

FORELÆSNINGSNOTER  
TIL  
INTRODUKTION TIL BILLEDBEHANDLING

S. I. Olsen  
Datalogisk Institut, KU

10. november 2005

Dette skrift er 7. revision af et notesæt i digital billedbehandling. Noterne er tænkt som undervisningsmateriale i 3. års-kurset *Introduktion billedbehandling* på datalogistudiet. Noterne fokuserer på emnerne: Repræsentation og Filtrering af billeder, men omfatter også en introduktion til detektion af kanter og andre “features” i billeder, samt segmentering af billeder. I forhold til tidligere versioner er der dels sket en omstrukturering, dels er en række emner og afsnit udeladt og nye tilføjet.

Formålet med noterne er at give en praktisk såvel som teoretisk indføring i en række metoder til behandling af digitale billeder. Noterne henvender sig til 3. og 4. års studerende, der som baggrund har bestået første år af datalogistudiet samt kurserne *Lineær Algebra* og *Mat Intro*. Det forudsættes at læseren har et modent (operationelt) forhold til matematik.

Notesættet er under stadig udvikling. Således mangler mange illustrationer. Det anbefales derfor, at noterne suppleres med en egentlig lærebog. Et lille udvalg af lærebøger er nævnt sidst i litteraturlisten (side 173). Fælles for bøgerne er at ingen af disse dækker præcis samme emnerfelter med samme detalierungsgrad som noterne. Derimod er de ofte mere gennemarbejdede og indeholder mange gode illustrationer.

For at opnå en beherskelse af metoderne inden for billedbehandling kan noternes forholdsvis teoretiske gennemgang af disse ikke stå alene. Det er forfatterens holdning, at uden en solid praktisk erfaring med anvendelse af metoderne, vil undervisningen være spildt. Derfor vil undervisningen blive suppleret med opgaver, hvoraf mange kræver implementation og eksperimenter med de gennemgåede metoder. Læseren anbefales indtrængende ikke at stille sig tilfreds med en teoretisk forståelse af en given metode, men derimod eksperimentelt at afprøve metodens styrker og begrænsninger.

Mange metoder til billedanalyse kan beskrives i et statistisk begrebsapparat, i begrebsapparat hentet fra kontinuert funktionsanalyse (herunder mål og integralteori), med begreber fra analyse af diskrete funktioner, og/eller ved brug af begreber fra løsning af differentiaalligninger. For visse metoder er et bestemt begrebsapparat oplagt mest anvendeligt, men for mange metoder kan disse udtrykkes forskelligt - men lige godt - i flere begrebsapparater. I nærværende noter skal vi i høj grad benytte os af den klassiske kontinuerte matematik, herunder mål og integralteorien. Dette betyder i praksis at vi ofte vil starte med at formulere problemet for kontinuerte billeder, og først når den kontinuerte løsning er fundet vil vi formulere en diskret implementation.

## Tak

Jeg vil især gerne takke Jon Sparring, men også en lang række af tidligere studerende for gennem en årrække at have korrigeret fejl, dårlig sprogbrug og kommasætning mv. Uden disse rettelser og forbedringer ville noterne være langt ringere end den version, som du nu læser.

## Notation

Nedenstående afsnit opsummerer de konventioner for notation, som er forsøgt fulgt gennem noterne.

$\star$	Foldningsoperatoren.
$\circ$	Korrelationsoperatoren.
$\boxtimes$	Symbol for “har den fouriertransformerede”.
$\mathcal{R}$	Kursiverede store bogstaver betegner mængder.
$f$	Små bogstaver $f, g$ etc. betegner funktioner i stedområdet.
$f^*$	Den komplekst konjugerede af $f$ .
$\mathbf{f}$	Fede bogstaver $\mathbf{f}, \mathbf{g}$ etc. betegner funktioner i frekvensområdet.
$F\{\cdot\}$	Fouriertransformationen.
$\mathbf{X}$	Fede store bogstaver sidst i alfabetet betegner matricer.
$\mathbf{x}$	Fede små bogstaver $\mathbf{x}, \mathbf{y}$ betegner vektorer
$(x_{ij})$	Matrix af elementer $x_{ij}$ i række $i$ og søjle $j$ .
$\mathbf{x}_{\cdot j}$	Søjlevektor af matrix.
$\mathbf{x}_{i\cdot}$	Rækkevektor af matrix.
$\mathbf{x}^t$ og $\mathbf{A}^t$	Vektor- og matrixtransponering.
$\bar{\mathbf{x}}$	Overstreget fede små bogstaver betegner gennemsnit af en vektor.
$\hat{\mathbf{x}}$	Hattede fede små bogstaver betegner et estimat.
$E(\cdot)$	Operatoren $E$ betegner “forventet værdi af”.
$p(\cdot)$	Operatoren $p$ betegner “sandsyligheden for”.
$p(A B)$	Betinget sandsyligheden for $A$ givet $B$ .



# Indhold

<b>1</b>	<b>Introduktion til digital billedbehandling</b>	<b>9</b>
<b>2</b>	<b>Foldning og Fouriertransformation</b>	<b>15</b>
2.1	Foldninger . . . . .	15
2.1.1	Impulsfunktionen . . . . .	17
2.1.2	Den diskrete foldning . . . . .	18
2.2	Fouriertransformationen . . . . .	18
2.2.1	Den kontinuerte fouriertransformation . . . . .	19
2.2.2	Den diskrete fouriertransformation . . . . .	21
2.2.3	Egenskaber ved fouriertransformationen . . . . .	22
2.2.4	Foldningssætningen . . . . .	25
2.2.5	Korrelationssætningen . . . . .	26
2.2.6	Samplingsætningen . . . . .	27
2.3	FFT-algoritmen . . . . .	31
<b>3</b>	<b>Filtrering af billeder</b>	<b>39</b>
3.1	Filtrering i frekvensområde . . . . .	39
3.1.1	Lav-pas filtrering . . . . .	40
3.1.2	Høj-pas filtrering . . . . .	43
3.1.3	Bånd-pas filtrering . . . . .	44
3.2	Filtrering i intensitetsområde . . . . .	44
3.3	Ikke-lineære filtre . . . . .	47
3.3.1	Filtrering i intensitetsområdet . . . . .	47
3.3.2	Filtrering i frekvensområdet: Homomorfisk filtrering . . . . .	48
3.4	Positionsafhængig filtrering . . . . .	49
3.4.1	Anisotrop filtrering . . . . .	50
3.4.2	Normaliseret foldning . . . . .	54
<b>4</b>	<b>Billedrestauring</b>	<b>57</b>
4.1	En simpel model for billedforringelse . . . . .	57
4.1.1	Forvrængning . . . . .	58
4.1.2	Støj . . . . .	58
4.2	Gendannelsesproblemet . . . . .	60

4.3	Det inverse filter . . . . .	60
4.4	Wienerfilteret . . . . .	62
4.5	Det geometriske middelværdifilter . . . . .	65
4.6	Bayes Restoration . . . . .	65
4.7	Iterative suboptimale Filtre . . . . .	68
<b>5</b>	<b>Billedtransformationer</b>	<b>69</b>
5.1	Klassiske repræsentationer . . . . .	71
5.1.1	Repræsentation ved fourier-transformation . . . . .	71
5.1.2	Repræsentation ved cosinus-transformation . . . . .	71
5.1.3	Repræsentation ved binære basisbilleder . . . . .	72
5.1.4	Z-transformationen . . . . .	73
5.2	Karhuen-Loeve repræsentationen . . . . .	73
5.2.1	ICA-transformationen . . . . .	76
5.3	Wavelet transformationen . . . . .	77
5.4	Pyramide repræsentationer . . . . .	79
5.4.1	Quad-tree repræsentation . . . . .	80
5.4.2	Gauss-pyramide . . . . .	81
5.4.3	Burtpyramide . . . . .	81
5.4.4	Laplace-pyramide . . . . .	81
<b>6</b>	<b>Histogrammer og simple statistiske metoder</b>	<b>83</b>
6.1	Histogram linearisering . . . . .	83
6.1.1	Direkte histogramspecifikation . . . . .	86
6.1.2	Lokal histogram linearisering . . . . .	86
6.2	Grænseværdibestemmelse . . . . .	87
6.2.1	Normalfordelte populationer . . . . .	87
6.2.2	Dybeste dal -algoritmen . . . . .	88
6.3	Kvantificering . . . . .	89
6.3.1	Optimal kvantificering . . . . .	90
6.4	K-means algoritmen . . . . .	91
6.5	Mixture models . . . . .	92
<b>7</b>	<b>Geometrisk opretning af billeder</b>	<b>95</b>
7.1	Warping . . . . .	98
7.2	Geometriske projektioner . . . . .	100
7.3	Homeografier . . . . .	101
<b>8</b>	<b>Filtrering ved Matematisk morfologi</b>	<b>105</b>
8.1	Ekspansion og Erosion . . . . .	106
8.2	Åbning og Lukning . . . . .	107
8.3	Morfologiske operationer . . . . .	109
8.3.1	Hit-or-Miss transformationen . . . . .	109

8.4	Gråtone-morfologi . . . . .	110
8.5	Morfologiske gråtoneoperationer . . . . .	111
<b>9</b>	<b>Regularisering og diffusion</b>	<b>113</b>
9.1	Mumford-Shah . . . . .	115
9.2	Markov Random Fields . . . . .	116
9.3	Simuleret udglødning . . . . .	120
9.4	Diffusion . . . . .	122
9.4.1	Perona-Malik . . . . .	122
<b>10</b>	<b>Detektion af kanter i billeder</b>	<b>127</b>
10.1	Kantmodeller . . . . .	127
10.2	Støjproblemet . . . . .	129
10.3	Numeriske metoder til måling af afledede . . . . .	130
10.4	Beudet operatorer . . . . .	132
10.5	Detektionsmetoder . . . . .	135
10.5.1	Differentialgeometri . . . . .	138
10.6	Laplace-Gauss-filtret . . . . .	140
10.7	Canny's kantoperator . . . . .	142
10.8	Deriche's kantoperator . . . . .	145
10.9	Detektion af linier . . . . .	147
10.10	Kvadraturfiltre . . . . .	148
<b>11</b>	<b>Detektion af hjørner og andre interessepunkter</b>	<b>151</b>
11.1	Simple interessepunktsdetektorer . . . . .	151
11.2	Hjørnedetektion . . . . .	152
11.2.1	Krumningsmål . . . . .	152
11.2.2	Egenværdianalyse . . . . .	153
11.2.3	Harris-detektoren . . . . .	154
11.3	Junction detektion . . . . .	154
11.4	Blobs . . . . .	157
11.5	Symmeteripunkter . . . . .	158
11.6	Hough-transformationen . . . . .	159
<b>12</b>	<b>Segmentering</b>	<b>163</b>
12.1	Segmentering ved grænseværdiafskæring . . . . .	163
12.2	Simpel segmentering ved nabosammenligninger . . . . .	164
12.3	Segmentering ved Split-metoder . . . . .	165
12.4	Segmentering ved merge-metoder . . . . .	165
12.5	Seed growing segmentering . . . . .	167
12.6	Watershed segmentering . . . . .	167
12.7	Segmentering ved K-means algoritmen . . . . .	168
12.8	EM algoritmen . . . . .	169

12.9 Andre metoder . . . . .	171
------------------------------	-----



# Kapitel 1

## Introduktion til digital billedbehandling

*Digital billedbehandling* kan defineres som den proces, hvor et digitalt billede transformeres til et nyt digitalt billede. I praksis er definitionen for snæver, men rent forståelsesmæssigt velegnet til at skelne digital billedbehandling ud fra de nært beslægtede fagområder: *grafik*, *billedsyntese* og *datamatsyn*. I grafik og billedsyntese er formålet at *fremstille* et billede ud fra en specifikation. I datamatsyn er formålet typisk at karakterisere den scene, som et givet billede viser. Mange af de teknikker som anvendes inden for de nævnte felter er dog ens.

Digital billedbehandling kan ikke entydigt rubriceres under nogen af de traditionelle discipliner (matematik, fysik, datalogi e.lign), men er multidisciplinær. Der indgår meget matematik og fysik i billedbehandling, men også fag som psykologi, neurofysiologi, geografi, fotogrammetri osv. Eksempelvis har perceptionspsykologien bibragt stor viden om hvorledes vi, som mennesker, behandler og fortolker de synsindtryk, der dannes på nethinderne, når vi ser. Denne viden har bl.a. haft betydning for udvikling af metoder til detektion af kanter i billeder, for kodning af billeder etc.

Ud fra en datalogisk synsvinkel er behandling og fortolkning af billeder interessant, fordi billeder indeholder samplede data, som regel uden en klart defineret semantik. Tværtimod er det ofte formålet med behandlingen at transformere billedet til en (mere kompakt) repræsentation, hvor det semantiske indhold er lettere tilgængeligt. Denne proces vil ofte involvere mere eller mindre avancerede matematiske teknikker. Rollen for den (matematisk træned) datalog er derfor dels at udvikle nye metoder, dels at kunne vælge og sammensætte de rigtige metoder til løsning af et givet problem.

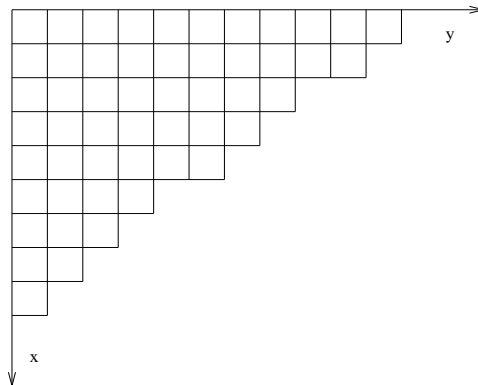
Digital billedbehandling anvendes i dag inden for et meget stort antal discipliner og fagområder. Et klassisk anvendelseseksempel er forbedring af røngten- eller ultralydsbilleder. Et andet eksempel er fra astronomien, hvor man typisk ønsker at fjerne den forvrængning af billedet, der opstår når lyset passerer gennem atmosfæren (eller ønsker at korrigere for den linsefejl, der gjorde de første billeder optaget af Hubble-teleskopet vanskelige at tyde). Et tredje eksempel er analyser af mikroorganismer og deres bevægelsesmønstre. Billedbehandling benyttes også i industriel sammenhæng. Kvalitetskontrol er en typisk anvendelse. Dette kan handle om at kontrollere en

svejsesøm; om at måle kvaliteten af et stykke høvlet træ ved at tælle antal knaster, at måle antallet af bakterier i en vandprøve, at sortere hønseæg efter kvalitet osv.

Digital billedbehandling starter som regel med en *filtrering* af billedet. Formålet er at fjerne eller nedtone uinteressante detaljer samt at fremhæve væsentlige karakteristika. Resultatet af filtreringen kan ved inspektion af et menneske danne udgangspunkt for en beslutning, eller kan danne grundlag for en yderligere datamatisk analyse. Første skridt i en sådan analyse er ofte at foretage en datakompression. Denne kan foretages ved at detektere egenskaber (eng. features) i/ved billedet, samt at knytte mål til de observerede “features”. Eksempler på “features” er kanter/kontraster, højderygge i intensitetslandskabet, små pletter med markant afvigende intensitet i forhold til omgivelserne etc. De tilknyttede mål er ofte metriske: størrelse, orientering e.lign. Sidste trin er typisk at drage slutninger ud fra de observerede “features” samt prædefinerede prototyper af “features” for det, som analysen skal finde eller måle i billedet. Eksempler på sådanne *genkendelse af mønstre* kan være: Findes der en indsnævret blodåre i billedet? Hvor mange forekomster af en given art er synlige i billedet? Disse noter vil fokusere på grundlaget for sådanne analyser.

## Hvad er et digitalt billede

Et digitalt billede er en (rektangulær) opstilling af billedelementer. Hvert billedelement kaldes en *pixel*. Vi betragter et billede  $I$  med  $N$  rækker og  $M$  søjler som et to-dimensionalt array af dimension  $N \times M$ . Man siger at  $I$  har en *opløsning* på  $N \times M$ . Idet vi benytter rækkenummeret  $x$  hhv. søjlenummeret  $y$  på en pixel som første- hhv. andenkoordinat vil vi referere til pixelen ved  $I[x][y]$  eller  $I(x, y)$ . Vi antager altså at billedet er indlagt i et sædvanligt retvinklet *koordinatsystem* med førsteaksen rettet “lodret nedad” og andenaksen orienteret “vandret mod højre”. Hvert

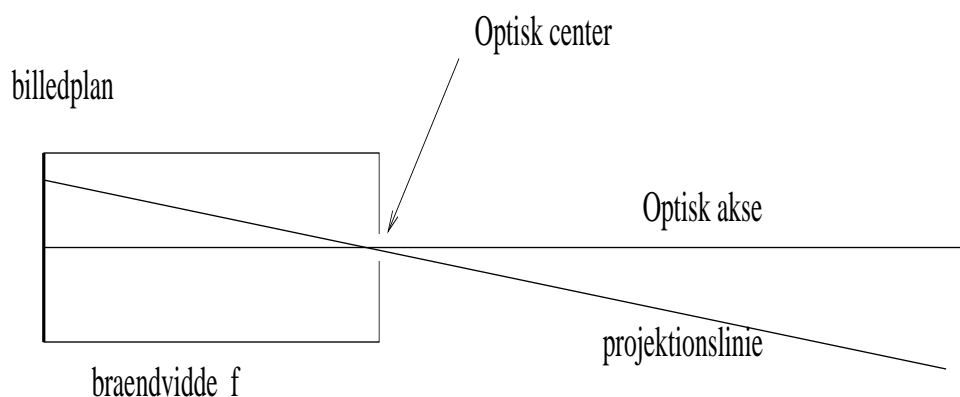


billedelement kan rumme et eller flere tal. Hvis tallet tilhører mængden  $\{0, 1\}$  kaldes billedet binært. Typisk skal vi i disse noter beskrive operationer på *gråtonebilleder*. De billeddannende sensorer, der oftest benyttes, kvantificerer de enkelte billedværdier til 8 pixel, dvs. til værdier mellem 0 og 255 (begge inkl.). Bl.a. i medicinske anvendelser benyttes ofte en kvantificering til

12 bit eller 16. For RGB-farvebilleder rummer hver pixel tre 8 bit størrelser angivende mængden af rødt, af grønt og af blå. Multispektrale billeder indeholder mange (fra 5 op til flere hundrede) farvebånd, ofte fra den infrarøde del af det elektromagnetiske spektrum og ikke kun rødt, grønt og blå. Multispektrale billeder optages ofte af skannere monteret i fly eller satellitter. Fortolkningen af sådanne billeder afhænger naturligvis af et grundigt kendskab til sensoren, der har produceret billedet. Endelig benyttes billeder indeholdende flydende (pseudo-reelle) tal, eller flydende komplekse tal.

De fleste billeddannende sensorer er følsomme inden for et vist bånd af det elektromagnetiske spektrum. Således er et almindeligt CCD-baseret kamera (CCD = Charge Coupled Device) følsomt over for synligt lys i intervallet fra ca. 400 nm til 700 nm. Andre sensorer, f.eks. radar- og røngtenapparater er mere følsomme over for andre bånd i det elektromagnetiske spektrum. Endelig findes der sensorer, der ikke benytter elektromagnetiske bølger som medium. Eksempelvis benytter sonar- og ultralyds-skannere lydbølger. Det er meget vigtigt at lære det billeddannende instrument godt at kende, inden metoder til billedanalyser udvikles. Dette omfatter både sensorens følsomhed og dynamik samt typen og omfanget af støj i billederne. Endelig kræver det ofte en del oplæring at fortolke billederne på en brugbar måde.

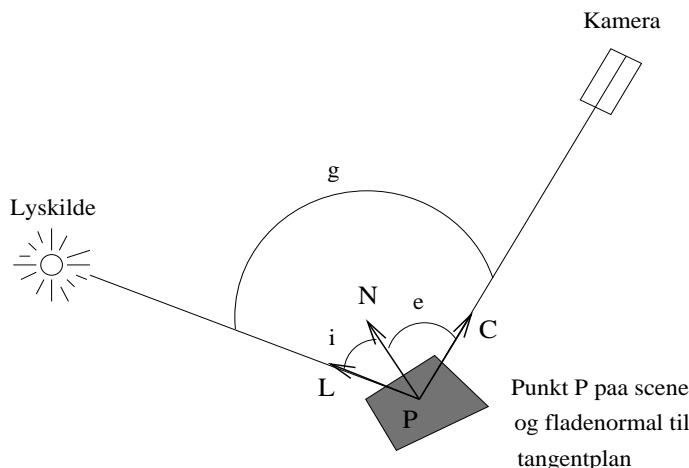
I det følgende skal vi koncentrere os om sensorer, der benytter synligt lys som medie. Som grundlæggende eksempel skal vi betragte et simplificeret kamera, et *pin-hole*-kamera. Dette består af en kasse, hvor den ene side udgør billedplanet, og hvor der i den modstående side er gjort et punktformigt hul, hvorigennem lyset udefra kan trænge ind i kassen.



Gennem hullet (det optiske center) bliver lyset projiceret og afbildet på billedplanet. Den del af scenen, der afbildes, er bestemt af brændvidden  $f$ . Hvis  $f$  er lille vil projektionen eksempelvis være vidvinklet. Da alle projektionslinier går igennem det optiske center er projektionen perspektivisk. Det grundlæggende problem ved et pin-hole kamera er, at der kun kan trænge få fotoner ind gennem det lille hul. Gøres hullet større bliver billedet uskarpt, fordi lyset som reflekteres fra et scenepunkt mod hullet "rammer" et helt område af billedplanet. Det kan bemærkes at gamle fotografiapparater faktisk var opbygget efter pin-hole modellen. Problemet med at samle lys nok blev løst dels ved at benytte magnesiumbomber (giver et stærkt lys) og ved at benytte

lang eksponeringstid. I dag benyttes et system af linser i stedet. I de datamatiske analyser, hvor en kameramodel er påkrævet, er pin-hole modellen dog ofte en god approksimation.

Det lys der afbildes stammer oprindeligt fra en lysgiver. Sædvanligvis er vi omgivet af mange lysgivere, hvoraf solen ofte er den kraftigste. I mange modeller antages det, at lyskilden er uendelig fjern og punktformig. Flere og større lyskilder kan derefter fås ved "integration". Antag derfor, at en enkelt, uendelig fjern og punktformig lyskilde er givet, og betragt et punkt på billedplanet. Til dette svarer en projektionslinie, der skærer scenen i et punkt  $P$ . Antag endvidere, at vi kan modellere scenefladen i en lille omegn om  $P$  ved en plan flade med normal  $N$ .



Når lyset rammer fladen vil en vis del af energien i lyset blive omsat til varme. Den res-tende del vil blive reflekteret. Brøkdelen  $\rho$  af mængden af reflekteret lys er materialeafhængig og kaldes *albedo*. Måden hvorpå lyset reflekteres er også materialespecifik. Den bidirektionale *reflektionsfordelingsfunktion*  $R$  angiver hvor meget lys, der reflekteres i en given retning, når fladen belyses fra en anden given retning. For de fleste materialer (de isotrope) er  $R$  en funktion af indfaldsvinklen  $i$  (vinklen mellem fladenormalen og en vektor rettet fra  $P$  mod lyskilden), af udfaldsvinklen  $e$  (defineret tilsvarende) og vinklen  $g$  mellem det indfaldende og udfaldende lys:

$$E = I\rho R_M(i, e, g)$$

hvor  $E$  er mængden af observeret lys, og  $I$  er mængden af lys udsendt af lysgiveren i retning mod  $P$ . Funktionen  $R$  er naturligvis materialespecifik. For et perfekt spejl gælder, at  $R = 1$ , hvis  $i = e$ , og  $g = i + e$ . Ellers er  $R = 0$ . Et andet grænsetilfælde er et perfekt diffust materiale, hvor lyset reflekteres ligeligt i alle retninger. Her er  $R = \cos(i)$ .

I digitale kamera med en CCD-sensor består chip'en primært af et array af små felter, der hver fungerer som ladningsreservoir. Ladningen i et felt vil være proportional med antallet af fotoner, der har ramt feltet siden det sidst blev afladet. Ved billedoptagelse skiftes ladningerne ud til en forstærker hvorefter det resulterende analoge signal lavpasfiltreres for at undgå aliasing,

samples efter en såkaldt pixel-klokke og diskretiseres til en digital værdi v.h.a. en analog-til-digital konvertering. I konsumkameraer opnås farvebilleder ved at 'coate' chipen med et mønster af røde, grønne og blå filtre (foran hvert reservoir). Et program analyserer efterfølgende det optagne billede og interpolerer vha. en såkaldt Bayer-maske farveværdier i alle pixels. Bemærk at den reelle spatiale opløsning af hver farvekomponent kun er en trediedel i forhold til en ucoated chip med samme antal pixels.

Et væsentligt element inden konstruktionen af et billedbehandlingssystem er at vurdere kvaliteten af og problemerne i de billeder, der skal analyseres. Hvis det er muligt at identificere centrale problemer, f.eks. sløringsgrad og form, omfang og art af eventuel støj, på et tidligt tidspunkt, er det lettere at vælge analysemetode. Mange problemer, der ellers ville opstå på et senere tidspunkt, kan så undgås. Det er håbet at læserne lærer at betragte billeder på en ny kritisk måde, således at de forelagt et billede hurtigt kan vurdere om en given analyse er mulig, samt hvilke metoder, der vil være hensigtsmæssige at anvende.

Når man betragter et digitalt billede er det vigtigt at huske den måde hvorpå billedet er vist (typisk på en monitor eller på printet papir). Disse medier er meget forskellige. Ofte er tætheden (pixels/tomme) forskellig, antallet af mulige gråtoner/farver er forskellige og baggrunden er forskellig (sort for en monitor, hvidt for papir). Dette kan give meget forskellige synsoplevelser. Eksempelvis kan mange printere ikke gengive sort korrekt. Også lyset i det lokale, hvor man betragter billede, kan have stor indflydelse på synsoplevelsen.

## Introduktion til filtrering

Ved filtrering af et billede ønsker man ofte af forbedre billedkvaliteten. Det givne billede kan eksempelvis være uskarpt, og filtreringens formål være at øge kontrasten i billedet. Et andet formål kan være at reducere eventuel støj i billedet eller at fremhæve kontrasterne i billedet. Et tredje formål kan være at ekstrahere netop de strukturer, der er relevant i en efterfølgende analyse.

Der findes mange klasser af filtreringsmetoder, der hver kan have deres fordele og ulemper. Ved valget af filtreringsmetode er det vigtigt at holde sig filtreringens formål for øje. Hvad er problemet med billederne? Hvordan er billederne karakteriseret? Kan der opstilles en model for billedforringelsen? Hvad skal de filtrerede billeder benyttes til?

Den simplest tænkelige filtrering af et billede er at modificere billedelementerne et for et. Dette er f.eks. tilfældet ved skalering af intensitetsværdierne i et billede sådan, at intensitetssområdet udnyttes bedst muligt. En anden klasse af filtre er dem hvor filterværdien i hver pixel bestemmes af en vægtet sum af billedværdierne i en omegn af pixelen.

$$O(x, y) = \sum_{(i,j) \in S} w_{ij} I(x-i, y-j) \quad (1.1)$$

hvor vægtene eller *filterkoefficienterne*  $w_{ij}$  bestemmer filterfunktionen, og  $S$  angiver *støtten* for filteret. Hvis støtten for filteret er lille siges filtreringen at være *lokal*. Der tales her om Finite Impulse Response (FIR)-filtrering. Hvis filterstøtten udgør hele billedet siges denne at være *global*.

Hvis filterkoefficienterne er konstante for alle pixel siges filteret at være *positionsinvariant*. Da filtreringen består i en vægtet sum, er denne *lineær*. Vægtene danner i sig selv en funktion kaldet *filterfunktionen*. En stor del af disse noter omhandler lineære positionsinvariante filtre. Mange af disse er globale i deres kontinuerte udgave, men implementeres ofte som diskrete filtre med lokal støtte fordi de fleste filterkoefficienter i praksis er så små, at de kan antages lig nul. Filtringen foretages ofte ved en *foldning* af billedet med filterfunktionen. Vi skal senere se hvordan samme filtrering også kan foretages i *frekvensområdet* ved at multiplicere de fouriertransformerede af billedet og filterfunktionen.

*Fouriertransformationen* er een blandt mange lineære globale transformationer. En anden beslægtet transformation, der bl.a. benyttes ved JPEG-billedkodning, er *Cosinustransformationen*. Fouriertransformationen giver et beskrivelsesværktøj, der ofte benyttes til at karakterisere signaler og billeder. Da transformationen kan implementeres meget effektivt, muliggør den desuden en effektiv implementation af filtreringen med meget stor støtte. Forståelse af fouriertransformationen og brugen af denne er centralt i noterne.

En lang række filtreringer kan ikke beskrives ved brug af lineære positionsinvariante filtre. Eksempelvis kan filterkoefficienterne variere fra pixel til pixel. Alternativ kan den vægtede sum erstattes af en ikke-lineær operation (f.eks. en maksimums-, minimums-, eller median-operation). Noterne introducerer også til denne type af filtreringer.

Endelig findes et stort antal billedbehandlingsoperationer, som ikke let kan beskrives ved anvendelse af noget simpelt filter, men som involverer mere komplekse algoritmer. Dette er i et vist omfang tilfældet ved detektion af kanter og andre *features* i billeder og i særdeleshed ved *segmentering* af billedet i områder eller objekter.

# Kapitel 2

## Foldning og Fouriertransformation

Dette kapitel beskriver det matematiske grundlag for filtrering af billeder ved lineære positionsinvariante filtre. Senere kapitler vil beskrive konkrete filtre og deres anvendelse. Den matematiske baggrund er den klassiske teori for *Mål- og integralregning*. Vi skal undervejs se hvorledes man kan formulere diskrete versioner af de kontinuerte varianter.

### 2.1 Foldninger

Foldningen af to en-dimensionale reelle funktioner  $f(x)$  og  $g(x)$  er defineret ved integralet:

$$f \star g(x) = \int_{-\infty}^{\infty} f(\alpha)g(x - \alpha)d\alpha \quad (2.1)$$

I det to-dimensionale tilfælde fås analogt:

$$f \star g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)g(x - \alpha, y - \beta)d\alpha d\beta \quad (2.2)$$

Definitionen udvides let til funktioner defineret på  $\mathbb{R}^n$ .

Foldningsbegrebet er centralt i billedbehandling fordi alle lineære og positionsinvariante filtreringer kan beskrives som foldning af det oprindelige billede med en filterfunktion. Som vi skal se senere, stiller den klassiske (kontinuerte) funktionsteori et overordentlig nyttigt værktøj til rådighed ved beskrivelse af mange billedbehandlingsoperationer. Foldninger og fouriertransformationer er de operationer, der er mest benyttede. Om foldningsoperationen gælder følgende regler, der alle er lette at eftervise:

$$f \star g = g \star f \quad (2.3)$$

$$f \star (g + h) = f \star g + f \star h \quad (2.4)$$

$$f \star (g \star h) = (f \star g) \star h \quad (2.5)$$

Yderligere gælder at resultatet af en foldning er begænset (dvs.  $< \infty$ ) og differentiabel (med begrænsede afledede) når blot en af funktionerne er det. I dette tilfælde gælder desuden at den afledede af foldningen er lig foldningen af en af funktionerne med den afledede af den anden:

$$D(f \star g) = Df \star g = f \star Dg \quad (2.6)$$

hvor  $D$  angiver en lineær differentialoperator, som f.eks.  $\frac{\partial}{\partial x}$ . Vi skal senere udnytte disse egenskaber intensivt.

Det ville have været overordentlig nyttigt om foldningsoperationen var forsynet med et 1-element, dvs. at der fandtes en invers foldning. Dette er desværre **ikke** tilfældet. I billedbehandling er det ofte tilfældet, at det aktuelle optagne billede  $O$  er forvrænget på en måde, der kan beskrives ved en foldning  $O = h \star I$ , hvor  $I$  angiver det ideelle billede. Selv om  $h$  er kendt eksakt er det sjældent muligt at rekonstruere  $I$  ud fra observationen  $O$ , fordi det ikke er muligt at *affolde* billedet (dvs. at folde baglængs). Dette er f.eks. tilfældet hvis  $h$  har elimineret information i  $I$ .

Som det ses af definitionen på den en-dimensionale foldning, består denne for hver værdi af  $x$  i følgende operationer:

- 1) En spejling:  $g(\alpha) \rightarrow g(-\alpha)$ .
- 2) En translation:  $g(-\alpha) \rightarrow g(-\alpha + x)$ .
- 3) Konstruktion af produktfunktionen  $t(\alpha) = g(-\alpha + x)f(\alpha)$ .
- 4) Integration af produktfunktionen  $h(\alpha) = \int t(\alpha)d\alpha$ .

### Eksempel

Betragt følgende to funktioner:

$$f(\alpha) = \begin{cases} 1 & \text{hvis } 0 \leq \alpha \leq 1 \\ 0 & \text{ellers} \end{cases} \quad (2.7)$$

$$g(\alpha) = \begin{cases} 1/2 & \text{hvis } 0 \leq \alpha \leq 1 \\ 0 & \text{ellers} \end{cases} \quad (2.8)$$

Ved spejling af  $g$  og translation med  $x \leq 0$  vil de områder af de to funktioner, der er forskellig fra nul, ikke lappe ind over hinanden. Produktfunktionen vil derfor være nulfunktionen, og integral-værdien være lig 0. Det samme argument holder når der translateres med en  $x$ -værdi større end 2. For  $0 \leq x \leq 2$ , vil de to funktioner  $f(x)$  og  $g(x - \alpha)$  overlape i større eller mindre grad. Overlapningen er maksimal for  $x = 1$ . I dette tilfælde får produktfunktionen værdien  $\frac{1}{2}$ , for  $0 \leq \alpha \leq 1$ , og ellers 0. Integralværdien er således lig  $\frac{1}{2}$ . Det er relativt let at udregne foldningsresultatet:

$$f \star g(x) = \begin{cases} x/2 & \text{hvis } 0 \leq x \leq 1 \\ 1 - x/2 & \text{hvis } 1 \leq x \leq 2 \\ 0 & \text{ellers} \end{cases} \quad (2.9)$$

Foldningen af to “firkantfunktioner” er således en “trekant”. Læseren opfordres til at beregne foldningen  $f \star f \star g(x)$ .



### 2.1.1 Impulsfunktionen

Impulsfunktionen  $\delta(x)$  er defineret som den *uegentlige funktion*, der har værdi lig 0 uden for  $x = 0$ , og som har integral lig 1. En impulsfunktion  $\delta(x - x_0)$  i  $x_0$  defineres tilsvarende:

$$\int_{-\infty}^{\infty} \delta(x - x_0) dx = \int_{x_0^-}^{x_0^+} \delta(x - x_0) dx = 1 \quad (2.10)$$

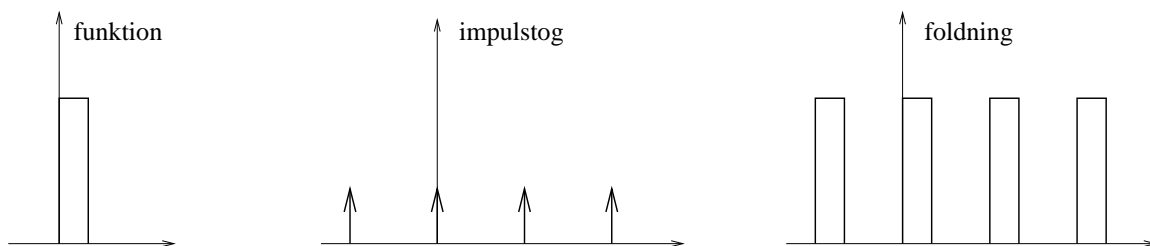
En impulsfunktion tegnes som en pil rettet opad fra  $x$ -aksen. Foldningen af en funktion  $f$  med en impulsfunktion i  $x_0$  er givet ved:

$$f \star \delta(x_0) = \int_{-\infty}^{\infty} f(x) \delta(x_0 - x) dx = f(x_0) \quad (2.11)$$

idet kun værdien i  $x = x_0$  vil bidrage til integrationen. Hvis vi har givet en kontinuert funktion  $f(x)$ , kan vi altså *sample* denne i  $x_0$  ved ovennævnte foldning. Et endeligt/uendeligt *impulstog* er defineret ved en endelig/uendelig følge af impulser:

$$\sum_{n \in \Omega} \delta(x - (nT + x_0)) \quad (2.12)$$

hvor  $\Omega$  er en delmængde af de hele tal, og  $T$  og  $x_0$  er reelle tal. Et impulstog (eller del af et impulstog) tegnes som et antal pile, der udgår fra  $x_0 - nT, \dots, x_0 - T, x_0, x_0 + T, \dots, x_0 + nT$ . Foldningen af en funktion med et impulstog kan betragtes som en kopiering. Tag f.eks. definitionen af  $f(\alpha)$  i (2.7), og antag at  $x_0 = 0$ , og at afstanden  $T$  mellem impulserne i ovenstående tog er større end 1. Da vil resultatet af foldningen være en funktion, der har værdien 1 i alle punkter  $x$ , således at  $(x \bmod T) \leq 1$ , og ellers værdien 0. Dette er illustreret i Figur 2.1. En vigtig an-



Figur 2.1: Foldning af en funktion med et tog af impulsfunktioner

vendelse af impulsfunktioner og impulstog er beskrivelse af sammenhængen mellem kontinuerte og diskrete funktioner. Vi skal senere (i afsnit 2.2.6 om *samlingsætningen*) se hvorledes man kan *sample* en kontinuert funktion ved at multiplicere denne med et impulstog. Impulsfunktionen  $\delta(x)$  er relateret til, men må ikke forveksles med, *indikatorfunktionen*  $\delta_\Omega(x)$  defineret ved:

$$\delta_\Omega(x) = \begin{cases} 1 & \text{hvis } x \in \Omega \\ 0 & \text{ellers} \end{cases} \quad (2.13)$$

### 2.1.2 Den diskrete foldning

Antag at  $f$  hhv.  $g$  er givet ved et endelig sæt af værdier  $(f(0), f(1), f(2), \dots, f(A-1))$  og  $(g(0), g(1), g(2), \dots, g(B-1))$ . Som vi skal se i næste afsnit om fouriertransformationen vil det ofte være en fordel at betragte hvert sæt af værdier som tilhørende en periode af en periodisk funktion.  $f(x)$  er periodisk med perioden  $N$  hvis der for alle heltallige  $k$  gælder at  $f(x + kN) = f(x)$ . Eksempelvis er  $\cos(x)$  periodisk med perioden  $2\pi$ . For at undgå at de første hhv. sidste værdier i en periode influerer på beregningen af de sidste hhv. første værdier af foldningen er det nødvendigt at udvide funktionerne  $f$  og  $g$  ved:

$$f_e(x) = \begin{cases} f(x) & \text{hvis } 0 \leq x \leq A-1 \\ 0 & \text{hvis } A \leq x \leq M-1 \end{cases} \quad (2.14)$$

og

$$g_e(x) = \begin{cases} g(x) & \text{hvis } 0 \leq x \leq B-1 \\ 0 & \text{hvis } B \leq x \leq M-1 \end{cases} \quad (2.15)$$

hvor  $M$  er den fælles periode af de udvidede signaler, og også bliver perioden af foldningsresultatet. Det kan vises at  $M$  skal vælges således at:

$$M \geq A + B - 1 \quad (2.16)$$

Hvis  $M$  vælges mindre end denne størrelse vil de enkelte perioder kunne overlappe (når den ene funktion spejles og translateres). Dette vil ofte give anledning til andre værdier end de forventede. Fænomenet kaldes *warp-around error*. Med ovenstående afklaringer er den diskrete to-dimensionale foldning defineret ved:

$$f_e(x, y) \star g_e(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m, n) g_e(x - n, y - m) \quad (2.17)$$

hvor  $M$  og  $N$  er valgt så de opfylder ovennævnte krav. Som tidligere nævnt kan ovennævnte foldning betragtes som et vægtet gennemsnit.

#### Eksempel

Lad  $f = [1, 1, 1]$  og  $g = [-1, 1, -1]$  forstået således, at  $f(i) = 1$  for  $x$  lig 0, 1 og 2, og ellers lig 0. Vi har altså at  $A = B = 3$  og får  $M = 5$ . En enkelt periode af de udvidede signaler er  $f_e = [1, 1, 1, 0, 0]$  og  $g_e = [-1, 1, -1, 0, 0]$ . Tre perioder af  $f_e$  har værdierne:  $[1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0]$ . Det overlades til læseren at verificere at foldningsresultatet bliver:  $f_e \star g_e = [-1, 0, -1, 0, -1]$ .

## 2.2 Fouriertransformationen

Fouriertransformationen er klassisk inden for al signal- og billedbehandling. Dette skyldes dels historiske forhold, dels inspiration fra fysikken, og dels at en række foldninger kan beregnes

væsentlig hurtigere ved brug af fouriertransformationen. Som beskrevet senere kan det vises, at der gælder en meget nyttig relation mellem foldningen af to funktioner og deres fouriertransformerede (foldningssætningen). Den fouriertransformerede af et signal beskriver frekvensgangen i signalet. Hvis signalet er en simpel sinusbølge, vil den fouriertransformerede være to symmetrisk positionerede impulsfunktioner. Således giver fouriertransformationen mulighed for visuelt at fortolke billedet på en ny måde og for at specificere visse filtre lettere. Matematisk set er fouriertransformationen blot en ud af mange baseskiftende transformationer. Nogle af de billedbehandlingsmæssigt interessante transformationer er beskrevet i kapitel 5.

