som forsøgspersonen har valgt som de der skilte sig ud.

Korrekte valg: Antallet af de valgte billeder, der også var ændrede.

Tidsforbrug: Den tid, målt i sekunder, der gik fra billederne blev vist, til forsøgspersonen klikkede på 'Næste'-knappen.

7.6 Databehandling

Med de indsamlede data har der været rig mulighed for at undersøge en lang række samspil mellem forskellige parametre. Vi har dog primært været interesseret i at finde de parameterværdier, der giver størst mulig tolerance, uden at dette har resulteret i tilsyneladende synlige ændringer af billedet. Bestemmelsen af disse parameterværdier er beskrevet i dette afsnit.

7.6.1 Fejlkilder

Under forsøgene har der vist sig et antal fejl, som vi har forsøgt at tage hensyn til under den videre behandling af forsøgsresultaterne.

I forbindelse med forsøgsperson nummer 15 viste det sig desværre, at StegoViewer havde en fejl i forbindelse med opsamling af data. Fejlen viste sig, når en forsøgsperson ikke valgte to billeder, men gik videre til det næste billedsæt i forsøget med ét eller ingen valgte billeder. I disse tilfælde gemte StegoViewer koordinater fra tidligere valgte billeder således, at der altid ville være to valgte billeder, hvis der tidligere i forsøget havde været valgt to billeder. Hvis forsøgspersonen valgte to billeder, ville disse valg blive gemt rigtigt. Et enkelt valgt billede ville også blive gemt rigtigt, blot ville der så også blive gemt fejlagtige koordinater, hvis et andet billede tidligere havde været valgt. Selvom denne fejl var nem at rette, lod vi StegoViewer fortsætte uændret gennem de sidste forsøg, så de sidste forsøgsresultater kunne sammenlignes med de første. Af samme årsag blev de sidste forsøgspersoner heller ikke gjort opmærksomme på fejlen.

Ovenstående fejl betyder, at de fleste sæt kan fortolkes som om forsøgspersonen var tvunget til at vælge to billeder, hvilket reelt ikke var tilfældet. At dette ikke gælder for alle billeder skyldes, at fejlen først optræder i det øjeblik forsøgspersonen har valgt to billeder, hvorfor fejlen typisk ikke vil vise sig for de første valg.

De fejlfyldte data gør, at der vil være registreret billeder som korrekt valgte, selvom forsøgspersonerne ikke har valgt dem. Vi har derfor foretaget to analyser af alle resultaterne fra forsøgene: Én analyse, hvor vi benytter de opsamlede data, og en anden analyse, hvor vi har forsøgt at fjerne de valg, som er opstået pga. fejlen.

En af forsøgspersonerne gjorde efter forsøget opmærksom på, at han var kommet til at springe et enkelt sæt af billeder over. For det pågældende sæt betyder dette, at det i teorien kunne have to valg mere registreret end det har fået. Vi har imidlertid valgt at se bort fra dette, da vi anser den

forskel som én forsøgsperson kunne have gjort på resultaterne for dette sæt, som værende af ubetydelig størrelse.

En anden forsøgsperson kom under forsøget til at lukke StegoViewer ned, da hun var nået ca. halvvejs. Da StegoViewer ved hver programkørsel beregner en rækkefølge for sættene pány, var det ikke muligt at genopstarte programmet, så forsøgspersonen kunne bedømme de resterende sæt. Vi har derfor valgt at se bort fra dette "halvt" gennemførte forsøg. Forsøgspersonen indvilgede i at udføre et nyt forsøg senere.

Nogle forsøgspersoner klagede over, at der blev vist fejlagtige røde skygger på nogle billeder, afhængig af deres placering på skærmen. Dette skyldes en mindre konvergensfejl ved skærmen, til trods for, at vi har forsøgt at indstille skærmen så godt som muligt. Da de billeder, der skilte sig ud, er blevet vist tilfældige steder på skærmen for alle forsøgspersoner, for bl.a. at mindske effekten af konvergensfejl, har vi valgt ikke at behandle denne fejlkilde yderligere.

7.6.2 Resultaternes validitet

Ved hvert sæt af billeder har forsøgspersoner haft mulighed for at markere op til to billeder. Herunder beskrives sandsynlighederne for valgene, hvis disse er tilfældige. Ved hver beregning angiver R et rigtigt valgt billede og F et forkert valgt billede.

Vælges der kun ét billede, er sandsynlighederne fordelt således:

$$P(R) = \frac{2}{9} = 22{,}22\%$$

$$P(F) = \frac{7}{9} = 77{,}77\%$$

Ved valg af to billeder er sandsynlighederne:

$$P(RR) = \frac{2}{9} \cdot \frac{1}{8} = 2{,}77\%$$

$$P(RF|FR) = \frac{2}{9} \cdot \frac{7}{8} + \frac{7}{9} \cdot \frac{2}{8} = 38{,}89\%$$

$$P(FF) = \frac{7}{9} \cdot \frac{6}{8} = 58{,}33\%$$

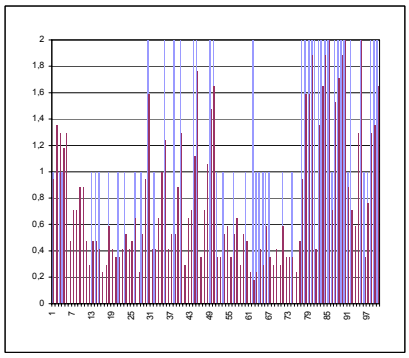
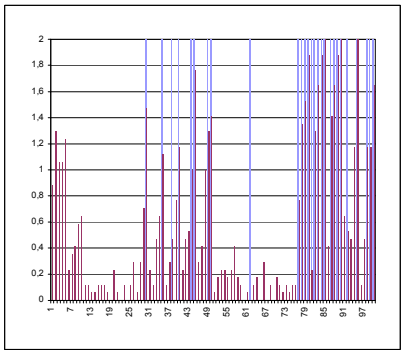
Som det fremgår af disse beregninger, er der stor sandsynlighed for, at en forsøgsperson tilfældigt kan komme til at vælge ét rigtigt billede i et sæt, hvadenten der kun vælges et billede, eller for-

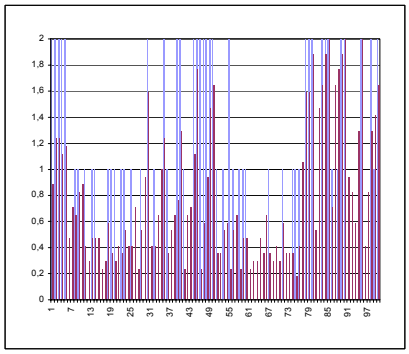
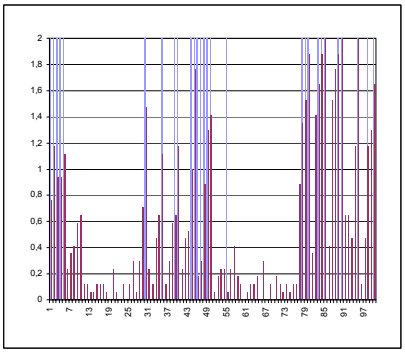
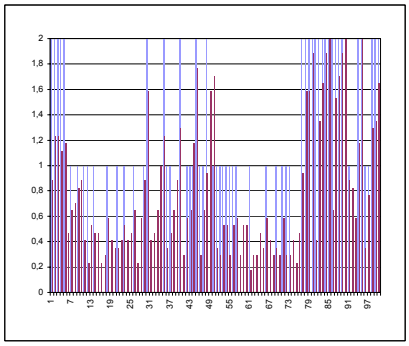
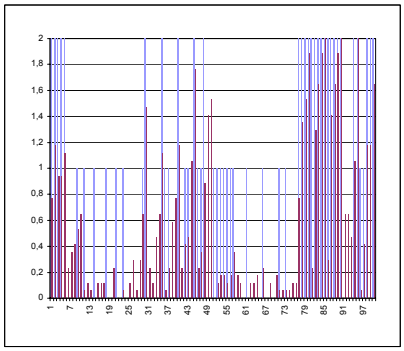
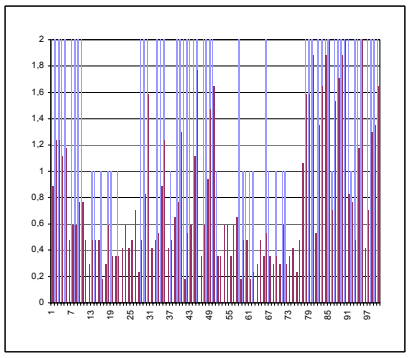
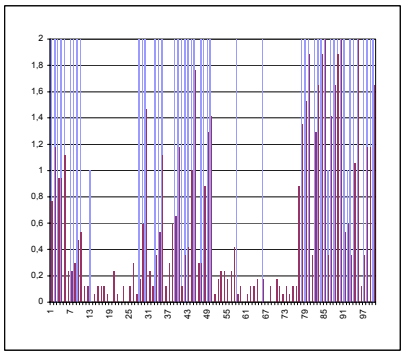
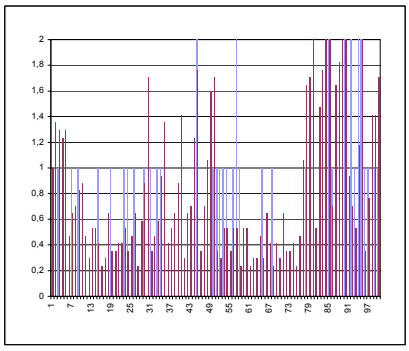
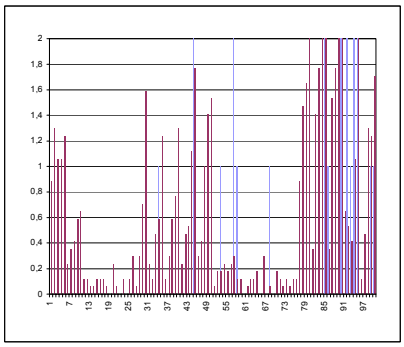
søgspersonen vælger to billeder. Da forsøgspersonerne imidlertid er blevet bedt om kun at markere de billeder, de mente skilte sig ud, og altså ikke skulle vælge billeder mere eller mindre tilfældigt, har vi bestemt ikke at behandle denne usikkerhed i den videre databehandling.

Baseret på resultaterne fra forsøgene, og på baggrund af de oplysninger om forsøgspersonernes syn, som de blev bedt om at give inden forsøget, er der flere af forsøgspersonerne der, af en eller anden grund, skilte sig ud fra resten:

- En forsøgsperson ser kun med ét øje.
- En forsøgsperson er farveblind (brun/rød/grøn).
- En forsøgsperson bedømte sit syn som værende dårligere end normalt syn.
- En forsøgsperson bedømte sit syn som værende bedre end normalt syn.
- En forsøgsperson fandt mærkbart færre billeder end de andre forsøgspersoner.

For at kunne vurdere, om disse forsøgspersoners resultater lignede de resterende forsøgspersoners, eller om de kunne betegnes som "støj" i de samlede data, har vi fremstillet to søjlediagrammer for hver af disse forsøgspersoner. I et søjlediagram indgår to serier af data: Den ene serie er antallet af den aktuelle forsøgspersons rigtigt valgte billeder for hvert sæt (markeret med blå i graferne i nedenstående figur). Den anden serie består af det gennemsnitlige antal rigtigt valgte billeder for hver serie, hvor gennemsnittet er beregnet på baggrund af alle forsøgspersoner, pånær den forsøgsperson som undersøgelsen gælder (markeret med rød i graferne i nedenstående figur). Da vi arbejder med henholdsvis de originale data, og data korrigerede for den programfejl, der omtales i afsnit 7.6.1, forekommer hvert søjlediagram i to versioner, en for hvert datasæt. I Figur 45 viser venstre søjle de oprindelige data, og højre søjle de korrigerede data.

Forsøgsperson ser kun med ét øje	
	

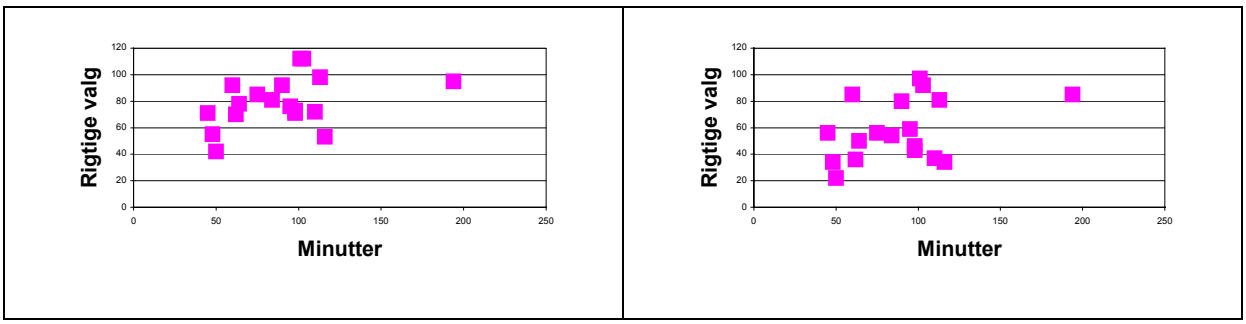
Forsøgsperson er farveblind	
	
Forsøgspersons syn er dårligere end normalt syn	
	
Forsøgspersons syn er bedre en normalt syn	
	
Forsøgspersons har valgt færre billeder end de andre forsøgspersoner	
	

Figur 45: Søjlediagrammer til sammenligning af forsøgspersoner.

Sammenligningen af disse forsøgspersoner med de andre forsøgspersoner kan nu gøres ved at se, om forsøgspersonernes søjler nogenlunde følger søjlerne for de andre forsøgspersoner. Et mere nøjagtigt mål for, om forsøgspersonerne skiller sig ud, kan naturligvis opnås ved udregning af korrelationen mellem forsøgspersonerne. Vi har dog bedømt, at den visuelle sammenligning er dækkende i denne situation.

Af diagrammerne kan det ses, at søjlerne for de første fem af ovenstående forsøgspersoner beskriver nogenlunde samme mønster som søjlerne for de resterende forsøgspersoner. For den femte forsøgsperson kan søjlerne fra de oprindelige data ikke umiddelbart vise, at der er nogen klar sammenhæng mellem forsøgspersonens valg, og de valg resten af forsøgspersonerne har truffet. I søjlerne for de korrigerede data ser det dog ud som de højeste søjler, hvor forsøgspersonen har valgt to rigtige, ligger i samme områder som de højeste søjler for de resterende forsøgspersoner. Ud fra ovenstående har vi bestemt ikke at udelukke resultater fra nogle af forsøgspersonerne i den videre databehandling.

Under forsøgene har StegoViewer registreret, hvor lang tid forsøgspersonerne har brugt på hvert enkelt sæt af billeder. Det er derfor nærliggende at undersøge, om antallet af rigtigt trufne valg er afhængig af den tid forsøgspersonerne har brugt. I Figur 46 er sammenhængen mellem rigtige valg og den brugte tid vist. Figuren viser to plot, hvor det venstre plot benytter de oprindelige data, og plottet til højre benytter de korrigerede data. Førsteaksen viser den brugte tid (målt i minutter), og andenaksen viser antallet af rigtigt trufne valg. Af figuren fremgår det, at de forsøgspersoner, der har brugt kortest tid på forsøget, også har haft en tendens til finde færre billeder, end de personer, der har brugt længere tid. Udover denne observation, er der ingen indlysende sammenhæng mellem den brugte tid og antallet af rigtigt trufne valg.



Figur 46: Sammenhæng mellem antal rigtigt trufne valg og brugt tid.

7.6.3 Bestemmelse af parameterværdier

Hvis et billede ændres, og det ændrede billede sammenlignes med det oprindelige billede, kan ændringen defineres som værende enten usynlig, skelnelig, synlig eller egentlig forstyrrende. Udfra personforsøgene har vi forsøgt at bestemme de parameterværdier, der skal indgå ved udledning af de perceptuelle tolerancer for en række billeder (se Tabel 7). Disse værdier er forsøgt valgt, så de giver den størst mulige ændring i de billeder, der har været anvendt i forsøget, således at denne ændring har været netop skelnelig. Bestemmelse af de parameterværdier, der giver netop skelnelig ændringer, er beskrevet herunder.

Lad V være en af de afprøvede parametre, der er uafhængig af andre afprøvede parametre, og som er afprøvet med parameterværdierne $v \in [v_{\min} \text{ K } v_{\max}]$, hvor højere værdier giver større ændringer i billederne. $n(v_i)$ er den funktion, der for værdien v_i af V, angiver antallet af gange forsøgspersoner har udvalgt billedet, der er genereret med værdien v_i, som et billede der skiller sig ud. Lad N angive antallet af forsøgspersoner. Da hvert billede genereret med v_i vises netop to gange for hver forsøgsperson, kan $n(v_i)$ ikke blive større end $2N$. For min-max-billederne defineres v_{minmax} som den højeste værdi af V, hvorom der gælder:

$$\forall v_i, v_{\min} \le v_i \le v_{minmax} : \\ n(v_i) \le N$$

Tilsvarende defineres v_{arb} for arbitrærbillederne som den højeste værdi af V, hvorom der gælder:

$$\forall v_i, v_{\min} \le v_i \le v_{arb} : \\ n(v_i) \le \frac{N}{2}$$

Den parameterværdi, v_{bedst}, som skal benyttes til den endelige bestemmelse af toleranceintervallerne, er da givet ved:

$$v_{bedst} = \min(v_{minmax}, v_{arb})$$

Vi har defineret v_{bedst} således, at ændringer med denne parameterværdi skulle være netop skelnelige. At $n(v_i)$ sammenlignes med N i forbindelse med min-max-billederne og $N/2$ når det gælder

arbitrærbillederne skyldes, at forsøgspersonerne formentlig vil have sværere ved at detektere forskelle i arbitrærbillederne.

Som med de uafhængige parametre skal de parvis afhængige parametre bestemmes, så de resulterer i de største ændringer i billedet, der netop er skelnelige. I de parvis afhænge parametre indgår en tærskelværdi (t) og en ændringsfaktor (f). Bestemmelse af sættet (t_{anv}, f_{anv}), der skal indgå i den endelige bestemmelse af toleranceværdier for billederne i Tabel 7 , sker ved følgende trin:

1. For min-max-billedet bestemmes først sættet (t_{val}, f_1), for den mindste faktorværdi f_1. Om (t_{val}, f_1) gælder:

$$\forall t_i, t_1 \le t_i \le t_{val} : \\ n(t_i, f_1) \le N$$

2. Dernæst bestemmes sættet (t_{val}, f_{val}) ud fra følgende:

$$\forall f_j, f_1 \le f_j \le f_{val} : \\ n(t_{val}, f_{val}) \le N$$

3. For arbitrærbilledet bestemmes (t'_{val}, f'_{val}) som (t_{val}, f_{val}); dog skal der her gælde:

$$n(t'_{val}, f'_{val}) \le \frac{N}{2}$$

4. Det sæt (t_{anv}, f_{anv}), der skal anvendes til den endelige bestemmelse af den perceptuelle tolerance, er da givet ved:

$$\forall (t_i, f_j), \\ t_1 \le t_i \le \min(t_{val}, t'_{val}), \\ f_1 \le f_j \le \min(f_{val}, f'_{val}) : \\ n(t_i, f_j) \le N \quad \text{(for min} - \text{max} - \text{billedet)} \\ n(t_i, f_j) \le \frac{N}{2} \quad \text{(for arbitrærbilledet)}$$

7.7 Resultater

I afsnit 7.6.3 er beskrevet, hvordan vi har bestemt de parameterværdier, vi har benyttet i forbindelse med den endelige bestemmelse af toleranceintervaller. Disse parametre, der er samlet i sættet *P*, består af:

$$P = \{p_{jnd}, p_{kant}, p_{H_v}, p_{H_h}, p_{assim,vægtet}, p_{assim,middel}\}$$

For de parametre, hvor der kun skal bestemmes en enkelt parameterværdi, er dette gjort ud fra grafer, der beskriver antallet af rigtigt valgte billeder som funktion af parameterværdierne. Dette betyder, at vi ikke kun har valgt at fastsætte parameterværdier, som vi har afprøvet for, men gerne andre parameterværdier, som graferne kan godtgøre for.

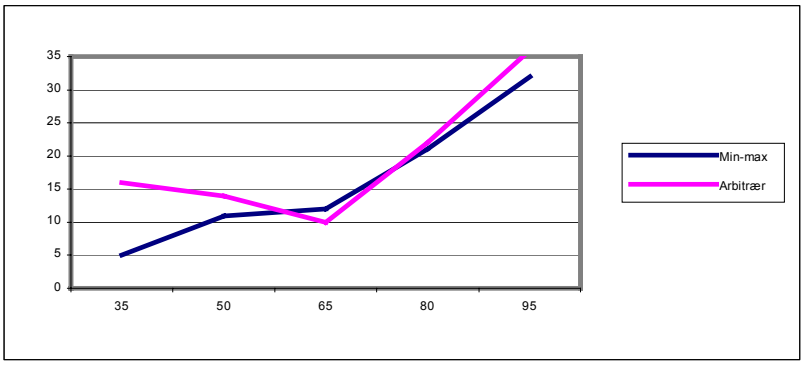
For de parametre, hvor vi har bestemt et par af parameterværdier, er dette blevet gjort ud fra antallet af gange, billeder baseret på de pågældende par, er blevet valgt. Parameterværdierne er altså ikke bestemt ud fra grafer. Her er graferne altså udelukkende medtaget for at illustrere, hvordan værdierne er fordelt.

Alle parametre er illustreret med en graf samt en tabel, der indeholder de værdier, som ligger til grund for grafen. Alle værdier og grafer kan desuden findes på den medfølgende cd-rom. For alle parametre har vi valgt at bestemme de anvendte parameterværdier ud fra de korrigerede data. Vi har dog valgt at vise grafer og tabeller for både de oprindelig og de korrigerede data.

7.7.1 Just noticeable distortion

Herunder viser grafen for min-max-billederne tydeligt, at jo større en ændring i min-max-billederne er, desto flere gange er det ændrede billede blevet valgt. For arbitrærbillederne ser tendensen dog ikke ud til at være ligeså tydelig. For de sidste tre parameterværdier følger kurven for arbitrærbillederne udemærket kurven for min-max-billederne; men for de første to parameterværdier i arbitrærbillederne, ser det ud til, at ændringerne baseret på disse værdier har været lettere at se, end ændringerne baseret på den tredie værdi.

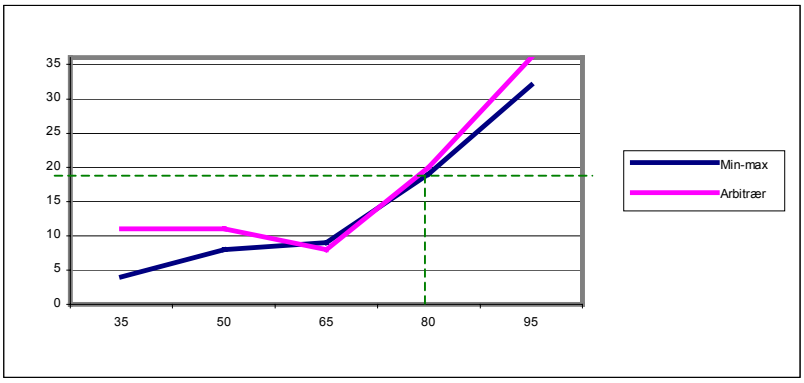
Værdi	Min-max	Arbitrær
35%	5	16
50%	11	14
65%	12	10
80%	21	22
95%	32	36



Tabel 9: JND – oprindelige data.

I graferne for de korrigerede data viser grafen for min-max-billederne samme tendens som for de oprindelige data. For arbitrærbillederne er tendensen for de parameterværdier, der giver den mindste ændring, lidt mindre tydelig end i grafen for de oprindelige data, selvom den stadig er der.

Værdi	Min-max	Arbitrær
35%	4	11
50%	8	11
65%	9	8
80%	19	20
95%	32	36



Tabel 10: JND – korrigerede data.

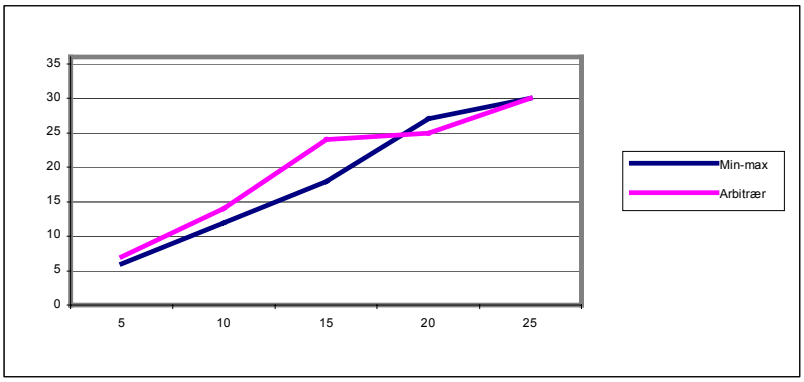
Da grafen for arbitrærbillederne udviser en anden opførsel, end vi kunne forvente, har vi valgt alene at bestemme p_{jnd} ud fra grafen for min-max-billederne. Den bestemmes her til:

$$p_{jnd} = 79$$

7.7.2 Kantpixels

I begge grafer for de oprindelige data ses en tydelig tendens: Jo større ændring der er i et billede, desto flere gange er billedet blevet valgt. Hvad der kan undre lidt ved graferne er, at det ikke ser ud til, at der er særlig forskel på, om ændringerne er sket i et min-max-billede, eller om de er sket i et arbitrærbillede.

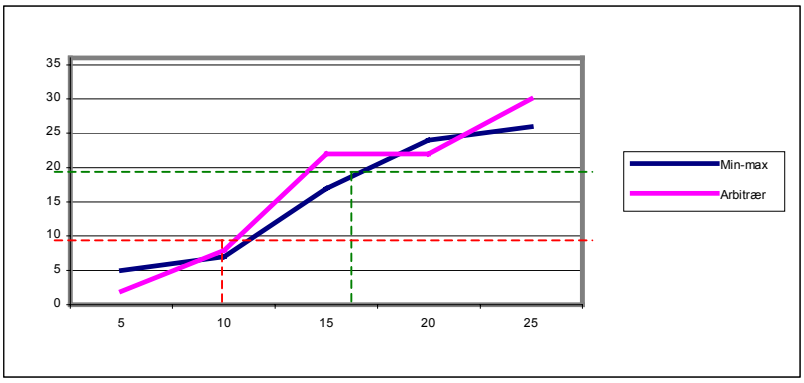
Værdi	Min-max	Arbitrær
5%	6	7
10%	12	14
15%	18	24
20%	27	25
25%	30	30



Tabel 11: Kantpixels – oprindelige data.

Graferne for de korrigerede data viser samme tendenser som graferne for de oprindelige data: Begge grafer er monotont stigende med flere valgte billeder når der er større ændringer i billederne.

Værdi	Min-max	Arbitrær
5%	5	2
10%	7	8
15%	17	22
20%	24	22
25%	26	30



Tabel 12: Kantpixels – korrigerede data.

I følge ovenstående grafer kan der bestemmes to værdier for p_{kant}. Som beskrevet i afsnit 7.6.3 vælger vi den mindste af disse værdier. Parameterværdien bestemmes derfor til:

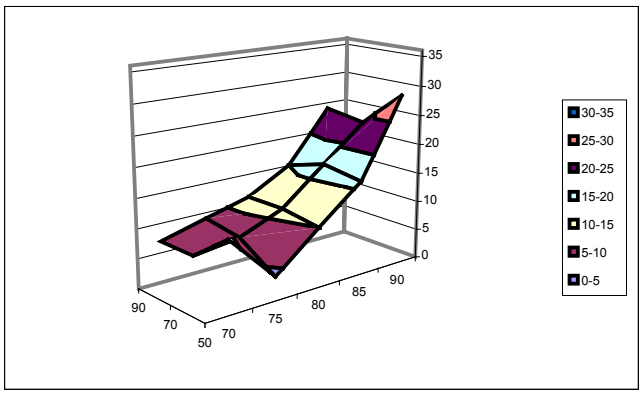
$$p_{kant} = 10\%$$

7.7.3 Homogenitet i value

I grafen for de oprindelige værdier, taget for min-max-billederne, ses afhængigheden af ændringsfaktoren ikke særligt tydeligt. For nogle tærskelværdier stiger antallet af valgte billeder, når ændringsfaktoren stiger, mens antallet for andre tærskelværdier falder. Afhængigheden af tærskel-

værdier er derimod ganske tydelig: For alle tre ændringsfaktorer ses en tydelig tendens til, at højere tærskelværdi, der giver flere ændrede pixels, resulterer i, at flere billeder bliver valgt.