### 2.2.1 Den kontinuerte fouriertransformation

Lad  $f(x)$  være en kontinuert reel funktion af en reel variabel  $x$ . Antag endvidere at  $f$  er numerisk integrabel, dvs at  $\int_{-\infty}^{\infty} |f(x)| dx < \infty$ . Da er den fouriertransformerede af  $f$  defineret ved.

$$F\{f\} = \mathbf{f}(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx \quad (2.18)$$

hvor  $i = \sqrt{-1}$  er den imaginære enhed, og  $u$  er *frekvensvariablen*. Transformationen sender altså en funktion over i en anden funktion. Ofte skrives:

$$f \bowtie \mathbf{f} \quad (2.19)$$

hvor  $\bowtie$  betyder “har den fouriertransformerede”. Den fouriertransformerede er en kompleks funktion da integranden  $f(x)e^{-2\pi i u x}$  involverer den komplekse eksponentialfunktion. Givet den fouriertransformerede  $\mathbf{f}(u)$  kan  $f(x)$  beregnes ved den *inverse fouriertransformation*:

$$F^{-1}\{\mathbf{f}\} = f(x) = \int_{-\infty}^{\infty} \mathbf{f}(u) e^{2\pi i u x} du \quad (2.20)$$

Da  $\mathbf{f}(u)$  er kompleks, kan den skrives som en sum af en realdel og en imaginærdel:

$$\mathbf{f}(u) = R(u) + iI(u) \quad (2.21)$$

Ofte skrives  $\mathbf{f}(u)$  dog på formen:

$$\mathbf{f}(u) = |\mathbf{f}(u)| e^{i\phi(u)} \quad (2.22)$$

hvor

$$|\mathbf{f}(u)| = \sqrt{R^2(u) + I^2(u)} \quad (2.23)$$

betegnes *fourierspektret*. Kvadratet

$$S(u) = |\mathbf{f}(u)|^2 \quad (2.24)$$

på denne størrelse kaldes for *styrkespektret* for  $f$ .  $S(u)$  angiver hvor meget energi i signalet, som er knyttet til frekvensen  $u$ . Størrelsen:

$$\phi(u) = \arctan\left(\frac{I(u)}{R(u)}\right) \quad (2.25)$$

kaldes *fasevinklen* (eller blot fasen). At  $u$  kaldes for en frekvensvariabel er inspireret af *Eulers* formel :

$$e^{2\pi i u x} = \cos(2\pi u x) + i \sin(2\pi u x) \quad (2.26)$$

Den to-dimensionale kontinuerte fouriertransformerede er defineret på tilsvarende måde:

$$F\{f\} = \mathbf{f}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2\pi i (ux + vy)} dx dy \quad (2.27)$$

### Eksempel

Lad:

$$f(x) = \begin{cases} H & \text{hvis } A \leq x \leq B \\ 0 & \text{ellers} \end{cases} \quad (2.28)$$

Funktionen  $f$  er klart numerisk integrabel. Den fouriertransformerede er:

$$\mathbf{f}(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx \quad (2.29)$$

$$= H \int_A^B e^{-2\pi i u x} dx \quad (2.30)$$

$$= \frac{-H}{2\pi i u} [e^{-2\pi i u x}]_A^B \quad (2.31)$$

$$= \frac{-H}{2\pi i u} (e^{-2\pi i u B} - e^{-2\pi i u A}) \quad (2.32)$$

I tilfældet  $A = 0$  får vi:

$$\mathbf{f}(u) = \frac{-H}{2\pi i u} (e^{-2\pi i u B} - 1) \quad (2.33)$$

$$= \frac{-H}{2\pi i u} [e^{-\pi i u B} - e^{\pi i u B}] e^{-\pi i u B} \quad (2.34)$$

$$= \frac{H}{\pi u} \sin(\pi B u) e^{-\pi i u B} \quad (2.35)$$

Den transformerede ses at være kompleks. Styrkespektret for  $f$  er bestemt ved:

$$|\mathbf{f}(u)| = H B \left| \frac{\sin(\pi B u)}{\pi B u} \right| \quad (2.36)$$

I tilfældet at  $A = -B$  får vi:

$$\mathbf{f}(u) = \frac{-H}{2\pi i u} (e^{-2\pi i u B} - e^{2\pi i u B}) \quad (2.37)$$

$$= 2BH \frac{\sin(2\pi B u)}{2\pi B u} \quad (2.38)$$

Her ser vi, at den transformerede er reel. Vi husker at der om en lige hhv. ulige funktion gælder at  $f(-x) = f(x)$  hhv.  $f(-x) = -f(x)$ . Det kan vises, at en hver lige funktion har en fouriertransformeret med imaginærkomponent lig nul. Omvendt gælder, at enhver reel funktion har en fouriertransformeret, der er lige. Er  $f$  ulige ses tilsvarende at  $\mathbf{f}$  vil være rent imaginær.

### Eksempel

Den fouriertransformerede af en impulsfunktion i  $x_0$  er givet ved:

$$\mathbf{f}(u) = \int_{-\infty}^{\infty} \delta(x - x_0) e^{-2\pi i u x} dx \quad (2.39)$$

$$= e^{-2\pi i u x_0} \quad (2.40)$$

Hvis  $x_0 = 0$  fås altså at  $\mathbf{f}(u)$  er lig konstanten 1. Prøv at udregne den fouriertransformerede af  $\delta(x - x_0) + \delta(x + x_0)$ .

### Eksempel

Lad  $f(x) = e^{-\alpha|x|}$ . Verificer at den fouriertransformerede bliver:

$$\mathbf{f}(u) = \int_{-\infty}^{\infty} e^{-\alpha|x|} e^{-2\pi i u x} dx \quad (2.41)$$

$$= \int_0^{\infty} e^{2\pi i u x - \alpha x} dx + \int_0^{\infty} e^{-2\pi i u x - \alpha x} dx \quad (2.42)$$

$$= \int_0^{\infty} 2 \cos(2\pi u x) e^{-\alpha x} dx \quad (2.43)$$

$$= \frac{2\alpha}{\alpha^2 + 4\pi^2 u^2} \quad (2.44)$$

## 2.2.2 Den diskrete fouriertransformation

Antag at der er givet  $N$  stikprøver (samples)  $(f(0), f(1), f(2), \dots, f(N-1))$  udtaget med ækvidistant afstand fra en en-dimensional funktion  $f$ . Hver af de samlede værdier kan være reelle eller komplekse. Den *diskrete* fouriertransformation og dens inverse er da givet ved:

$$\mathbf{f}(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-2\pi i u x / N} \quad (2.45)$$

$$f(x) = \sum_{u=0}^{N-1} \mathbf{f}(u) e^{2\pi i u x / N} \quad (2.46)$$

for  $u$  hhv.  $x$  lig  $0, 1, 2, \dots, N - 1$ . Tilsvarende formler gælder i det to-dimensionale tilfælde:

$$\mathbf{f}(u, v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-2\pi i(ux/N + vy/M)} \quad (2.47)$$

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \mathbf{f}(u, v) e^{2\pi i(ux/N + vy/M)} \quad (2.48)$$

for  $u$  og  $x \in [0 \dots N - 1]$ , og for  $v$  og  $y \in [0 \dots M - 1]$ .

Beviser for, at ovenstående ligninger på fornuftig måde definerer den diskrete fouriertransformation, ligger uden for rammerne af disse noter. Det bør bemærkes, at der i det diskrete tilfælde ikke stilles krav om integrabiliteten, da den diskrete fouriertransformation består af endelige summer, og derfor altid er defineret.

I billedbehandling benyttes der ofte kvadratiske billeder ( $N=M$ ). I dette tilfælde er det sædvanen at gøre den *forlæns* og den *baglæns* fouriertransformation symmetriske ved at splitte normaliseringskonstanten ( $\frac{1}{N^2}$ ) op i to:

$$\mathbf{f}(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i(ux + vy)/N} \quad (2.49)$$

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \mathbf{f}(u, v) e^{2\pi i(ux + vy)/N} \quad (2.50)$$

Som vi senere skal se, eksisterer der en hurtig algoritme (FFT-algoritmen) til beregning af den fouriertransformerede. Denne metode kræver at såvel  $N$  som  $M$  er potenser af 2.

En grundlæggende antagelse er at de  $N$  samples ( $f(0), f(1), f(2), \dots, f(N - 1)$ ) er udtaget med konstant afstand  $\Delta x$ . Denne antagelse har dog kun betydning for en fortolkning af transformationerne. Tilsvarende beskriver de  $N$  værdier af den fouriertransformerede en fortolkning hvor frekvensvariablen er samlet med en konstant afstand  $\Delta u$ . Som vi skal se senere i afsnit 2.2.6 om *samplingsætningen*, gælder der følgende relation:

$$\Delta u = \frac{1}{N\Delta x} \quad (2.51)$$

For fast  $N$  ses, at jo mindre samplingafstanden  $\Delta x$  er, jo større bliver  $\Delta u$ . Da der ialt er  $N$  samples af frekvensvariablen  $u$ , betyder dette, at højere frekvenser af signalet bliver repræsenteret. For fast  $\Delta x$  betyder færre/flere samples  $N$  blot færre/flere samples af frekvensvariablen, men ikke at det repræsenterede frekvensområde ændres.

### 2.2.3 Egenskaber ved fouriertransformationen

Fouriertransformationen har en lang række af egenskaber, der er vigtige at kende ved praktisk anvendelse. Af definitionen på transformationen følger direkte, at denne er lineær:

$$F\{f + g\} = F\{f\} + F\{g\} \quad (2.52)$$

$$F\{\alpha f\} = \alpha F\{f\} \quad (2.53)$$

Desuden gælder der at:

$$f(\alpha x) \propto \frac{1}{|\alpha|} \mathbf{f}(u/\alpha) \quad (2.54)$$

Fouriertransformationen er separabel, idet der gælder:

$$F(u, v) = \frac{1}{NM} \sum_{x=0}^{N-1} e^{-2\pi i u x / N} \left[ \sum_{y=0}^{M-1} f(x, y) e^{-2\pi i v y / M} \right] \quad (2.55)$$

Dette betyder, at en implementation af den to-dimensionale fouriertransformation blot kræver, at vi kan foretage en en-dimensional fouriertransformation. For et kvadratisk billede og ved brug af normaliseringskonstanten  $1/N$  jvf. (2.50) gælder således:

$$\mathbf{f}(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \hat{f}(x, v) e^{-2\pi i u x / N} \quad (2.56)$$

$$\hat{f}(x, v) = N \left[ \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i v y / N} \right] \quad (2.57)$$

En anden egenskab ved fouriertransformationen er, at den komplekse eksponentialfunktion  $e^{-2\pi i u x / N}$  er periodisk med periode  $N$ . Dette betyder at:

$$\mathbf{f}(u, v) = \mathbf{f}(u + N, v) = \mathbf{f}(u, v + N) = \mathbf{f}(u + N, v + N) \quad (2.58)$$

Det er her værd at erindre (se afsnit 2.1.2), hvorledes vi ved den diskrete foldning udvidede signalerne, sådan at disse havde samme fælles periode.

Det er desuden let at vise, at hvis  $f$  er reel, da er den fouriertransformerede konjugeret symmetrisk, dvs.:

$$\mathbf{f}(u, v) = \mathbf{f}^*(-u, -v) \quad (2.59)$$

Dette betyder at:

$$|\mathbf{f}(u, v)| = |\mathbf{f}(-u, -v)| \quad (2.60)$$

Endelig skal det nævnes, at den fouriertransformerede af et billede, der er roteret  $\theta$  grader i forhold til en reference, også vil være roteret  $\theta$  grader i samme retning. Udtrykt i polære koordinater gælder altså:

$$f(r, \psi + \theta) \propto \mathbf{f}(\omega, \phi + \theta) \quad (2.61)$$

hvor  $(r, \psi)$  er tidsvariable og  $(\omega, \phi)$  er frekvensvariable.

### Lokalisering og værdi i Origo for fourierspektret

Et praktisk problem ved visuel inspektion af den fouriertransformerede af et kvadratisk billede er at Origo for den betragtede periode vil være lokaliseret i såvel  $(0,0)$ ,  $(0,N)$ ,  $(N,0)$  og  $(N,N)$ , hvor  $N$  (et multipla af 2) er kantlængden for billedet. Visuelt vil det være at foretrække at have Origo centreret i  $(N/2, N/2)$ . Generelt er translationsegenskaberne for den fouriertransformerede givet ved:

$$f(x, y)e^{2\pi i(u_0x+v_0y)/N} \propto F(u - u_0, v - v_0) \quad (2.62)$$

Idet vi lader  $u_0 = v_0 = N/2$ , og bemærker at:

$$e^{2\pi i(xN/2+yN/2)/N} = e^{\pi i(x+y)} = (-1)^{x+y} \quad (2.63)$$

ses, at translationen fremkommer ved fouriertransformation af  $(-1)^{x+y}f(x, y)$ .

Ved visuel inspektion af fourierspektret er vi ofte kun interesserede i at se styrkespektret  $|F(u, v)|$ . I praksis er en så stor del af energien i fourierspektret samlet i Origo, at ved en lineær skalering vil vi kun se Origo som en lys prik på en sort baggrund. Ofte benyttes derfor skaleringen:

$$V(u, v) = \log(1 + |F(u, v)|) \quad (2.64)$$

hvorpå  $V(u, v)$  skaleres lineært eller histogramlineariseres (se evt. afsnit 6.1), sådan at alle intensitetsværdier udnyttes.

Værdien af den fouriertransformerede i Origo er relateret til middelintensiteten i billedet. Dette følger direkte af definitionen på den diskret fouriertransformerede:

$$\mathbf{F}(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)e^{-2\pi i(ux+vy)/N} \quad (2.65)$$

ved substitution af  $(u, v)$  med  $(0, 0)$ . Den fundne relation bliver:

$$\mathbf{F}(0, 0) = N\bar{f}(x, y) \quad (2.66)$$

hvor  $\bar{f}$  er gennemsnittet af  $f$ . Som det ses, giver formelen mulighed for direkte at angive den ønskede middelintensitet i et filtreret billede, blot ved at sætte værdien af den fouriertransformerede i  $(0, 0)$ .

### Afledede

Som vi skal diskutere nøjere i kapitel 10 om kantdetektion, er har man ofte brug for at måle de partielle afledede af en intensitetsfunktion. Vi skal ikke her komme ind på de problemer der opstår, hvis billedet indeholder støj etc., men betragte det ideelle kontinuerte tilfælde, hvor billedet antages uendeligt ofte differentiabelt. Da gælder:

$$F\left\{\frac{\partial^n f}{\partial x^n}\right\} = (2\pi i)^n u^n F\{f\} \quad (2.67)$$

hvor  $F\{f\}$  her angiver fouriertransformation af  $f$ , og  $u$  er frekvensvariablen. Dette ses let af:



$$\frac{\partial^n f}{\partial x^n} = \frac{\partial^n}{\partial x^n} \int \mathbf{f} \cdot e^{2\pi i u x} du = \int (2\pi i u)^n \cdot \mathbf{f} \cdot e^{2\pi i u x} du = F^{-1}\{(2\pi i u)^n \cdot \mathbf{f}\}$$

Ønsker vi eksempelvis at anvende Laplaceoperatoren  $(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})$  på et billede  $I$  kan dette foregå ved transformationen:  $F^{-1}\{\mathbf{h}F\{I\}\}$ , hvor frekvensfiltret  $\mathbf{h}$  er givet ved:

$$\mathbf{h}(u, v) = -4\pi^2(u^2 + v^2) \quad (2.68)$$

Tilsvarende kan andre differentialoperatorer defineres i frekvensområdet. Det skal bemærkes at i praksis vil denne metode være problematisk, idet billeder ofte indeholder højfrekvent støj, og at denne fremhæves markant ved, som i ovenstående eksempel, at multiplicere frekvensbilledet med et polynomie  $(u^2 + v^2)$ .

### Energibevarelse ved fouriertransformation

Energien i et signal er defineret ved integralet af de kvadrerede signalværdier. *Parsevals sætning* udtrykker at at denne energi kan måles lige vel i sted- som i frekvensområdet:

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\mathbf{f}(u)|^2 du \quad (2.69)$$

Mere generelt gælder der:

$$\int_{-\infty}^{\infty} f(x)g^*(x)dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathbf{f}(u)\mathbf{g}^*(u)du \quad (2.70)$$

hvor  $g^*(x)$  er den komplekst konjugerede af  $g(x)$ .

### 2.2.4 Foldningssætningen

Som tidligere beskrevet kan en meget stor del af de klassiske (og nyere) metoder til billedfiltrering beskrives ved foldningen af billedet med et filter. En af de væsentligste grunde til at der er ofret så meget tid på fouriertransformationen er, at fouriertransformationen af en foldning er lig produktet af de fouriertransformerede:

$$f(x) \star g(x) \rightsquigarrow \mathbf{f}(u)\mathbf{g}(u) \quad (2.71)$$

og ækvivalent:

$$f(x)g(x) \rightsquigarrow \mathbf{f}(u) \star \mathbf{g}(u) \quad (2.72)$$

Dette resultat går under navnet *foldningssætningen*. Et bevis for sætningen ligger uden for rammerne af disse noter. I det diskrete tilfælde gælder tilsvarende sætninger. Ved anvendelse af foldningssætningen er det nødvendigt at sikre sig mod *warp-around error* ved at udvide billederne passende. For store billeder er så meget af energien i disse i praksis samlet i og omkring Origo,

at udvidelserne ikke altid foretages. Dette vil naturligvis resultere i en fejl, der dog ofte er så lille at den kan ignoreres.

Foldningssætningen siger altså, at vi kan formulere filtreringen i tidsområdet eller i frekvensområdet efter behag. Dette skal vi udnytte i de følgende afsnit. En anden fordel som sætningen giver, er, at visse filtreringer kan beregnes mere effektivt i frekvensområdet. Dette bygger på at FFT-algoritmen som beskrives i afsnit 2.3, har en kompleksitet på  $O(N \log N)$  for et signal på  $N$  elementer. Implementeres foldningen af de to signaler direkte fås en tidskompleksitet på  $O(N^2)$ . Ved først at fouriertransformere de to funktioner, derefter at multiplicere resultaterne element for element, og sluttelig invers fouriertransformere produktet fås igen en kompleksitet på  $O(N \log N)$ . For store værdier af  $N$  giver en implementation v.h.a. foldningssætningen altså mulighed for en “speed-up”. Denne kan være betragtelig, især hvis den fouriertransformerede af filteret kan beregnes på forhånd.

### Eksempel

Antag et billede af dimension  $N \times N$  og et filter med støtte  $M \times M$ , hvor  $N > M$  er potenser af 2. Implementeres foldningen direkte er det totale antal multiplikationer lig  $N^2 M^2$ . Benyttes fouriertransformationen (FFT-algoritmen) først på billedet samt det udvidede filter (så det får samme støtte  $N$  som billedet), og til slut på resultatet af den elementvise multiplikation af de to transformerede billeder, fås ialt  $N^2(6 \log(N) + 1)$  multiplikationer.

### 2.2.5 Korrelationssætningen

Korrelation eller mere præcist Krydskorrelation af to kontinuerte, reelle funktioner med begrænset numerisk integral er defineret ved:

$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f^*(\alpha) g(x + \alpha) d\alpha \quad (2.73)$$

Som det ses ligner foldninger og korrelationer hinanden meget. Forskellen er kun, at i korrelationen er funktionen  $f$  komplekst konjugeret og minuset i translationen er ændret til et plus. De fleste af sætningerne om foldningen og sammenhæng mellem foldning og fouriertransformation kan gentages her. Således gælder korrelationssætningen:

$$f(x) \circ g(x) \bowtie \mathbf{f}^*(u) \mathbf{g}(u) \quad (2.74)$$

og ækvivalent:

$$f^*(x) g(x) \bowtie \mathbf{f}(u) \circ \mathbf{g}(u) \quad (2.75)$$

Autokorrelationen af en funktion  $f$  er defineret ved  $f(x) \circ f(x)$ . Det er let at vise at der gælder:

$$f(x) \circ f(x) \bowtie |\mathbf{f}(u)|^2 \quad (2.76)$$

og

$$|f(x)|^2 \bowtie \mathbf{f}(u) \circ \mathbf{f}(u) \quad (2.77)$$

Korrelationer kan, som foldninger, implementeres effektivt ved brug af fouriertransformationen.

Korrelationer anvendes meget i billedbehandling til at finde “match” mellem en given prototype (eller ‘template’) og et forelagt billede. Typisk er prototype-billedet lille (f.eks.  $32 \times 32$ ). Vi ønsker nu, forelagt et (stort) billede at finde de positioner hvor billedet i det tilsvarende lille udsnit stemmer godt overens med prototypen. Dette gøres ved at detektere de lokale maksima i resultatet af korrelationen mellem prototypen og billedet. Denne anvendelse er typisk i *tracking*, hvor man søger korrespondance mellem punkter i en billedsekvens.

Hvis prototypen (som i ovenstående eksempel) er reel, da adskiller korrelationen sig kun fra foldningen ved den manglende spejling. I eksemplet fra afnit 2.1 om foldninger er det let at se, at maksimum i korrelationsfunktionen (korrelationen mellem  $f$  og  $g$ ) vil antages i  $x = 0$ , og ikke som ved foldningen i  $x = 1$ . Da funktionerne fra dette eksempel er symmetriske, vil formen af korrelationsfunktionen være ens med resultatet af foldningen, dvs. en “trekantsfunktion”.

Bemærk at der gælder at:

$$\int (f(\alpha) - g(x + \alpha))^2 d\alpha = (f \circ f)(0) + (g \circ g)(0) - 2(f \circ g)(x) \quad (2.78)$$

Maksimering af korrelationen er således ækvivalent med minimering af summen af kvadrerede differencer. Ofte ønsker at angive en grænseværdi for hvor dårligt de to funktioner må ligne hinanden. For minimering af summen af kvadratforskellene kan en absolut grænseværdi tit angives direkte. For maksimering af korrelationen er det sværere at angive en absolut minimal grænseværdi. I stedet benyttes ofte en relativ værdi  $\beta(f \circ f(1))$ , hvor  $\beta$  er en konstant. Her accepteres et match kun hvis funktionerne passer bedre sammen end  $f$  passer til en translation på en pixel af  $f$  selv.

Et alternativt til den rå korrelation er, at normere korrelationen ved at skalere funktionerne  $f$  og  $g$  i forhold til den energi, som signalerne indeholder:

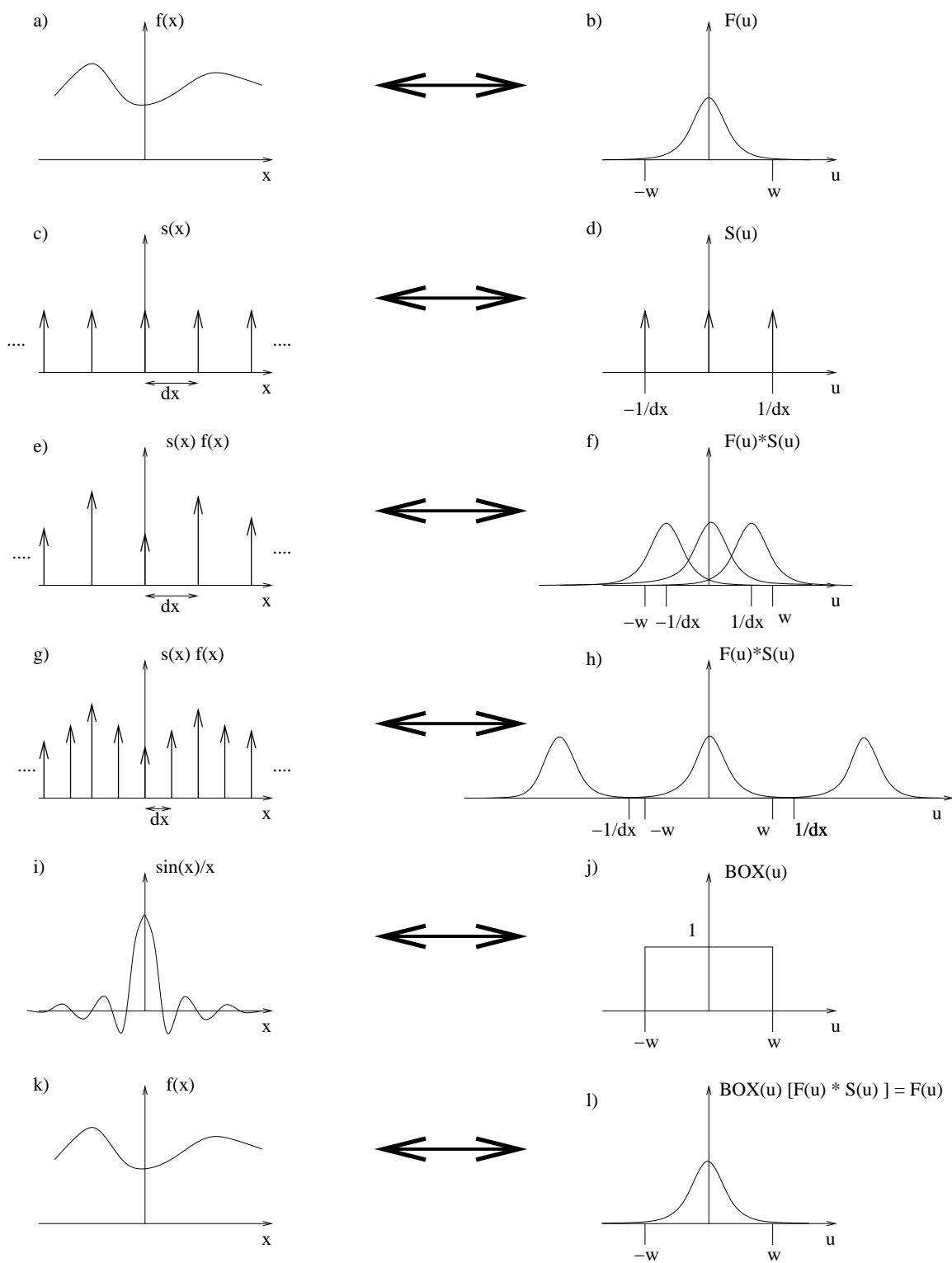
$$\tilde{f}(x) = \frac{f(x)}{\sqrt{(f \circ f)(0)}} \quad \text{og} \quad \tilde{g}(x) = \frac{g(x)}{\sqrt{(g \circ g)(0)}} \quad (2.79)$$

Det overlades til læseren at udlede en ligning svarende til (2.78) for de normerede funktioner.

### 2.2.6 Samplingsætningen

Samplingsætningen omhandler, hvor mange samples (stikprøver) af en kontinuert funktion vi skal tage for at sikre, at der ikke går information tabt i samplingen. Eksempelvis kræver en konstant funktion kun én sampel. I det følgende beskrives vha. tegneserier hvad der sker ved sampling af en en-dimensional funktion. Resultaterne for det to-dimensionale tilfælde er analoge.

Lad  $f(x)$  være en funktion, der antages defineret på hele den reelle akse. Funktionen  $f(x)$  kaldes for *båndbegrænset* med båndbredde  $w$ , hvis den fouriertransformerede  $\mathbf{f}(u)$  er konstant lig 0 uden for frekvensintervallet  $[-w, w]$ . Figur 2.2 a) og b) illustrerer en båndbegrænset funktion. Bemærk at der i figuren benyttes store bogstaver, f.eks.  $F(u)$  om den fouriertransformerede  $\mathbf{f}(u)$ .



Figur 2.2: Grafisk illustration af effekten af en uendelig sampling

### Den fouriertransformerede af et impulstog

I Figur 2.2 c) er der vist et uendeligt impulstog  $s(x)$ . Ved multiplikation af  $f(x)$  med dette tog fås samplingen vist i 2.2 e). Billederne på højresiden af figuren viser de fouriertransformerede af funktionerne vist i billederne på venstresiden af figuren. 2.2 c) og d) illustrerer, at den fouriertransformerede  $S(u)$  af et uendeligt impulstog  $s(x)$  også er et impulstog med impulsafstand, der er omvendt proportional med impulsafstanden i  $s(x)$ , dvs:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad \Leftrightarrow \quad S(u) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta\left(u - \frac{n}{T}\right) \quad (2.80)$$

Dette resultat er ikke oplagt. Et bevis falder uden for rammerne af disse noter, dog kan følgende løse argumenter sandsynliggøre resultatet: Betragt funktionen  $S_1$  bestående af tre impulsfunktioner lokaliseret i  $-1/T$ , i 0 og i  $1/T$ . Af definitionen på fouriertransformationen følger direkte at den fouriertransformerede  $s_1$  af  $S_1$  vil være summen af en konstant fra den midterste impulsfunktion og en cosinus fra de to symmetrisk positionerede impulsfunktioner:

$$s_1(u) = 1 + 2 \cos(2\pi u/T) \quad (2.81)$$

Denne funktion har lokale maksima for  $u = pT$ , hvor  $p$  er et heltal. Konstruer nu  $S_2, S_3$  etc. ved at tilføje impulsfunktioner i  $-2T$  og  $2T$ , i  $-3T$  og  $3T$  etc. Herved adderer vi til den fouriertransformerede nye cosinusfunktioner med dobbelt, tredobbelt etc. frekvens. I maksimumspunkterne har cosinusfunktionerne med dobbelt, tredobbelt etc. frekvens også lokalt maksimum. I intervallerne mellem de lokale maksima vil funktionerne svinge hurtigere og hurtigere, og vil have en tendens til af ophæve hinanden. I grænsen fås:

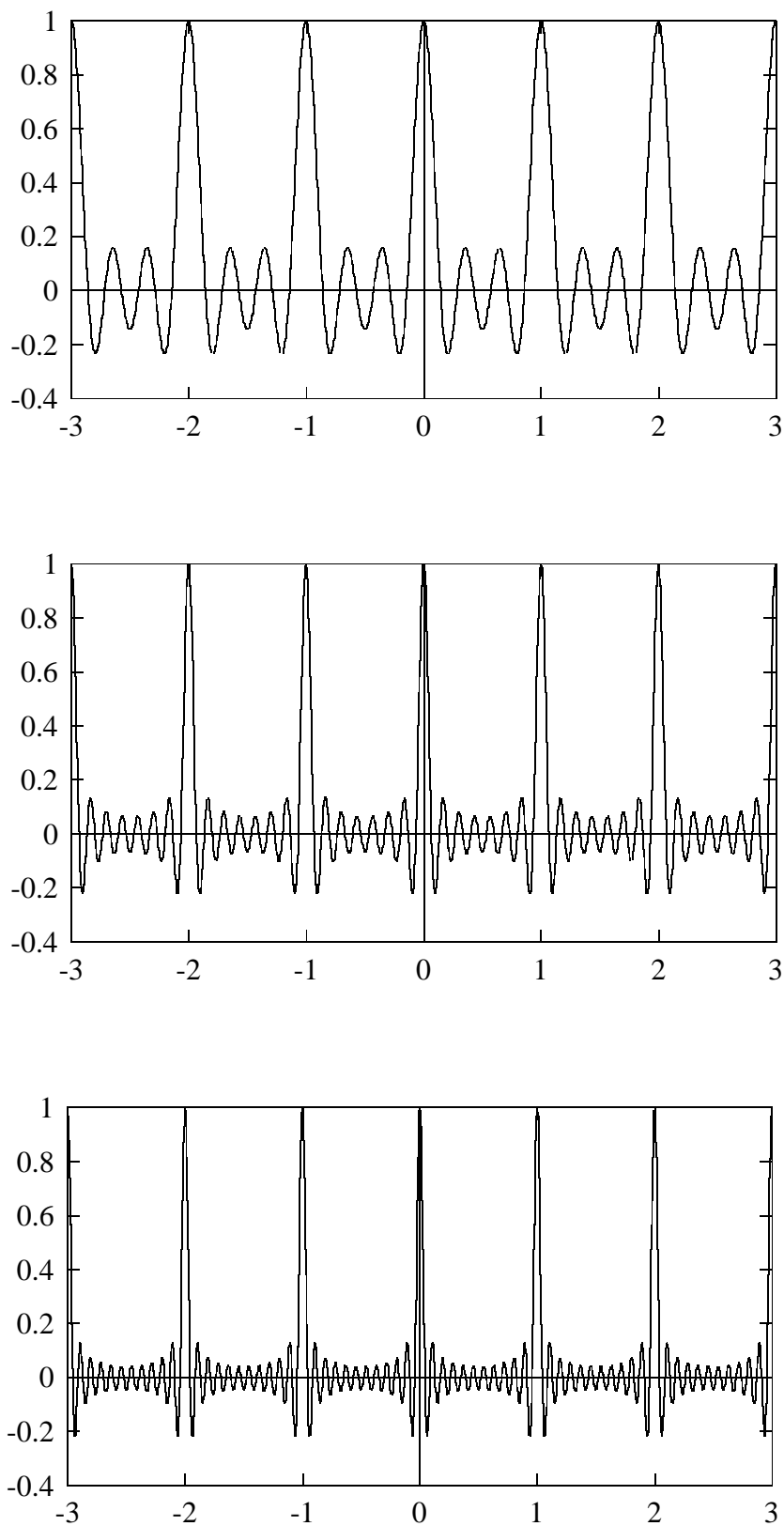
$$s_{\infty}(u) = \lim_{K \rightarrow \infty} \left[ 1 + 2 \sum_{n=1}^K \cos(2\pi n u/T) \right] = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (2.82)$$

Funktionerne  $s_3, s_7$  og  $s_{11}$  er vist nedenfor i Figur 2.3.

Vi kan nu fortsætte med beskrivelsen af samplingsætningen. Multiplikationen af  $f(x)$  med impulstog  $s$  svarer i fourierdomænet til en foldning af  $F(u)$  med  $S(u)$ . Denne foldning kan, som tidligere beskrevet, betragtes som en kopiering. Herved fremkommer 2.2 f). Som det ses er afstanden  $dx$  mellem samples  $s(x)$  valgt stor. Dette betyder at afstanden mellem de tilsvarende impulsfunktioner af den fouriertransformerede er lille. Herved kommer de enkelte perioder af foldningsresultatet til at overlappe. Dette fænomen kaldes *aliasing*. Aliasing vil umuliggøre en eksakt rekonstruktion af den kontinuerte funktion  $f(x)$ . For at undgå aliasing må der oplagt kræves at:

$$w \leq \frac{1}{2dx} \quad (2.83)$$

En situation hvor  $dx$  er valgt sådan, at ovenstående ulighed er opfyldt er vist i Figur 2.2 g) og h). Her er de enkelte perioder af foldningsresultatet vel separeret. Figur 2.2 j) og l) viser, hvorledes det ved multiplikation af foldningsresultatet med et box-filter er muligt at ekstrahere præcis en periode. Ved invers fouriertransformation af denne funktion fås den oprindelige *kontinuerte* funktion  $f(x)$ . Vi kan derfor opsummere:

Figur 2.3: Funktionerne  $s_3(u)$ ,  $s_7(u)$ , og  $s_{11}(u)$

Antag at den kontinuerte funktion  $f(x)$  er defineret for alle reelle  $x$  og er båndbegrænset med båndbredde  $w$ . Lad  $g(x)$  være en sampling af  $f$  med samplingafstand  $dx \leq 1/(2w)$ . Da kan den kontinuerte funktion  $f$  rekonstrueres eksakt ud fra de diskrete samples ved at invers fouriertransformere produktet af et box-filter med støtte  $[-w, w]$  og den fouriertransformerede af  $g$ .

### Sampling af endelige funktioner

Antag at  $f(x)$  er defineret på hele den reelle akse, og at  $f(x)$  er båndbegrænset med båndbredde  $w$ , og betragt en sampling således, at samplingsætningen er opfyldt. Dette er illustreret i Figur 2.4 a) til f). I Figur 2.4 g) og h) ses hvorledes et endeligt antal samples kan ekstraheres ved multiplikation med et vindue. Som tidligere beskrevet svarer dette (i fourierdomænet) til en foldning med den fouriertransformerede af vinduesfunktionen. Figur 2.4 j) viser resultatet af denne foldning. Da  $H(u)$  har frekvensværdier, der fortsætter mod uendelig, forvrænges frekvensspektret i forhold til det viste i b). Selv om samplingen opfylder betingelserne for samplingsætningen, er det derfor ikke længere muligt at rekonstruere  $f$ , fordi det oprindelige fourierspektrum ikke kan rekonstrueres. Bemærk at ved invers fouriertransformation fås en funktion, der er defineret på hele den reelle akse, og som *ikke* er lig nul uden for det område, hvori  $f$  oprindeligt er samplet. Således gælder der at:

Der eksisterer ingen funktioner med endelig støtte som er båndbegrænsede. Omvendt gælder, at enhver båndbegrænset funktion har uendelig støtte.

### Periodicitet af den diskrete fouriertransformation

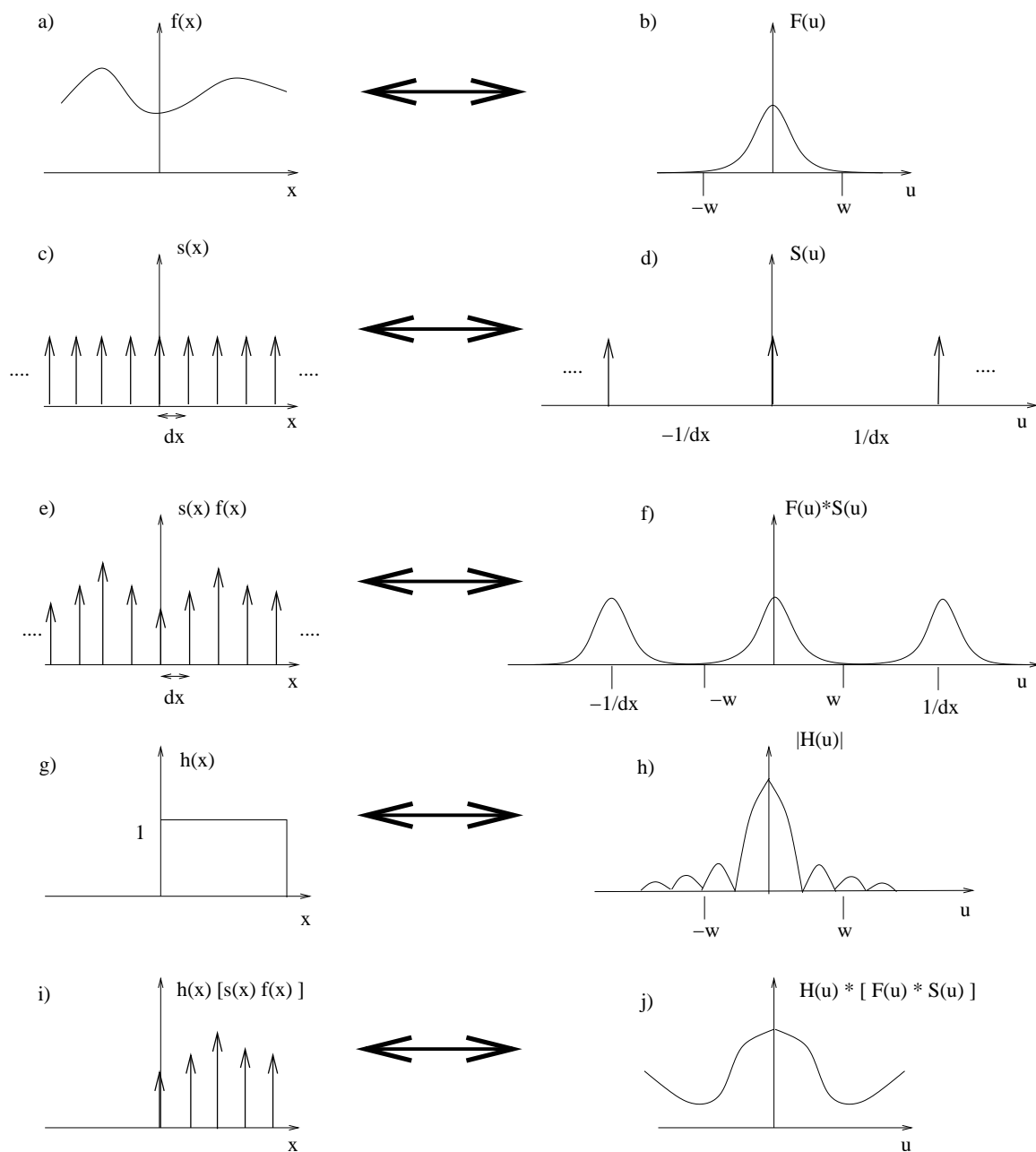
I den foregående diskussion har alle funktioner i frekvensområdet været kontinuerte. For at opnå en diskret repræsentation af en funktion i frekvensområdet multiplicerer vi blot denne med et impulstog. Figur 2.5 illustrerer (som en fortsættelse af Figur 2.4) hvorledes denne sampling foregår. I  $x$ -domænet svarer multiplikationen til en foldning med et impulstog med impulsafstand på  $1/du$ . Derved bliver foldningen, som vist i Figur 2.5 e), periodisk med periode  $1/du$ . Dette forudsætter naturligvis at  $du$  er valgt tilstrækkelig lille jvf. samplingsætningen. Funktionen  $f$  såvel som dens diskret fouriertransformerede er repræsenteret ved  $N$  samples. Disse er udtaget ækvidistant, således at hver periode er dækket. Dette betyder at  $Ndx = X$  i  $x$ -domænet og tilsvarende at  $Ndu = 1/dx$  i frekvensdomænet. Der gælder altså at:

$$du = \frac{1}{Ndx} \quad (2.84)$$

## 2.3 FFT-algoritmen

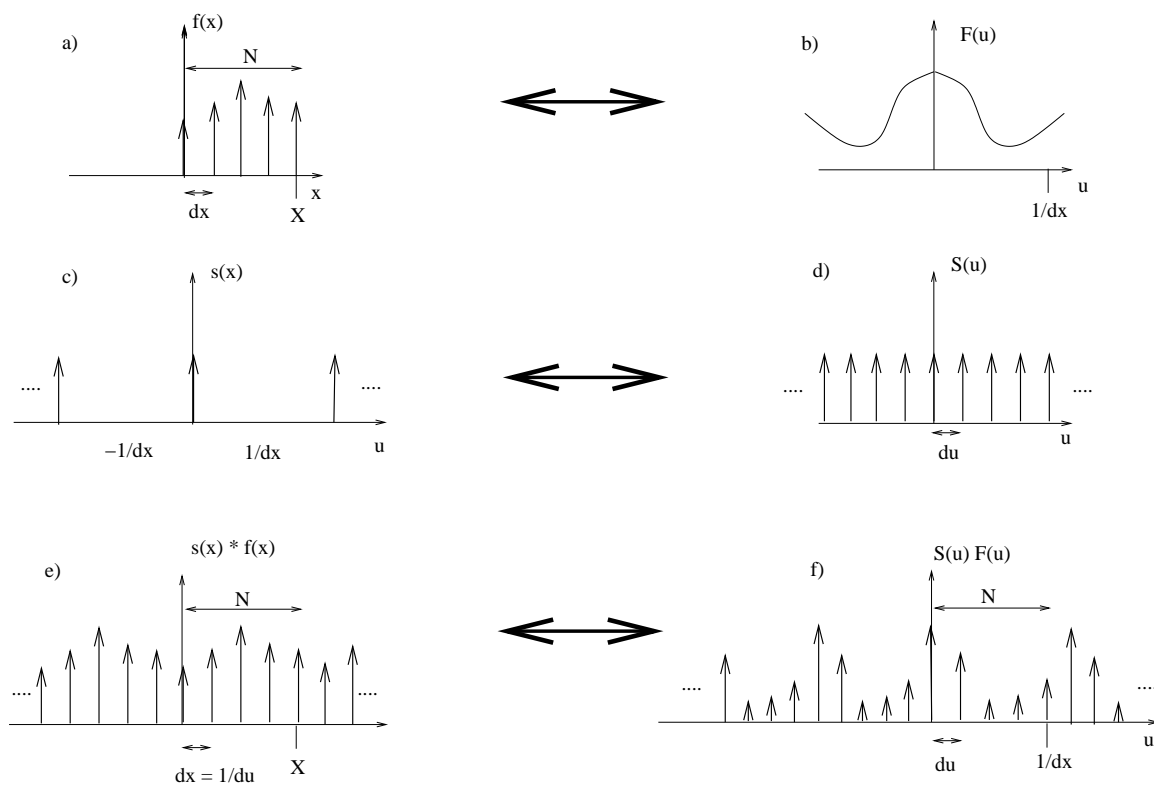
Betragt den diskrete en-dimensionale fouriertransformation:

$$\mathbf{f}(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) W_N^{ux} \quad (2.85)$$



Figur 2.4: Grafisk illustration af effekten af en endelig sampling





Figur 2.5: Grafisk illustration af den diskrete fouriertransformation

hvor  $N$  er antal samples, hvor:

$$W_N = e^{-2\pi i/N} \quad (2.86)$$

og hvor faktoren  $W_N^{ux}$  skal læses “ $W_N$  ophævet til potensen  $ux$ ”. For hver af de  $N$  værdier af  $u$  omfatter transformationen  $N$  komplekse multiplikationer af  $f(x)$  med  $W_N^{ux} = e^{-2\pi i x u/N}$  og  $N - 1$  additioner. Multiplikationer mv. til udregning af de  $N$  værdier af den komplekse eksponentialfunktion medregnes ikke, da disse kan tabelleres på forhånd. Den en-dimensionale fouriertransformation har derfor en tidskompleksitet på  $O(N^2)$ .