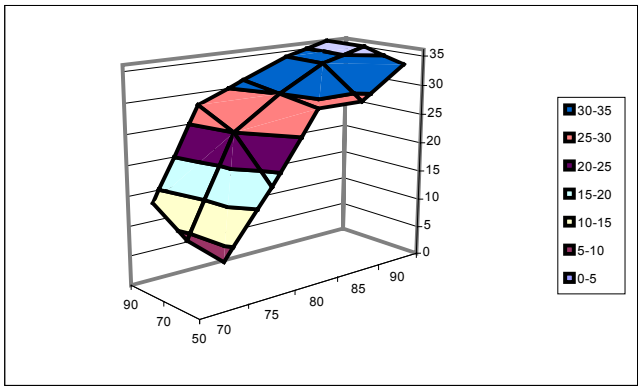
Tærskel	50%	70%	90%
70%	12	7	7
75%	4	8	9
80%	10	11	11
85%	16	17	15
90%	29	23	24



Tabel 13: Homogenitet i value, min-max-billeder – oprindelige data.

For arbitrærbillederne ses afhængigheden af både tærskelværdi og ændringsfaktor lidt tydeligere end tilfældet var med min-max-billederne: For alle tærskelværdier giver en øget ændringsfaktor også et øget antal valgte billeder. Igen er den mest tydelige afhængighed dog afhængigheden af tærskelværdien. For to af de tre ændringsfaktorer har alle forsøgspersoner valgt begge de billeder, der er baseret på den højeste tærskelværdi.

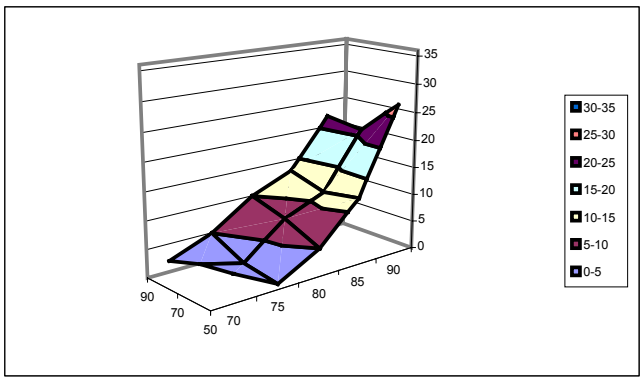
Tærskel	50%	70%	90%
70%	8	9	13
75%	18	25	28
80%	29	30	31
85%	29	34	34
90%	34	36	36



Tabel 14: Homogenitet i value, arbitrærbilleder – oprindelige data.

For de korrigerede data er der ikke den store forskel i beskrivelsen af grafen for min-max-billederne. Det mest tydelige er, at grafen er sænket i forhold til grafen for de oprindelige data, da der er færre rigtige valg i de korrigerede data.

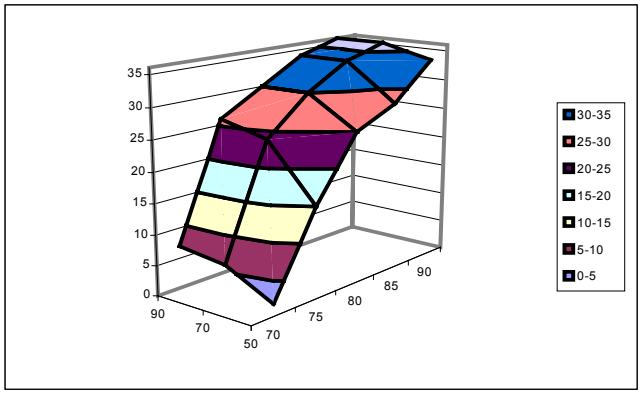
Tærskel	50%	70%	90%
70%	5	4	2
75%	1	2	5
80%	5	8	10
85%	12	11	13
90%	27	21	22



Tabel 15: Homogenitet i value, min-max-billeder – korrigerede data.

For arbitrærbillederne skiller grafen for de korrigerede data sig heller ikke meget fra grafen for de oprindelige data. Dog er afhængigheden af tærskelværdierne blevet endnu tydeligere, da den lave ende af grafen er blevet endnu lavere, hvilket gør grafen mere stejl.

Tærskel	50%	70%	90%
70%	2	6	7
75%	15	24	26
80%	25	30	30
85%	28	34	34
90%	34	36	36



Tabel 16: Homogenitet i value, arbitrærbilleder – korrigerede data.

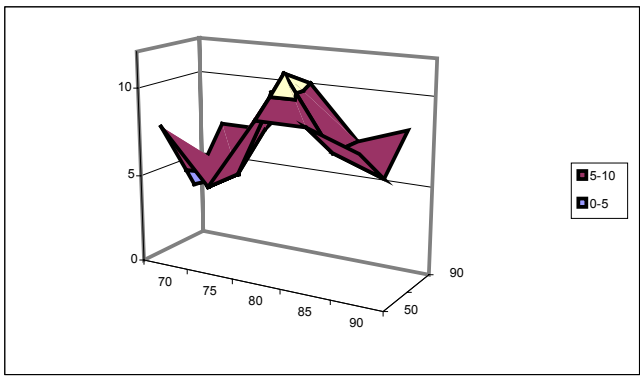
Ved at følge beskrivelsen i afsnit 7.6.3 bestemmes p_{H_v} til:

$$p_{H_v} = (70\%, 90\%)$$

7.7.4 Homogenitet i hue

Grafen for min-max-billederne, baseret på de oprindelige data, ser ikke ud til at kunne give nogle anvendelige resultater. De givne resultater ser mest af alt ud til at stamme fra støj i forsøget.

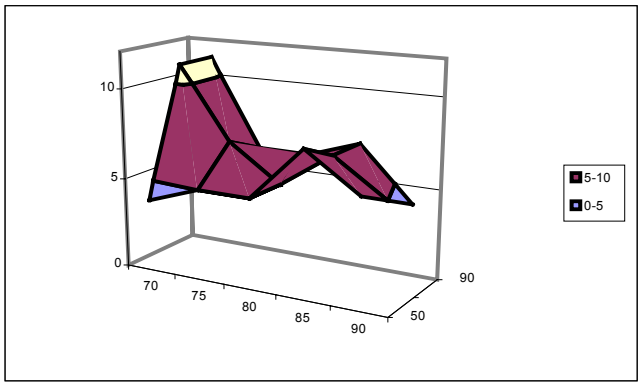
Tærskel	50%	70%	90%
70%	8	4	7
75%	5	5	7
80%	9	11	10
85%	9	7	7
90%	8	6	8



Tabel 17: Homogenitet i hue, min-max-billeder – oprindelige data.

Som for grafen for min-max-billederne ser også grafen for arbitrærbillederne, baseret på de oprindelige data, ud til at være baseret på støj i forsøgsresultaterne.

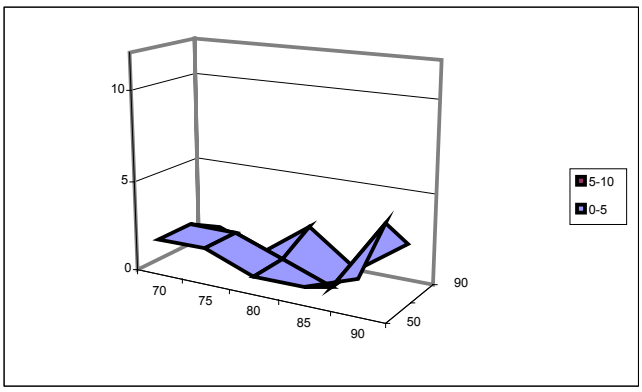
Tærskel	50%	70%	90%
70%	8	4	7
75%	5	5	7
80%	9	11	10
85%	9	7	7
90%	8	6	8



Tabel 18: Homogenitet i hue, arbitrærbilleder – oprindelige data.

Betragtes grafen for min-max-billeder, baseret på de korrigerede data, ses at ingen af billederne er blevet valgt mere end 18 gange. I følge afsnit 7.6.3 bestemmes p_{H_0} på baggrund af de sæt, for hvilke antallet af billedvalg netop overstiger 18.

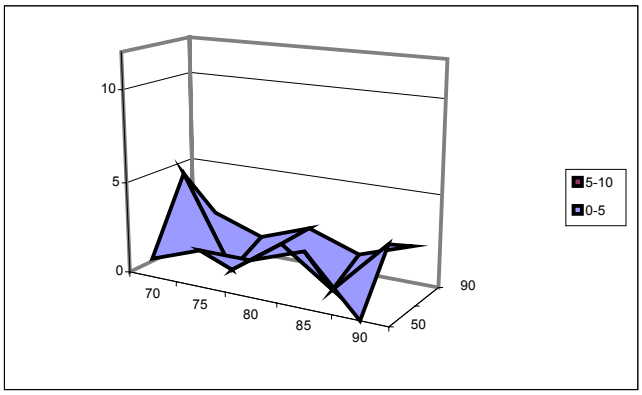
Tærskel	50%	70%	90%
70%	2	2	1
75%	2	2	0
80%	1	1	2
85%	1	0	0
90%	2	4	2



Tabel 19: Homogenitet i hue, min-max-billeder – korrigerede data.

Ingen af arbitrærbillederne er, ifølge de korrigerede data, blevet valgt mere end de 9 gange, der bestemmer p_{H_b}.

Tærskel	50%	70%	90%
70%	1	5	2
75%	2	0	1
80%	2	2	2
85%	3	0	1
90%	0	3	2



Tabel 20: Homogenitet i hue, arbitrærbilleder – korrigerede data.

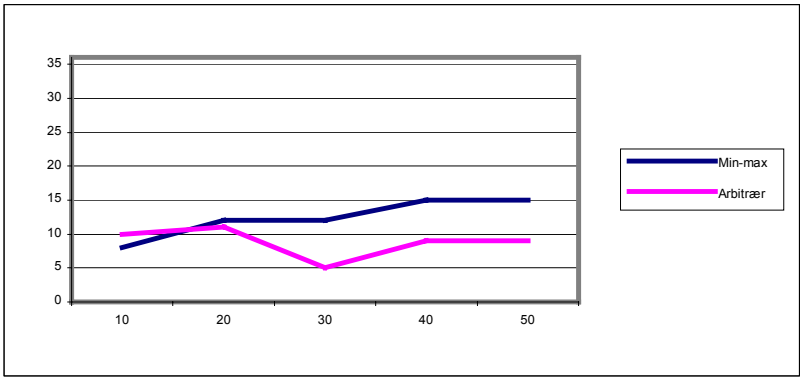
Da ingen af graferne for henholdsvis min-max-billeder og arbitrærbilleder kan bestemme en værdi for parret p_{H_b}, bestemmes begge værdier i parret som de værdier, inden for de afprøvede intervaller, der vil give de største ændringer i et billede. Parret gives altså værdierne:

$$p_{H_b} = (90\%, 90\%)$$

7.7.5 Assimilering – vægtet

Grafen for min-max-billederne, baseret på de oprindelige data, er monotont stigende med større ændringer i billederne. Grafen for arbitrærbillederne viser desværre ikke denne tendens.

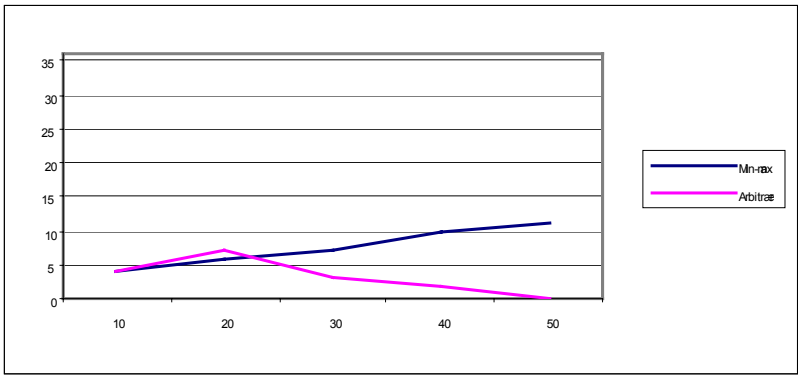
Værdi	Min-max	Arbitrær
10%	8	10
20%	12	11
30%	12	5
40%	15	9
50%	15	9



Tabel 21: Assimilering, vægtet – oprindelige data.

For de korrigerede data ser grafen for min-max-billederne stadig ud til at være monotont stigende. Desværre kommer grafen på intet tidspunkt over de 18 rigtigt valgte billeder, der bestemmer $p_{assim,vægtet}$. Grafen for arbitrærbillederne ser ikke bedre ud: Det billede der er ændret mest, er slet ikke blevet valgt.

Værdi	Min-max	Arbitrær
10%	4	4
20%	6	7
30%	7	3
40%	10	2
50%	11	0



Tabel 22: Assimilering, vægtet – korrigerede data.

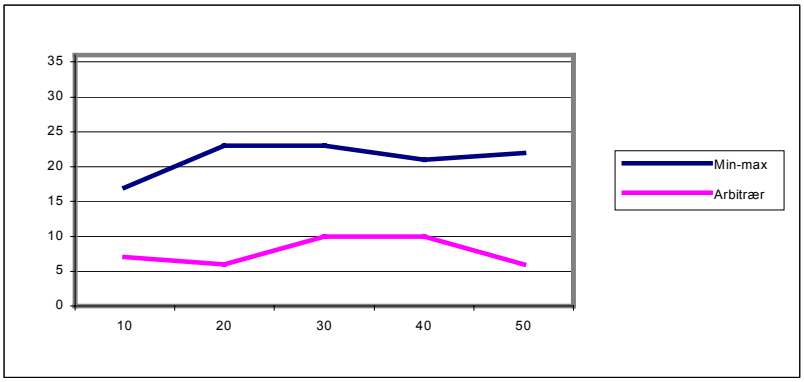
Da hverken grafen for min-max-billederne eller grafen for arbitrærbillederne ser ud til at kunne bruges til bestemmelse af $p_{assim,vægtet}$ sættes denne til den største parameterværdi, der er blevet afprøvet for:

$$p_{assim,vægtet} = 50\%$$

7.7.6 Assimilering – middelværdi

Ingen af graferne for de oprindelige data opfører sig som forventet. Grafen for min-max-billederne har en svag ændring opad i antallet af valgte billeder, når ændringen stiger, mens grafen for arbitrærbillederne viser den omvendte tendens.

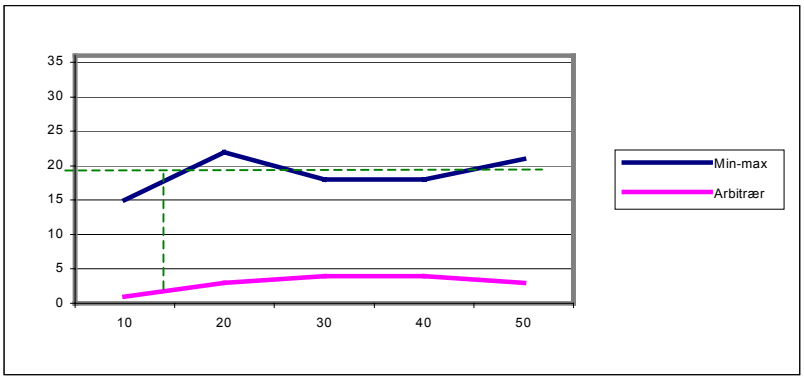
Værdi	Min-max	Arbitrær
10%	17	7
20%	23	6
30%	23	10
40%	21	10
50%	22	6



Tabel 23: Assimilering, middelværdi – oprindelige data.

Korrigering af data har sænket begge graferne. Desværre er der stadig ingen af graferne der viser den forventede tendens, hvor øget ændring resulterer i flere rigtigt valgte billeder.

Værdi	Min-max	Arbitrær
10%	15	1
20%	22	3
30%	18	4
40%	18	4
50%	21	3



Tabel 24: Assimilering, middelværdi – korrigerede data.

En bestemmelse af $p_{assim,middel}$ vil, ifølge afsnit 7.6.3, være den højeste værdi, hvor alle tidligere værdier for min-max-billeder har resulteret i mindre end 18 valgte billeder. I følge grafen giver dette:

$$p_{assim,middel} = 15\%$$

7.8 Konklusion på personforsøg

Forud for personforsøgene havde vi en tese om, at jo mere vi ændrede i de billeder, der indgik i forsøget, jo flere forsøgspersoner ville kunne se, at billederne var ændret. Denne tese viste sig kun at holde, uden undtagelser, ved ændringer af kantpixels. For just noticeable distortion og den

vægtede assimilering holdt tesen kun for min-max-billederne, mens den for homogenitetn i value kun holdt for arbitrærbillederne. For min-max-billederne, for homogeniteten i value, er sammenhængen med ændringer i pixel, for fastholdt tærskelværdi, ikke tydelig. Derimod holder tesen for ændringer i tærskelværdien, der også resulterer i ændringer i billedet. For homogeniteten i hue holder tesen slet ikke, og resultaterne her ser nærmest tilfældige ud. Efter forsøgene kan vi altså konkludere, at tesen ikke nødvendigvis er usand, men at der nok skal både flere parameterværdier, og mere parameterspecifikke forsøg til at dække disse parameterværdier, før tesen kan bevises.

En anden tese var, at der ville være stor forskel på antallet af gange et min-max-billede og et arbitrærbillede blev valgt, for samme parameterværdi. Denne tese holder for assimileringen baseret på middelværdien, og ser i øvrigt ikke ud til at holde for de andre filtre. Her kan vi konkludere, at der tilsyneladende ikke er belæg for at behandle min-max-billederne anderledes end arbitrærbillederne ved bestemmelsen af de parameterværdier, der skal anvendes ved udregningen af den perceptuelle tolerance. Om der stadig er grund til udarbejdelse af to slags billeder er lidt mindre klart. Da der er stor forskel på opfattelsen af min-max-billederne og arbitrærbillederne for assimileringen baseret på middelværdien, kunne noget tyde på, at nogle effekter er lettere at finde i min-max-billederne end i arbitrærbillederne, mens andre effekter skjules lige godt i begge slags billeder.

Den programfejl, der er beskrevet i afsnit 7.6.1 gjorde databehandlingen en hel del mere besværlig, da vi ikke ved hvad de "rigtige" data var. De oprindelige data var baseret på for mange rigtigt valgte billeder, og de korrigerede data fik formentlig for få. Vi lavede en grafisk brugergrænseflade, så vi for hvert valg, hvor koordinaterne på et billede gik igen fra tidligere valg, kunne udvælge hvilke af de valgte billeder, der var falsk positive. Vi undlod dog hurtigt at bruge denne funktion, da vi dels ikke havde nødvendige forudsætninger for at kunne vælge, hvilke billeder der var forkert valgt, og dels da datasættet baseret på disse valg formentlig heller ikke ville blive korrekt.

Som beskrevet i afsnit 7.6.3 skilte en del af forsøgspersonerne sig ud fra de andre forsøgspersoner. Vi havde med vilje valgt forsøgspersonerne så dette ville være tilfældet. På denne måde kunne forsøget omfatte en forholdsvis bred forsøgsgruppe, til trods for, at gruppen ikke var særlig stor. Da vi mener, at ingen af forsøgspersonerne skilte sig så meget ud, at deres resultater ikke stemte overens med de resterende forsøgspersoners, har vi valgt at lade alle forsøgspersoners resultater indgå i databehandlingen. Hvis vi havde valgt at udelade nogle af forsøgspersonernes resultater, kunne dette måske have givet andre resultater en de beskrevet i afsnit 7.7, så her er mulighed for videre undersøgelser.

På grund af størrelsen af det resulterende dataset har vi ment, at vi kunne nøjes med at underkaste resultaterne en mindre statistisk analyse. Denne analyse er fortrinsvis gået på at aflæse grafer, og udvælge resultater fra forsøgene. Til et større forsøg ville disse metoder ikke være tilstrækkelige, da vi f.eks. ikke kan bedømme, hvor meget vi kan stole på de data vi arbejder med. Behovene i forbindelse med den analyse vi har udført, har dog ikke været særligt store, hvorfor vi mener, at den udførte databehandling har været tilstrækkelig.

Da bestemmelsen af et billedes perceptuelle tolerance er afhængig af resultaterne af en række forskellige filtres kumulative effekt, er det ikke givet, at resultaterne af flere filtre, eller undersøgelse af flere parametre, ville gøre større effekt på tolerancen. I kombinationsmetoden giver selv ganske små forskelle i tolerancen dog store forskelle i den steganografiske kapacitet, så flere filtre, eller en bedre optimering af de anvendte filtre, ville måske her kunne have en stor effekt.

8. Behandling af spread spectrum

Som nævnt i afsnit 4.4 kan bestemmelse af den perceptuelle tolerance bruges til at forbedre metoder, der ændrer på billeder således, at disse ændringer kun foretages, så de ikke kan ses. Spread spectrum (se bl.a. [Cox et al 95]) er en metode til at gemme vandmærker i bl.a. billeder, og som ved andre vandmærkningsmetoder ændrers det billede metoden anvendes på. Vi har derfor valgt at undersøge, om spread-spectrum kan optimeres efter den perceptuelle tolerance på en sådan måde, at der enten kan gemmes flere data ("et større vandmærke") i et givent billede, eller så den mængde data, der gemmes, bliver gemt på en sådan måde, at billedets perceptuelle tolerance ikke overskrides.

8.1 Metode

Ved steganografi på kommunikationslinier er spread spectrum en metode til at camouflere signaler. Dette gøres ved at sende et smalt signal (det camouflerede) over en bredere båndbredde, således, at den energi det smalle signal har ved en given frekvens er så lille, at den kan tilskrives støj. Spread spectrum i vandmærker benytter den samme metode, hvor et billede konverteres til en frekvensrepræsentation, som skal udgøre værtsmediet. Ved camoufleringen ændres frekvenskoefficienterne i henhold til det vandmærke, der skal camoufleres. I [Cox et al 95] beskrives en implementering af spread spectrum på billeder, som vi har valgt at bygge videre på. Her benyttes Fourier-transformation til konvertering til frekvensrummet. Også [O'Ruanaidh et al] benytter Fourier-transformationen, der som resultat giver komplekse koefficienter. Her benyttes en ændring i koefficienternes fase fremfor deres størrelse, da eksperimenter har vist, at disse ændringer har mindst synlig effekt.

Vi har valgt at benytte den mere beregningssimple Walsh-transformation (jf. [Gonzales&Woods 93]) i vor implementering, hvor vi desuden har valgt at vi kun benytter value-værdier bestemt ved HSV-modellen. Walsh-transformationen er givet ved:

$$W(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \prod_{i=0}^{n-1} (-1)^{[b_i(x) b_{n-1-i}(u) + b_i(y) b_{n-1-i}(v)]}$$

I ovenstående er $W(u, v)$ Walsh-repræsentationen af koefficienten ved en given frekvens, $f(x, y)$ er value-værdien for pixelen på plads (x, y) og $b_k(z)$ er den k'te bit i den binære repræsentation af z.

Den inverse Walsh-transformation er givet ved:

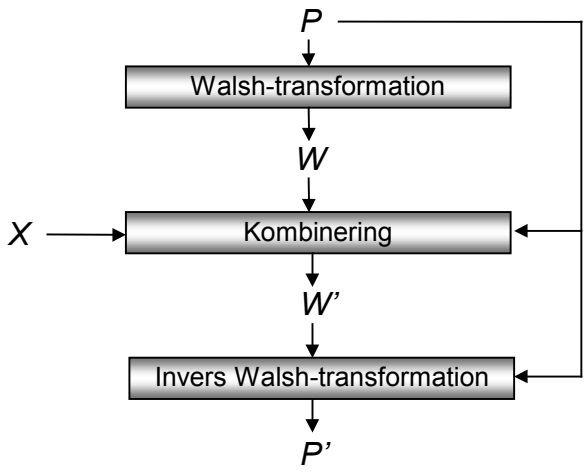
$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}W(u,v)\prod_{i=0}^{n-1}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

Som det kan ses af ovenstående, er det muligt at benytte samme algoritme til både almindelig og invers Walsh-transformation, da disse transformationer i al væsentlighed foregår på samme måde.

I Figur 47 ses på skematisk form, hvordan indsættelse af vandmærker med spread spectrum og Walsh-transformationen foregår. Et vandmærke konstrueres som en sekvens af reelle tal $X = x_1, \ldots, x_k$, der så kombineres med lige så mange Walsh-koefficienter, $W = w_1, \ldots, w_k$, der er bestemt på baggrund af et billede *P*. I [Cox et al 95] angives tre måder at kombinere *X* og *W* på:

$$w'_i = w_i + \alpha x_i$$
$$w'_i = w_i(1+\alpha x_i)$$
$$w'_i = w_i e^{\alpha x_i}$$

Hvis *w*-værdierne varierer meget, er den første metode ikke velegnet. F.eks. vil det ikke gøre den store forskel, hvis der lægges 100 til $w_i=10^6$, mens 100 lagt til $w_i=10$ kunne formodes at ødelægge billedet. Ingen af de sidste metoder er følsomme over for den slags varierende *w*-værdier. Da metode nummer to umiddelbart er den, der kræver færrest regneopreationer, vælger vi at benytte den. I *W* erstattes de *k* Walsh-koefficienter med de *k* kombinationer af w_i og x_i, og det resulterende sæt af Walsh-koefficienter kaldes *W'*. *W'* udsættes herefter for en invers Walsh-transformation, og resultatet er et vandmærket billede P'.



Figur 47: Indsætning af vandmærke med spread spectrum.

I [Cox et al 95] beskrives, udover det ovenfor nævnte, også hvordan et vandmærke er i stand til at modstå en række angreb på billedet. Alt dette skulle ikke blive ændret væsentligt af vores optimering, hvorfor vi ikke vil beskrive det yderligere her.

8.2 Optimering

Det er ikke umiddelbart ligetil at vælge en parameter α til brug i ovenstående beregninger, da det ikke er til at forudsige, hvilken værdi der kan bruges på alle w_i i et givent billede, uden at billedet synligt forringes: Jo større α-værdi der benyttes, desto bedre er vandmærket beskyttet, og jo større er sandsynligheden for, at vandmærkets tilstedeværelse afsløres. I [Cox et al 95] har man valgt at sætte α til 0,1, men man er opmærksom på problemet, og giver forskellige løsningsmetoder.

At vælge et fast sæt $A = \alpha_1, \ldots, \alpha_k$ gældende for hele billedet strider dog mod vor antagelse om, at ikke alle dele af et billede opfattes ligeligt. For at imødekomme dette, kan billedet opdeles i mindre områder, hvorefter der kan bestemmes et $A = \alpha_1, \ldots, \alpha_k$ for hvert af disse områder. Da Walsh-transformationen her kræver, at der behandles områder af størrelsen $N=2^n$, har vi valgt at behandle områder med en sidelængde på 8 pixels. De w-værdier der ændres i hvert 8×8 område er givet ved følgende (u, v)-par:

(1, 0), (0, 1), (0, 2), (2, 0), (2, 1), (1, 2), (3, 1), (1, 3)

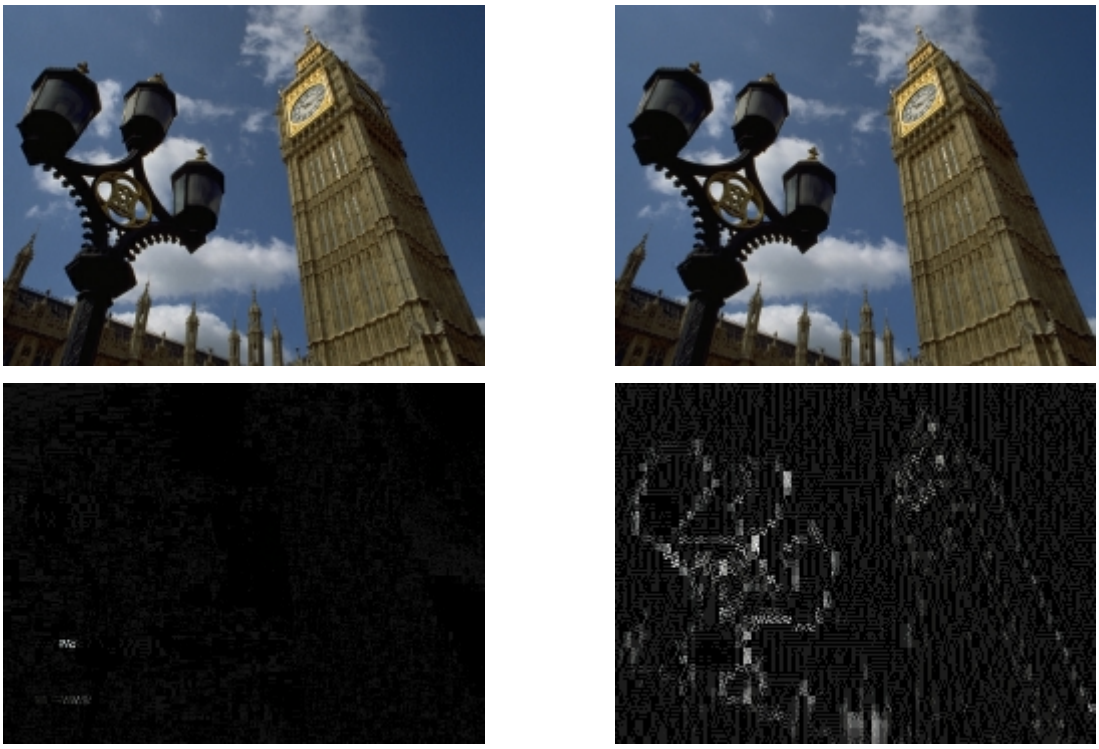
Disse w-værdier kombineres med x-værdier, der skiftevis består af værdien 0 og værdien 1.