I praksis beregnes fouriertransformationer ved anvendelse af *den hurtige fouriertransformation*: FFT-algoritmen. Det kan vises at antallet af (komplekse) multiplikationer hhv. additioner for denne metode er  $\frac{N}{2} \log N$  hhv.  $N \log N$ . I det følgende skitseres kernen af den FFT-algoritme, der er baseret på den såkaldte “successive doubling”-metode. Dette er den mest generelle og mest benyttede FFT-algoritme. En grundigere behandling findes f.eks. i [6].

En begrænsende antagelse for FFT-algoritmen er, at antallet af samples er en potens af 2, dvs:  $N = 2^n$ , hvor  $n$  er et positivt heltal. I dette tilfælde gælder, at  $N = 2M$ . Jævnfør ovenstående definition af den diskrete fouriertransformation gælder der derfor:

$$\mathbf{f}(u) = \frac{1}{2M} \sum_{x=0}^{2M-1} f(x) W_{2M}^{ux} \quad (2.87)$$

$$= \frac{1}{2} \left[ \frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_{2M}^{u(2x)} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_{2M}^{u(2x+1)} \right] \quad (2.88)$$

$$= \frac{1}{2} \left[ \frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_M^{ux} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_M^{ux} W_{2M}^u \right] \quad (2.89)$$

hvor vi blot har opdelt summen i to led efter lige og ulige værdier af argumentet til  $f$ , og udnyttet at  $W_{2M}^{2ux} = W_M^{ux}$ . Det er nu oplagt at definere:

$$\mathbf{f}_{lige}(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_M^{ux} \quad (2.90)$$

$$\mathbf{f}_{ulige}(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_M^{ux} \quad (2.91)$$

for  $u = 0, 1, 2, \dots, M - 1$ . Med disse forkortelser kan vi skrive:

$$\mathbf{f}(u) = \frac{1}{2} [\mathbf{f}_{lige}(u) + \mathbf{f}_{ulige}(u) W_{2M}^u] \quad (2.92)$$

$$\mathbf{f}(u+M) = \frac{1}{2} [\mathbf{f}_{lige}(u) - \mathbf{f}_{ulige}(u) W_{2M}^u] \quad (2.93)$$

hvor minuset i den sidste ligning fås fordi  $W_{2M}^M = -1$ . Formlerne beskriver en effektiv rekursiv metode til beregning af fouriertransformationen, idet en  $N$ -samples transformation opdeles i to grupper. I den første gruppe beregnes  $\mathbf{f}(0)$  til  $\mathbf{f}(N/2 - 1)$  ved evaluering af to  $N/2$ -samples transformationer. Værdierne af  $\mathbf{f}(N/2)$  til  $\mathbf{f}(N - 1)$  fås direkte vha. ovenstående formel uden brug af yderligere transformationer.

### Eksempel

En to-punkts fouriertransformation (dvs.  $n = 1$ ) kræver beregning af  $\mathbf{f}(0)$ , dvs.  $\mathbf{f}_{lige}(0)$  og  $\mathbf{f}_{ulige}(0)$ . Da begge disse er 1-punkts transformationer, og da fouriertransformationen af et enkelt værdi er lig værdien selv, kræver bestemmelsen af disse størrelser ingen multiplikationer eller additioner. Derefter skal  $\mathbf{f}_{ulige}(0)$  multipliceres med  $W_2^0$ . Heraf ses, at det totale antal af multiplikationer hhv. additioner er  $\text{mult}(n = 1) = 1$  og  $\text{add}(n = 1) = 2$ .

For  $n = 2$  opdeles de fire værdier i to grupper som før. Den første gruppe kræver udregning af to 2-punkts transformationer. Dette kræver  $2\text{mult}(1)$  hhv.  $2\text{add}(1)$  multiplikationer hhv. additioner. Dertil kommer to multiplikationer og to additioner til at beregne  $\mathbf{f}(0)$  og  $\mathbf{f}(1)$ . Endelig kræves to additioner til beregning af  $\mathbf{f}(2)$  og  $\mathbf{f}(3)$ . Det totale antal operationer bliver således:  $\text{mult}(2) = 2\text{mult}(1) + 2 = 4$  og  $\text{add}(n = 2) = 2\text{add}(1) + 4 = 8$ .

Ved at fortsætte med  $n = 3, 4, \dots$  er det let at indse at antallet af multiplikationer hhv. additioner for  $N = 2^n$  (hvor  $n \geq 1$ ) punkter bliver:

$$\text{mult}(n) = 2\text{mult}(n-1) + 2^{n-1} \quad (2.94)$$

$$\text{add}(n) = 2\text{add}(n-1) + 2^n \quad (2.95)$$

De tidligere nævnte størrelser for tidskompleksiteten af algoritmen følger nu let ved induktion.

### Den inverse FFT

Den forlæns og baglæns en-dimensionale fouriertransformation adskiller sig, ud over en division med  $N$ , kun ved fortegnet på den komplekse eksponentialfunktion  $W_N$ . I overstående udledning af FFT-algoritmen antog vi, at disse tal var tabelleret på forhånd. For at kunne genbruge tabellerne i IFFT-algoritmen er det derfor oplagt at kompleks konjugere begge sider af ligningen (for definitionen af den inverse transformation). Ved tilsvarende at dividere med  $N$  fås:

$$\frac{1}{N}f^*(x) = \frac{1}{N} \sum_{u=0}^{N-1} \mathbf{f}^*(u) W_N^{ux} \quad (2.96)$$

Vi kan derfor benytte FFT-algoritmen til at beregne den inverse fouriertransformation. Blot skal vi før og efter anvendelsen kompleks konjugere  $\mathbf{f}(u)$  og  $f(x)$ . Til slut skal resultatet multipliceres med  $N$ .

Det skal bemærkes, at en helt tilsvarende måde kan benyttes i det to-dimensionale tilfælde. En forskel er dog, at der ikke her skal multipliceres med  $N$  til sidst. Implementeres den to-dimensionale transformation ved gentagende passager af den en-dimensionale transformation, er det vigtigt kun at foretage kompleks konjugeringen før og efter *hele* transformationen, og altså ikke efter transformationen af hver række eller søjle. Yderligere skal det bemærkes at  $f(x)$  ofte vides at være reel, hvorfor den afsluttende kompleks konjugering er overflødig.

### Bit-spejlings-metoden

Antag at vi benytter FFT-algoritmen med succesiv doubling på en tabel med 8 punkter:  $\{f(0), f(1), f(2), \dots, f(7)\}$ . Af beskrivelsen af FFT-algoritmen fremgår det at der for alle værdier af  $u$  gælder at  $f$ -værdier

Oprindeligt array	Oprindeligt binært indeks	Omordnet binært indeks	Omordnet array
$f(0)$	0 0 0	0 0 0	$f(0)$
$f(1)$	0 0 1	1 0 0	$f(4)$
$f(2)$	0 1 0	0 1 0	$f(2)$
$f(3)$	0 1 1	1 1 0	$f(6)$
$f(4)$	1 0 0	0 0 1	$f(1)$
$f(5)$	1 0 1	1 0 1	$f(5)$
$f(6)$	1 1 0	0 1 1	$f(3)$
$f(7)$	1 1 1	1 1 1	$f(7)$

Tabel 2.1: Omordning af array efter “bit-reversal” reglen

med lige index indgår i beregningen af  $f_{\text{lige}}$ , og at  $f$ -værdier med ulige index indgår i beregningen af  $f_{\text{ulige}}$ . Inddatasættet kan altså opdeles i to grupper:  $\{f(0), f(2), f(4), f(6)\}$  og  $\{f(1), f(3), f(5), f(7)\}$ . Tilsvarende kan de to 4-punkts grupper opdeles videre:

$$\{f(0), f(4)\}, \{f(2), f(6)\}, \{f(1), f(5)\}, \{f(3), f(7)\}$$

For at gøre FFT-algoritmen hurtig er det en fordel om tabellen af inddataværdier kunne omordnes efter ovenstående skema, idet operationerne til indeksering herved kan spares. Heldigvis viser det sig, at der gælder en simpel regel (*the bit-reversal rule*) for, hvorledes inddataværdierne skal omordnes. Dette er illustreret i Tabel 2.1. Det ses, at for gyldigt index  $x$  fås det korrekte index i det omordnede array ved at udtrykke  $x$  binært og opskrive disse i omvendt rækkefølge. Yderligere diskussion af denne, og andre metoder, der er nyttige ved implementation af FFT-algoritmen findes mange andre steder f.eks. [6].

Et problem ved FFT-algoritmen er, at antal sampels (sidelængderne) antages at være en potens af 2. Dette er naturligvis ikke altid tilfældet. Der findes tre løsninger på dette problem:

- At benytte den generelle fouriertransformation (2.15). Denne er let at implementere, men langsom.
- At udvide billederne sådan at sidelængderne bliver en potens af 2. For at undgå aliasing skal man i dette tilfælde huske at benytte en vinduesfunktion, da den kunstigt frembragte diskontinuitet ved udvidelsen ellers vil forårsage et stort omgang af højfrekvent energi, der ikke er relateret til billedinformationen, men til diskontinuiteten (Se evt. videre diskussion i afsnit 5.1).
- At benytte særlige numeriske implementationer (som kendt fra implementationen af *Fastest Fouriertransform in The West* (se <http://www.fftw.org>). Her opdeles signalet i mindre stykker sådan, at antal sample i hvert stykke er et lille primtal. Ved brug af særligt specialiserede

metoder kan transformationen foretages væsentlig hurtigere end ved den direkte implementation, men også væsentlig langsommere end hvis signalet have haft en kantlængde med en potens af 2.



# Kapitel 3

## Filtrering af billeder

Lineære positionsinvariante filtre er formodentlig de mest benyttede filtre. Et filter  $f$  er lineært, hvis filtreringen kan skrives som et lineært udtryk  $O = FI$ , hvor  $I$  hhv.  $O$  er input hhv. output billeder organiseret som søjlevektorer, og  $F$  er en kvadratisk matrix af filterværdier. Filteret er positionsinvariant, hvis alle billedpunkter filtreres på samme måde. Dette betyder at række nummer  $j$  i  $F$  er identisk med række  $j - 1$  bortset fra en cyklisk højreskift på en position. Den matematiske teori bag lineære filtre er særdeles veludviklet. Dette gælder specielt vurderinger af filterets egenskaber i frekvensområdet. Filtreringen kan også beskrives ved foldning af billedet med punktspredningsfunktionen for filtret.

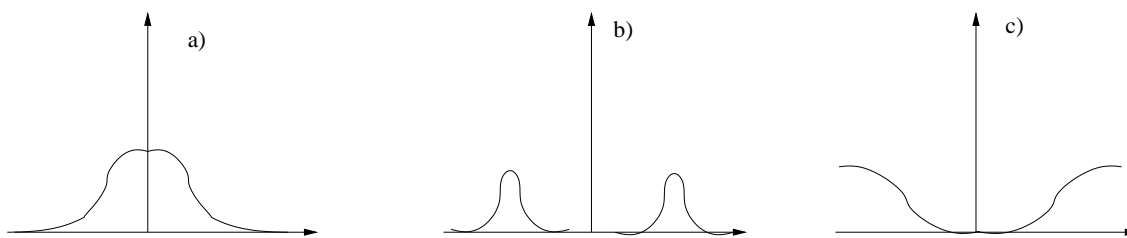
På trods af alle de nydelige egenskaber, der gælder for lineære positionsinvariante filtre er disse ikke altid hensigtsmæssige. Dette gælder f.eks. hvis der ønskes en lavpasfiltrering, der bevarer kontrasterne (kanterne) i billedet. Her er det nødvendigt at benytte enten positionsafhængige eller ikke-lineære filtre. Eksempler på sådanne filtre er anisotrope filtre, hvor filterfunktionen bestemmes pixel for pixel. Et eksempel på et ikke-lineært filter er medianfilteret.

### 3.1 Filtrering i frekvensområde

Det kan ofte være en fordel at specificere filtreringen i frekvensområdet, fordi visse støjtyper har en markant frekvenskarakteristik, eller fordi formålet med filtreringen netop er at fremhæve eller dæmpe visse frekvensområder. Det matematiske grundlag for frekvensfiltrering er foldningssætningen. Frekvensfiltreringen foretages ved at øge eller dæmpe bestemte frekvenser svarende til værdierne af frekvensfilteret  $h$ . I de følgende afsnit beskrives en række klassiske filtre, hvis formål er at:

- Dæmpe de høje frekvenser og lade de lave frekvenser passere
- Dæmpe de lave frekvenser og lade de høje frekvenser passere
- Dæmpe udvalgte frekvenser og lade et bånd af frekvenser passere

Sådanne filtre er illustreret i Figur 3.1. Vi vil udelukkende betragte filtre, der ikke berører fasen af billederne, men kun styrkespektret.



Figur 3.1: Styrkespektrere for a) Et lav-pas filter; b) Et bånd-pas filter; c) Et høj-pas filter.

### 3.1.1 Lav-pas filtrering

Ved en lav-pas filtrering dæmpes de middelstore og høje frekvenser. Dette betyder også at kontrasten i et billede forringes, fordi højfrekvent energi fra billedkontraster ikke kan skelnes fra højfrekvent energi fra støj. Visuelt kan højfrekvent støj være meget generende. Hvis en reduktion i kontrasten kan accepteres, er lav-pas filtrering en effektiv metode til at undertrykke støjen. Grænsen mellem de frekvenser, der skal passere, og de, der skal undertrykkes bestemmes ved “cut-off”-frekvensen  $D_0$ , der defineres som den frekvens, hvor filterværdien antager halvdelen af sin maksimale værdi.

Som regel vil næsten al energi i et billede være koncentreret i og omkring origo i den fouriertransformerede af billedet. Dette betyder, at den meste energi bevares selv om “cut-off”-frekvensen vælges meget lille. Imidlertid er de høje frekvenser særdeles bestemmende for det visuelle indtryk. Selv om blot få procent af energien i et billede afskæres ved at undertrykke de højeste frekvenser, kan det filtrerede billede ofte have ringe lighed med det oprindelige billede. Hvis formålet med filtreringen er at frembringe et billede, der skal betragtes af et menneske, da er det som regel en fordel at bevare en vis del af det højfrekvente indhold af billedet, selv om dette betyder, at billedet er støjfyldt. Lavpas filtrering er illustreret i Figur 3.2. I det følgende skal vi se på nogle elementære lav-pass filtre.

**Idealfiltret:** Lad i det følgende  $D = \sqrt{u^2 + v^2}$ . Ideal lav-pass filtret, med “cut-off”-frekvens  $D_0$  er givet ved:

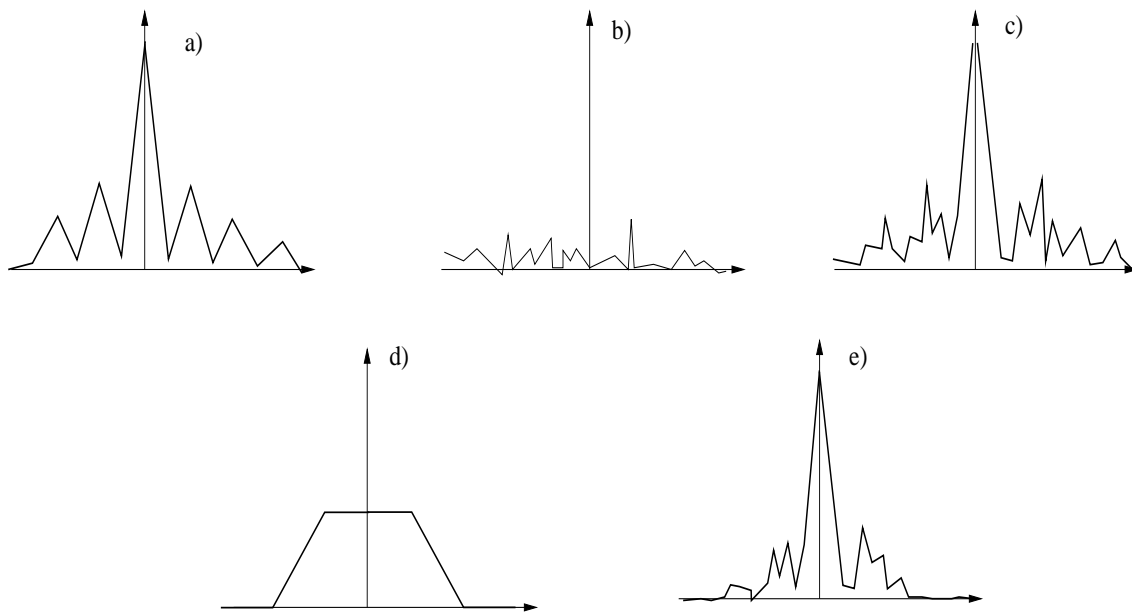
$$\mathbf{h}(u, v) = \begin{cases} 1 & \text{hvis } D < D_0 \\ 0 & \text{ellers} \end{cases} \quad (3.1)$$

Idealfiltret er rotationssymmetrisk, og undertrykker alle frekvenser større end  $D_0$  effektivt, hvorimod de lavere frekvenser passerer uændret.

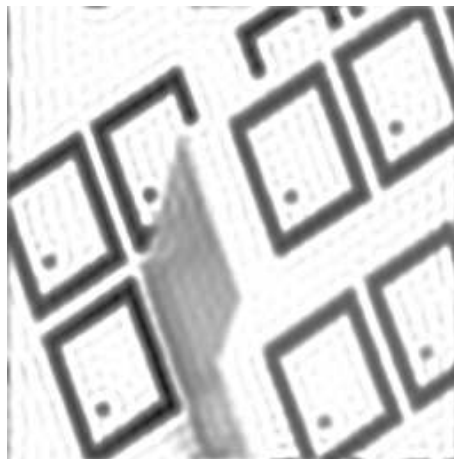
Ved invers fouriertransformation af et idealfiltreret billede vil man, som illustreret i Figur 3.3, ofte kunne observere en kraftig *ringningseffekt* omkring kraftige kontraster i billedet. Denne skyldes at funktionen  $\mathbf{h}(\mathbf{x})$  er diskontinuert i  $D_0$ . Ringningseffekten optræder især ved kraftige kontraster i billedet. Årsagen til ringningseffekten kan forstås ud fra foldningssætningen. I det en-dimensionale tilfælde er den inverst fouriertransformerede af ideal-frekvensfiltret er givet ved:

$$h(x) = \frac{\sin(2\pi D_0 x)}{\pi x} \quad (3.2)$$



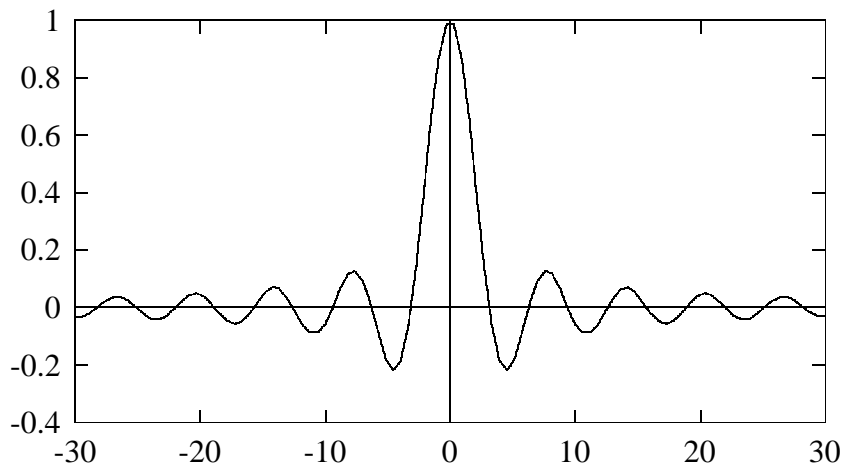


Figur 3.2: Frekvensfiltrering: Styrkespektre for a) Et signal uden støj; b) Støj; c) Et signal med støj; d) Et frekvensfilter; e) Et filtreret signal.



Figur 3.3: Ringningseffekten

Funktionen er skitseret i Figur 3.4: Da foldningen kan betragtes som en kopiering, betyder dette, at markante kontraster vil blive “kopieret” ud til positionerne af de lokale ekstrema i  $h(x)$ . I praksis vil ringningseffekten være så generende, at idealfiltre sjældent kan benyttes. I stedet benyttes filtre, hvor overgangen mellem de frekvenser der passerer, og de der undertrykkes er mere glat.

Figur 3.4: Funktionen  $\sin(x)/x$ 

**Butterworthfilteret:** Et  $n$ 'te ordens lav-pas butterworth filter med "cut-off"-frekvens  $D_0$  er i frekvensområdet givet ved ligningen:

$$\mathbf{h}(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}} \quad (3.3)$$

Butterworthfiltret producerer (for lille  $n$ ) kun i meget ringe grad ringningseffekt. Årsagen er, at frekvensovergangen omkring "cut-off"-frekvensen  $D_0$  er glat, og at halen af filteret tillader en vis portion af de høje frekvenser at passere. Dette betyder, at selv om højfrekvent støj dæmpes ved filtreringen, vil kanterne i det filtrerede billede stadig være bevaret i nogen grad afhængig af valget af  $D_0$  og  $n$ .

Filtre som Butterworthfiltret kan være meget effektive til støjfiltrering, hvis denne er højfrekvent og har lille amplitude. Butterworthfiltre benyttes ofte som delefiltre ved opdeling af et lydssignal i delsignaler, der hver styrer bas-, mellemtone- og diskant-højtalere.

### Hanningfilteret:

Hanning filteret er endnu et klassisk lav-pas filter. Dette er givet ved:

$$\mathbf{h}(u, v) = \frac{1}{2} \left[ 1 + \cos \left( \frac{\pi D}{2D_0} \right) \right] \quad (3.4)$$

for  $D \leq 2D_0$ . Uden for denne cirkelskive er filterværdien nul.

**Gauss filteret:**

Det i billedbehandling formodentlig mest benyttede filter i sted- såvel som i frekvensområdet er det rotationssymmetriske to-dimensionale Gauss filter:

$$\mathbf{h}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{D^2}{2\sigma^2}} \quad (3.5)$$

hvor  $\sigma$  angiver spredningen af filteret. Det overlades til læseren at udlede udtryk for cut-off frekvensen som funktion af  $\sigma$ .

Et grundlæggende problem ved alle de nævnte filtre er, at de undertrykker høje frekvenser relateret til kontraster i billederne på samme måde som høje frekvenser relateret til støj. Det er således ikke muligt at foretage en frekvensbaseret støjfiltrering uden samtidig at mindske kontrasten i det resulterende billede.

**3.1.2 Høj-pas filtrering**

Høj-pas filtrering foretages når formålet med filtreringen er at fremhæve kontrasterne i billedet. Et høj-pas ideal-, hhv. første ordens Butterworthfilter kan analogt til formlerne i foregående afsnit skrives:

$$\mathbf{h}(u, v) = \begin{cases} 0 & \text{hvis } D < D_0 \\ 1 & \text{ellers} \end{cases} \quad (3.6)$$

$$\mathbf{h}(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}} \quad (3.7)$$

hvor  $D(u, v) = \sqrt{u^2 + v^2}$ , og “cut-off”-frekvensen  $D_0$  som før er bestemt ved den frekvens hvor filtret antager halvdelen af sin maksimale værdi.

Ved høj-pas filtrering vil vi ofte observere at kanter fremhæves og at alle områder med glat varierende (oprindelig) intensitet vil få ens intensitet efter filtrering. Dette uafhængigt af, at områderne før filtreringen havde markant forskellig (men glat varierende) intensitet. Årsagen er, at de lave frekvenser er for effektivt undertrykt. En metode til at undgå dette er at addere en konstant  $\epsilon$  til høj-pas filtret (i frekvensområdet). Denne metode kaldes “at fremhæve de høje frekvenser” (eng. high frequency emphasis).

$$\mathbf{h}(u, v) = \begin{cases} 1 & \text{hvis } D = 0 \\ \epsilon + \frac{1}{1 + [D_0/D(u, v)]^{2n}} & \text{hvis } D > 0 \end{cases} \quad (3.8)$$

Et yderligere raffinement er at foretage en histogram linearisering af intensitetsværdierne i det inverst fouriertransformerede høj-pass filtrerede billede. Histogram linearisering er beskrevet i afsnit 6.1.

### 3.1.3 Bånd-pas filtrering

Ved en bånd-pas filtrering tillades kun et bestemt frekvensbånd at passere. Passagen kan ske uhindret som i et ideal bånd-pas filter eller dæmpes glat ved overgangen mellem de frekvenser, der passerer, og de, der undertrykkes. Et ideal bånd-pas filter vil have formen:

$$h(u, v) = \begin{cases} 1 & \text{hvis } D_0 - W < D < D_0 + W \\ 0 & \text{ellers} \end{cases} \quad (3.9)$$

hvor  $D = \sqrt{u^2 + v^2}$ ,  $D_0$  er “midterfrekvensen” af det frekvensbånd der skal passere, og  $2W$  er bredden af frekvensbåndet.

Ofte benyttes et båndafvisningsfilter (eng. band reject filter), der i det ideale tilfælde fremkommer af ovenstående ligning ved ombytning af betingelserne i de to tilfælde.

Hvis der forekommer meget regelmæssig støj i et billede, f.eks. en overlejret sinus bølge med en vis orientering og frekvens, vil denne støj optræde som punkter  $(u_0, v_0)$  og  $(-u_0, -v_0)$  med numerisk stor værdi i styrkespektret for billedet. Det er ofte muligt at se sådanne punkter i et logaritmisk plot af styrkespektret. Algoritmisk kan man først bestemme middelværdien  $M(u, v)$  af styrkespektret som funktion af radius  $D$ , dernæst bestemme de punktpar  $(u_0, v_0)$  og  $(-u_0, -v_0)$  hvor styrkespektrets værdi i begge punkter er markant større end  $M(u_0, v_0)$ . Der kan nu bestemmes et (ideal-båndafvisnings-) filter med værdi 0 i to små cirkelskiver centreret i de detekterede punkter og ellers med værdi 1. Ved filtreringen kan den systematiske støj fjernes fuldstændigt uden synlig beskadigelse af billedet iøvrigt.

$$h(u, v) = \begin{cases} 0 & \text{hvis } (u - u_0)^2 + (v - v_0)^2 < C_0 \quad \text{eller} \quad (u + u_0)^2 + (v + v_0)^2 < C_0 \\ 1 & \text{ellers} \end{cases} \quad (3.10)$$

Som alternativ kan man f.eks. benytte et første ordens bånd-afvisnings Butterworth filter til dæmpning af alle frekvenser i et cirkelbånd (omkring origo):

$$h(u, v) = \frac{1}{1 + \left[ \frac{D(u, v)W}{D^2(u, v) - C_0^2} \right]^2} \quad (3.11)$$

hvor  $C_0 = \sqrt{u_0^2 + v_0^2}$  angiver afstanden fra origo til centret af cirkelskiverne.

## 3.2 Filtrering i intensitetsområde

Filtrering i intensitetsområdet foretages med fordel, når støtten for filtret er lille, eller når specielt foldningsmaskineri er til rådighed. Fordelen ved filtrering i intensitetsområdet bliver særlig stor, hvis filtret er separabelt, dvs. når filtret  $F(x, y)$  kan skrives:

$$F(x, y) = f_1(x)f_2(y) \quad (3.12)$$

Eksempelvis er kernen for box filtret og gauss filtret (se nedenstående afsnit), samt kernen for fouriertransformationen alle separable.

Antag i det følgende, at filtreringens formål er at fjerne støjen og at denne er additiv og hvid. Hvid støj er karakteriseret ved at have konstant middelværdi lig 0, konstant varians  $\sigma^2$  og ved at støjbidraget i to vilkårlige pixels er uafhængige. Støjmodellen  $g = f + \eta$  diskuteres videre i afsnit 4.1.

Ved lineær filtrering i intensitetsområdet foretages filtreringen ved beregning af et lokalt vægtet gennemsnit. Hvis støtten for filtret er stor, vil filtrets egenskab mht. undertrykkelse af støj som regel være god, idet støjen i mange billedpunkter indgår i beregningen, og fordi den forventede værdi  $E\{\eta\}$  af støjen er lig nul.

$$F * g = F * f + F * \eta \approx E\{f\} + E\{\eta\} \approx E\{f\} \quad (3.13)$$

I områder hvor  $f$  kun varierer lidt vil  $f - F * g$  defor nærme sig nul for voksende støtte. I områder hvor  $f$  varierer meget vil  $f - F * g$  imidlertid vokse med voksende støtte. Der er således tale om en afvejning mellem et filters støjfiltreringsegenskab og dets egenskab mht. bevarelse af det "oprindelige" billede. Generelt er det nødvendigt at øge støtten jo ringere signal-støjforholdet i et billede er. Signal-støjforholdet er defineret ved forholdet mellem energien i signalet og energien i støjen. I dette tilfælde er mere global information nødvendig. Hvis filtreringens formål er at producere et billede til efterfølgende menneskelig inspektion, vil det imidlertid ofte være en fordel at holde støtten lille. Mennesker er som regel i stand til at abstrahere fra ret store mængder af støj, men har svært ved at beskrive et billede, hvor strukturene er udviskede. Filtrets støtte er naturligvis ikke enebestemmende for filtrets egenskaber. Filterformen er overordentlig vigtig. I de følgende afsnit beskrives to meget benyttede filtre samt deres egenskaber.

### Box-filtret:

Box-filtre har kvadratisk støtte og konstant værdi. Filtret antages at have integral 1.0 således at et  $N \times N$  box-filtret har værdier  $1/N^2$ . Box-filtret er ofte benyttet da det er separabelt og let at implementere. Den fouriertransformerede af et box-filtret  $f$  er givet ved:

$$\mathbf{f}(u, v) = \frac{\sin(2\pi Nu)}{\pi Nu} \frac{\sin(2\pi Nv)}{\pi Nv} \quad (3.14)$$

Heraf ses, at nogle høje frekvenser (svarende til nulpunkterne i den fouriertransformerede) vil blive dæmpet totalt, hvorimod andre høje frekvenser (svarende til ekstrema) kun dæmpes moderat. Da amplituden af den fouriertransformerede aftager som  $1/(\pi Nv)$  ses, at støjdem্পningen vil være moderat med mindre  $N$  vælges meget stor. Til gengæld vil filtreringen så også berøre det lave og det mellemstore frekvensindhold. I praksis vælges  $N$  ofte lille da filtreringen så bliver meget hurtig.

### Gauss-filtret:

Gauss-filtret er vel nok det mest interessante af alle filtre. Vi skal møde gauss-filtret et stort antal gange, både her hvor det optræder som støjfilter, ved detektion af kanter i billeder samt ved kodning og repræsentation af billeder. Det to-dimensionale gauss-filtret er givet ved:

$$D_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.15)$$

Gauss-filtret er rotationssymmetrisk, separabelt og parametriseret ved kun een parameter  $\sigma$ , der angiver spredningen af filtret. Parameteren  $\sigma$  kaldes ofte for *skalaparameteren*. Den fouriertransformerede af en gaussfunktion med spredning  $\sigma$  er selv en gaussfunktion, og har spredning  $1/2\pi\sigma$ .

Gaussfunktionen er positiv, men aftager så hurtigt, at den i beregningsmæssige sammenhænge ofte trunkeres til nul, for  $r = \sqrt{x^2 + y^2}$  større end  $2\sigma$  eller  $3\sigma$ . For store værdier af  $\sigma$  vil støtten for filtret i intensitetsområdet derfor blive stor, og selv om separabiliteten udnyttes vil foldningen tage forholdsvis lang tid. I dette tilfælde kan filtreringen ofte foretages hurtigere ved implementation i frekvensområdet.

Generelt vil der gælde at jo mindre støtte en funktion har i intensitetsområdet, jo større støtte vil dens fouriertransformerede have i frekvensområdet. Således er den fouriertransformerede af en impulsfunktion jo en konstant. Graden af koncentration kan f.eks. måles ved spredningen af funktionen eller ved “cut-off”-frekvensen. Man kunne da spørge hvilket filter der var mest muligt koncentreret i såvel intensitetsområdet som i frekvensområdet. Det kan vises at dette filter netop er gauss-filtret.

Som vi skal diskutere i kapitel 10 om kantdetektion, har gaussfiltret andre enestående egenskaber, f.eks. at det er løsning til varmeledningsligningen. Betragt vi et billede  $I$  som repræsenterende temperatur hvor stor hhv. lille intensitet repræsenterer varme hhv. kulde, og betragt vi et kontinuum af foldninger  $G_{\sigma} * I$  som funktion af variansen  $\sigma^2$  (tiden), vil de varme hhv. kolde områder langsomt udlignes når tiden går ( $\sigma^2$  øges). Kolde områder vil blive varmere og varme områder koldere. For  $\sigma^2 \rightarrow \infty$  vil temperaturen (intensiteten) overalt tilnærme middeltemperaturen (middelintensiteten). Udvidelsen af signalet/billedet med den ekstra dimension  $\sigma$  (tiden) kaldes for *skalarummet*.

Betragtes et billede som funktion af  $\sigma$ , vil man kunne iagttage, at det bliver gradvist enklere (med færre detaljer). Der kan vises en lang række egenskaber ved hvorledes forskellige features som f.eks. lokale ekstrema, kanter etc. gradvist forsvinder i skalarummet. Sådanne analyser involverer ofte ganske avanceret matematik, herunder katastrofeteori, og ligger uden for rammerne af disse noter. En gren i forskningen søger at finde en robust og kompakt repræsentation af signaler/billeder i skalarum. En sådan vil f.eks. kunne omfatte en ikke-ækvivalent sampling af punkter  $(x, y, \sigma)$  og vil give mulighed for en hierarkisk beskrivelse af billedet.

En tredje enestående egenskab ved gaussfiltret er, at dette filter vil fremkomme ved gentagne filtrering af et vilkårligt filter med sig selv. Dette er en naturlig konsekvens af *De store tals lov* i sandsynlighedsregningen og kan f.eks. udnyttes hvis specielt hurtigt “hardware” til eksempelvis  $3 \times 3$  -boxfiltrering er til rådighed. En acceptabel approximation til en gaussfiltrering kan da opnås ved gentagne box-filtrering. Ofte vil et lille antal iterationer være tilstrækkeligt. En anden konsekvens er, at foldningen af to gaussfiltre igen bliver et gaussfilter. Eksempelvis vil:

$$G_{\sigma_1} * G_{\sigma_2} = G_{\sqrt{\sigma_1^2 + \sigma_2^2}} \quad (3.16)$$

## 3.3 Ikke-lineære filtre

Filtrering med ikke-lineære filtre kan naturligvis ikke foretages ved nogen foldning. Ikke-lineære filtre kan virke bedre end lineære filtre, fordi de bedre kan tage hensyn til abrupte ændringer i intensitetsfunktionen. Dette bliver eksemplificeret ved medianfiltret og Nagaofiltret, der begge er udviklet til at foretage støjfiltrering i intensitetsområdet uden at ødelægge kanter i intensitetsfunktionen. Ikke-lineær filtrering kan også foretages i frekvensområdet. Dette eksemplificeres ved homomorfisk filtrering, hvor ideen er at separere filtreringen af belysnings- og reflektionskomponenten af intensiteten. Nyere forskning har vist at mange ikke-lineære filtreringer kan beskrives som løsning til sæt af koblede partielle differentialligninger. Videre diskussion af denne type af metoder ligger uden for rammerne af disse noter.

### 3.3.1 Filtrering i intensitetsområdet

Ved ikke-lineær filtrering i intensitetsområdet er filtersvaret en ikke-lineær funktion af billedværdierne i en (lille) omegn af punktet. Alternativt foretages den ikke-lineære operation på et sæt af værdier beregnet ved lineære filtreringer. Dette er tilfældet ved Nagaofiltret, der er retningsspecifikt. Ikke-lineære filtreringer foretages som regel i intensitetsområdet, fordi disse filtre er lettest at formulere i intensitetsområdet.

#### Medianfiltret:

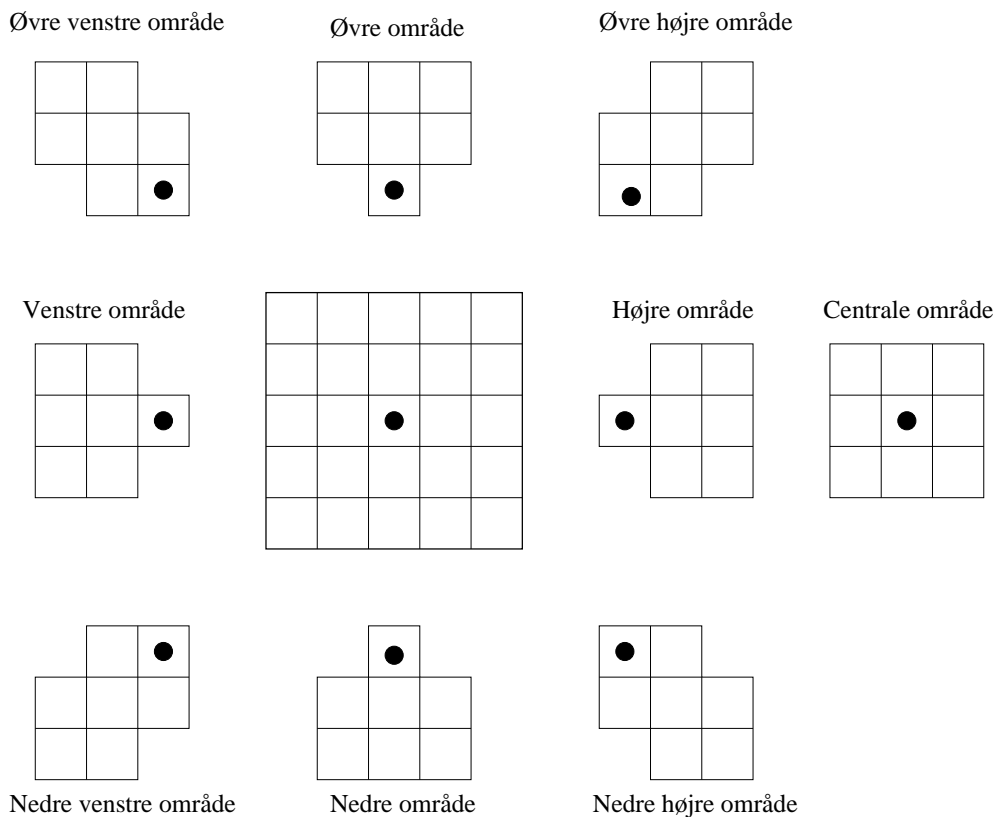
Filtersvaret for et  $N \times M$  medianfilter er bestemt ved medianen af de  $NM$  intensitetsværdier i støtten. Filtret er overordentlig effektivt til at fjerne støj, hvor intensiteten i en lille del af billedpunkterne er meget støjfyldt som f.eks. ved "salt og peber"-støj. Abrudte ændringer i intensitetsfunktionen efterlades skarpe men ofte flossede. Ofte kan filtret anvendes til egentlig kontrastfremhævning.

Den mest oplagte implementation af filtret består i at sortere de  $NM$  værdier og udvælge af den midterste værdi. Denne implementation benyttes ikke i praksis da den er meget langsom. I stedet benyttes en glidende modifikation af et histogram over intensitetsværdierne.

Støjfiltreringsegenskaben øges med filterstørrelsen. Ofte vil strukturen af kanter i billederne imidlertid blive forringet (kanterne bliver flossede). Alternativt kan filtreringen itereres et lille antal gange. Det bør endvidere bemærkes at støtten for filtret ikke behøver at være rektangulært, men f.eks. kan have form af et kors e.lign.

#### Nagaofiltret:

Nagaofiltret er et retningsspecifikt filter, hvis formål er at foretage støjfiltrering uden at ødelægge kanter i intensitetsfunktionen. Filterstørrelsen er sædvanligvis  $5 \times 5$ . Ofte itereres filtreringen et lille antal gange. Støtten opdeles i 9 (overlappende) retningsspecifikke områder, der eksempelvis kan benævnes: det centrale, det øvre-venstre, det øvre-højre, det nedre-venstre, det nedre-højre,



Figur 3.5: De 9 betragtede områder i nagafilteret

det øvre, det venstre, det højre og det nedre område. Disse er illustreret i nedenstående Figur 3.4, hvor billedelementet hvis filterværdi er under beregning er markeret ved en sort klat.