Desuden indfører vi en α-værdi for hver af de k w-værdier, hvor k altså højst kan blive $8^2 = 64$. At bestemme hvilke værdier $A = \alpha_1, \ldots, \alpha_k$ skal antage kan så komme ud på en undersøgelse af P.

Vi har valgt at lade spread spectrum-metoden gennemløbe en løkke, hvor α-værdierne for hvert gennemløb tilrettes, så de får den største værdi, der ikke resulterer i et billede, hvis værdier ligger uden for toleranceintervallerne. Denne løkke gennemløbes til forskellen på α-værdierne for et gennemløb og α-værdierne for det tidligere gennemløb, ikke overskrider en given værdi.

8.3 Resultater

Vi har implementeret ovenstående vandmærkningsalgoritme, og har sammenlignet resultaterne fra ét billede, der har fået indlagt vandmærker på to forskellige måde: med alle α_i sat til værdien 0,1 (som i [Cox et al 95]), og hvor $A = \alpha_1, \ldots, \alpha_k$ er bestemt som beskrevet i afsnit 8.2. Figur 48 viser forskellen på de to stegobilleder. Billedet til venstre er ændret med alle α_i sat til værdien 0,1, mens α-værdierne i billedet til højre varierer.



(a) (b)

Figur 48: Resultat af optimering af spread spectrum. (a) viser camouflering med fast α (0,1); i (b) er anvendt variabel α.

Da billede (b) med variable værdier for $A = \alpha_1, \ldots, \alpha_k$ er ændret, så ændringerne ikke skulle overskride den perceptuelle tolerance, skulle det ikke være umiddelbart muligt at se forskel på de to viste billeder. Tages forskellen mellem hvert af disse billeder, og det originale billede, er det dog

muligt at se forskel på de to stegobilleder. De nederste to billeder i figuren viser forskellene mellem stegobillederne og værtsbilledet. Denne forskel er forstærket ved at strække værdierne i begge billeder. Strækket er foretaget med Paint Shop Pro 6.02. Det skulle nu være tydeligt at se, at billedet til højre er ændret mere end billedet til venstre.

Hvis spread spectrum-metoder benytter sig af den perceptuelle tolerance, kan det altså være muligt at ændre mere på stegobilledet end uden brug af den perceptuelle tolerance. Dette kan muligvis benyttes til større eller mere robuste vandmærker.

9. Konklusion

Denne rapport har behandlet emnet steganografi ved først at give en introduktion til emnet, og derefter behandle steganografi i digitale billeder baseret på menneskesynet. I den forbindelse har vi introduceret begreberne perceptuel tolerance, der angiver, hvor meget et billede kan ændres, samt steganografisk kapacitet, der er afhængig af dels billedet og dels af den steganografiske metode, og som fortæller, hvor store mængder data, der kan camoufleres i et billede.

Litteraturen angiver mange steganografiske metoder, der delvis er baseret på HVS' behandling af det der ses. Vi mener dog, at afsnit 6 viser, at en grundigere undersøgelse af HVS' behandling af billeder vil kunne resultere i en større steganografisk kapacitet for disse metoder, samt en bedre udnyttelse af kapacitetens spatielle fordeling. I afsnit 5 beskrives en del af de synsfænomener der gør, at man kan ændre på et billede, uden at ændre perceptionen af det. Kun ganske få af disse fænomener benyttes til bestemmelse af billedets perceptuelle tolerance i afsnit 6. Ved en grundigere undersøgelse af HVS, f.eks. en bestemmelse af de områder i et billede, der slet ikke fokuseres på, vil man formentlig kunne opnå endnu større perceptuelle tolerancer, end de der er bestemt for billederne i Tabel 7.

Alle de billeder, der vises i Tabel 7, har været behandlet med samme parametersæt. Hvis billederne blev kategoriseret, og der for hver kategori blev bestemt et parametersæt, ville der formentlig kunne opnås bedre resultater, hvor den perceptuelle tolerance var tilpasset billedets karakteristik. Således ville f.eks. stregtegninger blive udsat for en mindre JPEG-komprimering end naturlige billeder.

Vi har gennemført en række personforsøg, for at fastslå, hvordan et antal filtre, der efterligner HVS' måde at behandle synsindtryk på, skal indstilles. Ud fra disse forsøg har vi indstillet et antal parameterværdier i filtrene, der så er blevet anvendt på billederne i Tabel 7. Desværre viste det sig, at forsøgsresultaterne ikke helt overholdt de teser, vi havde opstillet inden forsøgene, og der er derfor en del usikkerhed på de bestemte parameterværdier. Da resultaterne fra forsøgene, som beskrevet i afsnit 7.1, kun skal betragtes som vejledende, gør usikkerheden på parameterværdierne dog ikke, at vi ikke kan benytte dem til bestemmelse af den perceptuelle tolerance for en række billeder.

Baseret på den perceptuelle tolerance og kombinationsmetoden kan der camoufleres ganske store mængder data i de anvendte billeder. Sammenlignet med offentliggjorte metoder anvendt til

vandmærkning af billeder (f.eks. de metoder, der beskrives i [Bender et al]) giver kombinationsmetoden steganografiske kapaciteter, der er mange størrelsesordner større. Det er dog her værd at bemærke, at kombinationsmetoden er særdeles uanvendelig som vandmærkningsalgoritme, da den bl.a. ikke er robust over for ret mange angreb på billederne.

Til generering af de billeder, der har været anvendt til personforsøgene, samt ind- og afkodning af data, har vi fremstillet programmet StegoShop. Programmet er skrevet i Java 2, der har vist sig at være velegnet til behandling af billeder. At programmet er skrevet i Java gør naturligvis, at det tager længere tid at udføre billedgenerering og –behandling end hvis programmet var skrevet i et ikke-fortolket sprog som f.eks. C/C++. Til gengæld er Java platformsuafhængigt, hvilket gør, at StegoShop kan køres på en hvilken som helst computer, der er forsynet med en Java-fortolker. Kildekoden til StegoShop kan findes på den medfølgende cd-rom.

Vi mener, at man kan udnytte den perceptuelle tolerance til andet end camouflering af data med kombinationsmodellen. For at vise dette har vi valgt at implementere og optimere en spread spectrum algoritme. Optimeringen består i at lade en parameter brugt til indkodningen være bestemt af den lokale perceptuelle tolerance, således at der ikke bliver perceptuel forskel på værtsbilledet og det ændrede billede. Denne parameter har i tidligere beskrevne metoder været ensartet for et helt billede ad gangen, og denne fastsættelse har derfor ikke udnyttet den forskel, der kan ligge i forskellige del af billedet. Vores undersøgelse har vist, at der kan varieres ganske meget på parameteren afhængig af billedets lokale sammensætning.

Den overordnede konklusion på denne rapport er, at anvendelse af en model for human perception med fordel kan anvendes til bestemmelse af den perceptuelle perception for et givent billede. Den perceptuelle perception kan så anvendes til bl.a. camouflering af data med en metode som kombinationsmetoden, eller til optimering af vandmærkningsmetoder, således at der kan indsættes robuste vandmærker, uden at billedet ændres perceptuelt.

10. Efterskrift

Vort arbejdet med denne rapport har strakt sig over mere end tre år. Dette har bl.a. betydet, at den opmærksomhed vi på verdensplan har oplevet omkring emnerne steganografi og vandmærkning, er steget betydeligt. Vi begyndte med at have ganske få artikler om emnet, og havde da ikke held med at finde flere. Efterhånden som arbejdet skred frem, er der dog kommet mange artikler om emnerne, og da vi mod afslutningen af forløbet blev tilmeldt en mailingliste, der specifikt omhandler vandmærkning[15], fik vi et indtryk af, at emnet efterhånden var kommet til optage mange mennesker over hele verden.

Den stigende interesse har naturligvis ført til belysning af mange forskellige aspekter af emnerne steganografi og vandmærkning. Vi har derfor undret os over, at der tilsyneladende stadig kun er ganske få, der interesserer sig for det direkte samspil mellem HVS og størrelsen af ikke-synlige ændringer i et billede. Der har været ganske få tiltag indenfor denne tilgangsvinkel til steganografien, og de der har været, har næsten alle sammen beskæftiget sig med begrænsninger i frekvensdomænet. Vi har derfor måttet finde størstedelen af den litteratur, der omhandler HVS bearbejdelse af billeder, inden for fysiologi og psykofysik. Da dette er emneområder, vi ikke tidligere har arbejdet med, og som ligger forholdsvis fjernt fra den datalogiske sfære, har tilegnelsen af dette stof ikke altid været lige nem. Vi er derfor særdeles taknemmelige over for den hjælp, vi har fået med disse emneområder.

Den omtalte kombinationsmetode benytter sig af camouflering i det spatielle domæne. Det kunne umiddelbart være et forsøg værd at undersøge, om metoden også egnede sig til camouflering i frekvensdomænet, hvor kombinationer af tilladte koefficienter kan benyttes. Da disse koefficienter tilsyneladende har lettere ved at overleve angreb på et billede, end billedets pixelkomponenter har, er det ikke umuligt, at metoden kan benyttes til lettere former for vandmærkning. Da dette emne ikke er umiddelbart indenfor rammerne af dette speciale, har vi dog valgt ikke at undersøge det nærmere.

Et andet emne, der naturligvis kan undersøges meget mere, er en bedre fastsættelse af parameterværdier for de anvendte filtre, samt en bestemmelse af flere, og bedre, filtre. De foretagne personforsøg er kun at betragte som vejledende, og er derfor langt fra dækkende for en bestemmelse af de mulige filtre.

---

[15] Se '`ltssg3.epfl.ch:1248/kutter/watermarking`'.

I arbejdsprocessen har vi kun fundet ganske få firmaer, der kommercielt benytter sig af steganografi eller vandmærkning. Anvendelsen har oftest været begrænset til vandmærkning af billeder i kommercielle billedbehandlingsprogrammer som Paint Shop Pro eller Photoshop. Et enkelt firma har dog beskæftiget sig med vandmærkning af medicinske billeder for et engelsk hospital. Vores forventninger til dette emneområder er dog, at man i den kommende tid vil se flere tiltag på den kommercielle front også.

Litteratur

Steganografi og vandmærkning

[Anderson] Ross Anderson, *Stretching the Limits of Steganography*

[Bender et al] Walter Bender, Daniel Gruhl, Norishige Morimoto, *Techniques for data Hiding*

[Berghel&O'Gorman] Hal Berghel, Lawrence O'Gorman, *Digital Watermarking*

[Blundo&Santis 98] Carlo Blundo, Alfredo de Santis, *Visual Cryptography Schemes with Perfect Reconstruction of Black Pixels*, Computers & Graphics, Vol. 22, No. 4, 1998

[Cox et al 95] Ingemar J. Cox, Joe Kilian, Tom Leighton og Talal Shamoon, *Secure Spread Spectrum Watermarking for Multimedia*

[Cox et al 99] Ingemar J. Cox, Matthew L. Miller, Andrew L. McKellips, *Watermarking as Communications with Side Information*, Proceedings of the IEEE, Vol. 87, No.7, July 1999

[Cox&Linnartz] Ingemar J. Cox, Jean-Paul M.G. Linnartz, *Some general methods for tampering with watermarks*, IEEE J. on Selected Areas of Communications

[Cox&Miller 97] Ingemar J. Cox og Matt L. Miller, *A review of watermarking and the importance of perceptual modeling*, Proceedings of Electronic Imaging '97, February 1997

[Darmstaedter et al. 98] V. Darmstaedter, J.-F. Delaigle, J.J. Quisquarter, B. Macq, *Low cost spatial watermarking*, Computers & Graphics, Vol. 22, No. 4, 1998

[Hansen&Andersen 98] Klaus Hansen, Jens Damgaard Andersen, *Image steganography using spread spectrum techniques*, DIKU, København, 1998

[Hansen&Andersen 99] Klaus Hansen, Jens Damgaard Andersen, *Some Properties of the Steganographic Capacity of Images Derived from Watermarking Bechmarks*, DIKU, København, juni 1999

[Hartung et al 98] Frank Hartung, Peter Eisert, Bernd Girod, *Digital Watermarking of MPEG-4 Facial Animation Parameters*, Computers & Graphics, Vol. 22, No. 4, 1998

[Hartung&Girod 96] Frank Hartung, Bernd Girod, *Digital Watermarking of Raw and Compressed Video*, Proceedings SPIE 2952, October 1996

[Hartung&Kutter 99] Frank Hartung, Martin Kutter, Multimedia Watermarking *Techniques, Proceedings of the IEEE*, Vol. 87, No. 7, July 1999

[Kutter et al] Martin Kutter, Frédéric Jordan, Frank Bossen, *Digital Signature of*

Color Images using Amplitude Modulation

[Kutter&Petitcolas 99] M. Kutter, F.A.P. Petitcolas, *A fair benchmark for image watermarking systems*, Electronic Imaging '99

[Ó Ruanaidh et al 95] J.J.K. Ó Ruanaidh, W.J. Dowling, F.M. Boland, *Watermarking Digital Images for Copyright Protection*

[Ó Ruanaidh et al 96] J.J.K. Ó Ruanaidh, W.J Dowling, F.M. Boland, *Phase watermarking of images*, IEEE International Conference on Image Processing, Lausanne, Switzerland, September 1996

[Ó Ruanaidh et al] J.J.K. Ó Ruanaidh, F.M. Boland, O. Sinnen, *Watermarking Digital Images for Copyright Protection*

[Petitcolas et al 99] Fabien A.P. Petitcolas, Ross J. Anderson, Markus G. Kuhn, *Information Hiding – A Survey*, Proceedings of the IEEE, Vol. 87, No. 7, July 1999

[Pfitzmann] Birgit Pfitzmann, *Information Hidding Terminology*

[Piron et al 99] Laurent Piron, Michael Arnold, Martin Kutter, Wolfgang Funk, Jean Marc Boucqueau, Fiona Craven, *OCTALIS benchmarking: Comparison of four watermarking techniques*, Security and Watermarking of Multimedia Contents, Vol. 3657, San Jose, Californien, USA, January 1999

[Puate&Jordan] Joan Puate, Fred Jordan, *Using fractal compression scheme to embed a digital signature into an image*

[Servetto et al 98] Sergio D. Servetto, Christine I. Podilchuck, Kannan Ramchandran, *Capacity issues in digital image watermarking*, Proceedings of the 1998 IEEE International Conference on Image Processing, 1998

[Swanson et al 98] Mitchell D. Swanson, Mei Kobayashi, Ahmed H. Tewfik, *Multimedia Data-Embedding and Watermarking Technologies*, Proceedings of the IEEE, vol. 86, no. 6, June 1998

[Tirkel&Osborne] Anatol Z. Tirkel, Charles F. Osborne, *Image Watermarking – A Spread Spectrum Application*

[Voyatzis&Pitas 98] G. Voyatzis, I. Pitas, *Digital image watermarking using mixing systems*, Computers & Graphics, Vol. 22, No. 4, 1998

[Wolfgang et al 99] Raymond B. Wolfgang, Christine I. Podilchuk, Edward J. Delp, *Perceptual Watermarks for Digital Images and Video*, Proceedings of the IEE, Vol. 87, No. 7, July 1999

[Zhao&Koch 98] Jian Zhao, Eckhard Koch, *A generic digital watermarking model*, Computers & Graphics, Vol. 22, No. 4, 1998

Billedbehandling

[Ahumada] A.J. Ahumada, Jr, *Object Detection in a Noisy Scene*, Human Vision, Visual Processing, and Digital Display VII, SPIE Proceedings Volume 2657

[Brink 96] A.D. Brink, *Using spatial information as an aid to maximum entropy image threshold selection*, Pattern Recognition Letters 17, 29-36, 1996

[Cohen 98] Aaron Michael Cohen, *How Do I Process Images with Java?*, Dr. Dobb's Journal, #287, July 1998

[DICOM 93] *Digital Imaging and Communications in Medicine (DICOM)*, National Electrical Manufacturers Association, 1993

[Foley et al, 96] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, *Computer Graphics – Principles and Practice, Second edition in C*, Addison-Wesley, 1996

[Gonzales&Woods 93] Rafael C. Gonzales, Richard E. Woods, *Digital Image Processing*, Addison-Wesley, 1993

[Otterloo 88] P.J. van Otterloo, *A Contour-Oriented Approach to Digital Shape Analysis*, CIP-Data Koninklike Bibliotheek, Den Haag, 1988

[Solomon et al 94] J.A. Solomon, A.B. Watson, A. Ahumada, *Visibility of DCT basis functions: Effects of contrast masking*, Proceedings, Data Compression Conference, Snowbird, Utah, IEEE Computer Society Press, 1994

[Stark 99] J. Alex Stark, *Adaptive Image Contrast Enhancement using Generalizations of Histogram Equalization*, IEEE Transactions on Image Processing, 1999

[Zhu et al.] Song Chun Zhu, Yiang Nian Wu, David Mumford, *Minimax Entropy Principle and Its Application to Texture modeling*

[Zhu&Mumford] Song Chun Zhu, David Mumford, *Learning Generic Prior Models for Visual Computation*, IEEE Trans. on Pattern Analysis and Machine Intelligence

Fysiologi og psykofysik

[Beek] Jacob Beek, *The Perception of Surface Color*, Scientific American

[Campbell&Maffei] Fergus W. Campbell, Lamberto Maffei, *Contrast and Spatial Frequency*, Scientific American

[Ellis et al 86] Hadyn D. Ellis, Malcolm A. Jeeves, Freda Newcombe, Andy Young, *Aspects of Face Processing*, NATO ASI, 1986

[Fiorentini et al. 90] Adriana Fiorentini, Günter Baumgartner, Svein Magnussen, Peter H. Schiller, James P. Thomas: *The Perception of Brightness and Darkness*, Visual Perception: The Neurophysiological Foundations, Aca-

demic Press, inc., 1990

[Gilchrist] Alan L. Gilchrist, *The Perception of Surface Blacks and Whites*, Scientific American

[Laursen 76] Arne Mosfeldt Laursen, *Sanserne og de sensoriske funktioner i centralnervesystemet*, F.A.D.L.s forlag, 1976

[Livingstone] Margaret S. Livingstone, *Art, Illusion and the Visual System*, Scientific American

[Moses&Hart 87] Robert A. Moses, William M. Hart, Jr., *Physiology of the eye, Clinical Application*, The C.V. Mosby Company, 1987

[Parascope 97] *Subliminal Suicide*, http://www.parascope.com/articles/0497/sublim5.htm.

[Shapley et al 90] Robert Shapley, Terrence Caelli, Stephen Grossberg, Michael Morgan, Ingo Rentschler, *Computational Theories of Visual Perception*, Visual Perception: The Neurophysiological Foundations, Academic Press, inc., 1990

[Simmons] Gustavus J. Simmons, *The History of Subliminal Channels*

[Stark&Boyer 96] Louise Stark, Kevin Bowyer, *Generic Object recognition using Form & Function*, 1996, World Scientific

[Treisman et al 90] Anne Treisman, Patrick Cavanagh, Burkhart Fischer, V.S. Ramachandran, Rüdiger von der Heydt, *Form Perception and Attention*, Visual Perception: The Neurophysiological Foundations, Academic Press, inc., 1990

[Walraven et al 90] Jan Walaraven, Christina Enroh-Cugell, Donald C. Hood, Donald I. A. MacLeod, Julie L. Schnapf: *The Control of Visual Sensitivity*, Visual Perception: The Neurophysiological Foundations, Academic Press, inc., 1990

[Wilson et al 90] Hugh R. Wilson, Dennis Levi, Lamberto Maffei, Jyrki Rovamo, Russel DeValois, *The Perception of Form*, Visual Perception: The Neurophysiological Foundations, Academic Press, inc., 1990

[Zrenner et al 90] Eberhart Zrenner, Israel Abramov, Munehira Akita, Alan Cowey, Margaret Livingstone, Arne Valberg, *Color Perception*, Visual Perception: The Neurophysiological Foundations, Academic Press, inc., 1990

Andet

[Jayant et al 93] Nikil Jayant, James Johnston, Robert Safranek, *Signal Compression Based on Models of Human perception*, Proceedings of the IEEE, Vol 81-10, October 1993

[Kahn 96] David Kahn, *The Codebreakers*, Scribner, New York, 1996

[Lanners 75] Edi Lanners (på dansk ved Vagn Simonsen), *Illusionernes Verden*, Lademann Forlagsaktieselskab, 1975

[Murray&vanRyper 96] James D. Murray, William vanRyper, *Encyclopedia of Graphics File Formats*, 2. ed., O'Reilly & Associtcs, Inc., 1996

[Noll 97] Peter Noll, *MPEG Digital Audio Coding*, IEEE Signal Processing Magazine, September 1997

[Olivier 71] Anthony Olivier, *The Victorian Staffordshire Figure*, 1 udgave, Heinemann, London, 1971

[Pennebaker&Mitchell 93] William B. Pennebaker, Joan L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Rainhold, New York, 1993

[Qiao&Nahrstedt 98] Lintian Qiao, Klara Nahrstedt, *Comparison of MPEG Encryption Algorithms*, Computers & Graphics, Vol. 22, No. 4, 1998

[Schneier 96] Bruce Schneier, *Applied Cryptography*, 2. ed., John Wiley & Sons, Inc., 1996

[Shannon 48] C.E. Shannon, *A Mathematical Theory of Information*, The Bell System Technical Journal, Vol. 27, October 1948

[Sikora 97] Thomas Sikora, *MPEG Digital Video-Coding Standards*, IEEE Signal Processing Magazine, September 1997

[Wallace 91] Gregory K. Wallace, *The JPEG Still Picture Compression Standard*, IEEE Transactions on Consumer Electronics, 1991

[Wayner 96] Peter Wayner, *Disappering Cryptography*, Academic Press, 1996

Bilag A: Arbejdsbeskrivelse

Titel:	Steganografi i billeder baseret på en model af human perception	
Navne:	Christian Hammer	(chammer@bk.dk)
		110970-1579
	Lars Rosenberg Randleff	(lars@randleff.dk)
		110573-2097
Vejleder:	Klaus Hansen	(khan@diku.dk)
Aflevering:	15. august 2000	

Dette speciale omhandler emnet steganografi, med henblik på en undersøgelse af mængden af data, der kan camoufleres i et digitalt billede, udfra en model af menneskesynet.

Applikation:
I forbindelse med specialet udarbejdes en række billedbehandlingsrutiner, der samles i en Java-applikation. Denne applikation skal kunne beregne mængden af data, der kan camoufleres i et givent digitalt billede udfra en af os bestemt metode. Desuden skal applikationen kunne camouflere og udtrække camouflerede data af billedet.

Personforsøg:
For at undersøge, hvor meget et digitalt billede kan ændres, uden at ændringen kan ses, udføres et forsøg, hvor en række forsøgspersoner ser på billeder, der er blevet ændret i forhold til referencebilleder. Til opsamling af data fra personforsøget udvikles en Windows™ applikation, der kan fremvise billeder og opsamle resultaterne. Resultaterne indgår herefter i en statistisk analyse.

Rapport:
Til specialet skrives en rapport, der beskriver emnet steganografi. Rapporten beskriver også det ovenfor nævnte personforsøg, samt en anvendelse af resultaterne derfra.

Der gives en introduktion til begrebet steganografi. Denne introduktion vil bl.a. indeholde en sammenligning af begreberne kryptografi og steganografi, en historisk præsentation af steganografien, samt en beskrivelse af digitale vandmærker som en anvendelse af steganografi.

Dernæst behandles steganografi i digitale billeder. I denne forbindelse defineres og forklares en række begreber indenfor såvel steganografi som digitale billeder. Desuden angives metoder til at indkode og afkode camoufleret information fra billeder, og disse metoder illustreres med eksempler.

Da vi mener, at metoder til camouflering af information i billeder kan baseres på menneskesynets opfattelse af billederne, skal menneskesynet og synsfænomener beskrives således, at camouflering kan baseres herpå. Derefter beskrives, hvordan dele af et givent billede, der kan udnytte de omtale synsfænomener ved camouflering, kan bestemmes. For at bestemme, hvilke grænser menneskesynet i realiteten sætter for ændringerne i et billede, udfører vi det ovennævnte personforsøg.

Metoden, der benyttes til bestemmelse af de dele af et billede, der kan ændres, baserer sig på en kombination af forskellige teknikker, bl.a. kantbestemmelse. Det er ikke hensigten at bestemme og benytte mange teknikker, men derimod at godtgøre gevinsten ved en kombination af flere teknikker, i forhold til den mængde data, der kan camoufleres i et billede.

En anvendelse af steganografi baseret på menneskesynet kan være optimering af bl.a. vandmærkningsmetoder. Derfor vil vi beskrive, og forsøge at gennemføre, en optimering af spread spectrum-metoden på udvalgte billeder.

Bilag B: Terminologiliste

Angriber	En person, der ønsker at fjerne eller ødelægge et *vandmærke* eller anden camoufleret information.
Assimilering	En proces hvorved en genstand tilpasses sine omgivelser.
Brightness	Angiver den opfattede lyskomponent, og er altså en subjektiv størrelse. Anvendes også om lyskomponenten, der kan bestemmes objektivt, i *HSB-modellen*.
CAD/CAM	*Computer Aided Design / Computer Aided Manufacturing*. Computerstøttet design og fremstilling.
Camouflage	At udgive sig for at være noget andet.
CGL	Se *Corpus Geniculatum Laterale*.
Corpus Geniculatum Laterale	Center i hjernen, der behandler synsindtryk når de kommer fra øjnene, og før de behandles i *Visuel Cortex*.
Darkness	Det omvendte af *brightness*.
Farvemodel	De farver der er anvendt i et digitalt billede beskrives i et antal parametre. De parametre der anvendes beskriver farvemodellen. Se *RGB* og *HSV/HSB*.
Fiksérbillede	Et billede, som regel tegnet, hvori indgår elementer, som det ikke altid er lige nemme at se. Et eksempel kan ses på forsiden af denne rapport.
Fovea	Den centrale del af øjets nethinde, hvor det klareste syn forekommer. Området har ingen *stave* men til gengæld flere *tappe* end noget andet sted i øjet.
Ganglieceller	Celler i nethinden, der bearbejder signaler fra foto*receptorerne*.
HSB-modellen	Et billede repræsenteres ved tre værdier: Hue, Saturation og Brightness.
HSV-modellen	Samme som *HSB-modellen*, blot er B for Brightness blevet erstattet med V for Value.
Hue	Angiver farvekomponenten af en pixel, der er beskrevet ved *HSI-* eller *HSV-/HSB-modellen*. Hue går på en cirkulær skala fra rød, over grøn og blå, og tilbage til rød. Angivelsen af hue sker som en vinkel mellem rød og den aktuelle farve.
Human Visuel System	Det synsapparat, der behandler synsindtryk hos mennesket.
HVS	Se *Human Visual System*.
Intensitet	Angiver her lyskomponenten i *HSI-modellen*, hvor den vægter den røde, den

	grønne og den blå farvekomponent ligeligt.
Intervalbredde	Bredden af intervallet for den perceptuelle tolerance. Hvis en pixelkomponent antager en værdi indenfor komponentens interval vil billedet have en uændret perceptuel signifikans.
JND	Se *Just Noticeable Distortion*.
Just Noticeable Distortion	Bl.a. JPEG-komprimering benytter begrebet, der betyder, at ændringen i et givent billede kun må være netop synlig. Begrebet eksisterer også for lyd, hvor ændringen må være netop hørbar.
Kantkontrast	Kontrastfænomen, der opstår i områder omkring kanter.
Kontrastfarve	En farve, der kun kan opfattes afhængig af den kontrast farven indgår i.
Kryptografi	At gøre information ulæselig.
Lightness	Anvendes også i stedet for *brightness*.
Luminans	En objektiv størrelse, der angiver den mængde lys, som forlader en genstand.
Lysflux	Se *Luminans*.
Lysstyrke	Se *Luminans*.
Mætningsrenhed	Angiver hvor ren en farve er. Blandes en farve med hvid eller sort mindskes mætningsrenheden.
Områdekontrast	Kontrastfænomen, der ikke kun gælder kanten af et område, hvor kontrasten er størst, men gælder hele området.
Perceptuel signifikans	En scene, der opleves af en tilskuer, har en perceptuel signifikans, der er tilskuerens oplevelse af scenen. Hvis scenen ændres, uden at tilskueren oplever ændringen, er der ikke ændret i scenens perceptuelle signifikans.
Perceptuel tolerance	Et mål for, hvor meget et billede kan ændres, før det kan ses.
Pixelkomponent	En værdi, der beskriver en af et antal af pixelens egenskaber. Hvis en pixel f.eks. er beskrevet af *RGB-modellen*, har pixelen tre pixelkomponenter: R-, G- og B-komponenten.
Plombe	Data der indkodes i f.eks. et billede til plombering af billedet. Hvis data ikke kan genskabes i dele af et billede, er disse dele blevet ændret, og billedets plombering er brudt.
Primær visuel cortex	Bagerste del af hjernen, der behandler synsindtryk umiddelbart efter *Corpus Genicuilatum Laterale*.
Receptor	Celle der reagerer på udefrakommende stimulus. I øjets nethinde sidder fotoreceptorer, der reagerer på lys.