Metoden er for hvert område at beregne middelværdi og varians af intensitetsværdierne for billedpunkterne i området. Filterværdien bestemmes da som middelværdien for det område, der har mindst varians. Filtringen kan således betragtes som en “retningspecifik box-filtrering”. Retningen vælges sådan, at intensiteten i det givne punkt er mest i overensstemmelse med intensitetsværdierne i området svarende til retningen. Nagafiltret er beskrevet i [27].

### 3.3.2 Filtring i frekvensområdet: Homomorfisk filtring

Homomorfisk filtring er en ikke-lineær operation, der er baseret på en simpel antagelse om sammensætningen af frekvensspektret for et billede. Antagelsen er, at den observerede intensitet  $I$  kan beskrives ved produktet  $LR$  af en belysningskomponent  $L$  og en refleksionskomponent  $R$ . Logaritmen til  $I$  vil da være summen af logaritmerne til  $L$  og  $R$ . Om de fouriertransformerede af logaritmerne gælder da at:

$$\mathcal{F}\{\log I\} = \mathcal{F}\{\log L\} + \mathcal{F}\{\log R\} \quad (3.17)$$



Skrives ligningen  $\mathbf{i} = \mathbf{l} + \mathbf{r}$ , og betegnes filterfunktionen (i frekvensområdet) med  $\mathbf{h}$ , kan vi skrive:

$$I^{fil} = \mathcal{F}^{-1}\{\mathbf{h}\mathbf{i}\} = \mathcal{F}^{-1}\{\mathbf{h}\mathbf{l}\} + \mathcal{F}^{-1}\{\mathbf{h}\mathbf{r}\} = L^{fil} + R^{fil} \quad (3.18)$$

Tages nu eksponentialfunktionen af begge sider, kan resultatet igen skrives som produktet:

$$I^* = L^* R^* \quad (3.19)$$

Som ovenfor beskrevet består homomorfisk filtrering i først at fouriertransformere logaritmen til billedet. Den fouriertransformerede multipliceres med frekvensfiltret og resultatet invers fouriertransformeres. Endelig tages eksponentialfunktionen af den inverst fouriertransformerede.

Under antagelsen om ligningen  $I = LR$  ses, at hvis de “interessante” frekvensområder for  $L$  og  $R$  ikke overlapper, da kan filtret  $\mathbf{h}$  konstrueres således, at det modificerer frekvensområderne forskelligt. I praksis vil  $L$  som regel være lavfrekvent, hvorimod  $R$  vil have ikke forsvindende høje frekvenser.

Homomorfisk filtrering er meget anvendelig, hvis belysningen af den afbildede scene er ujævn. I så tilfælde skal filterværdierne vælges små for små frekvenser. Ønskes en samtidig kontrastfremhævnning vælges filterværdierne store for større frekvenser. Den eksakte form af filtret kan variere meget. Et muligt parametriseret filter er:

$$\mathbf{h}(u, v) = \lambda - \frac{\lambda - \epsilon}{1 + \alpha(u^2 + v^2)} \quad (3.20)$$

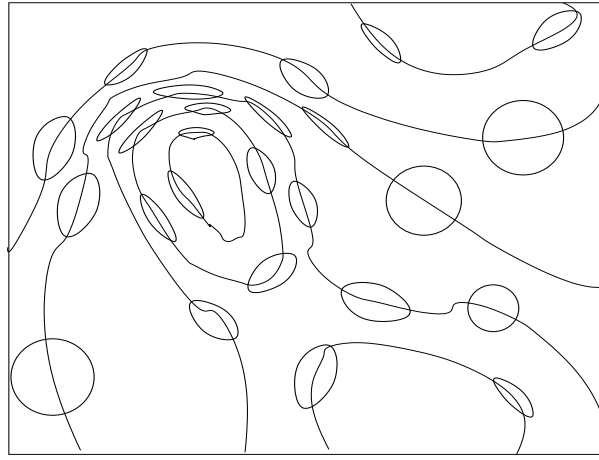
eller

$$\mathbf{h}(u, v) = \lambda - (\lambda - \epsilon)e^{-\frac{u^2 + v^2}{2\alpha^2}} \quad (3.21)$$

Her angiver  $\lambda$  og  $\epsilon$  maksimums- og minimumsværdien af filtret, og  $\alpha$  vil bestemme filterformen. I praksis er det relativt let at vælge fornuftige værdier af  $\lambda$  og  $\epsilon$  (eksempelvis til 2.0 og 0.1). Det kan være betydelig sværere at vælge en optimal værdi af parameteren  $\alpha$ . Igen skal det pointeres, at valget af parameterværdier afhænger af det konkrete billede samt af formålet med filtreringen.

## 3.4 Positionsafhængig filtrering

En filtrering kaldes positionsafhængig hvis filterformen varieres fra punkt til punkt. Typisk foretages to gennemløb af billedet. I det første bestemmes en eller flere parametre for hver pixel. I næste gennemløb foretages filtreringen med filtre, der er parametriseret af de fundne værdier. De typiske parametre, som bestemmes, er størrelsen på filteret og (hvis filteret ikke er rotationssymmetrisk) orienteringen og eccentriciteten af filteret.



Figur 3.6: Orientering og størrelse af et anisotropt filter overlejret på niveaukurverne for et billede.

**Eksempel** Et eksempel på et positionsafhængig (rotationssymmetrisk) gaussfiltrering kunne tage udgangspunkt i en indledende detektion af punkter med høj kontrast, f.eks. detekteret ved:

$$\left(\frac{\partial}{\partial x}I\right)^2 + \left(\frac{\partial}{\partial y}I\right)^2 \approx (I(x, y) - I(x-1, y))^2 + (I(x, y) - I(x, y-1))^2 > T^2 \quad (3.22)$$

hvor  $T$  er en brugerspecificeret grænseværdi. Efterfølgende bestemmes, for hver pixel, afstanden  $d(x, y)$  til den nærmeste detekterede kontrastpixel, og en parameter  $\sigma(x, y)$  bestemmes f.eks. ved  $d/2$ . Til slut filtreres det oprindelige billede med en gaussfunktion  $G_{\sigma(x, y)}$ . Filtreringen kan evt. udelades i de punkter  $(x, y)$  hvor  $\sigma < 1$ .

Et eksempel på et positionsafhængig anisotropt filter gives i nedenstående afsnit. Ulempen ved positionsafhængige filtreringer er oplagt: De er meget mere beregningskrævende end simple lineære positionsinvariante filtreringer. Fordelen er også oplagt: Filtreringen kan bedre styres efter det lokale billedindhold, og resultatet vil som regel være langt bedre.

### 3.4.1 Anisotrop filtrering

Formålet med en anisotrop filtrering er typisk at foretage en kontrast/struktur-bevarende lav-pas filtrering. Filteret er typisk aflangt og orienteret langs niveaukurven for intensitetsfladen. Bemærk at Nagao-filteret er et eksempel på et simpelt anisotropt (positionsafhængig) filter. Anisotrop filtrering - implementeret ved diffusion - diskuteres videre i afsnit 9.4.

Prototypen på et brugbart filter er et  $n$ -dimensionalt (multivariat) gaussfilter:

$$G(\vec{x}) = \frac{1}{(2\pi)^{n/2} |\mathbf{C}|^{1/2}} e^{-\frac{1}{2} \vec{x}^t \mathbf{C}^{-1} \vec{x}} \quad (3.23)$$

hvor matricen  $\mathbf{C}$  er *kovariansmatricen* for filteret,  $|\mathbf{C}|$  er determinanten af  $\mathbf{C}$ , superscript 't' angiver transponering, og hvor  $\vec{x} = (x, y)$  for  $n = 2$ . For  $n = 1$  degenererer filteret til det

sædvanlige en-dimensionale gaussfilter. For  $n = 2$  degenererer filteret til det sædvanlige to-dimensionale gaussfilter netop hvis kovariansmatricen  $\mathbf{C} = \sigma^2 \mathbf{I}$ , altså en diagonalmatrix med diagonalelementer lig variansen. Nøglen til forståelse af det generelle gaussfilter er kovariansmatricen  $\mathbf{C}$ . Niveaukurverne for gaussfilteret er ellipser. Størrelsen, orienteringen og eccentriciteten af disse er bestemt af  $\mathbf{C}$ . Som vi skal se, har matricen i 2D netop 3 frihedsgrader. Vi skal derfor starte med at forstå  $\mathbf{C}$ .

Antag at  $n = 2$ , og at der er givet  $N$  målinger  $(x_i, y_i)$ . Disse kan være koordinater, eller vilkårlige andre samhørende værdier. Som vi skal se senere, vil vi konkret benytte målinger af afledede i x- og y-aksens retninger. Lad  $\mathbf{m}_x$  hhv.  $\mathbf{m}_y$  være de forventede værdier  $E\{x_i\}$  og  $E\{y_i\}$  af x- hhv. y-koordinaterne for billedpunkterne. Lad endvidere:

$$\mathbf{x} = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_N & y_N \end{pmatrix} \quad \mathbf{m} = \begin{pmatrix} m_x & m_y \end{pmatrix} \quad (3.24)$$

Således angiver  $\mathbf{m}$  massemidtpunktet for segmentet.

Kovariansmatricen  $\mathbf{C}$  defineres da ved:

$$\mathbf{C} = E\{(\mathbf{x} - \mathbf{m})^t(\mathbf{x} - \mathbf{m})\} \quad (3.25)$$

$$= E\{\mathbf{x}^t \mathbf{x} - 2\mathbf{x}^t \mathbf{m} + \mathbf{m}^t \mathbf{m}\} \quad (3.26)$$

$$= E\{\mathbf{x}^t \mathbf{x}\} - \mathbf{m}^t \mathbf{m} \quad (3.27)$$

Idet vi approksimerer den forventede værdi med gennemsnittet af de tilgængelige målinger, får vi:

$$\mathbf{C} \approx \frac{1}{N} \left[ \sum_{i=1}^N \mathbf{x}_i^t \mathbf{x}_i \right] - \mathbf{m}^t \mathbf{m} \quad (3.28)$$

Kovariansmatricen er kvadratisk med størrelsen  $n \times n$  og har formen:

$$\mathbf{C} \approx \begin{pmatrix} a & c \\ c & b \end{pmatrix} \quad (3.29)$$

$$a = \sigma_x^2 = \frac{1}{N} \sum_{i=1}^N x_i^2 - \left( \frac{1}{N} \sum_{i=1}^N x_i \right)^2 \quad (3.30)$$

$$b = \sigma_y^2 = \frac{1}{N} \sum_{i=1}^N y_i^2 - \left( \frac{1}{N} \sum_{i=1}^N y_i \right)^2 \quad (3.31)$$

$$c = \text{cov}(x, y) = \frac{1}{N} \sum_{i=1}^N x_i y_i - \left( \frac{1}{N} \sum_{i=1}^N x_i \right) \left( \frac{1}{N} \sum_{i=1}^N y_i \right) \quad (3.32)$$

Kovariansmatricen  $\mathbf{C}$  er en statistisk beskrivelse af punktmængden  $(x_i, y_i)$ . En forståelse af matricen kan fås gennem en *egenværdianalyse* af denne. Vi husker (fra et grundlæggende kursus i

lineær algebra) at  $\mathbf{v}$  er en *egenvektor* med *egenværdi*  $\lambda$  for  $\mathbf{C}$  hvis der gælder at:

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v} \quad (3.33)$$

Fra lineær algebra husker vi også, at da  $\mathbf{C}$  er reel og symmetrisk, er egenværdierne alle reelle. Da  $\mathbf{C}$  er positiv definit (per konstruktion, idet  $\mathbf{x}$  har fuld søjlerang lig 2, er egenværdierne ikke-negative. Endelig er egenvektorerne ortogonale da  $\mathbf{C}$  er symmetrisk. For at finde egenværdierne opstilles det karakteristiske polynomium for  $\mathbf{C}$ :

$$\det(\mathbf{C} - \lambda\mathbf{I}) = \lambda^2 - (a+b)\lambda + (ab - c^2) = 0 \quad (3.34)$$

Egenværdierne er derfor lig:

$$\lambda = \frac{a+b}{2} \pm \frac{\sqrt{(a-b)^2 + 4c^2}}{2} \quad (3.35)$$

Vi kan finde egenvektoren  $\vec{v}_1 = \frac{1}{\sqrt{1+v^2}}(1, v)^t$  til egenværdien  $\lambda$  fra første ligning i definitionen:

$$\frac{1}{\sqrt{1+v^2}} \begin{pmatrix} a & c \\ c & b \end{pmatrix} \begin{pmatrix} 1 \\ v \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ v \end{pmatrix} \frac{1}{\sqrt{1+v^2}} \quad (3.36)$$

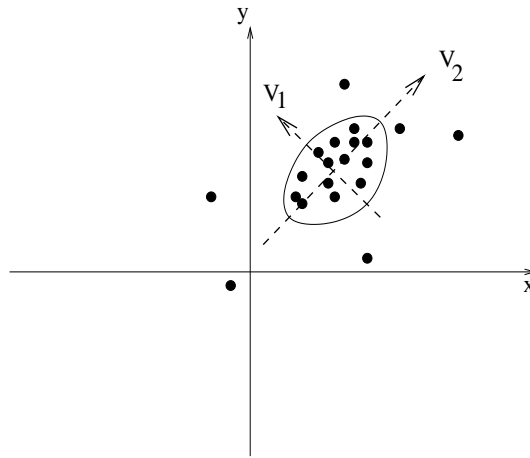
hvoraf vi får  $v = \frac{\lambda-a}{c}$ . Den anden egenvektor  $\vec{v}_2 = \frac{1}{\sqrt{1+v^2}}(-v, 1)^t$  fås som den ortogonale til  $\vec{v}_1$ . Den geometriske fortolkning af disse værdier er:

- Hvis de to egenværdier er lige store så er punktmængden  $(x_i, y_i)$  isotropt fordelt omkring centroiden (massemidtpunktet) for punkterne.
- Hvis den ene egenværdi er lig 0 så ligger punkterne på en ret linie gennem centroiden og med orientering givet ved egenvektoren med egenværdi  $> 0$ .
- Hvis  $0 < \lambda_1 < \lambda_2$  udgør punktmængden en mere eller mindre aflang 'pølse' med hovedakse givet ved  $\vec{v}_2$  (med varians  $\lambda_2$ ).

For det generelle (sidste af ovenvævnte) tilfælde kan punktskyen modelleres ved en ellipse med en storakse proportional med  $\sqrt{\lambda_2}$  og lilleakse proportional med  $\sqrt{\lambda_1}$ , og orienteret med storaksen efter  $\vec{v}_2$  og lilleaksen efter  $\vec{v}_1$ . Eccentriciteten er givet ved forholdet  $\sqrt{\lambda_2}/\sqrt{\lambda_1}$ . Ved for hver pixel at bestemme en (skaleret) kovariansmatrix kan vi derfor bestemme et generelt to-dimensionalt gaussfilter. Dette vil være orienteret langs med niveaukurven for intensitetsfunktionen gennem pixelen og med en eccentricitet, der er relateret til kontrasten i pixelen. Herved opnår vi en filtrering, der respekterer strukturerne i billedet, og undgår at udglatte på tværs af store kontraster. Tilbage står kun at bestemme den punktmængde  $(x_i, y_i)$  hvoraf kovariansmatrixen  $\mathbf{C}$  skal estimeres. En egenskab ved niveaukurverne er, at disse lokalt forløber vinkelret på gradienten. Oplagt er derfor at bestemme gradienten i alle nabopunkter (indenfor en vis radius). Et estimat af gradienten i punktet  $(i, j)$  er:

$$I_x(i, j) = I(i, j) - I(i-1, j) \quad (3.37)$$

$$I_y(i, j) = I(i, j) - I(i, j-1) \quad (3.38)$$



Figur 3.7: Egenvektorer for kovariansmatricen for en punktmængde, samt ellipsen med akser givet ved  $2\sigma_x$  og  $2\sigma_y$ .

Ovenstående gennemgang kan synes besværlig, men er vigtig - ikke kun af hensyn til at opnå en god filtrering - men også fordi samme metode kan anvendes i et stort antal tilfælde. To sådanne eksempler er detektion af hjørnepunkter og karakterisering af formen på et segmenteret objekt. Vigtigt i ovenstående filtreringsmetode er, at vi aldrig behøver udregne egenværdier og egenvektorer eksplicit. Alt hvad der er krævet er, at vi kan estimere gradienten, og at vi kan beregne kovariansmatricen.

### Strukturtensor

I kapitel 10 skal vi dels se hvorledes de partielle afledede  $I_x$  og  $I_y$  af intensitetsfunktionen  $I$  kan estimeres, og dels hvorledes matricen:

$$S = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (3.39)$$

indgår i bestemmelsen af en række forskellige mål for krumningen af intensitetsfladen i et givet punkt. Vi skal også se at en ofte benyttet metode til estimation af  $(I_x, I_y)$  er at folde  $I$  med de partielle afledede af en Gaussfunktion med skala  $\sigma_1$ . Denne parameter er bestemt af mængden af støj i billedet.

*Strukturtensor* er et Gaussisk vægtet gennemsnit af ovenstående strukturmatrix  $S$ , og kan skrives:

$$ST = G_{\sigma_2} \star \begin{bmatrix} (I_x^{\sigma_1})^2 & (I_x^{\sigma_1})(I_y^{\sigma_1}) \\ (I_x^{\sigma_1})(I_y^{\sigma_1}) & (I_y^{\sigma_1})^2 \end{bmatrix} \quad (3.40)$$

hvor  $\sigma_2$  er den skala hvorpå gennemsnittet tages, dvs. et mål for støtten af det lokale gennemsnit. Som det ses ligner ovenstående udtryk definitionen af kovariansmatricen (for de partielle afledede af  $I$ ) i ligning (3.28). En forskel er at ovenstående udtryk involverer et vægtet gennemsnit,

hvilket øger stabiliteten af målet. En anden forskel er at der i ovenstående udtryk ikke subtraheres middel-strukturmatricen  $\mathbf{m}^t \mathbf{m}$ . Struktur tensoren benyttes i mange metoder til detektion af linier og hjørnepunkter i billeder.

### 3.4.2 Normaliseret foldning

Normaliseret foldning er en nyttig metode hvis billedet indeholder mindre områder hvori intensiteten ikke er kendt. Ved normaliseret foldning kan sådanne områder udfyldes. De manglende data kan opstå under selve samplingprocessen eller være markeret i en tidligere analyse. Et typisk eksempel er retouchering af billeder, hvor visse områder manuelt er markeret. Tiltroen til billed-informationen i pixelen  $(x, y)$  kan modelleres ved en binær funktion  $c(x, y)$ . I visse tilfælde er tiltroen ikke binær, men graderet. I sådanne tilfælde kan  $c(x, y) \in [0 : 1]$  opfattes som en sandsynlighed for korrekte data i  $(x, y)$ . I begge tilfælde foretages foldningen ved et lokalt vægtet gennemsnit af de værdier i omegnen. Vægtene er valgt som produktet af sandsynligheden  $c(x, y)$  med en filterfunktion  $f(x, y)$ .

$$O(x, y) = \frac{[f(x, y) \cdot c(x, y)] \star I(x, y)}{f(x, y) \star c(x, y)} \quad (3.41)$$

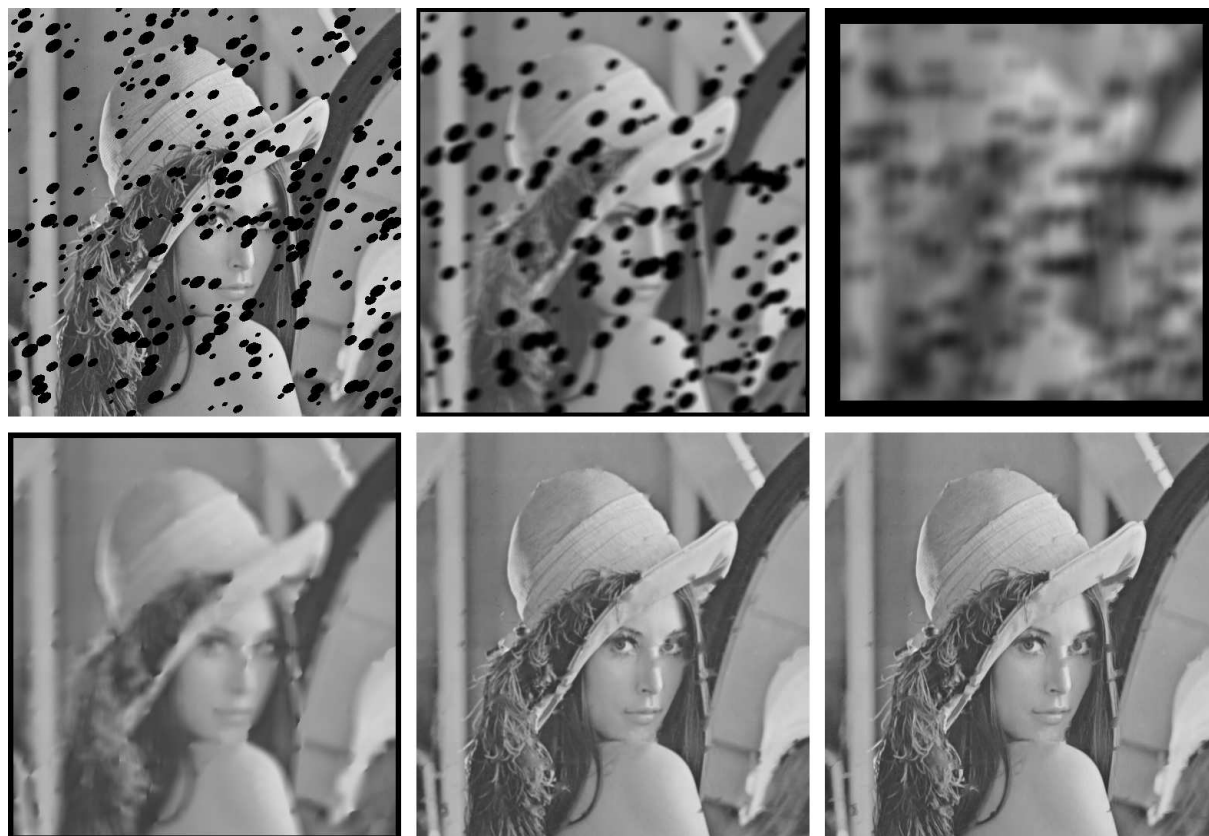
Filterfunktionen  $f(x, y)$  kan vælges vilkårligt. Et typisk eksempel er en todimensional gauss-funktion.

Selv om filterfunktionen  $f(x, y)$  er konstant, er normaliseret filtrering positionsafhængig fordi  $c(x, y)$  er det. I mange situationer kan  $f(x, y)$  med fordel gøres positionsafhængig, f.eks. ved at bestemme spredningen af gaussfilteret som halvdelen af afstanden til nærmeste pixel med veldefineret værdi ( $c(x, y) = 1$ ). I dette tilfælde vil værdien af pixels lokaliseret midt i et 'stort' område af udefinerede data blive bestemt af de veldefinerede data udenfor området. Pixels med veldefinerede naboer kan efterlades med (stort set) uændret værdi.

Væsentlig bedre rekonstruktion af de manglende data kan fås, hvis der yderligere benyttes et anisotrop gaussfilter. Herved kan strukturer/kontraster i veldefinerede pixels i omegnen få stor indflydelse på rekonstruktionen.

#### Eksempel

Nedenstående vises et originalbillede med områder, hvor data mangler. Til højre for dette er vist to billeder der fremkommer ved sædvanlig foldning med en gaussfunktion med konstant lille hhv. stor spredning. Det ses at en sædvanlig foldning ikke kan genskabe en acceptabel rekonstruktion af billedet. I anden række er vist resultater af normaliseret gauss-foldning med fast spredning, med spredning bestemt ud fra afstanden til nærmeste veldefinerede punkt, og med den fulde kovariansmatrice bestemt lokalt (anisotrop normaliseret gaussfiltrering). Det ses tydeligt at rekonstruktionerne i sidste række er bedre, og at sidstnævnte filtrering bedre end de øvrige formår at rekonstruere strukturerne i områderne med manglende data.



Figur 3.8: Rekonstruktion af et billede med manglende data (øverst tv.) ved sædvanlig foldning (øverst midt og th.) samt ved tre varianter af normaliseret gauss-foldning (se teksten for yderlige beskrivelse).





# Kapitel 4

## Billedrestaurering

### 4.1 En simpel model for billedforringelse

Ved opstilling af modeller for billedforringelse og for filtrering af billeder er det hensigtsmæssigt at formulere modellerne for kontinuerte billeder. Dette muliggør anvendelse af den sædvanlige mål- og integralteori. I det følgende antages derfor, at det observerede billede  $g$  er en kontinuert, to-dimensional, reel funktion. Vi vil antage, at  $g$  fremkommer ved forvrængning af en ideel intensitetsfunktion  $f$  efterfulgt af en addition af en støjfunktion  $\eta$ :

$$g = H[f] + \eta \quad (4.1)$$

hvor  $H$  er en operator på mængden af kontinuerte 2-dimensionale reelle funktioner, og  $\eta$  er en endelig reel funktion. Det antages altså, at støjen er uafhængig af billedet  $f$  og af forvrængningen  $H$ .

Antag videre, at operatoren  $H$  er lineær og positionsinvariant. Hvis støjen  $\eta$  et øjeblik antages lig nulfunktionen, kan vi skrive:

$$g(x, y) = H \left[ \int \int f(i, j) \delta(x - i, y - j) di dj \right] \quad (4.2)$$

hvor  $\delta$  er den sædvanlige deltafunktion (impulsfunktion). Da  $H$  er lineær og  $f(i, j)$  er uafhængig af  $x$  og  $y$  fås:

$$g(x, y) = \int \int f(i, j) H[\delta(x - i, y - j)] di dj \quad (4.3)$$

Fordi  $H$  er positionsinvariant kan vi skrive:

$$g(x, y) = \int \int f(i, j) h(x - i, y - j) di dj \quad (4.4)$$

hvor  $h(x, y) = H[\delta(x, y)]$  er operatoren  $H$  anvendt på impulsen  $\delta(x, y)$ . Funktionen  $h$  kaldes forvrængningens impulsrespons/impulssvar eller punktspredningsfunktionen for forvrængningen. Det ses at hvis forvrængningen er positionsinvariant, da er denne fuldstændigt karakteriseret ved svaret på en impuls i punktet  $(0,0)$ .

Af ovenstående ligning ses, at  $g$  kan skrives som foldningen af  $f$  med  $h$ . I det generelle tilfælde, hvor der optræder additiv støj, får vi:

$$g = h * f + \eta \quad (4.5)$$

Ovenstående model er, på trods af dens simple form, fundamentet for en lang række metoder til filtrering af billeder. Generelt er opgaven at finde en operator  $G$  således at  $f \approx Gg$ . Dette problem benævnes gendannelsesproblemet (eng. restoration). I praksis kræves en vis viden om støjen  $\eta$  og om punktspredningsfunktionen  $h$  for at kunne foretage en approksimativ gendannelse. En eksakt gendannelse er sjældent mulig. F.eks. kan forringelsen af billedet (og kvantificeringen af billedværdier til heltal) have fjernet visse detaljer helt. I det simple tilfælde, hvor  $h$  er lig en impulsfunktion (dvs.  $g = f + \eta$ ), tales om støjfiltrering.

### 4.1.1 Forvrængning

Et typisk eksempel på en forvrængning af et billede opstår ved fotografering af en scene gennem et røgfyldt eller tåget medium. Dette kan være tilfældet ved satellitbilleder, hvor et tyndt skylag bevirker en spredning af de reflekterede lysstråler. Et andet eksempel er et billede optaget med et kamera, der ikke er fokuseret på scenen. Bemærk at det her ikke altid kan antages at forvrængningen er positionsinvariant. I begge tilfælde vil det observerede billede være uskarpt (mere eller mindre udtværet) i forhold til det ideelle billede. I sådanne tilfælde vil forvrængningen ofte være rotationssymmetrisk og monotont aftagende. En prototype på en sådan punktspredningsfunktion er den to-dimensionale gaussfunktion.

I nogle tilfælde vil punktspredningsfunktionen ikke være rotationssymmetrisk. Et eksempel er en afbildning af et objekt i bevægelse som f.eks. et billede af en bil i fart. Hvis eksponeringstiden er for stor, vil billedet af bilen være udtværet i retningen givet ved den projicerede bevægelse.

Generelt er det meget svært at bestemme punktspredningsfunktionen. Indeholder det ideelle billede et sort punkt på en hvid baggrund (eller omvendt) kan funktionen approximeres ved en skaleret udgave af det observerede punkt. Denne teknik kan også anvendes hvis  $I_0$  indeholder en ideel trappetrinskant. Her kan vi dog kun måle forvrængningen vinkelret på kanten og kun indirekte ved et integral af punktspredningsfunktionen. Som beskrevet tidligere er det imidlertid ikke let at gendanne det ideelle billede, selv om punktspredningsfunktionen er kendt med stor nøjagtighed. I det følgende vil vi, med mindre andet nævnes, antage at det billede, som vi vil filtrere, ikke har været udsat for nogen forvrængning under optagelsen.

### 4.1.2 Støj

Overordnet antages, som tidligere beskrevet, at støjen er uafhængig af det ideelle billede og af en evt. forvrængning. Desuden antages støjen at være en stokastisk proces, hvor støjkomponenterne i to forskellige billedpunkter er uafhængige. Ofte er støjen additiv, men visse sensorer, som f.eks. radar, har multiplikativ støj. Hvis støjen i hver pixel, ud over at være uafhængig, kan antages at være trukket fra ens fordelinger siges støjen at være i.i.d. (independent identically distributed). Hvis såvel middelværdien som variansen af støjen er konstant over billedet, siges støjen at være

*stationær*. Ofte antages at middelværdien af støjen er lig nul. I så tilfælde siges støjen at være *hvid*.

Væsentlige er også støjens statistiske egenskaber. Tre ofte anvendte støjtyper er:

1. Uniformt fordelt støj. Alle støjværdier er lige sandsynlige.
2. Gaussisk -eller normalfordelt støj. Denne støjtype er typisk for elektriske kredsløb, fordi denne kan beskrives som summen af mange små ensfordelte støjbidrag. Ifølge *De store tals lov* i sandsynlighedsregningen vil summen da være normalfordelt uafhængig af fordelingen af bidragene.
3. Salt og pebber støj. Støjen kan beskrives ved overskrivning af en vis procentdel af de oprindelige billedværdier med den mindst hhv. den størst mulige billedværdi. Denne støjtype kan f.eks. forekomme i slidte/gamle kameraer, hvor nogle celler er “faldet ud” og andre giver værdi som ved en overeksponering.

Det er meget vigtigt ved konstruktion af metoder til bortfiltrering af støj og til billedbehandling generelt at undersøge støjens natur, således at den korrekte model for forringelsen kan opstilles. Omvendt er det vigtigt ved anvendelse af et filter, at gøre sig klart hvilke antagelser om støjens karakter filtret forudsætter, samt hvilken støj der aktuelt er til stede i billedet.

Mængden af støj i et signal eller billede måles ofte ved *signal-støjforholdet* SNR. Dette er defineret ved *energien* i billedet divideret med energien i støjen:

$$\text{SNR}_{ms} = \frac{\int \int f^2 dx dy}{\int \int \eta^2 dx dy} \quad (4.6)$$

Ofte benyttes  $\text{SNR}_{rms}$  (eng. root mean square signal to noise ratio) defineret ved kvadratroden af  $\text{SNR}_{ms}$ -størrelsen. Under antagelsen om at støjen er hvid, er det let at vise at variansen af støjen er lig middelen energien i støjen.

$$\sigma_\eta^2 = E(\eta^2) - E^2(\eta) = E(\eta^2) = \frac{1}{||A||} \int_A \eta^2 dx dy \quad (4.7)$$

Ligeledes kan man vise, at styrkespektret for hvid støj (i teorien) er en konstant funktion, hvis værdi netop er lig variansen.

Et andet mål er PSNR (*Peak signal to noise ratio*), defineret ved:

$$\text{PSNR} = 20 \log_{10} \left( \frac{\text{Max}}{\text{SNR}_{rms}} \right) \text{dB} \quad (4.8)$$

hvor Max er den størst mulige værdi (ofte 255.0). PSNR måles i decibel (dB). Bemærk at en forøgelse af PSNR på 20 dB svarer til en 10 gange mindre værdi af rms-værdien.

Der findes en række metoder til estimation af mængden af støj i billeder. Estimationen er vanskelig, fordi det er problematisk at skelne mellem egentlig støj og kontraster, f.eks. relateret til tekstur i billederne. Metoderne vil derfor ofte overestimere mængden af støj. En anvendelig metode er først at markere de billedpunkter, hvor størrelsen af intensitetsgradienten er lille (se evt. Kapitel 11), f.eks. kan man vælge at markere de 10 procent billedpunkter med mindst gradientstørrelse. For hver af disse punkter subtraheres et estimat af den lokale middelværdi fra intensiteten, og variansen af de fremkomne tal benyttes som estimat af variansen af støjen.

## 4.2 Gendannelsesproblemet

Udgangspunktet for gendannelse af forringede billeder er den tidligere beskrevne simple model (4.1). Gendannelsesproblemet består i at finde en operator  $G$  der, anvendt på et forringet billede  $g$ , vil tilnærme det oprindelige ideelle billede  $f$ . Jævnfør den simple model  $g = H[f] + \eta$  bliver estimatet:

$$\hat{f} = G[g] = G[H[f]] + G[\eta] \quad (4.9)$$

Gendannelsesfejlen bliver således:

$$e = \|f - \hat{f}\| = \|(1 - GH)[f] - G[\eta]\| \quad (4.10)$$

hvor “1” angiver identitetsoperatoren. Af ovenstående ligning fremgår, at  $G$  dels bør vælges således at  $G[\eta]$  er lille, dels som den (approximativt) inverse til operatoren  $H$ . Disse to ønsker er ofte uforenelige. Desuden er det på ingen måde sikkert, at den inverse til  $H$  eksisterer. Hvis f.eks.  $H$  er idealfilteret, vil informationen knyttet til de eliminerede frekvenser ikke kunne genskabes. Selv for operatorer hvis punktspredningsfunktion har en strengt positiv fouriertransformeret vil rekonstruktionen ofte være utilfredsstillende. Dette skyldes, at repræsentationen af flydende tal kun har en begrænset nøjagtighed. Desuden er hverken  $H$  eller  $\eta$  normalt kendt eksakt, men må estimeres.

I det følgende beskrives en række klassiske metoder til gendannelse. Metoderne bygger alle på operationer i frekvensdomænet. I det simpleste tilfælde, hvor støjen  $\eta$  antages lig nul, vælges  $G$  som den (approximativt) inverse af  $H$ . Dette kaldes *invers filtrering*. Generelt er det nødvendigt at medtage støjkomponenten i modellen. Et eksempel på et sådan filter er *Wienerfilteret*. De nævnte filtre kan udledes og beskrives ved brug af flere forskellige matematiske eller statistiske metoder. Vi skal her benytte den algebraiske tilgang.

## 4.3 Det inverse filter

Lad  $g$  være et observeret billede fremkommet ud fra et idelt billede  $f$  ved forvrængning  $H$  og addition af uafhængig støj  $\eta$ . Som tidligere antages at  $H$  er lineær og positionsinvariant. Ved at nummerere alle billedpunkter (f.eks. række for række) kan funktionerne  $f$ ,  $g$ , og  $\eta$  opskrives som tre søjlevektorer. Indeholder disse billeder  $N$  punkter ses, at operatoren  $H$  kan opskrives som en

$N \times N$  matrice (da  $H$  er lineær).

I resten af kapitlet benyttes store bogstaver  $H$ ,  $W$  etc. for matricer og små bogstaver  $f$ ,  $g$  etc. for vektorer. Små fede bogstaver  $\mathbf{f}$ ,  $\mathbf{g}$  etc. angiver vektorer i frekvensområdet. Med denne notation kan den simple model for billedforringelse skrives som et sæt af lineære ligninger:

$$\eta = g - Hf \quad (4.11)$$

Idet vi på sædvanlig måde definerer kvadrat-normen  $\|x\|^2$  af en vektor  $x$  ved det indre produkt  $x^t x$ , hvor  $x^t$  angiver den transponerede af  $x$ , kan vi skrive:

$$\|\eta\|^2 = \|g - Hf\|^2 \quad (4.12)$$

I udledningen af det inverse filter antages, at der ikke eksisterer nogen a priori viden om støjvektoren  $\eta$ . I stedet søges den løsning  $\hat{f}$ , der minimerer normen af støjleddet. Anderledes sagt søges  $\hat{f}$  således at  $H\hat{f}$  approksimerer  $g$  bedst muligt i mindste kvadraters betydning, dvs. som den funktion, der minimerer:

$$J(\hat{f}) = \|g - H\hat{f}\|^2 \quad (4.13)$$

Den søgte løsning,  $\hat{f}$ , findes på sædvanlig måde ved at differentiere  $J$  mht.  $\hat{f}$  og ved at sætte resultatet lig nul:

$$\frac{\partial J(\hat{f})}{\partial \hat{f}} = -2H^t[g - H\hat{f}] = 0 \quad (4.14)$$

Under antagelse at  $H^t H$  har fuld rang fås:

$$\hat{f} = H^\# g = [H^t H]^{-1} H^t g \quad (4.15)$$

Filtret  $H^\# = [H^t H]^{-1} H^t$  kaldes den *pseudoinverse* til  $H$ . Hvis det videre antages at  $H^{-1}$  eksisterer, dvs. at determinanten af matricen  $H$  er forskellig fra nul, reduceres ovennævnte filter til *det inverse filter*:

$$[H^t H]^{-1} H^t = H^{-1} (H^t)^{-1} H^t = H^{-1} \quad (4.16)$$

Antag nu at  $H$  er en  $N \times N$  matrice, og definer  $N \times N$  matricen  $W$  ved:

$$W(j, k) = e^{\frac{2\pi i}{N} jk} \quad (4.17)$$

hvor  $i = \sqrt{-1}$ . Det kan vises at  $W^{-1}$  eksisterer og at  $W^{-1}(j, k) = e^{-\frac{2\pi i}{N} jk}$ . Desuden vil det gælde, at  $H$  kan skrives som:

$$H = W D W^{-1} \quad (4.18)$$

hvor  $D$  er en diagonalmatrice, og hvor diagonalelementerne (egenverdierne for matricen  $H$ ) netop er givet ved den fouriertransformerede  $\mathbf{h}$  af punktspredningsfunktionen  $h$  for  $H$ . Det tekniske bevis for ovenstående er forholdsvis omstændeligt. Der udnyttes at  $h$  antages positionsinvariant,

hvilket betyder at  $H$  er blokcirkulær. Interesserede læsere henvises til f.eks. [14, Kapitel 5, p. 257-268]. Endelig kan det vises at:

$$W^{-1}f = \mathbf{f} \quad (4.19)$$

hvor  $\mathbf{f}$  er den diskret fouriertransformerede af  $f$ . Ved udnyttelse af ovennævnte resultater kan den inverse filtrering skrives:

$$\begin{aligned} \hat{f} &= H^{-1}g \\ &= (WDW^{-1})^{-1}g \\ &= WD^{-1}W^{-1}g \end{aligned} \quad (4.20)$$

og dermed:

$$W^{-1}\hat{f} = D^{-1}W^{-1}g \quad (4.21)$$

Da  $D$  er en diagonalmatrice med elementer  $\mathbf{h}(u, v)$  betyder ovenstående ligning, at den inverse filtrering kan formuleres i frekvensområdet ved:

$$\hat{\mathbf{f}}(u, v) = \frac{\mathbf{g}(u, v)}{\mathbf{h}(u, v)} \quad (4.22)$$

Denne meget ufuldstændige og dog omstændelige gennemgang bekræfter dermed, hvad man intuitivt ville have forventet ud fra foldningssætningen.

Ud over antagelsen om at  $H$  er kendt, er der i udledningen af det inverse filter gjort to skrappe antagelser, nemlig at  $H^{-1}$  eksisterer og at støjen er lig nul. Hvis  $\mathbf{h}(u, v)$  er forsvindende eller lig nul, er divisionen i ovennævnte ligning ikke veldefineret. Hvis problemet kun opstår i et lille antal punkter i frekvensområdet er en brugbar metode at definere resultatet af divisionen til nul. Problemet med støjen er større. Hvis  $\eta$  ikke er lig nul fås:

$$\hat{\mathbf{f}}(u, v) = \mathbf{f}(u, v) - \frac{\boldsymbol{\eta}(u, v)}{\mathbf{h}(u, v)} \quad (4.23)$$

Typisk vil støjen være højfrekvent, dvs. at  $\boldsymbol{\eta}(u, v)$  er ikke-forsvindende for store værdier af  $u$  og  $v$ , mens  $\mathbf{h}(u, v)$  typisk aftager meget stærkt som funktion af  $u$  og  $v$ . Dette betyder at sidste led i ovennævnte ligning bliver dominerende. Anderledes sagt vil gendannelsen  $\hat{f}$  kun have lidt til fælles med  $f$ . Af ovennævnte grunde er det inverse filter ikke anvendeligt i almindelighed. Smartere filtre, der i højere grad kan tage hensyn til eksistensen af støj, er nødvendige for en stabil gendannelse.