RGB-modellen	Et billede repræsenteres ved fire forskellige værdier: Rød, Grøn, Blå og α. Da α-værdien kun spiller ind ved sammensætning af billeder (konstruktion), beskrives RGB-billeder ofte kun ved de tre værdier Rød, Grøn og Blå.
Saturation	Angiver, mætningsgraden af farvekomponenten af en pixel, der er beskrevet ved *HSI-* eller *HSV-/HSB-modellen*. Hvis en pixel har en maksimal saturation, har den en ren farve. Er saturation derimod mindre end maksimum indeholder farven en del hvid.
Simultankontrast	Et område kan synes at have forskellige intensiteter, afhængig af kontrasten til omgivelserne.
Spread spectrum	En metode til bl.a. at camouflere vandmærker i billeder. Oprindeligt hentet fra kommunikationsteori.
Stave	Foto*receptor* der er placeret i øjets nethinde. Stave er bedst til at opfange ændringer i lys.
Steganografi	At camouflere information eller data.
Steganografisk kapacitet	Et mål for, hvor stor en mængde data, der kan camoufleres i et billede.
Stegobillede	Det billede der indeholder (camouflerede) data.
Tappe	Fotoreceptor der er placeret i øjets nethinde. Tappe er bedst til at opfange ændringer i farve.
Tolerancebillede	Et billede, der sammenfatter den perceptuelle tolerance for pixels, eller blokke i et billede.
Value	Anvendes om lyskomponenten i *HSV-modellen*.
Vandmærke	Data, der kan indkodes i et billede, og som er robuste over for ændringer i billedet. Kan bl.a. bruges til bestemmelse af ophavsrettigheder.
Visuel cortex	Se *Primær visuel cortex*.
Værtsbillede	Det oprindelige billede hvis *perceptuelle tolerance*, og dermed *steganografiske kapacitet* beregnes. Billedet bruges til camouflering af information.

Bilag C: Anvendelser af steganografi

I dagligdagen støder man nogle gange også på ting, der kan være steganografiske. I dette bilag beskrives en række mere eller mindre kendte fænomener, der faktisk er af steganografisk karakter.

Tekst-TV

I et TV-signal, hvis hovedformål jo er at bringe billeder og lyd til seerne, er der desuden gemt alle de oplysninger, der kan hentes frem på Tekst-TV (TTV). TTV-data er ikke gemt i selve billederne, men i en del tilgængelig del af signalet, hvor de overføres som styrekoder.

Den telepatiske tryllekunstner

En hjælper går rundt blandt publikum, og låner forskellige personlige genstande af disse. Nu bliver tryllekunstneren adspurgt, om han kan identificere denne genstand, og et øjeblik efter kommer svaret – det korrekte vel at mærke. Tricket består i, at tryllekunstneren og hjælperen på forhånd har indøvet et system, der beskriver genstanden. F.eks. kan spørgsmålet fra hjælperen: "Hvad er denne genstand så?" betyde, at han står med en skindpung, mens spørgsmålet: "Hvilken ting holder jeg nu i hånden?" betyder at han står med et armbåndsur.

The bible code

Her har forfatteren fundet ud af, at der i biblen er gemt en kode, der beskriver fremtidige begivenheder antageligt "indkodet" af Gud. Metoden tager sit udgangspunkt i den jødiske udgave af biblen, og bogstaver udvælges ud fra en given algoritme. Disse bogstaver danner tilsammen ord, der så skulle henvise til fortidige, nutidige og fremtidige begivenheder. Der er dog mange punkter hvor metoder ikke ligefrem er særligt videnskabelig endsige metodisk. Det jødiske skriftsprog indeholder ingen vokaler, og det er derfor op til fortolkeren selv at indsætte vokaler mellem konsonanterne. Desuden har de "gudgivne" meddelelser en tendens til at være affattet på engelsk og ikke på Jiddisch, som man ellers kunne vente når resten af den "ægte" bibel er skrevet på dette sprog. En kritiker har udført samme metode på en moderne skønlitterær bog, og har fundet masser af lignende beskeder.

The Comforter Dogs

De såkaldte *Comforter Dogs* er et par vemodigt udseende porcelænshunde, begge med hovedet vendt fremad, men med siden vendt til. Disse var i midten af 1800-tallet populære gaver fra søens farende svende til deres kærester eller hustruer. Til dem er også knyttet en lille fortælling ,der går som følger: Når manden (typisk en fisker eller sømand) var borte, blev de to hunde stillet i vindu-

et, med snuden vendt mod havet. Deres sørgmodige udtryk skulle så vise at de (og beboerne i huset) savnede manden, og blot ventede på, at han kom hjem. Når han så kom hjem, blev hundene vendt, så de nu sad med ryggen til havet. Dette blev af nogle udnyttet til at signalere til elskeren (elskerne), at nu var manden hjemme, og det ikke var så smart at komme forbi. Spejdede hundene efter manden, kunne elskeren så frit komme og gå.

Lastbilchauffører

Har man engang overhørt en radiosamtale mellem lastbilchauffører, kunne man have hørt følgende meddelelse: "Det er isglat på Køge Landevej", selvom samtalen foregik midt om sommeren! Denne kode dækker over noget helt andet, nemlig at en chauffør har opdaget, at der er politipatrulje på Køge Landevej, og han vil gerne advare andre om det.

Barneleg

En leg de fleste nok har prøvet i de små klasser i folkeskolen er: "Tag den ring og lad den vandre". Her går en person, detektiven, udenfor døren, og alle de tilbageværende sætter sig i en rundkreds. En anden person bliver udvalgt til at være ringbærer og detektiven kaldes ind. Nu skal alle i rundkredsen synge: "Ser du jeg har ingen", men den udvalgte skal synge "Ser du jeg har ringen". Detektiven skal nu, med eet gæt, ved at gå fra person til person og lytte nøje efter, finde ud af hvem der har ringen. Her skjules en besked i noget støj, der til forveksling kunne være beskeden, og kun nuancer adskiller information og støj.

Bilag D: Dokumentation af personforsøg

I forbindelse med personforsøgene er blevet anvendt et program til fremvisning af billeder og opsamling af data. Desuden har forsøgspersonerne modtaget skriftlige instruktioner og udfyldt en samtykkeerklæring. Programmet, kildetekst, instruktioner og samtykkeerklæringer kan findes på den medfølgende cd-rom.

Beskrivelse af forsøgsopstilling

Under rækken af forsøg har vi benyttet samme computer, samme skærm og så vidt muligt samme lokale. Betingelserne i lokalet var:

- Temperatur: 20° - 24°C
- Persienner lukket
- Dør lukket

Alt software er kørt på en Microsoft NT4.0 platform. Computeren brugt til vores forsøg beskrives ved:

- Compaq DeskPRO Pentium III
- 288 MB RAM
- Grafikkort: ATI 3D RAGE PRO AGP 2X (GT-C2U2) 8 MB

Grafikkortets drivere er:

- `ati.sys`, Version 5.1.118 (02 July 1998 08:11:51)
- `ati.dll`, Version 4.0 (02 July 1998 08:11:51)

Skærmen var en *SONY Multiscan G500 Trinitron* med følgende opsætning:

- 16 millioner farver
- 1600×1200 pixels
- Lille font
- Refresh rate: 70 Hz
- Farvetemperatur: 9300 K

Instruktion til forsøgspersoner

Hver forsøgsperson fik nedenstående instruktion inden forsøget:

Instruktion til forsøgspersoner

Denne instruktion gælder forsøgspersoner til specialet "Steganografi i billeder baseret på model af human perception" af Christian Hammer og Lars Rosenberg Randleff.

Under forsøget vil du komme til at se på billeder på en computerskærm, og du skal bruge en computermus, for at gennemføre forsøget. Forsøget går ud på at udpege billeder, der er ændret i forhold til andre billeder. I et sæt af billeder vises ni næsten ens billeder. Ud af disse ni billeder vil to billeder kunne skille sig ud ved ændringer i små detaljer. Din opgave går ud på at vælge disse to billeder, hvis du kan se, at de skiller sig ud.

Hvis to billeder skiller sig ud, vil de altid være ens. Der vil ikke være mere end to billeder i et sæt, der skiller sig ud, og i nogle sæt er der ingen billeder, der skiller sig ud. Hvis du ikke kan se, at to billeder skiller sig ud, kan du nøjes med at vælge et eller intet billede. Når du har valgt, det/de billeder, som du mener skiller sig ud, klikker du på knappen 'Næste', og det næste sæt af billeder vises. Der vises i alt 106 sæt af billeder.

Forsøget forventes at tage ca. 1½ time. Det er dog intet krav, at du bruger 1½ time på forsøget, hvis du kan gennemføre det på kortere tid, ligesom du gerne må være mere end 1½ time om det, hvis det er nødvendigt.

Under forsøget bør du overholde følgende retningslinier:

- Afstanden mellem skærmen og dine øjne bør ikke være mindre end 30 cm, og ikke større end 70 cm.
- Hvis du har kommentarer til et sæt af billeder, kan du skrive dem på papir under forsøget. Husk at referere til sættets nummer, som du kan se i højre side af skærmen.
- Det gælder ikke om at vælge billederne så hurtigt som muligt. Giv dig den tid du mener der skal til, og husk at holde pause, når du bliver træt i øjnene.
- Husk at blinke.

Tak for hjælpen.

Christian Hammer og Lars Rosenberg Randleff
Nærum, 26. august 2000.

Samtykkeerklæring

Alle forsøgspersoner har udfyldt en samtykkeerklæring. Disse erklæringer opbevares af forfatterne, og kan findes indscannet på den medfølgende cd-rom. Samtykkeerklæringen så således ud:

Samtykkeerklæring

Denne samtykkeerklæring gælder forsøgspersoner til specialet "Steganografi i billeder baseret på model af human perception" af Christian Hammer og Lars Rosenberg Randleff.

Navn:

_____

Køn:
___ Mand
___ Kvinde

Bruger du briller/kontaktlinser?
___ Nej
___ Ja

Er du farveblind?
___ Nej
___ Ja
Hvis 'Ja', hvordan? _____

Hvordan bedømmer du dit syn (evt. med briller/kontaktlinser)?
___ Bedre end normalt syn
___ Normalt syn
___ Dårligere end normalt syn

"Jeg bekræfter hermed at ovenstående oplysninger er rigtige, og giver hermed min tilladelse til, at oplysningerne, samt resultaterne fra min deltagelse i forsøget, må indgå i ovennævnte speciale. Jeg er bekendt med, at specialet vil være offentligt tilgængeligt.

Jeg bekræfter ligeledes at have læst 'Instruktion til forsøgspersoner' i forbindelse med forsøget."

Dato: _____

Underskrift: _____

Bilag E: Programbeskrivelse – StegoShop

StegoShop minder en smule om Paint Shop Pro, dog noget simplere og med det formål at behandle steganografisk materiale.

Programmet er udviklet i Java 2 på Intel-platformen, under Microsoft NT4.0, og er for det meste oversat med IBMs Java oversætter, kaldet *jikes*. Alt benyttet Java-relateret materiale er inkluderet på cd-rom'en.

Som en del af de udviklingsværktøjer Sun Microsystems stiller til rådighed, er en hjælpefil generator, der kan læse valide Java-kildefiler og levere en komplet beskrivelse af metoder, klassevariabler og klasse-hierarki. Denne er brugt, og dokumentationen er at finde på cd-rom'en. Derfor er der ikke lavet yderligere dokumentation af programmets struktur her. Dog skal de til Java-oversætteren og hjælpefilgeneratoren benyttede parametre nævnes her. Se i øvrigt hvordan det kan gøres på Windows NT i "mk.bat" i StegoShop biblioteket på cd-rom'en.

- Til *javac* (standard Java oversætter) er ingen parametre benyttet.
- Til *jikes* (IBMs Java oversætter) bruges "+M" (Makefile dependencies) udover "-classpath ...", der skal pege på placeringen af JAI-bibliotekerne.
- Til *javadoc* (Standard Java hjælpefil-generatoren) er "-private" sat, så samtlige klassevariable og klassemetoder beskrives.
- Til *java* (Standard Java bytekode-fortolkeren) har vi sat opstarts heap-hukommelsen til 6 MB og maksimum heap-hukommelsen til 200 MB ("-Xms6m" henholdsvis "-Xmx200m")

Opsætningen af selve programmet foregår for en stor del i "resource.properties". De vigtigste properties her er "menu" og "macro": *menu* definerer de menuer, der er tilgængelige i programmet, mens *macro* opremser alle de *properties*, der er kendt som makrokald. Et makrokald kan bruges direkte i menuen, og vil via nogle standard Java mekanismer, omforme disse til f.eks. en filtrering eller lignende. Makrokaldende havner i sidste ende i *StegoMacroInterpreter*-klassen, der har til opgave at behandle makroen. Alternativt kan en makro være sammensat af andre makrokald, og den kan dermed udføre temmeligt komplekse opgaver. Denne opgave varetager *StegoMacroInterpreter*'s baseklasse *MacroInterpreter*.

Selve billedbehandlingen er baseret på Java Advanced Imaging, som er et udvidelsesbibliotek til Standard Java. Det er også kendt som JAI, og vi har benyttet version 1.0.2 af det – under den indledende programudvikling benyttede vi dog en pre-release udgave (0.7b).

Vi vil ikke gå ind i en nærmere forklaring af, hvordan programmet bruges – men det burde være rimeligt ligetil, da det, som nævnt, i basis-funktionalitet minder om Paint Shop Pro og lignende billedbehandlingsværktøjer. En ting er dog værd at bemærke – selvom programmet kan foretage ind- og af-kodning (ud fra min og max billeder) bliver inddatafilen, og der er kun een, bestemt i *resource.properties* filen under navnet "datafile". Under afkodning bliver de afkodede bits opsamlet under filnavnet bestemt ved "datafile.out".

Bilag F: Programbeskrivelse – StegoViewer

StegoViewer er udviklet til vores personforsøg – til at fremvise *N* billeder ad gangen, hvoraf forsøgspersonen skal vælge op til *n* billeder, der opfattes som ændrede. Informationer om de valgte hhv. de ændrede billeder gemmes, og der kan således laves statistisk behandling af resultaterne.

Da vores krav til forsøgsrækken var et dobbeltblindforsøg, benytter StegoViewer `rand` funktionen fra Microsoft Visual C++ 6.0 (MVC) til udvælgelse af de enkelte billeder, samt til at bestemme positionen af de ændrede billeder. Endvidere startes tilfældighedsgeneratoren med `srand(time(NULL))`.

Skelettet til programmet er bygget ved hjælp af en af MVCs automatiske kodegeneratorer (MFC Application Wizard til en "Single Document" applikation). Reelt har vi kun arbejdet i et par af de genererede filer, ChildView.h og ChildView.cpp, og vi har tilføjet en lille klasse til vores 'Brugernavn' dialogboks, der vises ved opstart.

Ud fra et princip om enkelthed er formatet for inddatafilerne og uddatafilerne faktisk ens, og i øvrigt strengt tekstbaseret. Opsætningsværdier, og dataværdier der *ikke* er billedpar, foranstilles altid med et '@', hvorefter værdinavnet optræder, efterfulgt af en tabulering, og til sidst efterfulgt af værdien. Herunder vises et lille udpluk af de vigtigste:

`@Image.h`	`360`	Højden af de enkelte billeder
`@Image.w`	`480`	Bredden af de enkelte billeder
`@Select`	`2`	Maksimale antal valgte billeder
`@Show.h`	`3`	Antal billeder der vises horisontalt
`@Show.w`	`3`	Antal billeder der vises vertikalt

De øvrige er enten ren information til brug ved statistikken, eller sekundære opsætninger, såsom bredden af rammerne omkring billederne og afstanden mellem dem etc.

For at have styr på hvilke billedpar der arbejdes med, er alle parrene listet i filen 'settings.txt', der så også indeholder standardopsætningerne, som læses ind i starten af kørslen. Eftersom vi gerne vil kunne rekonstruere forsøgspersonens serie, kopieres hele opsætningsfilen (bortset fra kommentarlinier) ind i uddatafilen.

Hvert sæt af billeder får registreret det valgte billedpar, både som indeks i forhold til opsætningsfilen, og direkte som et par af filnavne. De ændrede og valgte billeders placering, antallet af korrekt valgte billeder, samt den tid forsøgspersonen har brugt på billedet. Den forbrugte tid skal tages med det forbehold, at diverse pauser og lign. også tælles med. Forsøgspersonerne skulle ikke være under et tidspres, og derfor var de ikke vidende om tidtagningen.

Nedenstående er et eksempel på et enkelt (let redigeret) valg fra en tilfældigt udvalgt forsøgsperson:

```
@Test.3.Index         14
@Test.3.Files         bora.bmp              RandHValue90-90.bmp
@Test.3.Changed       (1,1) (2,3)
@Test.3.Selected      (2,1) (3,1)
@Test.3.Correct       0
@Test.3.Time          3.034000 seconds
```

I ovenstående kigger vi på den tredie test, der viser billedpar nummer 14, der indeholder billederne 'bora.bmp' og 'RandHValue90-90.bmp'. De to ændrede billeder (RandHValue90-90.bmp) er placeret på (x,y) = (1,1), hhv. (x,y) = (2,3), og forsøgspersonen har (på 3 sekunder) valgt billede (2,1) og (3,1), og ingen af dem er rigtige.

Lidt primitivt, har vi valgt at lade det indtastede brugernavn afgøre hvilken af tre tilstande programmet køres i: normal, administrativ eller test. Test er forbeholdt demonstration af programmet, og arbejder da også på en noget mindre opsætningsfil (kun ca. 15 billeder mod normalt 106). Denne tilstand er ment som en kort præsentation af samtlige billedtyper og programmets virkemåde, og er blevet brugt før hver forsøgsperson gik i gang med den egentlige test. Normaltilstanden er så den almindelig testserie, mens den sidste tilstand er til at gennemse de enkelte forsøgspersonens sekvenser, samt til at lave statistik på resultaterne. Valget mellem de enkelte tilstande foregår ved, i 'Brugernavn' dialogboksen, at skrive "Tester" for test-tilstand og "Admin" for den administrative tilstand. Alt andet bliver opfattes som normaltilstand.

'Spøgelsesvalg'

Efter forsøgsrækken er gennemført har vi tilføjet en 'spøgelsesdetektion' i programmet, da det desværre viste sig, at programmet ikke slettede de valgte billede-positioner. Hvis forsøgspersonen efterfølgende valgte færre end det maksimalt mulige, ville programmet gentage de sidste valg, som om de var foretaget for den nye test.

Nedenstående eksempel viser en situation hvor forsøgspersonen *ingen* billeder har valgt, hvorfor 'selected' er gentaget i sin helhed.

```
@Test.1.Index         13
@Test.1.Files         bigben.bmp            RandHHue90-90.bmp
@Test.1.Changed       (2,2) (1,3)
@Test.1.Selected      (1,1) (2,1)

@Test.2.Index         9
@Test.2.Files         lena.bmp              RandAAv50.bmp
@Test.2.Changed       (3,2) (1,3)
@Test.2.Selected      (1,1) (2,1)
```

Eftersom programmet intern arbejder med en FIFO-kø til opsamling af valgte billeder, vil nye valg blot blive tilføjet i slutningen af 'selected' mens de første slettes. Derfor vil et antal valg, der er mindre end det maksimalt tilladte betyde, at nogle af de forreste positioner kan være spøgelsespositioner. Desværre er det ikke helt entydigt om det vitterligt er en spøgelsesposition eller ej – for forsøgspersonen kunne reelt have valgt samme billedposition igen, på samme plads. Derfor kan programmet ikke automatisk frasortere spøgelsesvalg. Følgende princip ligger til grund for at programmet kan sandsynliggøre en spøgelsesposition:

- Den forrige valgte sekvens sammenlignes med den nuværende, er der intet overlap mellem disse, er der intet problem.
- Er der overlap mellem den forreste del af det nuværende valg og den bagerste del at forrige valg, kan vi risikere, at vi køen indeholder et spøgelsesvalg.

Lidt mere formelt: De gamle valg repræsenteres ved $P = \{P_1, \mathrm{K}\ , P_k, \mathrm{K}\ P_n\}$ og de nye valg ved $Q = \{Q_1, \mathrm{K}\ , Q_k, \mathrm{K}\ Q_n\}$. Lad $P' = \{P_k, \mathrm{K}\ P_n\}$ og $Q' = \{Q_1, \mathrm{K}\ , Q_{n-k}\}$. Da vil kandidaterne til spøgelsesvalg være at finde i Q' hvis $Q' = P'$. Disse kandidater præsenteres for 'Admin'-brugeren der, sammenholdt med de øvrige data, må afgøre hvor mange af valgene der er spøgelser.

Bilag G: Programbeskrivelse – Spread Spectrum

Programmet SpreadSpectrum er vores forsøg på at behandle camouflering med spread spectrum-metoden ud fra nærmere bestemte toleranceintervaller. Kernen af programmet er opbygget som et kommandolinie-værktøj, der kan indlæse et antal PNM formaterede billedfiler, for derefter at behandle disse, enten enkeltvis eller samlet. Koden er ren C++ med en del STL (Standard Template Library) til at håndtere data-sammenhæng.

Vi valgte at lade formatet være PNM, eftersom dette er noget af den nemmeste at indlæse og formålet med dette program *ikke* var at lave en avanceret billedindlæsning/udskrivning. Til gengæld kan alle PNM formater indlæses, konverteres og udskrives i et hvilket som helst andet PNM format.

Selve behandlingen af billederne foregår i 'HandleImages', der først leder efter de tre billeder som behandlingen kræver: original-, minimal- og maksimal-billederne. Herefter udsættes originalbilledets frekvensrepræsentation for vores optimereingsalgoritme, og den optimerede alfa-værdi benyttes, til sidst, til at indkode originalbilledet. Alle øvrige billeder fjernes fra listen af billeder, og det nye, ændrede billede indsættes, så det bliver skrevet i en uddatafil.

Bilag E: Opbygning af cd-rom

Den medfølgende cd-rom indeholder billeder, dokumentation af personforsøgene, fremstillede programmer og denne rapport i forskellige udgaver. Herunder beskrives de kataloger, der er på cd-rom'en.

- **Billeder**

  Dette katalog indeholder de billeder, der har været anvendt i arbejdet med specialet. En del billeder, i form af figurer fra rapporten, kan findes under 'Rapport\Figurer\'.

  - **Diverse**
  - **Inkodning**

    Billeder, der har været anvendt til ind- og afkodning ved brug af kombinationsmetoden.
  - **Khan**

    Nogle af Klaus Hansens billeder, som vi bl.a. har bestemt den steganografisk kapacitet for.
  - **Kodak**

    Billeder hentet fra Kodaks hjemmeside (www.kodak.com). Nogle af disse billeder går igen i personforsøgene og i rapporten.
  - **Originale**

    De billeder, der bl.a. har været anvendt til bestemmelse af steganografisk kapacitet og til optimering af SpreadSpectrum.
  - **SpreadSpectrum**

    Resultater fra optimeringen af spread spectrum-metoden.
  - **Tolerance**

    Tolerancebilleder beregnet for billederne i 'Billeder\Originale\'.
- **Forsoeg**

  Kataloget indeholder de billeder, der har været anvendt til personforsøgene, de udfyldte samtykkeerklæringer, samt resultaterne fra forsøgene.

  - **Originale billeder**
  - **Anvendte billeder**
  - **Samtykkeerklæringer**
  - **Resultater**
- **Java**

  Java 2 SDK, Standard Edition, v 1.2.2

- **Links**

  De links, der er beskrevet i 'Bilag F: Steganografi på WWW' er gengivet her i HTML-format.

- **Programmer**

  De programmer, der er blevet fremstillet i forbindelse med specialet, er blevet samlet i dette katalog.

  - **Delphi**

    I starten af arbejdet med specialet blev der fremstillet små programmer med Delphi. Disse programmer havde bl.a. til formål at undersøge nogle af de egenskaber, som forskellige filtre har.

    - **Filter**
    - **RGB-HSI**
    - **Stoej**
  - **SpreadSpectrum**

    Kildekoden til programmet, der forsøger at optimere spread spectrum-modellen.
  - **StegoShop**

    Det Java-program, der har stået for bl.a. beregning af billeders perceptuelle tolerancer, bestemmelse af den steganografiske kapacitet for kombinationsmetoden, og ind- og af-kodning af data.

    - **StegoShop dokumentation**

    Dokumentationen for StegoShop består af en rækkeNogle af fautomatisk genererede HTML-dokumenter.
  - **StegoViewer**

    Det program, der har været anvendt under personforsøgene.
- **Rapport**

  Denne rapport i flere forskellige formater

  - **Figurer**

    Figurerne fra rapporten er anbragt i dette katalog, så de kan ses med et billedbehand-lingsprogram, hvor papirversionen af rapporten ikke viser detaljer i billederne tydeligt nok. En del af figurerne er fremkommet vha. små programmer, eller ved sammensætning af et antal billeder. I disse tilfælde er figuren angivet som et katalog for sig, hvori de in-volverede filer befinder sig.

Bilag F: Steganografi på WWW

I arbejdet med denne rapport, har vi ofte fundet henvisninger til URL'er, der giver informationer omkring emner som steganografi, vandmærkning og billedbehandling. I dette bilag har vi valgt at bringe en del af disse URL'er, opdelt i kategorierne: Publikationer, henvisninger, billeder, kommercielle og diverse. Listen er er også at finde på den medfølgende cd-rom. Der tages forbehold for forandrede og forældede henvisninger:

Publikationer

cui.unige.ch/~vision/Publications/watermarking_publications.html
external.nj.nec.com/homepages/bloom
hrl.harvard.edu/people/postdocs/zhu/publication.html
nif.www.media.mit.edu/DataHiding/index.html
stud3.tuwien.ac.at/~e9625414
www.cs.ucla.edu/~gangqu
www.eece.unm.edu/~licks
www.gts.tsc.uvigo.es/~wmark/home.cgi
www.isse.gmu.edu/~njohnson/Steganography
www.lnt.de/~eggers/publications.html
www.neci.nj.nec.com/homepages/ingemar

Henvisninger

members.xoom.com/stathisv/dsplinks.htm
www.cl.cam.ac.uk/~fapp2/watermarking
www.cl.cam.ac.uk/users/rja14
www.nt.e-technik.uni-erlangen.de/~hartung/watermarkinglinks.html

Billeder

www.cs.cmu.edu/~cil/v-images.html
www.faqs.org/faqs/compression-faq/part1/section-29.html
www.kodak.com/digitalImaging/samples/classic.shtml
www.xray.hmc.psu.edu/dicom/faq.html

Kommercielle

www.digimarc.com
www.mediasec.com
www.wmsinc.com/prod02.htm

Diverse

ltssg3.epfl.ch:1248/kutter/watermarking

www.bcpl.net/~dbroida/mlkihad.html

www.ctr.columbia.edu/~cylin/watauth/sari.html

www.dspguide.com/pdfbook.htm

www.jjtc.com/stegdoc

www.lenna.org

www.parascope.com/articles/0497/sublim5.htm

www.research.att.com/~njas/doc/shannon.html