## 4.4 Wienerfilteret

Wienerfiltret hører til en klasse af betingede restaureringsfiltre. Før beskrivelsen af filtret skal vi kort gennemgå denne klasse. Som tidligere omtalt ønsker vi at finde en løsning  $\hat{f}$  således at  $(g - H\hat{f}) = \eta$ . Denne matrix-ligning består af  $N$  ligninger med  $2N$  ubekendte ( $N$  fra forvrængningen

og  $N$  fra støjleddet). Matrix-ligningen har derfor mange løsninger. I tilfældet med det inverse filter løste vi problemet ved at antage at halvdelen af de ubekendte (dem fra støjleddet) var lig nul. I klassen af betingede restaureringsfiltre tilføjes  $N$  nye (lineære) ligninger til de  $N$ , som de aktuelle data beskriver. Herefter kræves samtlige ligninger opfyldt bedst muligt. Denne metode går under navnet *regularisering*. I klassen af betingede rekonstruktioner formuleres de  $N$  nye ligninger ved:  $Q\hat{f} = 0$ , hvor  $Q$  er en lineær operator på  $f$ . En sammensat funktional  $J(\hat{f})$  opbygges ved anvendelse af en såkaldt *Lagransk multiplikator*  $\alpha$ :

$$J(\hat{f}) = \|Q\hat{f}\|^2 + \alpha(\|g - H\hat{f}\|^2 - \|\eta\|^2) \quad (4.24)$$

Løsningen til det sammensatte problem findes på sædvanlig måde ved at differentiere  $J$  og ved at sætte resultatet lig nul:

$$\frac{\partial J(\hat{f})}{\partial \hat{f}} = 2Q^t Q \hat{f} - 2\alpha H^t [g - H\hat{f}] = 0 \quad (4.25)$$

Løses denne ligning mht.  $\hat{f}$  fås:

$$\hat{f} = (H^t H + \gamma Q^t Q)^{-1} H^t g \quad (4.26)$$

hvor  $\gamma = 1/\alpha$ . Den endelige løsning findes ved at vælge  $\gamma$  således at kravet  $\|g - H\hat{f}\|^2 = \|\eta\|^2$  er opfyldt.

Det grundlæggende problem er at vælge en meningsfuld bibetingelse  $Q\hat{f}$ . I Wienerfiltret er matricen  $Q$  valgt således at:

$$Q^t Q = R_f^{-1} R_\eta \quad (4.27)$$

hvor

$$R_f^{-1} = E\{f f^t\}^{-1} \quad (4.28)$$

$$R_\eta = E\{\eta \eta^t\} \quad (4.29)$$

hvor  $E\{\}$  angiver forventet værdi, og  $f$  hhv.  $\eta$  er søjlematricer af billedværdier hhv. støjværdier.  $R_f^{-1}$  hhv.  $R_\eta$  angiver således korrelationsmatricerne for  $f$  hhv.  $\eta$ . Typisk aftager korrelationen mellem billedelementerne meget stærkt som funktion af afstanden mellem disse. Dette betyder at matricerne  $R_f^{-1}$  og  $R_\eta$  langs hoveddiagonalen typisk vil indeholde et bånd af værdier, der er forskellig fra nul. Uden for dette bånd vil matriceværdierne være lig nul. For sådanne matricer kan det vises (beviset falder langt uden for rammerne af disse noter) at matricerne kan approksimeres med blokcirkulære matricer. Dette betyder at vi kan skrive:

$$R_f^{-1} = W A W^{-1} \quad (4.30)$$

$$R_\eta = W B W^{-1} \quad (4.31)$$

hvor  $A$  hhv.  $B$  er diagonalmatricer indeholdende de fouriertransformerede af korrelationselementerne i  $R_f^{-1}$  og  $R_\eta$ . Disse værdier, der i det følgende betegnes  $S_f$  hhv.  $S_\eta$ , er igen lig

værdierne af styrkespektret for  $f$  hhv.  $\eta$ . Idet vi udnytter at:

$$H = WDW^{-1} \quad (4.32)$$

$$H^t = WD^*W^{-1} \quad (4.33)$$

hvor  $D^*$  angiver den kompleks konjugerede af  $D$ , kan vi nu opskrive Wienerfiltret:

$$\hat{f} = (H^t H + \gamma R_f^{-1} R_\eta)^{-1} H^t g \quad (4.34)$$

$$= (WD^*DW^{-1} + \gamma WA^{-1}BW^{-1})^{-1} WD^*W^{-1}g \quad (4.35)$$

Ved multiplikation af begge sider med  $W^{-1}$  og ved reduktion fås:

$$W^{-1}\hat{f} = (D^*D + \gamma A^{-1}B)^{-1} D^*W^{-1}g \quad (4.36)$$

Idet såvel  $D$  som  $A$  og  $B$  er diagonalmatricer kan vi derfor opskrive frekvensfilteret:

$$\hat{\mathbf{f}}(u, v) = \left[ \frac{\mathbf{h}^*(u, v)}{|\mathbf{h}(u, v)|^2 + \gamma [\mathbf{S}_\eta(u, v)/\mathbf{S}_f(u, v)]} \right] \mathbf{g}(u, v) \quad (4.37)$$

$$= \frac{1}{\mathbf{h}(u, v)} \left[ \frac{|\mathbf{h}(u, v)|^2}{|\mathbf{h}(u, v)|^2 + \gamma [\mathbf{S}_\eta(u, v)/\mathbf{S}_f(u, v)]} \right] \mathbf{g}(u, v) \quad (4.38)$$

Det 'ægte' Wienerfilter fremkommer ved valg af  $\gamma = 1$ . Hvis  $\gamma$  betragtes som variabel tales om *det parametriske Wienerfilter*. Bemærk, at hvis der ingen støj er tilstede, da vil styrkespektret  $\mathbf{S}_\eta = 0$ . I dette tilfælde reducerer filtret altså til det inverse filter. Bemærk også ved valget  $\gamma = 1$  er der ikke længere nogen garanti for at betingelsen  $\|g - H\hat{f}\|^2 = \|\eta\|^2$  er opfyldt. Derimod kan det vises at udtrykket  $E\{[f(x, y) - \hat{f}(x, y)]^2\}$  antager minimal værdi for dette valg.

For at kunne anvende wienerfiltret er det et krav at såvel styrkespektret  $\mathbf{S}_f$  for det ideelle signal  $f$  er kendt, og at styrkespektret  $\mathbf{S}_\eta$  for støjen er kendt. Dette er ikke normalt tilfældet. Hvis støjen er hvid, kan det vises at  $E\{\mathbf{S}_\eta\} = \sigma_\eta^2$ . Dette løser imidlertid ikke problemet omkring kendskabet til  $\mathbf{S}_f$ . I praksis ses nedenstående approksimation ofte anvendt:

$$\hat{\mathbf{f}}(u, v) \approx \frac{1}{\mathbf{h}(u, v)} \left[ \frac{|\mathbf{h}(u, v)|^2}{|\mathbf{h}(u, v)|^2 + K} \right] \mathbf{g}(u, v) \quad (4.39)$$

hvor  $K^{-1}$  angiver signal-støjforholdet. I praksis virker Wienerfiltret meget bedre end det inverse filter, hvis det oprindelige billede indeholder blot anelse støj. Af formel (4.38) fremgår, at frekvenserne i estimatet vægtes med det forventede støj/signal-forhold. Effekten af at minimere  $\|Q\hat{f}\|$  bliver da at betone de frekvenser i estimatet, hvor det forventede signal/støj-forhold er højt, og undertrykke de frekvenser, hvor forholdet er lavt. Denne effekt kan samtidig forklare, hvorfor Wienerfiltret i praksis har en tendens til at opbløde kanter i billedet. Hvis støjen er hvid vil  $\mathbf{S}_\eta$  (i teorien) være konstant. Derimod vil  $\mathbf{S}_f$  aftage med afstanden fra origo. Resultatet bliver at de lave frekvenser betones, mens de høje frekvenser undertrykkes. Filtret har altså lav-pas karakteristik.



## 4.5 Det geometriske middelværdifilter

Man kunne ønske at udvikle et filter, der lå “tæt ved” det inverse filter, men ikke døjede med singulariteterne i dette filter. En mulig parametrisering kunne være:

$$\hat{f} = (\text{invers filter})^\alpha (\text{parametrisk wiener filter})^{1-\alpha} g \quad (4.40)$$

hvor  $0 \leq \alpha \leq 1$ . I fourierområdet kan denne ide formuleres ved filteret:

$$\hat{f}(u, v) = \left[ \frac{\mathbf{h}^*(u, v)}{|\mathbf{h}(u, v)|^2} \right]^\alpha \left[ \frac{\mathbf{h}^*(u, v)}{|\mathbf{h}(u, v)|^2 + \gamma [\mathbf{S}_\eta(u, v)/\mathbf{S}_f(u, v)]} \right]^{1-\alpha} \mathbf{g}(u, v) \quad (4.41)$$

Det geometriske middelværdifilter fremkommer nu ved valg af  $\alpha = 1/2$ . Hvis punktspredningsfunktionen er symmetrisk vil  $\mathbf{h} = \mathbf{h}^*$ . I dette tilfælde får filteret formen:

$$\hat{f}(u, v) = \left[ \frac{\mathbf{S}_f(u, v)}{\mathbf{S}_f(u, v) |\mathbf{h}(u, v)|^2 + \gamma \mathbf{S}_\eta(u, v)} \right]^{1/2} \mathbf{g}(u, v) \quad (4.42)$$

I visse tilfælde virker dette filter bedre end både det inverse og wienerfilteret.

## 4.6 Bayes Restoration

En teoretisk anden måde at bestemme restaurationsfilteret på er ved at betragte billederne som stokastiske felter (markov random fields), hvor de enkelte billedelementer er stokastiske variable. Dette muliggør anvendelsen af *Bayes sætning*, idet man så kan tale om sandsynligheden for observation af bestemte billeder. Bayes sætning siger at:

$$p(f|g) = \frac{p(g|f)p(f)}{p(g)} \quad (4.43)$$

hvor  $p(f|g)$ ,  $p(g|f)$ ,  $p(f)$ , og  $p(g)$  er fire forskellige sandsynlighedsfordelinger. I det konkrete tilfælde kan ligningen læses: Sandsynligheden for det restaurerede billede  $f$  (vores model/fortolkning) givet det observerede billede  $g$  er lig sandsynligheden for observationerne givet modellen ganget med sandsynligheden for modellen og delt med sandsynligheden for observationerne. Givet  $g$  ønsker vi at bestemme det billede  $f$ , der er mest sandsynligt. Dvs. vi ønsker at finde det  $f$ , der maksimerer den *a posteriori sandsynlighed*  $p(f|g)$ . Dette kaldes et *Maksimum A Posteriori estimat* (MAP). Da  $p(g)$  er konstant kan dette udtryk ignoreres i optimeringen. Opgaven er derfor at bestemme  $f$ , sådan at  $p(g|f)p(f)$  maksimeres. Dette kræver, at der opstilles analytiske udtryk for fordelingerne af vores prior  $p(f)$  og af likelihoodfunktionen  $p(g|f)$ .

En rimelig og generel a priori antagelse er at intensiteten  $f$  er multinomialt fordelt med middelværdi  $\bar{f}$  og kovariansmatrice  $C_{ff}$ :

$$p(f) = \frac{1}{(2\pi)^{n/2} |C_{ff}|^{1/2}} e^{-\frac{1}{2}(f-\bar{f})^t C_{ff}^{-1}(f-\bar{f})} \quad (4.44)$$

hvor  $f$  betragtes som en søjlevektor af de  $n = N \times M$  værdier i billedet (f.eks. opstillet række for række), og hvor:

$$\bar{f} = E\{f\} \quad (4.45)$$

$$C_{ff} = E\{(f - \bar{f})(f - \bar{f})^t\} \quad (4.46)$$

hvor  $E\{\cdot\}$  angiver statistisk forventet værdi.  $\bar{f}$  er altså en søjlevektor med længde  $n$  af de statistisk forventede intensiteter. Matricen  $C_{ff}$  er  $n \times n$ , hvor det  $(i,j)$ 'te element angiver kovariansen mellem intensiteterne i positionerne  $i$  og  $j$ . Bemærk at diagonalelementerne af  $C_{ff}$  indeholder varianserne af de statistisk forventede intensiteter i hvert billedpunkt.

En typisk observation i mange billedpunkter er, at intensiteten er sammenlignelig med intensiteterne i en lille omegn om punktet. Denne observation angiver en mulig måde at estimere  $\bar{f}$  på, og svarer til den typiske antagelse om, at  $f$  er glat. Tilsvarende vil kovariansen for punkter langt fra hinanden som regel være forsvindende. Dette betyder at  $C_{ff}$  vil have båndform med de største værdier liggende tæt omkring diagonalen. I det følgende skal vi antage den mest simple form af covariansmatricen, nemlig at  $C_{ff}$  er en diagonalmatrice med samme konstante værdi i alle diagonalelementer:

$$\text{Cov}(f_i, f_j) = 0 \quad \text{og} \quad \text{Var}(f_i) = \sigma_f^2 \quad (4.47)$$

Dette betyder at:

$$p(f) = \frac{1}{(2\pi)^{n/2} \sigma_f^n} e^{-\frac{1}{2\sigma_f^2}(f - \bar{f})^t(f - \bar{f})} \quad (4.48)$$

Det ses, at  $\sigma_f$  specificerer vores a priori antagelse om, hvor glat intensiteten  $f$  er. Jo større  $\sigma_f$  jo mindre glat. Problemet om specifikation af den a priori fordeling  $p(f)$  er nu blevet reduceret til et spørgsmål om estimation af  $\bar{f}$  og af  $\sigma_f$ .  $\bar{f}$  kan mest simpelt estimeres ved middelværdien af intensiteterne i de umiddelbare nabopunkter. Dette betyder at  $f - \bar{f}$  kan skrives som foldningen  $c(x, y) \star f(x, y)$ , hvor:

$$c(x, y) = \begin{cases} 1 & \text{hvis } x = y = 0 \\ -\frac{1}{4} & \text{hvis } (x, y) \in \{(0, 1), (0, -1), (1, 0), (-1, 0)\} \end{cases} \quad (4.49)$$

Alternativt kan vi skrive  $f - \bar{f} = Cf$ , hvor  $C$  er en  $n \times n$  matrix:

$$C = \frac{1}{4} \begin{bmatrix} 4 & -1 & & & -1 & & & \\ -1 & 4 & -1 & & & & & \\ & -1 & 4 & -1 & & & & \\ & & & 4 & & \ddots & & \\ -1 & & & & \ddots & & -1 & \\ & -1 & & & & & & -1 \\ & & \ddots & & & & & \\ & & & -1 & & 4 & & \\ & & & & -1 & -1 & 4 & -1 \\ & & & & & -1 & -1 & 4 \end{bmatrix} \quad (4.50)$$

Den betingede sandsynlighed  $p(g|f)$  beskriver fordelingen af støjen i billedet, idet vi antager at  $g$  er fremkommet ved forvrængning med punktspredningsfunktionen  $h$  og addition af støjen  $\eta$ :  $g = h \star f + \eta$ . Givet  $f$  kan vi derfor for enhver  $g$  udregne  $\eta$ . Vi vil i det følgende antage at støjen er normalfordelt med middelværdi 0 og med varians  $\sigma_\eta$ :

$$p(g|f) = \frac{1}{(2\pi) \sigma_\eta^2} e^{-\frac{1}{2\sigma_\eta^2}(g-Hf)^t(g-Hf)} \quad (4.51)$$

hvor  $H$  er den til punktspredningsfunktionen svarende matrice.

Ønsket om at maksimere  $p(g|f)p(f)$  kan ækvivalent udtrykkes som ønsket om at minimere  $J(f) = -2 \log(p(g|f)p(f))$  (da logaritmefunktionen er monotont voksende).

$$J(f) = \text{konstant} + \frac{1}{\sigma_f^2}(Cf)^t(Cf) + \frac{1}{\sigma_\eta^2}(g-Hf)^t(g-Hf) \quad (4.52)$$

Ved at løse  $\frac{dJ(f)}{df} = 0$  fås:

$$\frac{1}{\sigma_f^2} C^t C f + \frac{1}{\sigma_\eta^2} H^t (g - H f) = 0 \quad (4.53)$$

På samme måde som tidligere kan ligningen omskrives til at gælde i fourierområdet (ligningen har - bortset fra andet valg af matricen  $C$  - helt samme form som i udledningen af Wienerfilteret). Løsningen er derfor:

$$\hat{\mathbf{f}}(u, v) = \frac{\mathbf{h}^*(u, v)}{|\mathbf{h}(u, v)|^2 + \gamma |\mathbf{C}(u, v)|^2} \mathbf{g}(u, v) \quad (4.54)$$

hvor  $\mathbf{C}(u, v)$  er styrkespektret til funktionen  $c(x, y)$ , og hvor  $\gamma = \sigma_\eta^2 / \sigma_f^2$ .

I visse tilfælde kan det være hensigtsmæssigt at bestemme  $\bar{f}$  ved gennemsnittet af intensiteterne i en omegn, hvis form ændrer sig fra punkt til punkt. På denne måde vil man kunne undgå, at glathedskravet forhindrer rekonstruktion af skarpe kontraster. I dette tilfælde kan "løsningen" mest hensigtsmæssigt skrives:

$$f = \bar{f} + \gamma^{-1} H^t (g - Hf) \quad (4.55)$$

Udtrykket indikerer en beregningsmetode ved iterativ estimation, hvor  $f^{(0)}$  bestemmes ved  $g$  selv, eller - meget bedre - f.eks. ved invers eller wienerfiltrering. En god initialværdi er normalt afgørende for en hurtig konvergens. Derefter itereres (til konvergens):

$$f^{(i+1)} = \bar{f}^{(i)} + \gamma^{-1} H^t (g - Hf^{(i)}) \quad (4.56)$$

Sidste led i opdateringsformlen implementeres lettest i forurierområdet. Den iterative metode er naturligvis mere beregningskrævende men tillader positionsvariante elementer, hvilket kan forbedre kvaliteten af rekonstruktionen afgørende.

Som i de tidligere nævnte metoder til rekonstruktion benytter ovenstående metode sig også af kendskab til en række parametre. Det drejer sig om tallene  $\sigma_f^2$ ,  $\sigma_\eta^2$ , samt om punktspredningsfunktionen  $h$ . Som tidligere beskrevet kan variansen af støjen estimeres rimeligt ud fra billedet  $g$  selv. Variansen af  $g$  vil ofte være et overtal for variansen af  $f$ , men det afhænger af  $f$  og af mængden af støj. Ved iterative metoder kan det være en fordel at starte med at vælge  $\sigma_f^2$  lille svarende til antagelse om stor glathed af  $f$ . Værdien kan så øges op mod variansen af  $f^{(t)}$ .

## 4.7 Iterative suboptimale Filtre

Der findes mange måder hvorpå Wienerfiltret kan forbedres. En måde er at formulere bibetingelser  $Q\hat{f} = 0$ , der bedre er i stand til at bevare kanterne i rekonstruktionen, end valget i (4.27) er i stand til. En anden måde er iterativt at bestemme nogle af de parametre, der enten er ukendte eller dårligt estimerede. Som eksempel kræves i den generelle formulering, at  $\gamma$  vælges i overensstemmelse med betingelsen  $\|g - H\hat{f}\|^2 = \|\eta\|^2$ . Idet vi kan definere residualvektoren  $r = (g - H\hat{f})$  fås:

$$r(\gamma) = g - H(H'H + \gamma Q'Q)^{-1} H'g \quad (4.57)$$

Det kan vises at  $\|r\|^2$  vokser monotont som funktion af  $\gamma$ . Hvis vi kan finde et udtryk for  $\|\eta\|^2$ , er det derfor let at konstruere en procedure, hvorved  $\gamma$  iterativt justeres, indtil  $\|r\|^2 - \|\eta\|^2$  antager minimal værdi. Det kan vises at der gælder:

$$\|n\|^2 = N[\sigma_\eta^2 + \bar{\eta}^2] \quad (4.58)$$

hvor  $\bar{\eta}$  hhv.  $\sigma_\eta^2$  er middelværdien og variansen af støjen, og  $N$  er det totale antal billedelementer. For nærmere diskussion af denne metode refereres til [14, 19].

# Kapitel 5

## Billedtransformationer

Digitale gråtonebilleder kan repræsenteres på et utal af måder, der hver har sine fordele og ulemper mht. en given behandling af billedet. Det klassiske eksempel er *frekvensrepræsentationen*, der opnås ved brug af Fouriertransformationen. I dette kapitel introduceres til en række andre af de transformationer, der oftest benyttes ved behandling af gråtonebilleder. Desuden diskuteres metoder til dimensionalitetsreduktion af højdimensionale data (PCA, ICA).

I det følgende betragtes *lineære transformationer* af et billede  $f(x, y)$  resulterende i en repræsentation  $T(u, v)$ . Sådanne transformationer kan generelt beskrives ved:

$$T(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)g(x, y, u, v) \quad (5.1)$$

hvor vi her antager at billedet er kvadratisk med kantlængde  $N$ .

Funktionen  $g(x, y, u, v)$  kaldes *transformationskernen*. Vi antager at der findes en *invers kerne*  $h(u, v, x, y)$  således at  $T(u, v)$  kan transformeres tilbage til det oprindelige billede  $f(x, y)$ :

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u, v)h(u, v, x, y) \quad (5.2)$$

De to ligninger kaldes ofte *analyseligningen* hhv. *synteseligningen*. Første ligning beskriver hvorledes vi givet billedet  $f(x, y)$  kan beregne et sæt af *koefficienter*  $T(u, v)$ . Anden ligning beskriver hvorledes vi kan syntetisere et billede  $f(x, y)$  ud fra et koefficientsæt.

Kernen  $g(x, y, u, v)$  siges at være *seperabel* hvis:

$$g(x, y, u, v) = g_1(x, u)g_2(y, v) \quad (5.3)$$

og symmetrisk hvis  $g_1 = g_2$  er identiske. Vi skal primært beskrive transformationer (med tilhørende repræsentationer), der er *seperable* og *symmetriske*. I dette tilfælde kan ovenstående transformationen opskrives som en simpel matrixoperation:

$$\mathbf{T} = \mathbf{A}'\mathbf{F}\mathbf{A} \quad (5.4)$$

hvor  $\mathbf{F}$  hhv.  $\mathbf{T}$  er  $N \times N$  billedmatricer, og  $\mathbf{A}$  er en  $N \times N$  transformationsmatrix med elementer  $a_{ij} = g_1(i, j)$ . For den en inverse kerne,  $h(u, v, x, y) = h_1(u, x)h_1(v, y)$ , kan på tilsvarende måde danne matricen  $\mathbf{B}$  med elementer  $b_{ij} = h_1(i, j)$ . Skrives:

$$\mathbf{B}'\mathbf{T}\mathbf{B} = \mathbf{B}'\mathbf{A}'\mathbf{F}\mathbf{A}\mathbf{B} \quad (5.5)$$

ses, at hvis  $\mathbf{B} = \mathbf{A}^{-1}$  vil:

$$\mathbf{F} = \mathbf{B}'\mathbf{T}\mathbf{B} \quad (5.6)$$

Dette er således kravet til, at vi kan rekonstruere  $f$  fuldstændigt ud fra  $T$ . De senere beskrevne repræsentationer opfylder alle dette krav.

En vigtig fortolkning af den inverse transformation (synteseligningen) er en beskrivelse af billedet  $f$  som en sum af *projektioner* på en mængde af *basisbilleder*  $B$  for transformationen.

$$f(x, y) = \sum_u \sum_v T(u, v) B_{(u, v)}(x, y) \quad (5.7)$$

Billedet  $f$  beskrives altså ved en linearkombination af billederne  $B(u, v)$ . Værdien  $T(u, v)$  hørende til basisbilledet  $B(u, v)$  kaldes i denne sammenhæng for koefficienten til basisbilledet. Koefficienterne findes vha. analyseligningen ved at projicere billedet  $f$  ned på underrummet  $U$  hørende til basisbilledet. Vi kan derfor skrive:

$$F = U_1 \oplus U_2 \oplus \cdots \oplus U_{N^2} \quad (5.8)$$

hvor  $F$  udgør mængden af billeder  $f$ , og  $\oplus$  angiver direkte sum. Hvis ovennævnte dekomposition af  $f$  er entydig, dvs. hvis det indre produkt  $\langle a, b \rangle$  mellem to elementer  $a \in U_i$  og  $b \in U_j$  er lig nul (for  $i \neq j$ ), siges dekompositionen at være ortogonal. Vi er primært interesserede i sådanne dekompositioner.

Den information, der er repræsenteret i et billede, er naturligvis også repræsenteret i koefficienterne til de til transformationen hørende basisbilleder. For forskellige transformationer er fordelingen af informationen på koefficienter forskellig. Visse transformationer formår bedre end andre at koncentrere informationen på få koefficienter. Kodningsmæssigt er det af interesse at bestemme ortogonale transformationer, der pakker mest mulig information i koefficienterne for de  $k < N^2$  første underrum/basisbilleder. Givet  $k$  kan billedet  $f$  approksimeres ved at ignorere informationen i de  $N^2 - k$  resterende koefficienter:

$$\hat{f} = \sum_{i=0}^k f_i \quad (5.9)$$

hvor  $f_i \in U_i$ . Ved rekonstruktion ud fra en sådan trunkering begås en fejl  $\|f - \hat{f}\|$ . Jo mindre denne fejl er, jo bedre er transformationen mht. kodning af billedet. Som vi skal se findes netop én transformation, der pakker informationen i et billede mest mulig ned i de første koefficienter. Denne transformation kan benyttes generelt på data på vektorform, og benyttes intensivt ved reduktion af dimensionaliteten af data.

Det skal understreges at der ikke findes én bedste transformation. Ved billedkodning ønskes således en transformation med den egenskab at rekonstruktionen  $\hat{f}$ , ud over at omfatte få koefficienter, subjektivt ligner originalen  $f$  godt. Et andet krav kunne være at strukturerne, f.eks. kanterne i et billede, var velbevarede i rekonstruktionen  $\hat{f}$ . Sådanne kriterier vil ofte lede til forskellige transformationer.

## 5.1 Klassiske repræsentationer

I dette afsnit beskrives nogle transformationer, der ofte refereres i litteraturen. Transformationerne beskrives kun kort uden diskussion af algoritmer til hurtig implementation af transformationerne (Se evt. [31, 13]). For alle transformationer gælder at de er separable og symmetriske. Vi skal derfor kun angive de en-dimensionale transformationer.

### 5.1.1 Repræsentation ved fourier-transformation

Fouriertransformationen antages velkendt, og beskrives derfor kun summarisk. I det endimensionelle tilfælde er kernen  $g(x, u)$  givet ved:

$$g(x, u) = \frac{1}{\sqrt{N}} e^{-2\pi i u x / N} \quad (5.10)$$

hvor vi betragter et signal med  $N$  stikprøver, og hvor  $i = \sqrt{-1}$ . Den inverse kerne er bestemt ved:

$$h(u, x) = e^{2\pi i u x / N} \quad (5.11)$$

Det kan vises, at fouriertransformationen er suboptimal i den forstand, at koefficienterne ikke er ukorrelerede. Dog vil graden af korrelation aftage når billedstørrelsen øges. Årsagen til transformationens popularitet er primært, at repræsentationen tillader en direkte fortolkning mht. frekvensopdeling. Dette muliggør en enkelt lineær positionsinvariant filtrering, sådan som det er beskrevet tidligere. En sekundær årsag er, at transformationen kan foretages relativ let og hurtigt vha. FFT-algoritmen. Kodningsmæssigt er transformationen uderlegen i forhold til både cosinus-transformationen og Karhuen-Loeve transformationen. En ulempe ved fouriertransformationen er antagelsen om at billedet er periodisk udvidet. Hvis intensiteterne i modsatte billedkanter er meget forskellige vil dette introducere høje frekvenser. Disse er altså forårsaget af kravet om periodisk udvidelse og er ikke relateret til diskontinuiteter i billedet selv. Dette fænomen giver ofte problemer ved analyser i denne repræsentation. Dæmpes de høje frekvenser, og invers fouriertransformeres det filtrerede billede, vil dette være tydeligt forvrænget. Problemet ses oftest ved små billeder eller ved blokkodning af billedet.

### 5.1.2 Repræsentation ved cosinus-transformation

Kernen for den diskrete cosinustransformation af et signal med  $N$  stikprøver er givet ved:

$$g(x, 0) = \frac{1}{\sqrt{N}} \quad (5.12)$$

$$g(x, u) = \sqrt{\frac{2}{N}} \cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

for  $x \in [0, N-1]$  og  $u \in [1, N-1]$ . Den inverse transformation er bestemt ved dekompositionen:

$$f(x) = \frac{1}{\sqrt{N}} C(0) + \sqrt{\frac{2}{N}} \sum_{u=1}^{N-1} C(u) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \quad (5.13)$$

hvor  $C(u)$  er transformationskoefficienterne til  $f$ .

Cosinustransformationen er meget benyttet ved kodning af billeder. En årsag er, som vi senere skal diskutere, at koefficienterne for transformationen er næsten dekorrelerede. Kodningsmæssigt er transformationen derfor approksimativt så god som Carhuen-Loeve-transformationen. Transformationen kan desuden implementeres hurtigt ved en algoritme, der meget ligner FFT-algoritmen. I forhold til fouriertransformationen antager cosinustransformationen ikke en periodisk udvidelse af billedet. I stedet er der tale om udvidelse ved spejling. Dette betyder at der kun sjældent opstår fejlfortolkninger af “frekvensinformationen”. Dette gør metoden mere attraktiv ved blokkodning end fouriertransformationen. Cosinustransformationen er valgt som international standard ved kompression af billeder (JPEG) og videosekvenser (MPEG) og visse lydkodere (MP). Især inden for kodning af musik benyttes et antal varianter af Cosinustransformationen (f.eks. *Den modificerede Cosinustransformation* der benyttes i Dolby AC3).

### 5.1.3 Repræsentation ved binære basisbilleder

*Walsh transformationen* og *Hardamard transformationen* er to transformationer, der begge, bortset fra en konstant, benytter basisbilleder hvis elementer er binære, dvs 1 og  $-1$ . Begge transformationer er lineære, seperable og symmetriske, og kan implementeres ved meget hurtige algoritmer. Transformationerne har dog begge dårlige egenskaber mht. dekorrelation, og benyttes sjældent. Der henvises til [13] for yderligere diskussion af metoderne.

Antag at billedstørrelsen  $N = 2^n$ , og at  $b_k(z)$  er defineret ved den  $k$ 'te bit i den binære repræsentation af  $z$ . Den endimensionale diskrete Walsh transformation er bestemt ved kernen:

$$g(x, u) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x)b_{n-1-i}(u)} \quad (5.14)$$

Bortset fra konstanten  $\frac{1}{N}$  er den inverse kerne identisk med  $g(x, u)$ .

Hardamard transformationskernen er givet ved:

$$g(x, u) = \frac{1}{N} (-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)} \quad (5.15)$$

hvor summationen i eksponenten foretages i modulo-2 aritmetik. Bortset fra konstanten  $\frac{1}{N}$  er den endimensionale inverse Hardamard transformation identisk med  $g(x, u)$ .



### 5.1.4 Z-transformationen

Z-transformationen er en generalisering af den diskrete fouriertransformation og benyttes intensivt i klassiske metoder til on-line analyse af lyd. Dette skyldes at metoden leder til såkaldte IIR-filtre (Infinite Impulse Response filters). Z-transformationen har, som fouriertransformationen, en lang række nyttige egenskaber og anvendelser. Z-transformationen defineres nedenstående, men beskrives ikke yderligere i disse noter. Interesserede læsere henvises til fx.[25, 26].

Z-transformationen af en sekvens  $x(n)$ ,  $n = 0, 1, 2, \dots, N$ , er defineret ved:

$$X(z) = \sum_{n=0}^N x(n)z^{-n} \quad (5.16)$$

hvor  $z$  er en kompleks variabel. Bemærk at hvis  $z$  erstattes af  $e^{2\pi i u}$ , da reducerer udstrykket til den sædvanlige fouriertransformation af  $x(n)$ . Som et specialtilfælde af ovenstående definition ses ved indsættelse at den z-transformerede af en impulsfunktion  $\delta(n - n_0)$  i  $n = n_0$  er givet ved  $z^{-n_0}$ .

#### Eksempel

Idet vi husker at en diskret funktion  $x(n)$  kan opskrives som en sum af vægtede deltafunktioner  $\sum_{k=0}^N x(k)\delta(n - k)$ , ses at signalet med værdier 1, 2, 3, og 4 for  $n$  lig 0, 1, 2, 3 og ellers med værdier lig 0 vil således have z-transformationen  $X(z) = 1 + 2z^{-1} + 3z^{-2} + 4z^{-3}$ .

## 5.2 Karhuen-Loeve repræsentationen

Karhuen-Loeve transformationen, går også under navnene Hotelling transformationen, Principalkomponent transformationen, PCA (efter engelsk: Principal component analysis). Transformationen adskiller sig fra de tidligere beskrevne transformationerne ved at være baseret på statistiske egenskaber ved det givne forelagte billede  $f(x, y)$ . Basisbillederne for transformationen er således afhængige af  $f$ . Dette, at basisbillederne ikke er underforståede, men bestemt ud fra  $f$  selv, gør metoden overlegen mht. dekorrelation af intensitetsværdierne i forhold til de tidligere beskrevne metoder. Generelt bliver informationen i et billede pakket bedre ned i færre koefficienter. Dette er, kodningsmæssigt dog af tvivlsom værdi, fordi rekonstruktionen ikke kan foretages uden yderligere kendskab til basisbillederne. I analyse-mæssig sammenhæng er transformationen derimod unik, da transformationen beskriver den mest kompakte (lineære) dekomposition af et billede i basisbilleder.

Vi betragter billedelementerne  $f(x, y)$  som stokastiske variable og opskriver disse ved passende

ordning som en  $N^2 \times 1$  søjlevektor:

$$\mathbf{x} = \begin{pmatrix} f(0, 0) \\ f(1, 0) \\ \vdots \\ f(N-1, N-1) \end{pmatrix} \quad (5.17)$$

Kovariansmatricen  $\mathbf{C}_x$  for  $\mathbf{x}$  har størrelse  $N^2 \times N^2$  og defineres ved:

$$\mathbf{C}_x = E\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^t\} = E\{\mathbf{x}\mathbf{x}^t\} - \mathbf{m}_x\mathbf{m}_x^t \quad (5.18)$$

hvor  $t$  angiver transponering,  $E\{\}$  angiver forventet værdi, og:

$$\mathbf{m}_x = E\{\mathbf{x}\} \quad (5.19)$$

Vektoren  $\mathbf{m}_x$  har således længden  $N^2$ . Lad nu  $\mathbf{e}_i$  hhv.  $\lambda_i$  være den  $i$ 'te egenvektor hhv. egenværdi for  $\mathbf{C}_x$  ( $i = 1, 2, \dots, N^2$ ), og lad  $\mathbf{A}$  være en transformationsmatrice, hvis rækker er bestemt ved egenvektorerne  $\mathbf{e}_i$ .

$$\mathbf{A} = \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1N^2} \\ e_{21} & e_{22} & \cdots & e_{2N^2} \\ \vdots & \vdots & \vdots & \vdots \\ e_{N^2 1} & e_{N^2 2} & \cdots & e_{N^2 N^2} \end{pmatrix} = \begin{pmatrix} \vec{\mathbf{e}}_1 \\ \vec{\mathbf{e}}_2 \\ \vdots \\ \vec{\mathbf{e}}_{N^2} \end{pmatrix} \quad (5.20)$$

således at  $e_{ij}$  er den  $j$ 'te komponent af den  $i$ 'te egenvektor. Karhuen-Loeve transformationen er da givet ved:

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x) \quad (5.21)$$

Denne transformation har to interessante egenskaber, nemlig:

$$\mathbf{m}_y = E\{\mathbf{y}\} = E\{\mathbf{A}(\mathbf{x} - \mathbf{m}_x)\} = \mathbf{A}E\{\mathbf{x}\} - \mathbf{A}\mathbf{m}_x = 0 \quad (5.22)$$

og

$$\begin{aligned} \mathbf{C}_y &= E\{(\mathbf{y} - \mathbf{m}_y)(\mathbf{y} - \mathbf{m}_y)'\} \\ &= E\{\mathbf{y}\mathbf{y}'\} \\ &= E\{(\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{m}_x)(\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{m}_x)^t\} \\ &= E\{\mathbf{A}(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^t\mathbf{A}^t\} \\ &= \mathbf{A}\mathbf{C}_x\mathbf{A}^t \\ &= \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{N^2}) \end{aligned} \quad (5.23)$$

Middelværdien  $\mathbf{m}_y$  af transformationskoefficienterne er altså lig nulvektoren. Fra lineær algebra genkender vi udtrykket for  $\mathbf{C}_y$  som diagonaliseringen af  $\mathbf{C}_x$ . Dette betyder at  $\mathbf{C}_y$  har formen:

$$\mathbf{C}_y = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_{N^2} \end{pmatrix} \quad (5.24)$$

At kovariansmatricen for koefficienterne  $\mathbf{y}$  til transformationen er en diagonalmatrice betyder, at koefficienterne er (fuldstændig) dekorreleret. Størrelsen af egenværdien  $\lambda_i$  angiver hvor stor en del af energien i det oprindelige billede, der er pakket i basisbilledet svarende til den  $i$ 'te egenvektor. Sædvanligvis ordnes basisbillederne således at  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{N^2}$ . Det kan yderligere bemærkes, at  $\lambda_i$  er lig variansen af det  $i$ 'te element af  $\mathbf{y}$ .

Da  $\mathbf{C}_x$  er reel og symmetrisk, vil  $\mathbf{A}$  altid være en ortonormal matrice. Heraf følger at  $\mathbf{A}^{-1} = \mathbf{A}^t$ . Den inverse transformation er derfor givet ved:

$$\mathbf{x} = \mathbf{A}^t \mathbf{y} + \mathbf{m}_x \quad (5.25)$$

Bemærk iøvrigt at transformationsmatricen  $\mathbf{A}$  *ikke* er separabel.

Antag at vi ønsker en rekonstruktion  $\hat{\mathbf{x}}$  på basis af de  $k$  første egenvektorer (basisbilleder) af  $\mathbf{C}_x$ . Vi har da.

$$\hat{\mathbf{x}} = \mathbf{A}_k^t \mathbf{y} + \mathbf{m}_x \quad (5.26)$$

Ved udregning af kvadratfejlen er det praktisk at udtrykke denne som sporet ( $Tr$ ) af det ydre produkt af differensen  $(\mathbf{x} - \hat{\mathbf{x}})$  i stedet for, som normalt, det indre produkt. Vi kan opskrive kvadratfejlen som:

$$\begin{aligned} e_k^2 &= \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \\ &= \|(\mathbf{A}^t - \mathbf{A}_k^t)\mathbf{y}\|^2 \\ &= Tr[E\{[(\mathbf{A}^t - \mathbf{A}_k^t)\mathbf{y}][(\mathbf{A}^t - \mathbf{A}_k^t)\mathbf{y}]^t\}] \\ &= Tr[E\{(\mathbf{A}^t - \mathbf{A}_k^t)\mathbf{y}\mathbf{y}^t(\mathbf{A} - \mathbf{A}_k)\}] \\ &= Tr[(\mathbf{A} - \mathbf{A}_k)^t \mathbf{C}_y (\mathbf{A} - \mathbf{A}_k)] \\ &= \sum_{j=k+1}^{N^2} \lambda_j \end{aligned} \quad (5.27)$$

Da egenværdierne  $\lambda_i$  antages sorteret efter aftagende værdi, ses at transformationen optimalt pakker mest mulig energi i koefficienterne svarende til de første basisbilleder.

### Eksempel

Metoden er meget benyttet i analyser af multispektrale billeder optaget fra satellitter eller fra flybårne skannere. Typisk optages der 3-100 billeder svarende til hvert sit spektrale bånd. Da de spektrale bånd overlapper hinanden en del, og da visse materialer reflekterer lys over et bredt spektrum af frekvenser, er informationsindholdet i billederne meget redundant. Det er således muligt at ekstrahere det meste af informationen (og eliminere meget støj) i sættet af billeder ved at foretage en principalkomponent analyse på sættet af de  $N$  billeder. Ved at sætte de mindste af de  $N$  egenværdier til nul, opnås en kompakt repræsentation, hvor meget information er bevaret, og hvor informationsindholdet i basisbillederne er dekorreleret. Ved syntese ud fra det reducerede sæt af basisbilleder kan det oprindelige billedsæt rekonstrueres med meget lille fejl. I mange anvendelser kan videre analyser indskrænkes til kun at betragte de første 2-3 billeder i det transformerede sæt.

### Eksempel

Mange billedanalyser har til formål at analysere og beskrive variationen inden for et sæt af billeder. Et eksempel er analyse af 2D røngtenbilleder eller 3D fMR-billeder af menneskers lever. Efter en forbehandling, hvor billedet af organet segmenteres og normaliseres for translation, rotation og størrelse, ønskes formen på typiske lever beskrevet. Ved PCA-analyse af et stort antal billeder optaget af såvel raske som syge patienter er det ofte muligt at beskrive alle variationer inden for datamaterialet godt ved brug af meget få (3-6) basisbilleder. De tilhørende koefficienter udgør en kodning af billedet og beskriver den aktuelle variation i forhold til et gennemsnitlig lever-billede. Ved analyse af koefficientværdierne for raske hhv. syge patienter vil man ofte se, at disse falder i forskellige grupper. Disse grupper kan derfor modelleres ved en position og størrelse. Givet et nyt forbehandlet leverbillede er det enkelt og hurtigt at projicere dette på de valgte basisbilleder og måle, hvor tæt koefficientvektoren ligger på klassen af raske hhv. syge patienter. I visse tilfælde kan koefficientværdierne ikke blot benyttes til diagnosticering, men kan også indikere arten af en eventuel sygdom.

Analogt til ovenstående eksempel er PCA-metoden blevet hyppigt anvendt ved genkendelse af objekter eller varianter af objekter. Et enkelt velfungerende realtidssystem er genkendelse af positurer af mennesker og efterfølgende syntetisering af en tegneseriefigur, der efteraber positurerne. Årsagen til effektiviteten af sådanne metoder er, at projektion af et billede på få basisbilleder, samt at den videre analyse af den lavdimensionale koefficientvektor, kan foretages hurtigt. Den forudgående analyse, hvor et meget stort sæt af træningsbilleder analyseres for at finde basisbillederne, er derimod ofte meget beregningskrævende. Endelig skal det huskes at et fornuftigt resultat ofte forudsætter en ikke-triviell normalisering af billederne.

I mange praktiske situationer har datamatricen  $\mathbf{x}$  dimensioner  $N \times M$  hvor  $M \ll N$ , svarende til at antallet af stikprøver er meget større end antallet af dimensioner i data. Ved egenverdianalyse af kovariansmatricen  $C_x$  i ligning (5.18) får matricen  $(\mathbf{x}\mathbf{x}^t)$  dimension  $N \times N$ . Da  $N$  ofte er stor er dette et beregningsmæssigt problem. Da der kun kan eksistere  $k \leq M$  egenverdier med værdi  $\neq 0$ , er det oplagt at undersøge  $\mathbf{x}^t\mathbf{x}$  i stedet for  $\mathbf{x}\mathbf{x}^t$ . Bemærk at hvis  $\mathbf{e}$  er egenvektor med egenverdi  $\lambda$  for  $\mathbf{x}\mathbf{x}^t$  da gælder:

$$\mathbf{x}\mathbf{x}^t\mathbf{e} = \lambda\mathbf{e} \quad (5.28)$$

$$(\mathbf{x}^t\mathbf{x})(\mathbf{x}^t\mathbf{e}) = \lambda(\mathbf{x}^t\mathbf{e}) \quad (5.29)$$

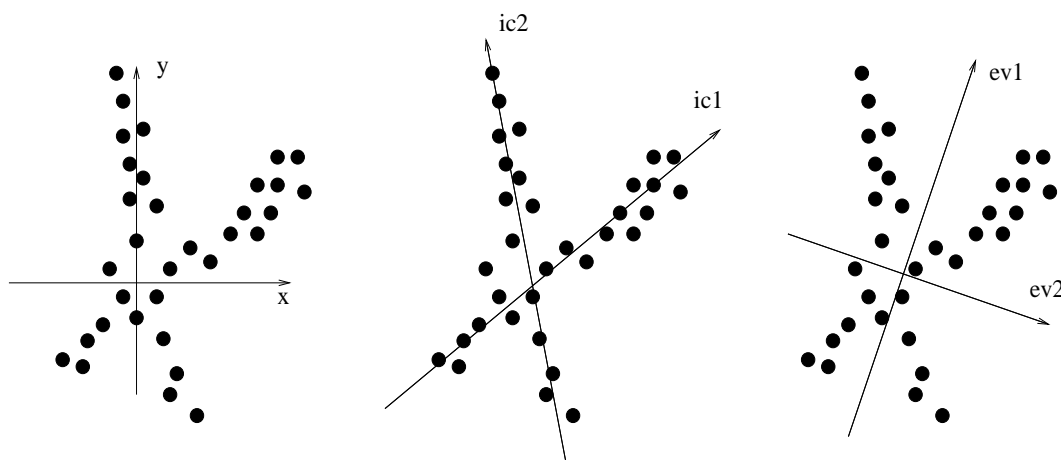
$$(5.30)$$

Det er derfor tilstrækkeligt at finde egenvektorer og egenverdier for den ofte langt mindre matrice  $\mathbf{x}^t\mathbf{x}$ . Dette kan have afgørende indflydelse på analysetiden.

### 5.2.1 ICA-transformationen

PCA-transformationen fra ovenstående afsnit beskriver en dekomposition i ortogonale komponenter. Dette er i visse situationer ikke optimalt. Nedenstående figur illustrerer, hvorledes et 2D-

datasæt med to tydelige komponenter bedre vil kunne beskrives ved en ikke-ortogonal dekomposition i uafhængige komponenter. *Independent Component Analysis* (ICA) er en metode til sådan



Figur 5.1: Punktsæt i koordinatsystemer givet ved observationerne selv (x,y); ved uafhængige komponenter (ic1, ic2) og ved egenvektorer (ev1, ev2).

decomposition. Metoden benytter sig af ikke-lineær analyse og implementationer af denne involverer ofte kompliceret og beregningskrævende optimering. Vi skal derfor ikke gå videre i detaljer med denne metode her.

## 5.3 Wavelet transformationen

Siden wavelet transformationen blev introduceret i 1980'erne har wavelet analyse udviklet sig til et selvstændigt felt inden for billedanalyse. Den matematiske teori bag wavelet transformationen er modnet og er i dag yderst velfunderet. Desværre vil en bare nogenlunde tilfredsstillende gennemgang af denne teori være ganske omfattende, hvorfor dette afsnit kun meget overfladisk introducerer området.

Lad  $f[n]$  være et diskret signal, og antag at antallet af sampels er en potens af 2. Betragt nu signalet som opdelt i tupler bestående af nabosampels og den transformation, hvor hver tupel i signalet erstattes af tuplen bestående af  $1/2$  gange summen hhv. differensen af sampelværdierne.

$$(a, b) \rightarrow \left( \frac{a+b}{2}, \frac{a-b}{2} \right) = (\alpha, \beta)$$

De to operationer er klart ortogonale, og det er oplagt, at det oprindelige signal kan rekonstrueres ud fra det transformerede. Ved omordning kan det transformerede signal opdeles i et  $\alpha$ -signal, og et  $\beta$ -signal, hver af den halve antal af det oprindelige signal. Disse signaler er oplagt af lavpas-hhv. højpas-karakteristik. Vi kan gentage ovenstående operation iterativt på  $\alpha$ -signalet. Processen er illustreret i nedenstående eksempel med 8 sampels:

**Eksempel**

Nedenstående skema angiver i første række det oprindelige signal. Efterfølgende par af rækker angiver for hver af de 3 iterationer i transformationsprocessen de signaler, der fremkommer ved anvendelse af de to simple operationer og den efterfølgende omordning.

0	4	2	2	4	10	6	18	2
1.a	3	1	3	-1	8	-2	10	-8
1.b	3	3	8	10	1	-1	-2	-8
2.a	3	0	9	-1	1	-1	-2	-8
2.b	3	9	0	-1	1	-1	-2	-8
3.a	6	-3	0	-1	1	-1	-2	-8

Det første element i det transformerede signal angiver middelværdien. De efterfølgende værdier beskriver i stadig mindre skala detaljer i signalet. Den sidste halvdel af signalet beskriver meget vellokaliseret information. Dette er en unik egenskab, som ikke deles af nogen af de tidligere beskrevne transformationer.

Ved ignorering af f.eks. den sidste halvdel af det transformerede signal kan det oprindelige signal approksimeres med tab af detaljerigdom.

0	6	-3	0	-1	0	0	0	0
1.a	3	9	0	-1	0	0	0	0
1.b	3	0	9	-1	0	0	0	0
2.a	3	3	8	10	0	0	0	0
2.b	3	0	3	0	8	0	10	0
3.a	3	3	3	3	8	8	10	10

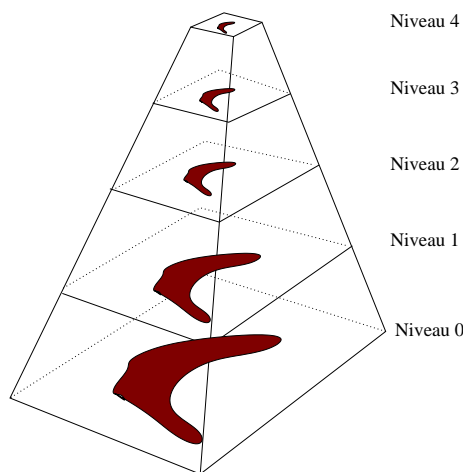
Bemærk at sidste del af signalet er rekonstrueret dårligt fordi numerisk store koefficienter er sat lig 0.

I wavelet-transformationen benyttes præcis samme metodik som i eksemplet ovenfor. Blot er de to filtre, der benyttes (lavpasfilteret  $\frac{1}{2}[1, 1]$  og højpasfilteret  $\frac{1}{2}[1, -1]$ ) udskiftet med andre filtre. Faktisk er ovenstående filtre ikke korrekte waveletfiltre, da konstanten  $\frac{1}{2}$  er forkert. De matematisk ikke-trivielle kriterier, som de to filtre skal opfylde, er ganske komplicerede og ofte vanskelige at udlede. Ovenstående gennemgang beskriver derfor ikke den smukke matematiske teori, der ligger bag waveletfiltrering, men tjener udelukkende til intuitiv illustration.

Bedømt som en multiskala-repræsentation er wavelet repræsentationen både unik fordi mange koefficienter beskrive vellokaliserede begivenheder og begrænsende, fordi den lavfrekvente information altid subsamples med faktoren 2 - skalabegivenheder herimellem kan ikke gøres eksplicitte. Den bedst kendte - og uden tvivl mest succesrige - anvendelse af wavelet transformationen er i kodning af enkeltbilleder, som beskrevet i JPEG2000-standarden, hvor den globale wavelet-transformation har erstattet blokkodning ved brug af Cosinustransformationen.

## 5.4 Pyramide repræsentationer

Den grundlæggende ide i pyramide repræsentationer er, at repræsentere billedet ved et sæt af trinvist “nedfotograferede” billeder. Dette er illustreret i figur 5.2. Repræsentationen omfatter det oprindelige billede selv og medfører derfor et øget lagerforbrug. Fordelen ved repræsentationen er, at den letter “grov-til-fin” analyser af billedet. Således muliggøres effektive globale analyser implementeret udelukkende ved brug af lokale operationer. Pyramide repræsentationer har især været benyttet ved visse segmenteringsmetoder, ved billedkodning, samt ved visse datamatsynsmetoder som f.eks. stereo- og bevægelsesanalyser.



Figur 5.2: Pyramiderepræsentation

Som beskrevet i et tidligere afsnit er *wavelet repræsentationen* en særlig pyramide repræsentation, der udmærker sig ved at beskrive billedet ved en ortogonal dekomposition og ved kun at tillade nedfotograferinger med faktoren 2. Vi skal her se nogle andre muligheder.

De fleste pyramide repræsentationer fremkommer ved sampling af et sæt af lav-pass filtrerede versioner af det oprindelige billede. Ved lav-pass-filtreringen reduceres båndbredden af det filtrerede billede. Det er derfor muligt at repræsentere dette ved færre pixels. Antag at det oprindelige billede har kantlængder  $N$  og  $M$ , og antag at det lav-pass filtrerede samplede billede har kantlængder  $\frac{N}{k} \times \frac{M}{k}$ . Da siges pyramiden bestående af det oprindelige og det samplede billede at have reduktionsfaktor  $k$ . Pyramider med kun 2 niveauer er sjældne. Ofte benyttes mange niveauer. Ofte benyttes reduktionsfaktoren 2. Hvis det mindste billede i en pyramidestruktur har kantlængde 1 tales om en *fuld pyramide*. I modsat fald tales om en *trunkeret pyramide*, eller om en pyramidestub. For en fuld pyramide med grundplan  $N \times M$  og reduktionsfaktoren 2, er det totale pladsforbrug givet ved:

$$N \times M \left[ 1 + \frac{1}{2^2} + \frac{1}{2^4} + \frac{1}{2^8} + \dots \right] < \frac{4}{3} N \times M \quad (5.31)$$

Pyramide repræsentationen er defineret ved reduktionsfaktoren og ved filteret. For gråtonebilleder

er filterstøtten ofte lille og filteret approksimerer en gaussfunktion. Repræsentationen vil derfor approksimere samplede snit i skalarummet for billedet.

### Grov til fin -analyse

En simpel men fundamental anvendelse af pyramidestrukturer er at foretage globale billedoperationer v.h.a. lokale operationer. Dette kaldes *grov-til-fin analyse*. Ideen er at foretage analysen gentagende gange, først på det mindste billede i strukturen, dernæst på det næstmindste osv. indtil analysen til sidst foretages på det oprindelige billede. Billedet på det øverste niveau beskriver det oprindelige billede globalt, og er, grundet lavpass-filtreringen, reelt støjfrit. Til gengæld er lokaliseringen af features i dette billede ringe. Et approksimativt resultat er imidlertid ofte tilstrækkeligt til at dirigere analysen på det billede på det næstøverste niveau i pyramidestrukturen. Da lavpass-filtreringen på dette niveau er mindre udtalt, vil eventuelle positionsangivelser kunne bestemmes med større nøjagtighed. Ved gentagende anvendelse af resultatet fra et niveau til vejledning af analysen på det næste niveau kan analysen til sidst foretages på det oprindelige billede.

En fordel ved grov-til-fin metoden er, at information kan spredes over store afstande selv om der kun anvendes lokale operationer. Et eksempel på dette er en adaptiv grænseværdiafskæring, hvor grænseværdien bestemmes i hver pixel  $(x, y, z)$  dels ud fra lokal information på niveau  $z$  dels ud fra tidligere beregnet information på niveau  $z - 1$ . En anden anvendelse er, at analysen kun behøver at blive foretaget i de områder af billedet, som den grove analyse har markeret som interessante (eng. focus of attention). Dette kan f.eks. udnyttes i metoder til detektion af ændringer i en sekvens af billeder. Hvis noget i scenen ændrer sig markant, vil intensiteten på øverste pyramideniveau ændre sig. Positionen af intensitetsændringen angiver hvor i billederne på lavere niveau, at en nærmere analyse skal foretages. Denne anvendelse er typisk i moderne videokodere. Man skal dog være opmærksom på, at mindre lokale ændringer i billedet på fuld skala ofte vil være filtreret ud på niveau  $z \gg 0$ . Ved blind grov-til-fin analyse vil man derfor risikere at overse sådanne ændringer.

#### 5.4.1 Quad-tree repræsentation

Quad-trees er en komprimeret version af en pyramiderepræsentation med reduktionsfaktor 2, hvor filteret er bestemt af gennemsnittet af de 4 elementer i et støtteområde på  $2 \times 2$ . Hvert billedelement  $(x, y, z)$  i quad-tree-pyramiden kan antage 2 former: Enten som 4 "pointere" til billedelementerne i støtten for elementet, eller en værdi. Sidstnævnte mulighed benyttes hvis de 4 elementer i støtten på niveau  $i - 1$  er ens, eller tilstrækkeligt ens. Et billede med konstant gråtoneværdi vil altså blive repræsenteret ved ét billedelement. Quad-tree repræsentationer af billeder benyttes fortrinsvis ved visse segmenteringsmetoder (se Kapitel 12). Metoden kan også være velegnet ifm. kodning af videosekvenser hvor store dele af billedet er statisk.



### 5.4.2 Gauss-pyramide

I en Gauss-pyramide er frembringerfunktionen en gaussfunktion. Gauss-pyramider er nok de mest benyttede pyramiderepræsentationer. For at undgå aliasing ved reduktionen bør spredningen  $\sigma$  af gaussfilteret vælges passende stor i forhold til reduktionsfaktoren  $k$ , f.eks.  $\sigma \geq \frac{2k}{\pi}$  eller omkring 1.27 for  $k = 2$ . Gausspyramider er attraktive dels fordi de bevarer strukturen i billedet bedre under nedfotograferingen end pyramider med mere simple frembringerfunktioner, dels fordi de er relativt hurtige at generere.

### 5.4.3 Burtpyramide

I [7] er beskrevet en særdeles hurtig og effektiv pyramidekodning, der kan ses som en approksimation til en gauss-pyramide med reduktionsfaktor 2, men som er udledt efter andre kriterier. Vi skal ikke her gennemgå udledningen, men kun referere den formodentlig mest benyttede frembringerfunktion  $g(x, y)$ , der er separabel, symmetrisk, og med en støtte på  $5 \times 5$  pixels.

$$g(x, y) = w_a(x)w_a(y) \quad (5.32)$$

$$w_a(t) = [1/4 - a/2, 1/4, a, 1/4, 1/4 - a/2, 1/4] \quad (5.33)$$

For  $a \approx 0.4$  minder filterformen meget om en gaussfunktion (med spredning 1.0). For  $a = 0.5$  har filtret en trekantsform. Beregningsmæssigt er approksimationen meget effektiv.

### 5.4.4 Laplace-pyramide

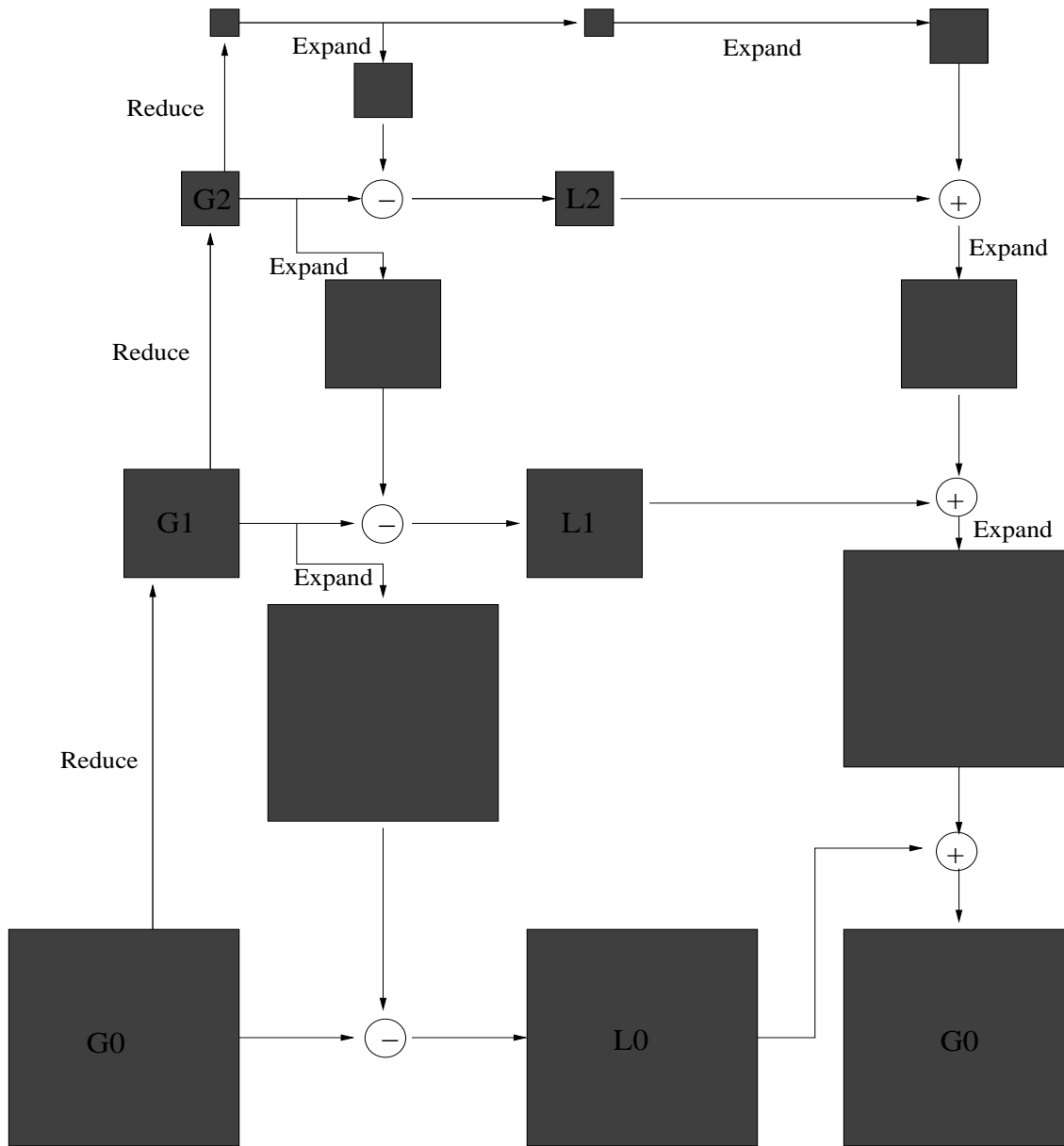
En anden vigtig pyramidetype er laplacepyramiden. Denne kan frembringes ud fra en gausspyramide ved niveauvis anvendelse af laplaceoperatoren  $L = \partial_{xx} + \partial_{yy}$  på billederne. Denne metode benyttes dog sjældent i praksis. I stedet approksimeres laplaceoperatoren (eller mere korrekt laplaceoperatoren på en gaussfunktion) ved differenser af to gaussfunktioner. Lad  $G_{(i)}$  være en gausspyramide, og lad  $\text{Expand}(\cdot)$  være en funktion, der givet et billede på niveau  $z$ , genererer et billede på niveau  $z-1$  ved interpolation. Eksempelvis vil en simpel (lineær) interpolation være givet ved:

$$G(x, y, z - 1) = \begin{cases} G(x/2, y/2, z) & \text{hvis } x \text{ og } y \text{ er lige} \\ \frac{G(\frac{x-1}{2}, y/2, z) + G(\frac{x+1}{2}, y/2, z)}{2} & \text{hvis } x \text{ er ulige og } y \text{ er lige} \\ \frac{G(x/2, \frac{y-1}{2}, z) + G(x/2, \frac{y+1}{2}, z)}{2} & \text{hvis } x \text{ er lige og } y \text{ er ulige} \\ \frac{G(\frac{x-1}{2}, \frac{y-1}{2}, z) + G(\frac{x+1}{2}, \frac{y+1}{2}, z)}{2} & \text{hvis } x \text{ og } y \text{ er ulige} \end{cases} \quad (5.34)$$

Laplacepyramiden  $L_{(i)}$  defineres ved:

$$L_i = G_i - \text{Expand}(G_{i+1}) \quad (5.35)$$

Dette er illustreret i figur 5.3. Laplacepyramider er ofte fulde, sådan at (1-pixels) billedet på største niveau er identisk med det øverste billede i gausspyramiden. Dette betyder at det oprindelige billede  $E$  kan skrives som summen af de ekspanderede laplace-billeder:



Figur 5.3: Gauss- og Laplacepyramider

$$E = L_0 + \text{Expand}(L_1 + \text{Expand}(L_2 + \dots + \text{Expand}(L_s))) \quad (5.36)$$

hvor  $s$  er niveaunummeret for det øverste billede i pyramiden. Bemærk endvidere at laplace-gaussfunktionen er et bånd-pass filter. Dette betyder at Laplacepyramiden er en repræsentation, hvor hvert billede i pyramiden er en repræsentation af et vist frekvensområde i det oprindelige billede. Laplacepyramider har været benyttet ved billedkodning samt ved skala-bestemt detektion features.

# Kapitel 6

## Histogrammer og simple statistiske metoder

Histogrambaserede metoder er formodentlig de mest simple operationer, der kan foretages. Ud fra billedværdierne selv eller ud fra en forbehandling af billedet optælles hvor mange pixels, der antager hver mulig værdi. På basis af histogrammet foretages så den videre analyse. To hyppige anvendelser er histogramlinearisering og grænseværdibestemmelse. I kodning benyttes analyser af histogrammer til indkodning med minimal middelvadratfejl. En generalisering af algoritmen hertil leder til k-means algoritmen, som vi senere skal benytte til segmentering af billeder.

### 6.1 Histogram linearisering

I det følgende beskrives en transformationsmetode hvorved intensiteterne ændres således, at tæthedsfunktionen for det resulterende billede vil være uniform. Metoden kaldes histogram linearisering (eng. histogram equalization). Metoden er ofte effektiv til at fremhæve detaljer, som ellers var usynlige for øjet, eller til at normalisere billeder med henblik på senere sammenligning. Derimod er en simpel histogramlinearisering sjældent anvendelig til at frembringe et billede, der forekommer kønnere for øjet.

I det følgende antages (midlertidigt) at intensitetsværdierne er skaleret til intervallet  $[0 : 1]$ , ved division med den maksimale værdi. For at udnytte resultaterne fra sandsynlighedsregningen formuleres metoden i det kontinuerte tilfælde, dvs. at alle intensitetsværdier  $r \in [0 : 1]$  er mulige. Intensitetsværdien  $r$  i et givet billedelement opfattes som en stokastisk variabel. Vi kan derfor tale om sandsynlighedsfordelingen  $p_r(r)$ . Transformationen kan skrives:

$$s = T(r) \tag{6.1}$$

Filturværdien fremkommer ved multiplikation af  $s$  med den ønskede maksimale værdi (typisk 255). Funktionen  $T$  antages at være monotont voksende, differentiabel og med kontinuerte afledede. Desuden antages at  $T$  afbilder intervallet  $[0 : 1]$  på sig selv. Dette betyder at afbildningen

bevarer ordningen fra “mørke” intensiteter til “lyse”, samt at vi kan tale om den inverse transformation  $T^{-1}$ . I det følgende vil vi få brug for nedenstående sætning fra sandsynlighedsregningen.

### Sætning

Lad  $X$  være en kontinuert stokastisk variabel med tæthed  $f$  og definitionsområde  $[0 : 1]$ . Lad  $\phi$  være en bijektiv funktion af  $[0 : 1]$  på  $[0 : 1]$ . Antag at  $\phi'$  eksisterer og er kontinuert. Lad  $\psi = \phi^{-1}$ . Da er tætheden  $g$  for  $Y = \phi(X)$  givet ved:

$$g(y) = f(\psi(y)) |\psi'(y)| \quad \text{for } y \in [0 : 1] \quad (6.2)$$

Først bemærkes at da  $\psi = \phi^{-1}$ , da vil  $\psi' = \frac{1}{\phi'}$ , samt at (den kummulative) fordelingsfunktion  $F(x)$  for tæthedsfunktionen  $f(x)$  er defineret ved  $F(x) = \int_0^x f(t) dt$ .

Beviset er i to dele: Først antages at  $\phi$  er voksende. Da er:

$$G(y) = P(Y \leq y) = P(\phi(X) \leq y) = P(X \leq \psi(y)) = F(\psi(y)) \quad (6.3)$$

Af kædereglen for differentiation fås:

$$g(y) = G'(y) = F'(\psi(y))\psi'(y) = f(\psi(y)) |\psi'(y)| \quad (6.4)$$

da  $\psi' > 0$  fordi  $\phi' > 0$ . Antag nu at  $\phi$  er aftagende. Da får vi:

$$G(y) = P(Y \leq y) = P(\phi(X) \leq y) = P(X \geq \psi(y)) = 1 - F(\psi(y)) \quad (6.5)$$

Derfor har vi:

$$g(y) = G'(y) = -F'(\psi(y))\psi'(y) = f(\psi(y)) |\psi'(y)| \quad (6.6)$$

Hermed er sætningen bevist.

Givet et billede med tæthedsfunktion  $p_r(r)$  samt transformationen  $s = T(r)$ , er tætheden  $p_s(s)$  for  $s$  (jvf. ovenstående sætning) givet ved:

$$p_s(s) = \left[ p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \quad (6.7)$$

Betrakt nu transformationen givet ved den (kummulative) fordelingsfunktion:

$$s = T(r) = \int_0^r p_r(w) dw \quad \text{for } 0 \leq r \leq 1 \quad (6.8)$$

For denne transformation er  $ds/dr = p_r(r)$ . Ved substitution i (6.7), ses at  $p_s(s) = 1$ . For den angivne transformation vil samtlige intensiteter i det transformerede billede altså være lige sandsynlige. Transformationen vil som regel kunne forbedre billedkvaliteten betydeligt. Dette kan især være tilfældet ved under- eller overbelyste billeder.

**Eksempel**

Lad  $p_r(r) = \frac{\pi}{2} \cos(\frac{\pi}{2}r)$ . For  $s = T(r) = \int_0^r p_r(w)dw = \sin(\frac{\pi}{2}r)$  fås at:  $r = \frac{2}{\pi} \arcsin(s)$ . Vi kan nu indsætte i formel (1.32):

$$\begin{aligned} p_s(s) &= \left[ p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \\ &= \left[ \frac{\pi}{2} \cos\left(\frac{\pi}{2}r\right) \frac{dr}{ds} \right]_{r=\frac{2}{\pi} \arcsin(s)} \\ &= \frac{\pi}{2} \cos(\arcsin(s)) \frac{2}{\pi} \frac{1}{\sqrt{1-s^2}} \\ &= 1 \end{aligned}$$

I praksis er intensiteterne i et digitalt billede ikke kontinuert fordelt. Dette nødvendiggør, efter beregning af de nye værdier, at tilnærme disse til de mulige intensitetsværdier. Herved kan det ske, at antal forskellige intensitetsværdier i det filtrerede billede bliver mindre end antal forskellige intensitetsværdier i det oprindelige billede. Væsentlig er også at bemærke, at alle pixels med samme intensitet før filtreringen naturligvis også vil have samme intensitet efter filtreringen.

**Eksempel**

Antag et billede med 8 mulige intensitetsværdier. Lad hyppighederne af de enkelte værdier være givet som angivet i tabellen:

<i>intensitet</i>	<i>hyppighed</i>	<i>accumuleret</i>	<i>interval</i>
0	0.00	0.00	0.000
1	0.09	0.09	0.125
2	0.01	0.10	0.250
3	0.30	0.40	0.375
4	0.29	0.69	0.500
5	0.10	0.79	0.625
6	0.11	0.90	0.750
7	0.10	1.00	0.875

I tabellen er også angivet den accumulerede sandsynlighed, samt den nedre intervalgrænse for tildelingen af den diskrete intensitetsværdi. Ved histogramlineariseringen ses at pixels med oprindelig intensitet 0, 1, og 2 alle får intensiteten 0 efter filtreringen. Afbildningen bliver:

$$\begin{array}{cccc} 0 \rightarrow 0 & 2 \rightarrow 0 & 4 \rightarrow 5 & 6 \rightarrow 7 \\ 1 \rightarrow 0 & 3 \rightarrow 3 & 5 \rightarrow 6 & 7 \rightarrow 7 \end{array}$$

I det filtrerede billede bliver intensitetsværdierne 1, 2, og 4 altså ikke udnyttet. Desuden vil man kunne se, at histogrammet over de transformerede værdier kun meget approksimativt er ligefordelt. Dette skyldes tillem্পningen af den kontinuerte teori til det diskrete tilfælde.

### 6.1.1 Direkte histogramspecifikation

I visse tilfælde er formålet med filtreringen at fremhæve bestemte udsnit af et billede eller generelt at forbedre kvaliteten, dog således at særlige detaljer fremstår klarere. I sådanne tilfælde er den lineære histogrammodifikation som regel utilstrækkelig. Hvad man ofte ønsker er interaktivt at kunne specificere et ønsket histogram. Dette er som regel muligt med moderne visualiseringsprogrammer men er ikke hensigtsmæssigt, når mange billeder skal filtreres således, at deres intensitetsfordeling er ens. Problemet er her at vælge den resulterende fordeling. Det er overordentlig svært at angive et sådant “ideelt” histogram selv med godt kendskab til det oprindelige billede.

Antag at formålet med filtreringen er at opnå et billede med tæthedsfunktion  $p_t(t)$ , f.eks. en normalfordeling med middelværdi lig 128 og med spredning lig 32. Igen vil vi kun betragte det kontinuerte tilfælde. Lad  $v = G(t)$  være den transformation, der lineariserer fordelingen af  $v$ :

$$v = G(t) = \int_0^t p_t(w)dw \quad (6.9)$$

Forelagt et billede med intensitetsfordeling  $p_r(r)$  kan dette transformeres på samme måde. Lad  $v = T(r)$  være den transformation, der lineariserer fordelingen af  $v$ . Vi kan derfor beskrive transformationen fra  $r$  til  $t$  ved transformationssammensætningen:

$$t = G^{-1}[T(r)] \quad (6.10)$$

I det kontinuerte tilfælde kan det være vanskeligt at finde et analytisk udtryk for  $G^{-1}$ . Dette problem er ikke til stede i det diskrete tilfælde, hvor vi kan beregne og foretage transformationen ved tabelopslag.

### 6.1.2 Lokal histogram linearisering

Lokal (i litteraturen ofte kaldt adaptiv) histogram linearisering er en metode, hvor billedet opdeles i en række områder, og hvor lineariseringen foretages separat for hvert område. Ved en opdeling af billedet i disjunkte områder vil man ofte kunne iagttage en “skakbord-effekt”. Dette er så generende at områderne i praksis altid vælges overlappende. Som regel er områderne kvadratiske og overlapper således, at modifikationen for hvert område kun berører billedelementet i midten af området. Områdernes størrelse er meget betydningsfulde for filtrets virkemåde. Ved valg af små områder vil enhver intensitetsændring blive forstærket. Dette kan virke som en egentlig støjforhøjelse, men vil også synliggøre meget fine/svage strukturer i billedet. Afhængig af formålet med filtreringen, fås den bedste virkning ofte ved valg af områdestørrelsen på mindst et par hundrede pixels.

Det skal bemærkes at beregningen af histogrammet (og transformationen) ikke behøver at ske på ny, hver gang området flyttes een pixel. Her vil jo blot ske det, at en “søjle” forsvinder

ud af området mens en ny kommer til syne. I praksis implementeres metoden ved løbende at modificere histogrammet for intensiteterne i området.

## 6.2 Grænseværdibestemmelse

I mange anvendelser ønsker man at finde en enkelt grænseværdi, der opdeler data. Et eksempel er kantdetektion (se Kapitel 11), hvor punkter med markant stor gradientstørrelse ønskes detekteret. I en sådan binarisering kan det i visse tilfælde antages, at værdierne for de to populationer har en kendt parametrisk form, f.eks. at gradientstørrelsen i ikke-kantpunkterne er normalfordelt og er eksponentialfordelt i kantpunkterne. I andre tilfælde er sådanne antagelser urimelige.

Som simpleste tilfælde gennemgås først grænseværdibestemmelse ved antagelse af to normalfordelte populationer. Dernæst skal vi se en enklere, men ofte mere effektiv, bestemmelse ved detektion af den/de dybeste dal/dale i histogrammet. Senere i kapitlet gennemgås mere generelle metoder.

### 6.2.1 Normalfordelte populationer

Antag at data består af to populationer (her kaldet objekt og baggrund) med  $N \cdot P_1$  hhv.  $N \cdot P_2$  elementer, hvor  $N$  er det totale antal observationer, således at  $P_1 + P_2 = 1$ . Lad  $p_1(x)$  hhv.  $p_2(x)$  være tæthedsfunktioner for de to fordelinger. Den samlede tæthedsfunktion (for hele billedet) vil da være givet ved:

$$p(x) = P_1 p_1(x) + P_2 p_2(x) \quad (6.11)$$

Lad  $\mu_1$  hhv.  $\mu_2$  være middelværdierne for  $p_1(x)$  hhv.  $p_2(x)$  og antag at  $\mu_1 < \mu_2$ .

Givet en grænseværdi  $T$  er sandsynligheden for fejlagtigt at klassificere baggrund som objekt:

$$e_1(T) = \int_T^\infty p_1(x) dx \quad (6.12)$$

Tilsvarende er sandsynligheden for fejlagtigt at klassificere objekt som baggrund, givet  $T$ , bestemt ved:

$$e_2(T) = \int_{-\infty}^T p_2(x) dx \quad (6.13)$$

Den samlede sandsynlighed for fejlklassifikation er derfor:

$$e(T) = P_1 e_1(T) + P_2 e_2(T) \quad (6.14)$$

Vi ønsker at bestemme  $T$  således at  $e(T)$  minimeres.

$$\frac{\partial e(T)}{\partial T} = -P_1 p_1(T) + P_2 p_2(T) = 0 \quad (6.15)$$

Givet funktionerne  $p_1(x)$  og  $p_2(x)$  samt tallene  $P_1$  og  $P_2$ , kan vi (evt. ved brug af numeriske metoder) bestemme den optimale grænseværdi  $T$  ved løsning af ovenstående ligning. Det bør bemærkes, at denne værdi ikke nødvendigvis svarer til positionen af den dybeste dal i  $p(x)$ .

**Eksempel**

Det er ofte rimeligt at antage at såvel  $p_1(x)$  som  $p_2(x)$  er normalfordelinger.

$$p_i(x) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \quad i = 1, 2 \quad (6.16)$$

I dette tilfælde er  $T$  altså løsning til ligningen:

$$\log(P_1\sigma_2) - \frac{(T - \mu_1)^2}{2\sigma_1^2} = \log(P_2\sigma_1) - \frac{(T - \mu_2)^2}{2\sigma_2^2} \quad (6.17)$$

Dette er en andengradsligning i  $T$ .

$$AT^2 + BT + C = 0 \quad (6.18)$$

Ved multiplikation med  $2\sigma_1^2\sigma_2^2$  på begge sider af ligningen fås koefficienterne:

$$\begin{aligned} A &= \sigma_1^2 - \sigma_2^2 \\ B &= 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2) \\ C &= \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \log\left(\frac{P_1\sigma_2}{P_2\sigma_1}\right) \end{aligned} \quad (6.19)$$

Der kan altså være tale om 0, 1 eller 2 reelle rødder. For  $\sigma_1 = \sigma_2 = \sigma$  degenererer ligningen til en førstegradsligning, der har løsningen:

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \log\left(\frac{P_2}{P_1}\right) \quad (6.20)$$

Det overlades til læseren at analysere og fortolke de andre tilfælde.

Som tidligere nævnt er det i visse situationer muligt at estimere tæthedsfunktionerne på forhånd. Kendskabet til hyppighederne  $P_1$  og  $P_2$  er imidlertid også nødvendige for en optimal bestemmelse af  $T$ . I visse situationer kan disse også estimeres på forhånd. Begge estimationer er (formodentlig) unøjagtige. Afhængig af tæthedsfordelingernes form er det muligt, at blot en lille estimationsfejl vil bevirke en stor fejl i beregnede værdi af  $T$ . Ved benyttelse af metoden bør man undersøge en sådan mulighed for ustabilitet.

**6.2.2 Dybeste dal -algoritmen**

En umiddelbart enklere metode til bestemmelse af grænseværdien er at finde bunden af den dybeste dal i histogrammet. Ofte vil histogrammet imidlertid have mange lokale minima. En udglatning af histogrammet vil derfor være nødvendig inden detektion af dalen. Spørgsmålet er dernæst, hvor meget histogrammet skal udglattes. Hvis histogrammet filtreres tilstrækkelig til kun at indeholde netop en dal, vil denne ofte være dårligt lokaliseret. En anvendelig metode er derfor iterativt at filtrere histogrammet og spore positionerne af alle dale over iterationsindeks.



Til slut vil kun én dal overleve, og positionen af denne for den mindste udglatning vil være direkte til rådighed.

I visse tilfælde vides, at billedet indeholder et kendt lille antal  $n$  forholdsvis ensfarvede objekter, som ønskes segmenteret. I sådanne tilfælde kan ovenstående metode oplagt modificeres således, at der detekteres  $n - 1$  grænseværdier. Alternativt til bestemmelsen af de  $n - 1$  grænseværdierne er bestemmelsen af de mellemliggende  $n$  værdier, som bedst repræsenterer fordelingen. Dette kaldes kvantificering.

## 6.3 Kvantificering

Ved kvantificering opdeles værdiområdet i et antal kvanter. Disse kan være lige store eller have en størrelse, der er bestemt ud fra fordelingen af de data som ønskes kvantificeret. Kvantificering benyttes i stort omfang ved kodning, men har langt bredere anvendelsesområde. Vi skal først opstille et udtryk for kvantificeringsfejlen i tilfældet hvor tæthedsfunktionen er kontinuert. Fra dette udtryk udledes en optimal kvantificering, der minimerer denne fejl. I det diskrete tilfælde kan en tilsvarende kvantificering opnås med *k-means algoritmen*, der benyttes intensivt i mange situationer hvor data skal opdeles i *klynger*.

Lad  $x$  være den størrelse som vi ønsker at kvantificere. Vi vil betragte  $x$  som en reel kontinuert stokastisk variabel og kan derfor tale om tæthedsfunktionen  $p(x)$  for  $x$ . Den kvantificerede værdi af  $x$  betegnes  $q$ . Kvantificeringsfejlen er derfor  $|x - q|$ . Generelt betragtes værdimængden af  $x$  som opdelt i et antal *kvanter* (intervaller)  $[z_i, z_{i+1}]$  således at der om den kvantificerede værdi  $q_i$  gælder, at  $z_i < q_i < z_{i+1}$ . Dette er illustreret i figur 6.1. Kvantlængden  $z_{i+1} - z_i$  betegnes  $\Delta_i$ .

Kvadratmiddelfejlen ved kvantificering af  $x \in [z_i, z_{i+1}]$  kan opskrives som:

$$e_i^2 = \int_{z_i}^{z_{i+1}} (x - q_i)^2 p(x) dx \quad (6.21)$$

Den samlede kvadratmiddelfejl ved opdeling af værdiområdet i  $K$  kvanter (svarende til  $K$  værdier  $q_i$ ) er da:

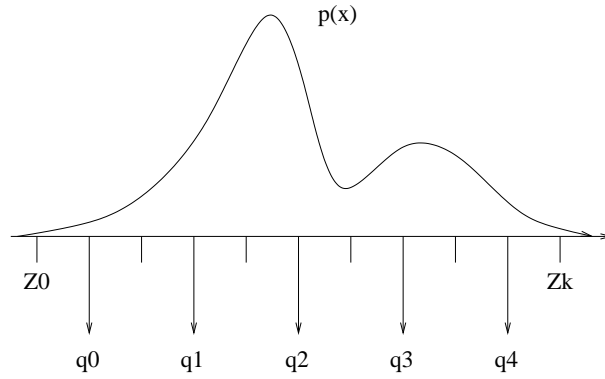
$$e_{ms}^2 = \sum_{i=0}^{K-1} \int_{z_i}^{z_{i+1}} (x - q_i)^2 p(x) dx \quad (6.22)$$

hvor  $z_0$  hhv.  $z_K$  muligvis kan være  $\pm\infty$  afhængig af værdimængden for  $x$ .

### Eksempel

Antag at  $x$  er ligefordelt i intervallet  $[0,1]$ , og antag at  $q_i = \frac{z_{i+1} + z_i}{2}$ . Da er:

$$\begin{aligned} e_{ms}^2 &= \sum_{i=0}^{K-1} \int_{z_i}^{z_{i+1}} (x - q_i)^2 p(x) dx \\ &= K \int_0^{1/K} (x - \frac{1}{2K})^2 dx \end{aligned}$$



Figur 6.1: Kvantificering af en stokastisk variabel med fordeling  $p(x)$ , og med værdiområde  $[Z_0: Z_k]$  i repræsentanter  $q_0, q_1, \dots, q_{k-1}$

.

$$= \frac{1}{12K^2}$$

Heraf ses er rms-fejlen er  $\frac{1}{\sqrt{12}K}$ .

### 6.3.1 Optimal kvantificering

Antag at fordelingsfunktionen  $p(x)$  er koncentreret omkring en bestemt værdi  $x_0$ , og at  $p(x)$  aftager hurtigt når  $|x - x_0|$  vokser. Ideen er da at øge kvantestørrelsen  $\Delta_i$  med afstanden fra  $x_0$ . Herved mindskes kvantificeringsfejlen for de fleste værdier. Vi søger netop den kvantificering, dvs. følgerne  $z_i$  og  $q_i$ , der minimerer  $e_{ms}^2$  fra ligning (6.22). Dette gøres på sædvanlig måde:

$$\frac{\partial}{\partial z_i} e_{ms}^2 = (z_i - q_{i-1})^2 p(z_i) - (z_i - q_i)^2 p(z_i) = 0 \quad i = 2, 3, \dots, K \quad (6.23)$$

$$\frac{\partial}{\partial q_i} e_{ms}^2 = -2 \int_{z_i}^{z_{i+1}} (x - q_i) p(x) dx = 0 \quad i = 1, 2, \dots, K \quad (6.24)$$

For  $p(z_i) \neq 0$  får vi:

$$z_i = \frac{q_{i-1} + q_i}{2} \quad (6.25)$$

$$q_i = \frac{\int_{z_i}^{z_{i+1}} x p(x) dx}{\int_{z_i}^{z_{i+1}} p(x) dx} \quad (6.26)$$

Resultatet er altså, at intervalgrænserne skal ligge midt imellem kvantificeringsværdierne  $q_i$ , og at kendskab til  $p(x)$  er nødvendig for at beregne disse. For  $K$  kvanter har vi således  $2K$  ligninger. Disse er imidlertid ikke lineære. Kun for særligt simple tæthedsfunktioner  $p(x)$  kan vi angive analytiske udtryk for kvantificeringsværdierne. Generelt er det nødvendigt at benytte numeriske metoder, der benytter sig af sekvenser af gradvist bedre og bedre gæt på værdierne. En sådan

algoritme - kaldet *K-means algoritmen* eller *Max-Lloyd algoritmen* er beskrevet i næste afsnit. Bemærk at i det særlige tilfælde, hvor  $p(x)$  er ligefordelt, vil  $q_i = \frac{z_i + z_{i+1}}{2}$ . Dette er ikke overraskende.

## 6.4 K-means algoritmen

Klyngeanalyse omhandler metoder til opdeling af data i klynger. Hidtil har vi kun diskuteret metoder til opdeling af en-dimensionale data. De fleste metoder til klyngeanalyse fungerer for multidimensionale data. Der eksisterer et hav af sådanne metoder. En af de simpleste, og bedst fungerende er K-means algoritmen også kaldt Max-Lloyd algoritmen. Vi skal først gennemgå denne. I et senere afsnit gennemgås en videreudvikling, EM-algoritmen, der som K-means algoritmen, har fået stor anvendelse inden for mange grene af billedanalyse.

I *K-means* algoritmen antages, at antallet  $K$  af klasser er kendt. Antag at der er givet  $n$  sampels af en featurevektor  $f$ . Vi generaliserer således situationen fra sidste afsnit til en vilkårlig dimension. Vi ønsker at bestemme de  $K$  klyngecentre  $\mathbf{C}_i$ , samt en navngivning af hver af de  $n$  sampels efter hvilken klynge de tilhører. Efterfølgende kan grænsen mellem to naboklynger bestemmes ved hyperfladen vinkelret på forbindelseslinien mellem centrene gennem midtpunktet af denne.

Før anvendelsen af K-means algoritmen er det nødvendigt at vælge  $K$  initielle klyngecentre. Disse trækkes ofte tilfældigt, men kan naturligvis bestemmes ved brug af andre metoder. K-means algoritmen består af en iterativ gentagelse af to beregninger. I C-trinet klassificeres alle  $n$  sampels efter den klynge, hvis center ligger nærmest. I E-trinet estimeres nye klyngecentre som massemidtpunktet for de sampels, som er klassificeret til klyngen. Derefter itereres navngivningen og genberegningen af klyngecentre, sålænge blot ét klyngecenter flytter sig.

**C-trin** Navngiv hver observation  $f$  med klyngenavnet  $j$ ,  
hvis der for alle  $i \neq j, 1 \leq i \leq K$  gælder at:

$$\|f - \mathbf{C}_j\| < \|f - \mathbf{C}_i\|$$

hvor  $\mathbf{C}_j$  er centeret for klynge  $j$ .

**E-trin** Beregn nye klyngecentre ved:

$$\mathbf{C}_i = \frac{1}{n_i} \sum_{f \in \text{klynge } i} f$$

hvor  $n_i$  er antallet af observationer navngivet med klynge  $i$ .

Det kan vises at K-means algoritmen konvergerer. I praksis er konvergensens forholdsvis hurtig. Derimod er der ingen garanti for, at algoritmen konvergerer mod det ønskede. Hvis data faktisk grupperer sig i  $K$  nogenlunde adskilte grupper, er dette dog normalt tilfældet. Det er

vigtigt at  $K$  er valgt korrekt, da det resulterende sæt af klyngecentre erfaringsmæssigt kan variere kraftigt som funktion af  $K$ . Hvis  $K$  kendes eksakt på forhånd, men klyngerne er dårligt separeret, kan det være en fordel at afvikle algoritmen et antal gange med forskellig valg af initiale klyngecentre. Typisk vil netop et af de resulterende klyngecentre (det korrekte) så fremkomme mange gange, hvorimod resten vil ligge sporadisk.

Ved anvendelse af K-means algoritmen på multidimensionale data skal man være opmærksom på valget af afstandsmål. Ved hovedløs anvendelse af det euklidiske afstandsmål vil resultatet sjældent svare til det forventede. Dette skyldes at visse feature- koordinatværdier kan variere mere end andre. Her er kuren at normalisere data inden anvendelse af algoritmen. Dette gøres ved for hver koordinat at subtrahere middelværdien af koordinaten og dividere med spredningen af koordinatværdien.

K-mean algoritmen kan modificeres på en lang række punkter således at nye klyngecentre kan introduceres, således at klynger med meget få “medlemmer” kan nedlægges etc. Sådanne algoritmer er ofte meget følsomme over for et større antal brugerspecificerede parametre. Der findes også metoder til automatisk bestemmelse af antallet  $K$  af klynger. K-means algoritmen kan også ses som et specialtilfælde af en mere generel algoritme kaldet *EM-algoritmen* (Expectation-Maximation algoritmen), der beskrives senere.

## 6.5 Mixture models

Der findes et overvældende antal metoder, der benytter antagelser og analyser af fordelinger. Et problem ved mange ældre metoder er at de grundliggende antagelser om simple fordelinger, f.eks. at støjen er normalfordelt, ikke er opfyldt i praksis. Det kan endvidere være vanskeligt at angive nogen parametrisk fordeling, der tilfredsstillende kan modellere det ønskede. En vej ud af dette problem er at beskrive den statistiske opførel af data ved en sum (mixture) af fordelinger. Nedenstående skitseres nogle situationer, hvor sådanne modeller er anvendelige. Senere i kapitel 12 om segmentering skal vi vende tilbage til problemstillingen.

Intensitetsforskellen mellem nabopixels er ofte lille og modelleres ofte godt ved en normalfordeling med middelværdi nul. Undtagelsen er i pixels omkring kanterne i et billede. Her kan forskellen bedre modelleres ved en eksponentialfordeling eller eventuelt en ligefordeling. Den samlede fordeling kan beskrives ved en vægtet sum af de to fordelinger. Ved analyser af billedsekvenser kan disse ofte opdeles i en baggrund, der er næsten konstant, og en forgrund, hvor der sker store ændringer. Betragtes intensiteten  $I(t)$  i en enkelt baggrundspixel, kan denne modelleres godt ved en normalfordeling med langsomt varierende middelværdi og spredning. For en forgrundspixel er en ligefordeling en bedre model, fordi der her kan ske store svært forudsigelige ændringer.

Ved segmentering af farvebilleder opdeles disse i områder med omtrentlig ens farve. Hvert segment kan ofte modelleres godt ved en middelværdivektor og kovariansmatrice for farvevektoren for segmentet. Samlet set vil billedet da have en fordeling, der kan beskrives ved et mixture af de enkelte normalfordelinger.

Ved tracking ønsker man at estimere en tilstandsvektor bestående af bl.a. position og bevægelsesvektor, begge som funktion af tiden, for de objekter, der er synlige i billedet. For et enkelt objekt er det oplagt at modellere tilstandsvektoren ved en flerdimensional normalfordeling, hvis parametre ændrer sig lineært over tid. Dette er grundlaget for en række klassiske metoder (Kalman-filtrering). Sådanne metoder har vist sig at fungere mindre godt i praksis, fordi data sjældent lader sig modellere så simpelt. Nyere metoder benytter modellering ved et mixtur af fordelinger. Ofte undgås at estimere parameterene i disse fordelinger. I stedet modelleres disse ved sampling. Sådanne metoder, der går under navnet partikel-filtrering, har vist sig at være væsentlig mere robuste.



# Kapitel 7

## Geometrisk opretning af billeder

I visse tilfælde har den forvrængning, som billedet under optagelsen har været udsat for, bevirket at billedkoordinatsystemet ikke svarer til det, som vi ønsker at betragte eller analysere billedet i. I tilfælde hvor et *referencebillede* eksisterer på forhånd, er den geometriske opretning eller *registrering* relativt enkel. Denne situation opstår eksempelvis ved analyser af fly- og satellitbilleder, hvor man ofte ønsker at sammenholde et nyt billede med et ældre. Opgaven er at transformere det nye billede, således at koordinatværdierne for en række udvalgte billedpunkter bliver identiske med koordinatværdierne for de samme billedpunkter i referencebilledet. Anderledes sagt ønsker man at bestemme den koordinattransformation, der afbilder det nye billede over i referencebilledet. Det er sædvanligt for hver koordinat at benytte polynomier af lav grad som transformationer. Afhængig af billederne og den måde hvorpå de er optaget, kan transformationerne dog være vilkårligt komplicerede.

Lad  $(x_0, y_0)$  være koordinater i det observerede billede  $I_0$  og lad  $(x, y)$  være koordinater i referencebilledet  $I$ , således at  $I(x, y)$  og  $I_0(x_0, y_0)$  afbilder samme punkt (område) i scenen. En 2. grads polynomiell koordinattransformation kan da beskrives ved:

$$\begin{aligned}x_0 &= a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 \\y_0 &= b_0 + b_1x + b_2y + b_3x^2 + b_4xy + b_5y^2\end{aligned}\tag{7.1}$$

En 1. grads transformation vil tillade en vilkårlig affin afbildning omfattende translation, rotation, skalering og skævvridning (således er det nye koordinatsystem ikke nødvendigvis retvinklet). Ovennævnte koordinattransformation kræver bestemmelse af 12 parametre. Hvis en 1. grads transformation kan beskrive forvrængningen, behøves kun 6 parametre. Det er sjældent nødvendigt at benytte transformationer af højere grad. I praksis estimeres parametrene ved at bestemme et stort antal (20 - 50 eller flere) "samhørende" koordinatværdier  $(x, y)$  og  $(x_0, y_0)$  i de to billeder. Disse punktpar kaldes i fotogrammetrien for *paspunkter* eller landmærker (eng. landmarks), og er ofte bestemt manuelt eller semiautomatisk vha. en interessepunktsdetektor. Sådanne detektioner diskuteres videre i kapitel 11.

Det system af lineære ligninger, der fremkommer af (7.1) kan normalt ikke løses. Der findes ingen sæt af parametre  $a_i$  og  $b_i$ , der opfylder alle ligninger. Vi vil derfor søge en approksima-

tiv løsning ved *mindste kvadraters metode*. Denne teknik er fundament for et meget stort antal metoder i billedbehandling. Metoden resulterer i en approksimation, der minimerer summen over alle ligninger af kvadratfejlen på hver ligning. Hvis støjen er normalfordelt kan det vises at dette er et såkaldt Maksimum-Likelihood estimat. Metoden gennemgås kort nedenstående. Yderligere beskrivelser kan findes i [28]. En tilbundsående gennemgang kan bl.a. findes i [23].

Antag at der benyttes  $N$  paspunkter. Dette giver  $2N$  ligninger. Lad  $\mathbf{C}_x$  hhv.  $\mathbf{C}_y$  være søjlevektorer bestående af de  $N$  x-koordinatværdier hhv. de  $N$  y-koordinatværdier i  $I_0$ -billedet. Lad  $\mathbf{A}$  hhv.  $\mathbf{B}$  være søjlevektorer af de 12 ukendte koefficienter. Lad endvidere  $\mathbf{X}$  være en  $N \times 6$ -matrice bestående af koordinatværdierne i  $I$ -billedet.

$$\mathbf{C}_x = \begin{pmatrix} x_0(1) \\ x_0(2) \\ \vdots \\ x_0(N) \end{pmatrix} \quad \mathbf{C}_y = \begin{pmatrix} y_0(1) \\ y_0(2) \\ \vdots \\ y_0(N) \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_5 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_5 \end{pmatrix} \quad (7.2)$$

$$\mathbf{X} = \begin{pmatrix} 1 & x(1) & y(1) & x^2(1) & x(1)y(1) & y^2(1) \\ 1 & x(2) & y(2) & x^2(2) & x(2)y(2) & y^2(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x(N) & y(N) & x^2(N) & x(N)y(N) & y^2(N) \end{pmatrix} \quad (7.3)$$

Vi kan nu skrive ligningerne (7.1) som:

$$\mathbf{C}_x = \mathbf{X}\mathbf{A} \quad \text{og} \quad \mathbf{C}_y = \mathbf{X}\mathbf{B} \quad (7.4)$$

Parametrene  $\mathbf{A}$  og  $\mathbf{B}$  kan altså findes separat. I de følgende vil vi finde  $\mathbf{A}$  ved anvendelse af *mindste kvadraters metode*. Bemærk at  $\mathbf{C}_x$  omdøbes til  $\mathbf{C}$ .

For givet valg af  $\mathbf{A}$  kan vi udregne kvadratfejlen:

$$e^2 = (\mathbf{C} - \mathbf{X}\mathbf{A})^t(\mathbf{C} - \mathbf{X}\mathbf{A}) = \|\mathbf{C} - \mathbf{X}\mathbf{A}\|^2 \quad (7.5)$$

hvor  $t$  angiver transponering. Vi ønsker at vælge  $\mathbf{A}$  således at  $e^2$  minimeres. Dette gøres på sædvanlig måde, ved at differentiere, og sætte resultatet lig nul.

$$\frac{\partial e^2}{\partial \mathbf{A}} = -2\mathbf{X}^t(\mathbf{C} - \mathbf{X}\mathbf{A}) = \vec{0} \quad (7.6)$$

Videre fås:

$$\mathbf{X}^t\mathbf{C} = \mathbf{X}^t\mathbf{X}\mathbf{A} \quad (7.7)$$

og dermed

$$\mathbf{A} = \mathbf{X}^\dagger\mathbf{C} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{C} \quad (7.8)$$



hvor matricen  $\mathbf{X}^\#$  kalden den *pseudoinverse* til matricen  $\mathbf{X}$ . Bemærk at den pseudoinverse kun indeholder talværdier (nemlig, de målte koordinatværdier for paspunkterne). Estimationen kræver således invertering af to  $6 \times 6$  matricer. Vi skal ikke her diskutere de numeriske problemer, som der kan være forbundet hermed.

Det skal bemærkes at der findes mange numeriske metoder til løsning af overbestemte ligningssystemer som (7.1). Disse behøver ikke eksplicit at konstruere den pseudoinverse matrix  $\mathbf{X}^\#$ . I [28] findes en ultrakort gennemgang af nogle få løsningsmetoder.

Selve transformationen foregår ved for hvert punkt  $(x, y)$  i referencekoordinatsystemet at bestemme den tilsvarende koordinat  $(x_0, y_0)$  i koordinatsystemet for det observerede billede  $I_0$  og derefter at “kopiere” intensitetsværdien:

$$I^{new}(x, y) := I_0(x_0(x, y), y_0(x, y)) \quad (7.9)$$

Herved opstår der det problem, at de estimerede koordinater  $x_0$  og  $y_0$  vil være reelle størrelse. Da intensiteten  $I_0$  kun kendes i billedpunkterne, dvs. for heltallige koordinatværdier, er det nødvendigt at foretage en interpolation af intensitetsfunktionen.

Interpolation af værdier ud fra diskrete stikprøver kan foretages på mange måder. Her skal nævnes tre metoder: *Nærmeste nabo interpolation*, *Bilineær interpolation* og *Kubisk interpolation*. Nærmeste nabo metoden er den simpleste og giver ofte et dårligt resultat. Kubisk interpolation er mest kompliceret og giver kun sjældent bedre resultat end ved bilinear interpolation. Væsentlig i denne sammenhæng er, at interpolationspunkter aldrig ligger længere fra et målepunkt end  $1/2$  pixel, samt at vi ikke her ønsker at indbygge nogen form for støjfiltrering. Dette forhold gør mere avancerede interpolationsmetoder mindre relevante.

**Nærmeste nabo interpolation** Bestem  $I_0(x_0, y_0)$ , hvor  $x_0$  og  $y_0$  er relle ved  $I_0(x_1, y_1)$ , således at  $x_1$  og  $y_1$  er heltallige, og således at  $|x_0 - x_1| \leq 1/2$  og  $|y_0 - y_1| \leq 1/2$ .

**Bilineær interpolation** Bestem  $I_0(x_0, y_0)$  ud fra de fire omliggende naboer ved:

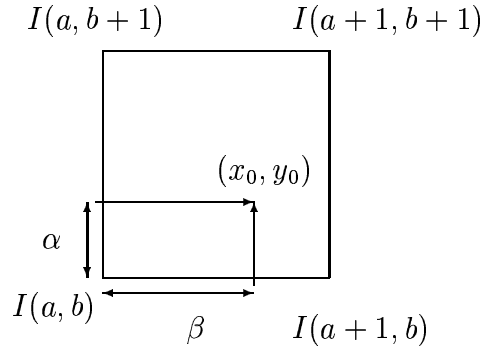
$$I_0(x_0, y_0) = (1 - \alpha)(1 - \beta)I_0(a, b) + (1 - \alpha)\beta I_0(a, b + 1) + \alpha(1 - \beta)I_0(a + 1, b) + \alpha\beta I_0(a + 1, b + 1) \quad (7.10)$$

hvor  $a = \lfloor x_0 \rfloor$ ,  $b = \lfloor y_0 \rfloor$ , og  $\alpha = x_0 - a$  samt  $\beta = y_0 - b$ . Dette er illustreret i figur 7.1.

**Kubisk interpolation** Ved kubisk interpolation bestemmes værdien ud fra 16 omgivende stikprøveværdier. Interpolationsformlen gives her i det en-dimensionelle tilfælde. Lad  $a$  og  $\alpha$  være givet som ved den bilineære interpolation. Den interpolerede værdi kan da bestemmes ved:

$$I_0(x_0, y_0) = \alpha(-1 + 2\alpha - \alpha^2)I_0(a - 1) + (1 - 2\alpha^2 + \alpha^3)I_0(a) + \alpha(1 + \alpha - \alpha^2)I_0(a + 1) + \alpha^2(-1 + \alpha)I_0(a + 2) \quad (7.11)$$

Det overlades til læseren at verificere formelen samt at udvide denne til det 2-dimensionelle tilfælde.



Figur 7.1: Skema til illustration af bilinear interpolation

**Eksempel** Billeder optaget med en billig optik er ofte geometrisk forvrænget. Den væsentligste forvrængning er radial, således at punkter langt fra billedkoordinaterne  $(x_c, y_c)$  af det optiske center afbildes tættere på dette (barrel distortion) eller længere væk fra dette (pincushion distortion) end for en ideel optik. Punktet  $(x_c, y_c)$  kaldes det principale punkt og kan estimeres ved en såkaldt kamerakalibrering. Ofte vil  $(x_c, y_c)$  ligge tæt på midten af billedet. Lad  $(x, y)$  være den sande position af et punkt observeret i koordinaterne  $(u, v)$ , og lad  $R = \sqrt{(u - c_x)^2 + (v - c_y)^2}$ . Da er en god model af forvrængningen givet ved:

$$\begin{aligned} x &= c_x + (u - c_x) \cdot (1 + k_2 R^2 + k_4 R^4) \\ y &= c_y + (v - c_y) \cdot (1 + k_2 R^2 + k_4 R^4) \end{aligned}$$

hvor  $k_2$  og  $k_4$  er konstanter, der som  $(x_c, y_c)$  kan bestemmes i en kamerakalibrering. I praksis kan man som regel antage at  $k_4 = 0$ . Ved udregning ses let at ligningerne er polynomielle og derfor kan benyttes til opretning helt analogt til det tidligere beskrevne.

## 7.1 Warping

Generelt kunne man ønske at warpe et billede på en måde, som ikke enkelt kan beskrives ved polynomier af lav grad. En ikke-parametrisk warping kan beskrives ved koordinatfunktioner  $u(x, y)$  og  $v(x, y)$ , sådan at  $(u, v)$ , givet punkt  $(x, y)$  i det warped billede, returnerer koordinaterne i referencebilledet:

$$(x_0, y_0) = (u(x, y), v(x, y))$$

$(u, v)$  er altså en vektorfunktion, der beskriver 'koordinatflowet' mellem referencebilledet og det warped billede. En måde at bestemme  $(u, v)$  på er ved interpolation ud fra et (stort) antal etablerede paspunkter. Vi ønsker at bestemme  $(u, v)$  sådan, at denne passer godt med paspunkterne og iøvrigt er pæn. Som vi skal se i kapitel 9 om regularisering, kan kravet om pænhed formuleres som et integral over de partialle afledede af  $u$  og  $v$ . Det kan bemærkes, at et sådant

krav ikke nødvendigvis leder til en bijektiv afbildning mellem billederne. Nedenstående skal vi opstille et af mange mulige udtryk for kravet om pænhed/glathed af  $u$  og  $v$  samt et udtryk for overensstemmelse med paspunkterne. Ved sammensætning fås et samlet godhedsmål for  $(u, v)$ . Minimering af dette vil lede til implicite ligninger, der definerer warpet.

Lad  $\Omega$  være mængden af paspunkter  $((x_0, y_0), (x, y))$  og lad  $\delta(x, y)$  være 1, hvis  $(x, y) \in \Omega$  og ellers 0. Lad  $x_0(x, y)$  hhv.  $y_0(x, y)$  være udefineret hvis  $(x, y) \notin \Omega$ . Vi ønsker da at:

$$\int \delta(x, y) \cdot [(x_0(x, y) - u(x, y))^2 + (y_0(x, y) - v(x, y))^2] dx dy$$

er lille. Årsagen til, at udtrykket opskrives som et integral (og til brugen af indikatorfunktionen  $\delta$ ) og ikke som en sum, er at vi ønsker at kombinere udtrykket med et glathedsmål for  $u$  og  $v$ . Et ofte brugt glathedsmål fås ved at integrere summen af de kvadrerede partielle afledede. I det aktuelle tilfælde fås:

$$\int [u_x^2(x, y) + u_y^2(x, y) + v_x^2(x, y) + v_y^2(x, y)] dx dy$$

Bemærk at begge ovenstående integral-størrelser er reelle positive tal. Jo mindre disse tal er for givet valg af  $u(x, y)$  og  $v(x, y)$ , jo bedre overensstemmelse er der i paspunkterne og jo glattere er 'flowfeltet'. Vi ønsker at bestemme  $(u, v)$ , sådan at begge ovenstående mål er små. En måde er at opstille en samlet funktional, hvor sidste led indgår med vægten  $\lambda$  i forhold til første led.  $\lambda$  angiver altså et tal for mistroen til paspunktsmålingerne. Jo mindre  $\lambda$ , jo mindre glat vil de optimale  $(u, v)$  være og jo bedre overensstemmelse vil der være med paspunkterne. Idet vi undlader angivelse af koordinater fås en sammensat funktional:

$$\Theta(u, v) = \int \delta[(x_0 - u)^2 + (y_0 - v)^2] + \lambda[u_x^2 + u_y^2 + v_x^2 + v_y^2] dx dy \quad (7.12)$$

De funktioner  $(u, v)$ , der minimerer  $\Theta$ , kan ikke findes ved sædvanlig differentialregning, da minimeringen foregår over et rum af funktioner. Ved brug af variationsregning er det muligt at optille to differentiallyigninger, hvis løsning er nødvendige krav, hvis  $(u, v)$  skal minimere ovenstående udtryk.

$$\begin{aligned} \delta(x_0 - u) + \lambda(u_{xx} + u_{yy}) &= 0 \\ \delta(y_0 - v) + \lambda(v_{xx} + v_{yy}) &= 0 \end{aligned} \quad (7.13)$$

Ved hjælp af variationsregning har vi transformeret et globalt krav om minimering til et sæt af koblede partielle differentiallyigninger, hvor vi for hver billedpunkt  $(x, y)$  har 2 ligninger. Man kan generelt ikke løse disse ligninger og opnå analytiske udtryk for  $u(x, y)$  og  $v(x, y)$ . Derimod leder ligningerne til en iterativ algoritme, der vil konvergere mod de optimale funktioner.

Bemærk først at  $u_{xx} + u_{yy} = \nabla^2 u$  (og tilsvarende for  $v$ ). En ofte benyttet diskret approksimation til  $\nabla^2 u_{x,y}$  i punktet  $(x, y)$  er  $4(\bar{u}_{x,y} - u_{x,y})$ , hvor  $\bar{u}_{x,y}$  er gennemsnittet af  $u$  i de fire nabopixels til  $(x, y)$ . Idet vi udelader punktindex  $(x, y)$ , kan vi derfor for hver  $(x, y)$  approksimere ovenstående ligninger med de diskrete versioner:

$$\begin{aligned} u \cdot (\delta + 4\lambda) &= \delta x_0 + 4\lambda \bar{u} \\ v \cdot (\delta + 4\lambda) &= \delta y_0 + 4\lambda \bar{v} \end{aligned} \quad (7.14)$$

Antag at vi har givet initielle estimater ( $t = 0$ ) af funktionerne  $u(x, y)$  og  $v(x, y)$ , f.eks. begge nulfunktioner. Da kan vi forbedre disse iterativt ved:

$$\begin{aligned} u^{t+1} &= \frac{\delta x_0 + 4\lambda \bar{u}^t}{\delta + 4\lambda} \\ v^{t+1} &= \frac{\delta y_0 + 4\lambda \bar{v}^t}{\delta + 4\lambda} \end{aligned} \quad (7.15)$$

Ved gentagende anvendelse vil vi for alle  $(x, y)$  finde værdier  $(u, v)$ , der stemmer godt overens med de fastlagte paspunkter, og som iøvrigt opfører sig pænt. For førstekoordinaten vil der efter konvergens gælde at:

$$u(x, y) = \begin{cases} \bar{u}(x, y) & \text{hvis } \delta(x, y) = 0 \\ \frac{x_0(x, y) + 4\lambda \bar{u}(x, y)}{1 + 4\lambda} & \text{hvis } \delta(x, y) = 1 \end{cases}$$

## 7.2 Geometriske projektioner

Mange billeder kan beskrives ved projektioner af et intensitetsmønster beliggende på en tre-dimensional flade på det to-dimensionale billedplan. Den mest benyttede projektion er den *perspektiviske projektion*. Simplifikationer af denne leder til den *affine projektion*.

Under antagelse om perspektiviske projektioner kan det vises, at to 2D-projektioner af samme 3D-punkter er relateret ved en såkaldt *homeografi*. Nedenstående gennemgås disse projektioner kort med vægt på opretning ift. en homeografi. For en tilbundsående beskrivelse af geometriske projektioner og deres anvendelser henvises til [17].

Lad  $\mathbf{X} = (X, Y, Z)^t$  være et punkt i et sædvanligt retvinklet 3-dimensionalt koordinatsystem  $\mathbf{O}_0$ , og lad  $\mathbf{O}$  være et andet lignende koordinatsystem fremkommet ved translation og rotation af  $\mathbf{O}_0$ . En sådan bevægelse kaldes *stiv*. For en ren translation kan punktet  $\mathbf{X}$  i det ny koordinatsystem  $\mathbf{O}$  beskrives ved  $\mathbf{X} + \mathbf{T}$ , hvor translationsvektoren er  $\mathbf{T} = (T_x, T_y, T_z)^t$ . For en ren rotation vil  $\mathbf{X}$  få koordinaterne  $\mathbf{R}\mathbf{X}$  hvor  $\mathbf{R}$  er en  $3 \times 3$  ortonormal matrice:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}$$

således at

$$r_i \cdot r_j = \begin{cases} 1 & \text{hvis } i = j \\ 0 & \text{hvis } i \neq j \end{cases}$$

Mere generelt kan kravet om ortonormalitet af  $\mathbf{R}$  fraviges, således at skalering og skævvridning kan modelleres. I det følgende antages at  $\mathbf{R}$  er vilkårlig.

Lad  $\mathbf{W} = [\mathbf{R}|\mathbf{T}]$  være den sammensatte  $3 \times 4$  matrix, der fremkommer ved til  $\mathbf{R}$  at tilføje søjlevektoren  $\mathbf{T}$ . Lad endvidere  $\mathbf{X}$  være udtrykt i *homogene koordinater*, dvs. ved  $(X, Y, Z, 1)$ . Da kan den samlede koordinatransformation beskrives ved:

$$\hat{\mathbf{X}} = (\hat{X}, \hat{Y}, \hat{Z})^t = \mathbf{W}\mathbf{X} \quad (7.16)$$

Den *perspektiviske projektion* fremkommer ved division af de første to koordinatværdier med den tredje koordinatværdi:

$$x = \frac{\hat{X}}{\hat{Z}} \quad \text{og} \quad y = \frac{\hat{Y}}{\hat{Z}} \quad (7.17)$$

Udskrevet fås:

$$\begin{aligned} x &= \frac{r_{11}X + r_{12}Y + r_{13}Z + T_x}{r_{31}X + r_{32}Y + r_{33}Z + T_z} \\ y &= \frac{r_{21}X + r_{22}Y + r_{23}Z + T_y}{r_{31}X + r_{32}Y + r_{33}Z + T_z} \end{aligned} \quad (7.18)$$

$$(7.19)$$

Den *affine projektion* fremkommer ved antagelserne  $r_3 = \vec{0}$ ,  $T_z = 1$  og  $Z = \text{konstant}$ . Dette svarer til at projektionen  $(x, y)^t$  kan beskrives ved en kombineret translation, rotation, skalering og skævvridning af  $(X, Y)^t$ , altså at dybdeinformationen kan ignoreres. Ovenstående projektligning reducerer til:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & T_x \\ r_{21} & r_{22} & T_y \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (7.20)$$

Affine projektioner kan være acceptable approksimationer til perspektiviske projektioner, hvis dybdeforskellene mellem de projicerede punkter er meget små i forhold til den gennemsnitlige dybde  $\bar{Z}$ .

Hvis den  $2 \times 2$  matrix  $\mathbf{R}$  er ortogonal kaldes afbildningen for ligedannet (eng. similarity transform). Hvis matricen yderligere er orthonormal kaldes afbildningen for euklidisk.

## 7.3 Homeografi er

Lad  $\mathbf{x}_i = (x_i, y_i, w)^t$  og  $\mathbf{x}'_i = (x'_i, y'_i, w')^t$  være perspektiviske projektioner (i homogene koordinater) af samme sæt af scenepunkter. Billedpunkterne svarer altså til  $(x_i/w, y_i/w)$  og  $(x'_i/w', y'_i/w')$ . En homeografi  $H$  er en projektiv afbildning mellem de to sæt af billedpunkter, således at  $\mathbf{x}'_i = H\mathbf{x}_i$ . Homeografien  $H$  er altså en  $3 \times 3$  matrix.  $H$  har 9 elementer, men er kun defineret op til en skalafaktor. Den har derfor kun 8 frihedsgrader. Da hver punktkorrespondance  $(\mathbf{x}, \mathbf{x}')$  giver 2 ligninger ses, at der mindst kræves 4 punktkorrespondancer for at finde  $H$ .

I det følgende skal vi se en simpel metode til estimation af  $H$ . Denne metode antager at punktkoordinaterne for sættet af korresponderende punkter er korrekte og målt med stor præcision. Typisk benyttes mange flere end de 4 minimalt krævede sæt af paspunkter.

Da homeografien kun er defineret op til en skalafaktor, er vektorene  $\mathbf{x}'_i$  og  $H\mathbf{x}_i$  parallelle, men ikke nødvendigvis ens. Krydsproduktet  $\mathbf{x}'_i \times H\mathbf{x}_i$  vil derfor være nulvektoren. Krydsproduktet kan skrives:

$$\mathbf{x}'_i \times H\mathbf{x}_i = \begin{pmatrix} y'_i \mathbf{h}_3^t \mathbf{x}_i - w'_i \mathbf{h}_2^t \mathbf{x}_i \\ w'_i \mathbf{h}_1^t \mathbf{x}_i - x'_i \mathbf{h}_3^t \mathbf{x}_i \\ x'_i \mathbf{h}_2^t \mathbf{x}_i - y'_i \mathbf{h}_1^t \mathbf{x}_i \end{pmatrix} = \mathbf{0}$$

hvor  $\mathbf{h}_j^t$  er den  $j$ 'te række af  $H$ . Da  $\mathbf{h}_j^t \mathbf{x}_i = \mathbf{x}_i^t \mathbf{h}_j$ , kan ligningen også skrives

$$\begin{bmatrix} \mathbf{0}^t & -w_i^t \mathbf{x}_i^t & y_i' \mathbf{x}_i^t \\ w_i^t \mathbf{x}_i^t & \mathbf{0}^t & -x_i' \mathbf{x}_i^t \\ -y_i' \mathbf{x}_i^t & x_i' \mathbf{x}_i^t & \mathbf{0}^t \end{bmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = \mathbf{0} \quad (7.21)$$

Ovenstående  $3 \times 9$  matrixligning,  $\mathbf{A} \mathbf{h} = \mathbf{0}$ , angiver en lineær metode til estimation af de 9 ubekendte  $h_j$ .

$$\mathbf{h} = \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} \quad \text{og} \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (7.22)$$

Bemærk at de tre ligninger ikke er uafhængige (pånær en skalafaktor fås den sidste ligning ved  $x_i'$  gange den første plus  $y_i'$  gange den anden), hvorfor den sidste ligning udelades. Vælges  $w_i' = 1$ , svarende til at  $(x_i', y_i')$  er koordinater i billedplanet, kan de sidste to ligninger skrives:

$$\begin{bmatrix} 0 & 0 & 0 & -x_i & -y_i & -1 & y_i' x_i & y_i' y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & x_i' x_i & x_i' y_i & x_i' \end{bmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (7.23)$$

Vi kan ikke normalt forvente at koordinatangivelserne er præcise. Ofte antages at støjen på disse er normalfordelt med en spredning på en eller nogle få pixels. Dette betyder, at vi for at opnå en rimelig løsning må benytte væsentlig flere end 4 paspunkter. For  $n$  paspunkter fås et homogent  $2n \times 9$  ligningssystem. Dette vil stadig have rang 8 (medmindre mindst  $n - 1$  af disse ligger på en ret linie). Vi er ikke interesserede i den trivielle løsning  $\mathbf{h} = \mathbf{0}$ , men i en med  $|\mathbf{h}| > 0$ . Da vi ved at løsningen kun kan bestemmes op til en skalafaktor, kan vi f.eks. vælge den med  $|\mathbf{h}| = 1$ . Den i mindste kvadraters forstand bedste løsning er givet ved den singulære vektor knyttet til den mindste singulære værdi i en (svd-) analyse af matricen  $\mathbf{A}$ . Hvis  $\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^t$  er en singular værdi dekomposition, og det antages at de singulære værdier i diagonalen af  $\mathbf{D}$  er arrangeret i aftagende rækkefølge, fås løsningen altså som den sidste søjle i matricen  $\mathbf{V}$ .

Givet estimation af matricen  $\mathbf{H}$  kan vi for alle punkter  $(x_i, y_i)$  i det ene billede beregne homografi-projektionen  $(x_i', y_i', w_i')$  i det andet billede og dernæst koordinaterne  $(x_i'/w_i', y_i'/w_i')$ .

Det bør bemærkes, at ovenstående metode til estimation af homeografien er meget følsom over for valg af koordinatsystem. For at opnå acceptable resultater er det sædvanen inden estimationen at translaterer alle punkter således, at de får middelværdi nul, samt at skalere koordinatværdierne, således at punkternes middelfastand til det ny origo er  $\sqrt{2}$ . Lad  $T$  og  $T'$  være de to normaliserende transformationer for de to punktmængder. Efter normalisering og estimation af  $\mathbf{H}$ , denormaliseres denne derefter ved:  $\tilde{\mathbf{H}} = T'^{-1} \mathbf{H} T$ .

Ovenstående metode er ofte kun det første trin i bestemmelsen af en homeofrafi. For at opnå bedre estimater er det ofte sædvane at anvende en efterfølgende ikke-lineær optimering. For en mere detalieret beskrivelse af metoden, samt alternative og udvidede metoder henvises til [17].

Estimation af homeografier for billedpar er en basal diciplin i 3D computer vision. Her ønsker man ofte ud fra et billedpar at bestemme scenens tre-dimensionale struktur. Det kan vises at homeografien indeholder essentiel information om hvorledes de to kameraer er orienteret i forhold til hinanden. Bestemmelsen af homeografien ud fra relativt få paspunkter kan f.eks. benyttes til etablering af nye korresponderende punkter. Givet et nyt punkt i et af billederne begrænser homeografien de mulige positioner af det korresponderende punkt i det andet billede til en ret linie. Dette letter søgningen efter det korreponderende punkt.





# Kapitel 8

## Filtrering ved Matematisk morfologi

Morfologi er læren om form (og stammer fra biologi). Matematisk morfologi dækker over en række metoder til filtrering af signaler og billeder. Metoderne kan ofte beskrives intuitivt ved hvorledes de ændrer formen af et signal/billede. Matematisk morfologi bygger på mængdelære. I de næste par afsnit skal vi begrænse os til binær morfologi, dvs. morfologiske operatoren anvendt på binære billeder. Et sådant billede kan beskrives ved den delmængde  $A \subset \mathbb{Z}^2$  af pixels der har værdi lig nul (er sorte). Ved anvendelse af en morfologisk operator dannes en ny mængde, der beskriver resultatet af filtreringen. Som vi skal se senere kan binær morfologi let udvides til gråtone- og farvemorfologi. Her betragtes blot mængder  $A \subset \mathbb{Z}^p$ . For gråtonemorfologi er  $p = 3$  for farvemorfologi er  $p = 5$ . Det skal også bemærkes at operationerne let kan formuleres i det euclidiske rum  $E^p$ . Der er ingen krav om numerabilitet af elementerne i mængderne. Derimod er det et krav, at det rum, hvori mængderne er defineret, er udstyret med et koordinatsystem. Således kan hvert element (punkt, pixel) i en mængde tilordnes en koordinat. Før de grundlæggende morfologiske operationer Ekspansion (eng. dilation) og Erosion beskrives, skal vi kort repetere nogle velkendte definitioner fra mængdelæren.

**Foreningsmængde og fællesmængde:** Foreningsmængden  $A \cup B$  hhv. fællesmængden  $A \cap B$  af to mængder  $A$  og  $B$  betegner de elementer (pixels), der tilhører en eller begge af de to mængder hhv. de elementer, der netop tilhører begge mængder:

$$A \cup B = \{x \mid x \in A \text{ eller } x \in B\} \quad (8.1)$$

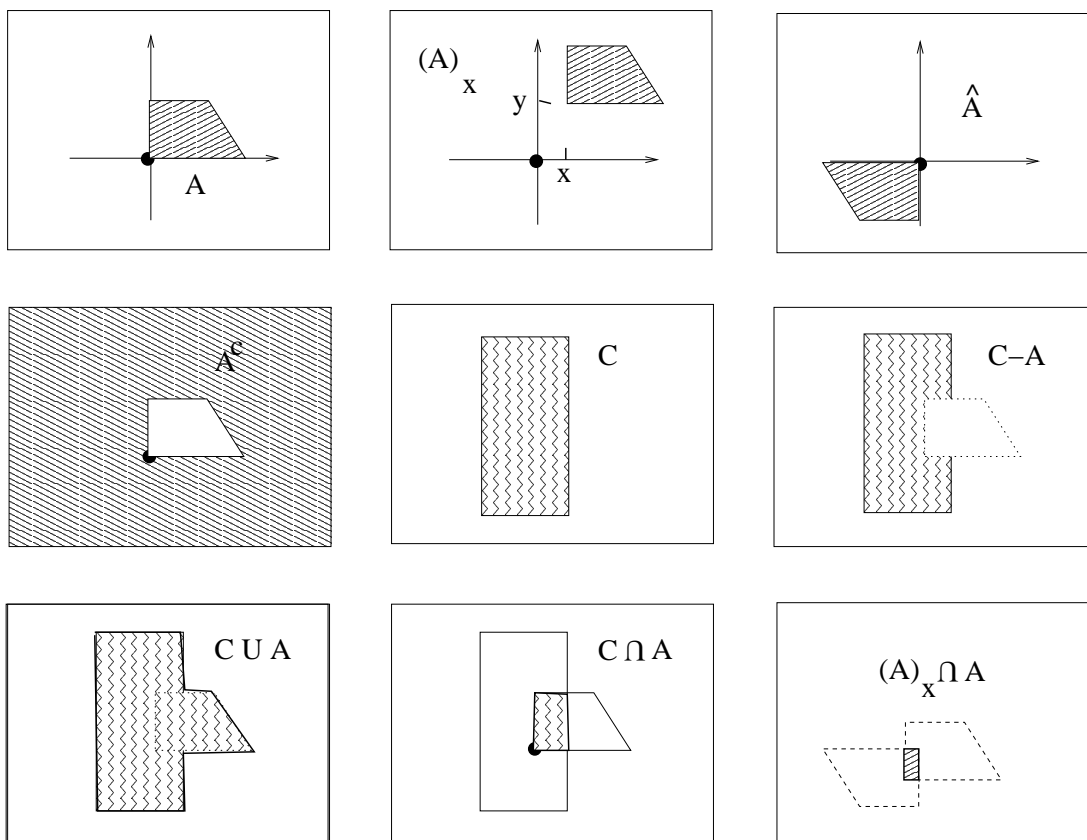
$$A \cap B = \{x \mid x \in A \text{ og } x \in B\} \quad (8.2)$$

**Translation:** Translationen  $(A)_x$  af en mængde  $A$  med en værdi  $x$  er givet ved:

$$(A)_x = \{c \mid c = a + x, \text{ for } a \in A\} \quad (8.3)$$

**Spejling:** Spejlingen  $\hat{B}$  af en mængde  $B$  er defineret ved:

$$\hat{B} = \{x \mid x = -b, b \in B\} \quad (8.4)$$



Figur 8.1: De grundlæggende mængdeoperationer.

**Komplement:** Komplementet  $A^c$  til mængden  $A$  er givet ved:

$$A^c = \{x \mid x \notin A\} \quad (8.5)$$

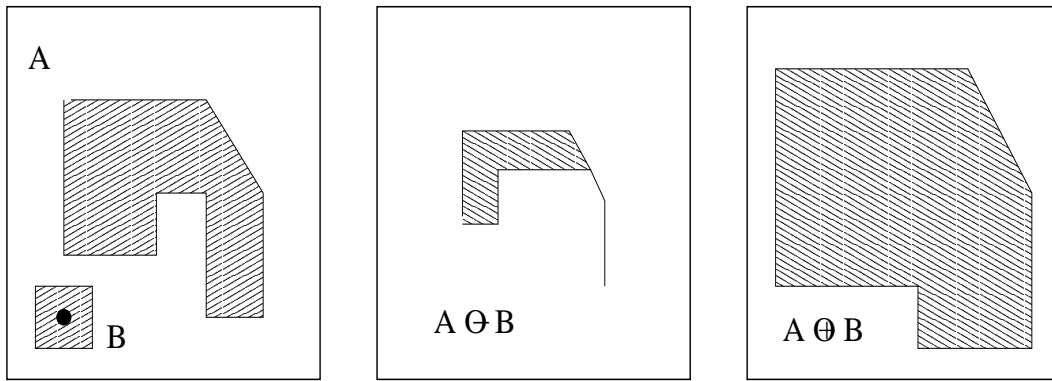
**Difference:** Differencen  $A - B$  mellem to mængder  $A$  and  $B$  er givet ved:

$$A - B = A \cap B^c = \{x \mid x \in A, x \notin B\} \quad (8.6)$$

De grundlæggende mængdeoperationer er illustreret i Figur 8.1.

## 8.1 Ekspansion og Erosion

Ekspansion og erosion er de to vigtigste basisoperationer inden for matematisk morfologi. I det følgende vil  $A$  betegne billedmængden, dvs. mængden af “sorte” pixels i billedet. Mængden  $B$  betegnes *strukturelementet* og svarer i en vis forstand til et filter. Ekspansionen af mængden  $A$



Figur 8.2: Erosion og ekspansion

med strukturelementet  $B$  er defineret ved:

$$A \oplus B = \{x \mid (\hat{B})_x \cap A \neq \emptyset\} \quad (8.7)$$

$$= \{x \mid (\hat{B})_x \cap A \subseteq A\} \quad (8.8)$$

Ekspansionen er således defineret ved mængden af alle forskydelser  $\vec{x}$  således at  $(\hat{B})_{\vec{x}}$  og  $A$  overlapper med mindst ét element. Som det ses, ligner definitionen meget udtrykket for en foldning. Erosion af mængden  $A$  med strukturelementet  $B$  er defineret ved:

$$A \ominus B = \{\vec{x} \mid (B)_{\vec{x}} \subseteq A\} \quad (8.9)$$

Erosionen er således defineret ved de punkter  $\vec{x}$  hvor  $B$ , translateret med  $\vec{x}$ , er helt indeholdt i  $A$ . Det kan let vises, at ekspansion og erosion er duale processer i den forstand at:

$$(A \ominus B)^c = A^c \oplus \hat{B} \quad (8.10)$$

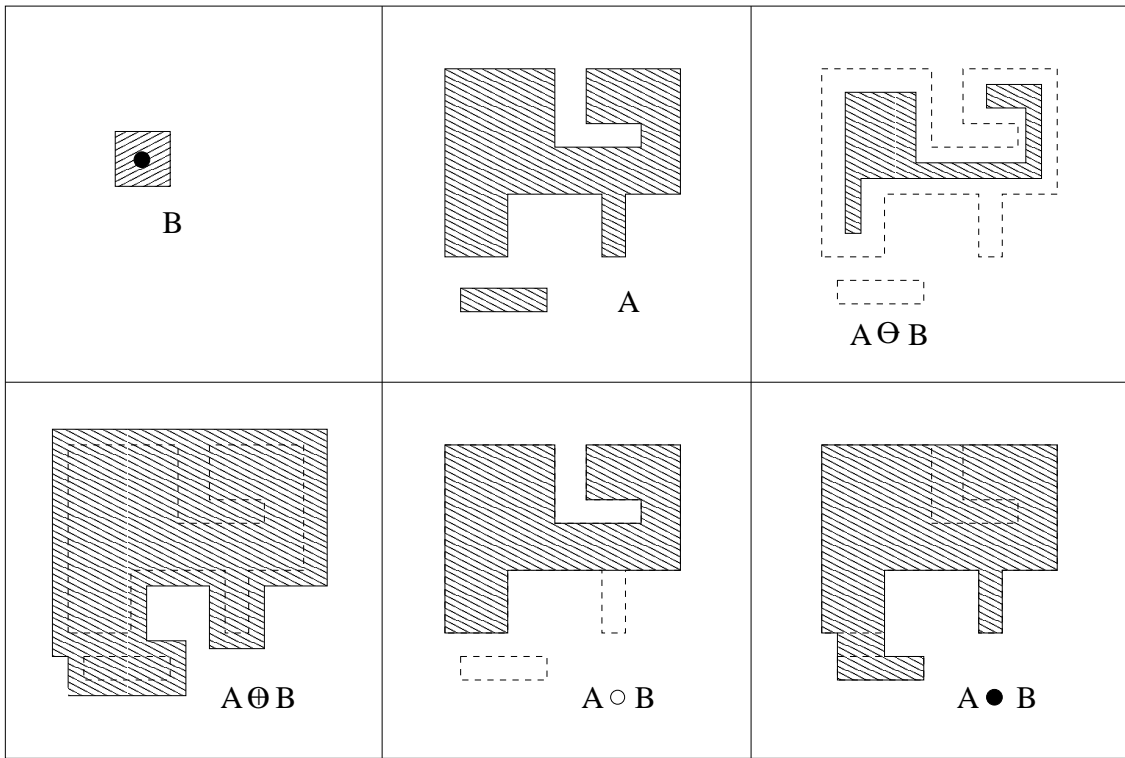
Bortset fra complementaritet er resultatet af at erodere “objektet”  $A$  med strukturelementet  $B$  således identisk med ekspansionen af “baggrunden” med det spejlede strukturelement. De to operationer er illustreret i Figur 8.2. Som det ses virker erosionen ved at “skrælle” et lag af “objekterne” i billedet. Herunder forsvinder mindre spidser og udbulinger. Ekspansionen, derimod, “lægger et lag på”. Herunder opfyldes mindre huller og indhak.

## 8.2 Åbning og Lukning

åbning og lukning af en mængde  $A$  med strukturelementet  $B$  er defineret ved:

$$\text{Åbning:} \quad A \circ B = (A \ominus B) \oplus B \quad (8.11)$$

$$\text{Lukning:} \quad A \bullet B = (A \oplus B) \ominus B \quad (8.12)$$



Figur 8.3: Åbning og lukning

Som det fremgår af definitionerne på erosion og ekspansion, er virkningen af såvel åbning som lukning at udglatte konturer. åbninger vil herunder eliminere tynde udbulinger, hvor lukninger vil fylde små indhak. Anderledes sagt: Åbninger fjerner små “positive” støjelementer, hvor lukninger fjerner “negative” støjelementer. Dette er illustreret i Figur 8.3. Som for ekspansion og erosion gælder der, at åbning og lukning er duale operationer mht. komplement og spejling, dvs.:

$$(A \bullet B)^c = (A^c \circ \hat{B}) \quad (8.13)$$

Desuden gælder der:

$$A \circ B \subseteq A \subseteq A \bullet B \quad (8.14)$$

$$(A \bullet B) \bullet B = A \bullet B \quad (8.15)$$

$$(A \circ B) \circ B = A \circ B \quad (8.16)$$

$$C \subseteq D \Rightarrow C \circ B \subseteq D \circ B \quad (8.17)$$

$$C \subseteq D \Rightarrow C \bullet B \subseteq D \bullet B \quad (8.18)$$

Der kan desuden vises en lang række sætninger, der beskriver relationer mellem udtryk involverende de morfologiske operationer og de grundlæggende mængdeoperationer. Interesserede henvises f.eks. til [16].

## 8.3 Morfologiske operationer

En lang række operationer på binære billeder kan konstrueres ud fra de ovennævnte grundlæggende morfologiske operatorer. Af de mere simple kan der nævnes:

1. Operationen  $f - (f \ominus B)$  ekstraherer indersiden af objekterne i et binært billede.
2. Operationen  $(f \oplus B) - f$  ekstraherer ydersiden af objekterne i et binært billede.
3. Operationen  $(f \circ B) \bullet B$  angiver en (formindsket) udglatning af objekterne i et binært billede.
4. Operationen  $(f \bullet B) \circ B$  angiver en (forstørret) udglatning af objekterne i et binært billede.
5. Operationen  $(f \bullet B) - (f \circ B)$  er et mål for støjen, herunder iregulariteten, af et binært billede.

Andre operationer er:

- Regionsfyldning (eng. region filling).
- Detektion af objektsammenhæng (eng. connected component analysis).
- Bestemmelse af det konvekse hylster.
- Udtynding af objekter (eng. thinning), herunder skelettransformationer.

Ovennævnte fire operationer implementeres ofte ved iterativ anvendelse af de mere basale operationer. En god beskrivelse af sådanne implementationer kan f.eks. findes i: *Gonzales and Woods: "Digital Image Processing", 1992*. En af de mere simple operationer *Hit-or-miss operationen* beskrives kort herunder.

### 8.3.1 Hit-or-Miss transformationen

Hit-or-Miss transformationen har til formål at bestemme et godt "match" mellem en prototype  $B$  og forekomster af samme figur i et billede  $A$ . Det er klart, at det er et krav til et objekt i billedet, at  $B$  er indeholdt i objektet. Der vil således gælde at  $M \subseteq (A \ominus B)$  er mængden af de mulige placeringer af prototypen i billedet. Hvis billedet indeholder et objekt, der er meget større end  $B$  vil  $M$  indeholde en delmængde af dette objekt, selv om der ikke er tale om noget match. For at udelukke disse situationer betragtes komplementærmængden  $A^c$ . Denne mængde må indeholde komplementærmængden til prototypen  $B$ . Lad  $W$  være et lille vindue, der omslutter  $B$ . Mængden  $W$  kan eventuelt være fremkommet ved ekspansion af  $B$ . Komplementærmængden til prototypen  $B$  approksimeres nu ved  $W - B$ . Der skal således gælde at:  $M \subseteq A^c \ominus (W - B)$ . Hit-or-Miss transformationen defineres ved:

$$A \odot B = (A \ominus B) \cap [A^c \ominus (W - B)] \quad (8.19)$$

$A \odot B$  angiver således de punkter, hvor  $B$  finder et match i  $A$  samtidig med, at  $(W - B)$  finder et match i  $A^c$ .

Væsenligt ved al matematisk morfologi er valget af strukturelement. Hvis dette er valgt uhensigtsmæssigt vil resultatet af filtreringen ofte være anderledes end forventet. Specielt er det vigtigt at kende skalaen for de “fænomener” i billederne som man ønsker at fremhæve eller bortfiltrere. Problemet med valg af strukturelement svarer helt til det tilsvarende problem ved valg af sædvanlige filterfunktioner ved lineær filtrering. Der er ingen lette løsninger. Dog vil det oftest være fortolkningsmæssigt fordelagtigt at vælge et strukturelement, der er symmetrisk om Origo, f.eks. et kvadrat.

## 8.4 Gråtone-morfologi

Gråtonemorfologi defineres let ud fra binær morfologi ved at udvide dimensionen af mængderne. Således kan de to-dimensionale gråtoneoperationer *ekspansion* og *erosion* defineres ved:

$$\begin{aligned} f \oplus b(s, t) &= \text{MAX} \{ f(s - x, t - y) + b(x, y) \mid (s - x, t - y) \in \text{Def}_f, (x, y) \in \text{Def}_b \} \\ f \ominus b(s, t) &= \text{MIN} \{ f(s + x, t + y) - b(x, y) \mid (s + x, t + y) \in \text{Def}_f, (x, y) \in \text{Def}_b \} \end{aligned} \quad (8.20)$$

hvor  $\text{Def}_f$  hhv.  $\text{Def}_b$  er definitionsområderne for billedet  $f$  hhv. for strukturelementet  $b$ . Som det fremgår, minder ovenstående udtryk meget om definitionerne af foldning og korrelation. Man kan forestille sig såvel billedet  $f$  som strukturelementet  $b$  som flader  $(x, y, f(x, y))$  i 3D. Ekspansionen og erosionen fremkommer nu ved at lade fladeflisen (strukturelementet)  $b$  glide hen over den øvre hhv. nedre side af fladen  $f$  og registrere de punktmængder, som Origo for  $\hat{b}$  hhv.  $b$  herved gennemløber. Det ses, at erosionen generelt vil gøre billedet mørkere. Ekspansionen vil gøre billedet lysere. Erosionen vil desuden fjerne relativt lyse pletter med bredde mindre end størrelsen af strukturelementet. Ekspansionen vil modsat fjerne mørke områder, der er mindre end strukturelementet.

En anden, måske lettere forståelig, fortolkning af de basale gråtonemorfologiske operationer baserer sig på definitionen af en **umbra** for en funktion. Lad  $f(x)$  være en en-dimensional funktion med reelle eller heltallige værdier. Umbra'en for  $f$  er da defineret ved den punktmængde der ligger under funktionen. Dette er illustreret i Figur 8.4. De morfologiske operationer på gråtonefunktionen  $f$  kan ækvivalent defineres som de tilsvarende binære definitioner på umbra'en for  $f$ . Denne fortolkning kan umiddelbart udvides til flere dimensioner.

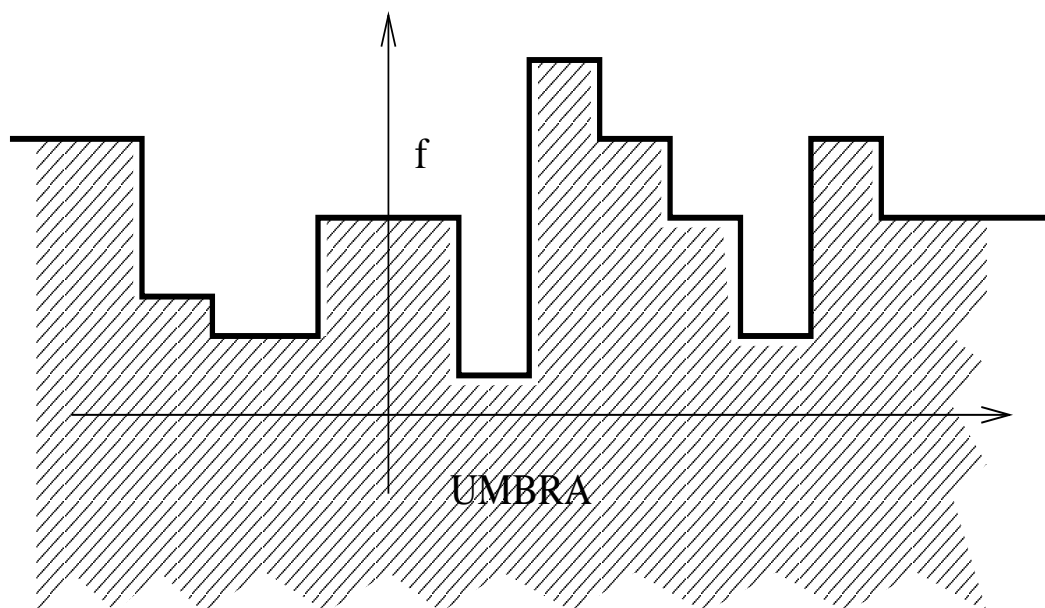
Definitionen af åbning og lukning af gråtone signaler eller billeder er helt identisk med definitionerne i det binære tilfælde, dvs.:

$$\text{Åbning:} \quad A \circ B = (A \ominus B) \oplus B \quad (8.21)$$

$$\text{Lukning:} \quad A \bullet B = (A \oplus B) \ominus B \quad (8.22)$$

Idet vi udnytter definitionerne  $f^c = -f(x, y)$  og  $\hat{b} = b(-x, -y)$  kan man vise gråtone-dualitetsrelationen:

$$-(f \bullet b) = (-f \circ \hat{b}) \quad (8.23)$$



Figur 8.4: Umbra for en en-dimensional funktion

Hvis man antager at strukturelementet er kugleformet, kan man forestille sig en gråtone åbning som den (øvre) 3D-flade som kuglen beskriver, når den bliver rullet hen over *undersiden* af fladen  $f$ . Tilsvarende kan lukningen beskrives ved den (nederste) flade som kuglen gennemløber når den rulles hen over *overfladen* af  $f$ .

## 8.5 Morfologiske gråtoneoperationer

Tre typiske morfologiske gråtoneoperationer er:

$$\text{Udglatning} \quad (f \circ b) \bullet b \quad (8.24)$$

$$\text{Gradient} \quad (f \oplus b) - (f \ominus b) \quad (8.25)$$

$$\text{Top-hat} \quad f - (f \circ b) \quad (8.26)$$

Det er ikke let at beskrive virkningen af de forskellige operationer i termer af frekvensoperationer. Visuelt har udglatningsoperationen lavpas-karakteristik, mens gradientoperationen og top-hat -operationen har højpas-karakteristik. Sidstnævnte operator er velegnet til at fremhæve detalier. Mere avancerede operationer kan sammenstykes ud fra de grundlæggende operationer ved iterativ anvendelse, ved kombination med de grundlæggende mængdeoperationer, samt med kombination af simple histogramteknikker og tælleoperationer. Interesserede læsere henvises til [32] for yderligere eksempler på anvendelse af matematisk morfologi.



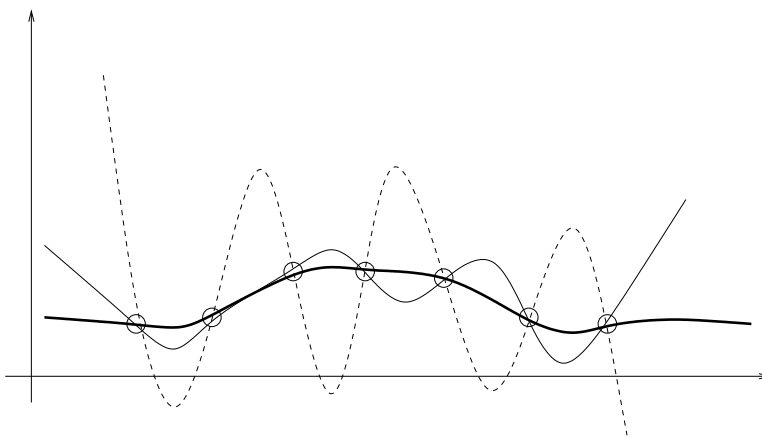


# Kapitel 9

## Regularisering og diffusion

Regularisering er en teknik, der kan benyttes, hvis man ønsker at tilføje a priori viden (betingelser) til et minimeringsproblem. Hvis der eksempelvis er for få ligninger i forhold til antallet af ubekendte, kan den a priori viden formuleres som et sæt af yderligere ligninger, således at det samlede problem kan løses. Hvis der er ligninger nok, men løsningen er ustabil (f.eks. ved at konditionen af en matrixligning er dårlig), da kan tilføjelsen af et regulariseringsled stabilisere løsningen. I det følgende skal vi se et eksempel på anvendelse af regularisering til rekonstruktion af støjfyldte signaler.

Lad  $c(x)$  være en reel funktion af en variabel, og antag at  $c(x)$  er fremkommet ud fra et ideelt signal  $g(x)$  ved tilføjelse af støj  $\eta(x)$ . Vi ønsker nu at bestemme et signal  $f(x)$  (løsningen), der dels ligger tæt på målingerne  $c(x)$  og dels “opfører sig pænt”. Sidstnævnte formuleres tit som et krav til kvadratet på den afledede af  $f$ , dvs. til  $|f'(x)|^2$ . For at forstå dette kan det være nyttigt at betragte de tre funktioner i figur 9.1. Figuren viser 7 punkter  $c(x)$ ,  $x = 1, 2, \dots, 7$  og tre kurver, der alle går igennem de 7 punkter.



Figur 9.1: Tre kurver, der alle passerer gennem samme 7 punkter.

Den fede kurve er klart den glatteste af de tre kurver. For punkter tæt på de 7 givne har den

fede kurve værdier, som ligger tæt på værdierne i de nærmeste givne punkter. Man vil derfor normalt foretrække denne frem for de andre kurver, der svinger mere. Den fede kurve er karakteriseret ved at have afledede, der er numerisk mindre.

Et standard mål for glatheden af en kurve er integralet af kvadratet på den afledede af kurven. Denne størrelse skal gerne være lille. Samtidig ønskes at kurven passerer igennem, eller tæt på, de givne målinger. Vi ønsker derfor sammenlagt at finde den funktion  $f(x)$ , der minimerer energifunktionen :

$$\Theta(f) = \int [ \|f(x) - c(x)\|^2 + \lambda \|f'(x)\|^2 ] dx \quad (9.1)$$

hvor  $\lambda$  kaldes *regulariseringsparameteren*. Jo større  $\lambda$  vælges, jo mere vægt lægges der på opfyldelsen af sidste term, dvs. på glathedskravet. Jo mindre  $\lambda$  jo større vægt på overensstemmelse med det givne signal. Det skal bemærkes at differentialoperatoren i glathedsledet kan omfatte mere end den/de førsteafledede. I to dimensioner eller mere er valget af differentialoperator ikke trivielt. En matematisk elegant metode til at finde den funktion  $f$ , som minimerer ovenstående funktional kaldes *variationsregning*. Denne (iøvrigt uundværlige) teknik er ikke del af dette kursus. Her skal vi derfor benytte en anden metode hvor vi først diskretiserer ligningen:

$$\text{Min} \sum \{ [f(x) - c(x)]^2 + \lambda [f(x+1) - f(x)]^2 \} \quad (9.2)$$

Bemærk først at ved at ekspandere summen vil  $f(x)$  indgå i tre termer, nemlig:  $[f(x) - c(x)]^2$ ,  $[f(x+1) - f(x)]^2$ , og  $[f(x) - f(x-1)]^2$ . Ved nu at differentiere mht.  $f(x)$ , og derefter at sætte resultatet lig nul fås:

$$2[f(x) - c(x)] - 2\lambda[f(x+1) - f(x)] + 2\lambda[f(x) - f(x-1)] = 0 \quad (9.3)$$

For løsningen  $f(x)$  til minimeringsproblemet skal ovenstående ligning være opfyldt for alle  $x$ . Dvs:

$$f(x)[1 + 2\lambda] = c(x) + \lambda[f(x+1) + f(x-1)] \quad (9.4)$$

Det er let at forestille sig hvorledes  $f(x)$  kan bestemmes ved ud fra en startsituation  $f^0(x) = c(x)$  at iterere nedenstående opdatering indtil der i alle punkter  $x$  gælder at  $f^{t+1}(x) = f^t(x)$ .

$$f^{t+1}(x) = \frac{c(x) + \lambda[f^t(x+1) + f^t(x-1)]}{1 + 2\lambda} \quad (9.5)$$

Alternativt til ovenstående *Jacobi*-opdatering kan man løse sættet af ligninger (9.3) direkte. Sættet af ligninger svarer til systemet  $\mathbf{X}\mathbf{f} = \mathbf{c}$  med  $N$  ligninger og ubekendte (hvor  $N$  er antallet af oprindeligt givne data). Lad  $\mathbf{c}$  hhv.  $\mathbf{f}$  være søjlevektorer bestående af de givne data hhv. de ønskede funktionsværdier. Under antagelse af at  $f(-1) = f(N+1) = 0$  ses at matricen  $\mathbf{X}$  får formen:

$$\mathbf{X} = \begin{pmatrix} 1+\lambda & -\lambda & & & \\ -\lambda & 1+2\lambda & -\lambda & & \\ & -\lambda & 1+2\lambda & -\lambda & \\ & & \ddots & \ddots & \ddots \\ & & & -\lambda & 1+2\lambda & -\lambda \\ & & & & -\lambda & 1+\lambda \end{pmatrix} \quad (9.6)$$

hvor matriceværdierne uden for det viste bånd er lig nul. Det ses at matricen har båndform og er diagonaldominant, hvilket garanterer konvergens af en Jacobi -eller en Gauss-Seidel opdatering.

I praksis giver regularisering, som beskrevet i ovennævnte eksempel, anledning til to problemer. Det første er at brugeren skal specificere en værdi af regulariseringsparameteren  $\lambda$ . Dette kan være vanskeligt og vil ofte kræve flere forsøg før et acceptabelt resultat er fundet. Det andet problem er mere fundamentalt: Diskontinuiteter i signalet vil blive udglattet på samme måde som støj. Denne observation er egentlig ikke overraskende fordi formuleringen af rekonstruktionen er lineær. Det er imidlertid ikke svært at modificere den basale regulariseringsligning (9.1), sådan at diskontinuiteter i signalet (til en vis grad) bliver bevaret, og ofte fremhævet. Årsagen til, at løsningen til regulariseringsligningen også kan formuleres som løsningen til et lineært ligningssystem, er, at alle termer i ligningen er kvadratiske. Ved *diskontinuert regularisering* er ideen at erstate det kvadratiske regulariseringsled med en mere kompliceret funktion af (f.eks.) den førsteafledede  $f'(x)$ .

$$\text{Min} \int [ \|f(x) - c(x)\|^2 + \lambda \rho(f'(x))] dx \quad (9.7)$$

Ved at vælge funktionen  $\rho$  passende er det muligt at bevare, og i visse tilfælde at fremhæve, diskontinuiteter i den rekonstruerede funktion  $f(x)$ . Et eksempel på en sådan funktion er:

$$\rho(t) = \begin{cases} t^2 & \text{hvis } t \leq T \\ T^2 & \text{ellers} \end{cases} \quad (9.8)$$

hvor  $T$  er en ny parameter. Virkningen af denne funktion er, at store grader i funktionen  $f$  ikke straffes ud over konstanten  $T^2$ . Dette betyder, at der ikke (i diskontinuitetspunkter) virker nogen kræfter på  $f(x)$ , som forsøger at udglatte funktionen. I stedet tilpasser funktionen  $f(x)$  sig helt til de givne data  $c(x)$  i sådanne punkter. Ovenstående model går i litteraturen under navnet *Den svage streng* (eng. weak string), fordi funktionen (strengen) knækker når data “hiver for meget” i den. En anden  $\rho$ -funktion er:

$$\rho(t) = \sigma \log(1 + t^2) \quad (9.9)$$

hvor  $\sigma$  er spredningen af støjen  $\eta$ . Ovenstående funktion går også under navnet *The Lorentzian estimator*.

Problemet ved brug af ovenstående metoder er, at løsningsrummet ikke længere er konvekst. Der kan med andre ord findes flere funktioner  $f_i$  der **lokalt** minimerer den opstillede funktional. Derfor er det nødvendigt at benytte algoritmer som *simuleret udglødning* (Se afsnit 9.3). Sådanne algoritmer er langsomme sammenlignet med lineære algoritmer og med metoder baseret på *Gradient descent* -metoden.

## 9.1 Mumford-Shah

Den teknik, som vi gennemgik i ovenstående afsnit, hvor kravene til løsningen specificeres gennem en funktional (9.1), er intensivt benyttet i en række viderudviklinger. Fælles er anvendelsen

af variationsregning, hvorved de implicit formulerede krav til løsningen transformeres til et sæt af koblede partielle differentiaalligninger. Disse løses som regel ved iterative metoder kendt fra sædvanlig numerisk analyse. Som tidligere nævnt er variationsregning ikke en del af dette kursus hvorfor vi ikke skal gennemgå sådanne metoder i detaljer. Dog skal vi kort se en enkelt af de mest refererede funktionaler, kendt som Mumford-Shah-funktionalet. Dette tager udgangspunkt i et ønske om samtidig udglatning og detektion af kanter (eller ækvivalent et ønske om segmentering). Et kantpunkt defineres som en pixel hvori størrelsen af gradienten er tilstrækkelig stor, dvs punkter  $(x, y)$  hvor  $\|\nabla(f)\| \geq T$ . Vi ønsker ikke at den søgte funktion  $f(x, y)$  er glat i et kantpunkt. Lad  $\mathcal{E}$  være mængden af kantpunkter og  $\mathcal{A}$  være mængden af ikke-kantpunkter. Idet koordinatangivelser udelades kan ønsket formuleres ved:

$$\Theta(f) = \int \int \|f - c\|^2 dx dy + \lambda \int_{\mathcal{A}} \int \|\nabla(f)\|^2 dx dy \quad (9.10)$$

Problemet ved ovenstående variant er, at  $\Theta$  vil antage minimal værdi lig nul, hvis vi vælger  $f = c$  og  $\mathcal{A} = \emptyset$ , altså hvis alle pixels er kantpunkter. Det er klart ikke, hvad vi ønskede. Vi må derfor indføre en straf for at introducere et kantpunkt svarende til funktionalen:

$$\Theta(f) = \int \int \|f - c\|^2 dx dy + \lambda \int_{\mathcal{A}} \int \|\nabla(f)\|^2 dx dy + \epsilon \int \int \delta(\mathcal{E}) dx dy \quad (9.11)$$

hvor  $\delta(\mathcal{E})$  er indikatorfunktionen på  $\mathcal{E}$ . Dette udtryk kaldes *Mumford-Shah-funktionen*. Minimering af udtrykket vil lede til to udtryk for hver pixel, og som begge evalueres for hver iteration i minimeringen. Det ene udtryk klassificerer punktet som kantpunkt eller ikke-kantpunkt. Det andet udtryk beregner den forbedrede værdi af funktionen  $f(x, y)$ .

Der findes yderligere et antal modifikationer af ovenstående funktional. Således kan den binære klassifikation af et punkt, som kantpunkt erstattes af et reelt tal  $e(x, y) \in [0 : 1]$ . Funktionalen bliver:

$$\Theta(f) = \int \int [\|f - c\|^2 + \lambda \cdot (1 - e) \|\nabla(f)\|^2 + \epsilon \cdot e] dx dy \quad (9.12)$$

Andre varianter omfatter udtryk hvorved kantgraden  $e(x, y)$  gøres afhængig af kantgraden i nabopunkter. Herved kan man formulere et ønske til de resulterende kanter om at være glatte og sammenhængende. Videre diskussion af sådanne metoder ligger uden for rammerne af disse noter.

## 9.2 Markov Random Fields

En stokastisk proces er en én- eller flerdimensional følge af indicerede stokastiske variable. For éndimensionelle processer  $x[i]$  betegner indices typisk en (ækvivalent samlet) tidsangivelse. For todimensionale processer  $x[i][j]$ , også kaldt *stokastiske felter*, tænkes ofte på billeder, hvor  $i$  og  $j$  angiver positionen og hvor værdien i hvert billedelement modelleres ved en stokastisk variabel. Modellering af indicerede størrelser vha. stokastiske processer er meget anvendte inden for signalanalyse og billedbehandling.

En stokastisk proces  $x[i]$ ,  $i = 1, 2, \dots$  kaldes *stationær* (eng. wide sense stationarity), hvis middelværdien  $m[i]$  og variansen  $\sigma^2[i]$  af  $x$  er den samme for alle  $i$ . Hvis  $x$  er flerdimensional erstattes kravet om ens varians, med et krav om at kovariansmatricen er konstant.

En stokastisk proces kaldes uafhængig, hvis sandsynligheden for at observere værdien  $x = t$  til tidspunktet (stedet)  $i$  er uafhængig af alle tidligere observerede værdier. Dette kan udtrykkes:

$$p(x[i] \mid x[i-1], x[i-2], \dots, x[0]) = p(x[i]) \quad (9.13)$$

Processen  $x[i]$  kaldes en *Markov process af  $n$ 'te orden*, hvis sandsynligheden for at observere værdien  $x[i] = t$  til tiden  $i$  kun er afhængig af de  $n$  sidst observerede værdier:

$$p(x[i] \mid x[i-1], x[i-2], \dots, x[0]) = p(x[i] \mid x[i-1], \dots, x[i-n]) \quad (9.14)$$

Mange fænomener kan modelleres godt ved markovprocesser. I billeder er intensiteten i et givet billedelement ofte tæt på intensiteten i nabo billedelementerne, hvorimod korrelationen med intensiteter i billedelementer, der ligger langt fra hinanden, er forsvindende. Antagelsen om markovegenskaben benyttes derfor ofte i signal og billedanalyse, samt ved kodning baseret på informationsteori.

Der findes en meget omfangsrig litteratur, såvel som et meget stort antal anvendelser, af stokastiske (markov) processer. Vi skal her kun omtale de såkaldte *Markov random fields* (MRF), og deres relation til *Gibbs random fields* (GRF) ved det såkaldte *Hammersley-Clifford teorem*.

Et markov random field (MRF) er et todimensionelt felt  $x[i][j]$  af stokastiske variable, om hvilken markovbetingelsen er at:

$$\begin{aligned} p(x[i][j] = t_{ij} \mid x[k][l] = t_{kl}, i \neq k, j \neq l) \\ = p(x[i][j] = t_{ij} \mid x[k][l] = t_{kl}, i \neq k, j \neq l, (k, l) \in \mathcal{N}) \end{aligned} \quad (9.15)$$

hvor  $\mathcal{N}$  er et naboområde. Et typisk anvendt naboområde er bestemt ved de fire naboer:  $(i-1, j)$ ,  $(i+1, j)$ ,  $(i, j-1)$ , og  $(i, j+1)$ . Et sådant MRF siges at være af orden 1. Hvis de 8 naboer benyttes siges feltet at være af orden 2.

Et Gibbs random field (GRF) mht. et naboområde  $\mathcal{N}$  er defineret ved:

$$p(x[i][j] = t) = \frac{1}{Z} \exp(-U(t)/T) \quad (9.16)$$

hvor  $Z$  er en normaliseringskonstant (så integralet af  $p$  bliver 1), hvor  $T$  er en parameter kaldet *temperaturen*, og hvor  $U(t)$  er en *energifunktion*, der har formen:

$$U(t) = \sum_k V_k(t) \quad (9.17)$$

Funktionen  $V_k(t)$  kaldes en *potentialfunktion*. Antallet af potentialfunktioner svarer typisk til antallet af bånd, der begrænser friheden af det stokastiske felt. Om  $V_k(t)$  gælder at den kun

afhænger af værdierne af  $x$  i naboområdet  $\mathcal{N}$  (dvs. en markovegenskab). Styrken ved formalismen er, at den tillader opskrivning af en sandsynlighedsfunktion, der er nem at håndtere, når blot de nødvendige energifunktioner kan formuleres. Det sidste er ofte forholdsvist let.

Det er ikke umiddelbart klart, at MRF's har noget som helst at gøre med GRF's. Der eksisterer imidlertid en sætning, kaldet *Hammersley-Clifford teoremet*, der siger, at et stokastisk felt er en GRF, hvis og kun hvis det er et MRF. De to formuleringer er altså ækvivalente.

### Eksempel

I stereoanalyse af billeder betragtes to billeder af samme scene. Antag at hvert scenepunkt, som er synligt i det ene billede også er synligt i det andet billede. Der eksisterer dermed en korrespondance mellem punkter  $P_L = (x_L, y_L)$  i det "venstre" billede og  $P_R = (x_R, y_R)$  i det "højre" billede.

Givet de to billeder er der ikke umiddelbart klart hvorledes punkterne i disse skal parres således at alle punktpar korresponderer. Vi skal i det følgende se et eksempel på en metode til at estimere korrespondancerne. Dette svarer til at bestemme to (diskrete) funktioner  $d_x$  og  $d_y$  således at:

$$\begin{aligned} x_R &= x_L + d_x(x_L, y_L) \\ y_R &= y_L + d_y(x_L, y_L) \end{aligned} \quad (9.18)$$

Funktionerne  $d_x$  hhv.  $d_y$  kaldes den horisontale hhv. vertikale disparitet. Det er ofte muligt at optage billederne således at den vertikale disparitet  $d_y()$  er konstant lig 0. Dette skal vi antage i det følgende. Lad billederne være af dimension  $N \times N$ . For hvert af de  $N^2$  punkter i det venstre billede er der således  $N$  mulige kandidater til løsning af korrespondanceproblemet, hvis vi antager at hvert venstre-punkt har netop ét korresponderende højre-punkt (denne antagelse er noget problematisk). Uden yderligere oplysninger er der derfor  $N^{2N}$  forskellige løsninger til problemet. I praksis findes to ofte benyttede a priori antagelser, der begrænser løsningsrummet.

- Intensiteten  $I_L$  og  $I_R$  i to korresponderende punkter  $P_L$  og  $P_R$  er tilnærmelsesvis ens. Bemærk at denne antagelse ikke holder for blanke eller spejlende materialer. På grund af støj i billedoptagelsen kan forskellen i intensitet ofte modelleres ved en normalfordeling med middelværdi 0. Støjen i to punkter antages at være uafhængig.
- Dispariteten  $d_x$  er tilnærmelsesvis ens i nabopunkter. Denne antagelse er rimelig hvis scenefladen er glat, men holder ikke de steder hvor afstanden til sceneobjekterne ændrer sig diskontinuert.

Nedenstående gives en simpel skitse til, hvorledes et estimat af dispariteten  $d_x$  kan findes. Vi skal modellere  $d_x(x_L, y_L)$  som en stokastisk variabel. Sandsynligheden for, at denne antager værdien  $D$  i punktet  $(x_L, y_L)$  givet intensiteterne  $I_L$  i  $(x_L, y_L)$  og  $I_R$  i  $(x_L + D, y_L)$ , kan opskrives som:

$$p(D|I_L, I_R) = \frac{p(I_L, I_R, D)}{p(I_L, I_R)} = \frac{p(I_R|D, I_L) p(D, I_L)}{p(I_R, I_L)} = \frac{p(I_R|D, I_L) p(D|I_L)}{p(I_R|I_L)} \quad (9.19)$$

Der er ikke i almindelighed grund til at tro, at dispariteten er afhængig af intensiteten  $I_L$ . Derfor er  $p(D|I_L) = p(D)$ . Nævneren i udtrykket er uafhængig af  $D$ . Den værdi, der maksimerer  $p(I_R|D, I_L) p(D)$  vil derfor også maksimere venstresiden af ovenstående udtryk. Vi skal nu udnytte den første af de tidligere nævnte antagelser, at forskellen i intensitet i korresponderende punkter er normalfordelt med middelværdi 0. Dette betyder at:

$$p(I_R|D, I_L) = p([I_R(x_L + D, y_L) - I_L(x_L, y_L)] = t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right) \quad (9.20)$$

Da støjen i to punkter er antaget uafhængig, er den samlede betingede sandsynlighed for at se det højre billede giver det venstre billede samt disparitetsfunktionen proportional med produktet (over alle billedpunkter) af ovenstående udtryk, dvs.

$$\prod_{(i,j) \in [N \times N]} \left[ p(D) \cdot \frac{1}{Z} \exp[-U(t)/T] \right] \quad (9.21)$$

For at kunne udnytte den anden antagelse, at funktionen  $d_x$  er glat, er det hensigtsmæssigt at modellere denne som et markov random field. Dette betyder, jvf. Hammersley-Clifford teoremet, at vi kan skrive:  $p(D)$  på samme form som ligning (9.16).

$$p(D = t) = \frac{1}{Z} \exp[-U(t)/T] \quad (9.22)$$

Vi skal her benytte den simplest tænkelige naboomegn  $\mathcal{N}$ , nemlig de fire nærmeste naboer. Da  $d_x$  antages omtrentlig ens i nabopunkterne, er det oplagt at definere:

$$V_C(t) = \sum_{i \in \mathcal{N}} [t - t_i]^2 \quad (9.23)$$

Størrelsen  $V_C$  måler således den lokale afvigelse fra antagelsen om konstant disparitet. I dette tilfælde har vi:

$$p(D) = \frac{1}{Z} \exp \left[ -\frac{1}{T} \sum_{k=-1,1} \sum_{l=-1,1} (d_x(i, j) - d_x(i+k, j+l))^2 \right] \quad (9.24)$$

Da vi kun ønsker at bestemme maksimum af ligning (9.19) kan vi ignorere konstanten  $Z$  i optimeringen. Lad  $\lambda = \frac{T}{2\sigma^2}$ . Da er aposteriori fordelingen  $p(D|I_L, I_R)$  proportional med:

$$\prod_{(i,j) \in [N \times N]} \exp \left( -\frac{\lambda}{T} [I_R(i + d_x(i, j), j) - I_L(i, j)]^2 \right) \exp \left( -\frac{1}{T} \sum_{k=-1,1} \sum_{l=-1,1} [d_x(i, j) - d_x(i+k, j+l)]^2 \right) \quad (9.25)$$

Idet logaritmfunktionen er monotont voksende, og temperaturen  $T$  er en konstant, ses at maksimum aposteriori estimeret af  $d_x(i, j)$  er bestemt ved minimum af udtrykket:

$$H(d_x) = \sum_{(i,j) \in [N \times N]} \left( \lambda [I_R(i + d_x(i, j), j) - I_L(i, j)]^2 + \text{Glat}(i, j) \right) \quad (9.26)$$

hvor

$$\begin{aligned} \text{Glat}(i, j) &= [d_x(i, j) - d_x(i - 1, j - 1)]^2 + [d_x(i, j) - d_x(i - 1, j + 1)]^2 \\ &= + [d_x(i, j) - d_x(i + 1, j - 1)]^2 + [d_x(i, j) - d_x(i + 1, j + 1)]^2 \end{aligned} \quad (9.27)$$

Hvad vi har vist er, at vi givet nogle relativt vage forudsætninger har været i stand til at opstille et relativt kompakt udtryk for kvaliteten  $H$  af en disparitetsfunktion  $d_x$ . Udtrykket angiver, for hver punkt i det venstre billede, den straf en given disparitetsværdi vil bidrage med. Udtrykket er ikke-negativt og lig 0 netop når forudsætningerne er eksakt opfyldt. Den funktion  $d_x$ , der minimerer funktionalen  $H$ , er et Bayes estimat af en mest sandsynlige disparitetsfunktion.

Det ses, at udtrykket  $H$  er parametriseret af konstanten  $\lambda$ . Denne parameter angiver et mål for troværdigheden af antagelsen om ens intensitet i korresponderende punkter i forhold til antagelsen om glathed af disparitetsfunktionen. En mulighed er at fastlægge en bestemt værdi af  $\lambda$  fra starten (fundet ved andre metoder eller gættet). En anden strategi er først at vælge en lille værdi af  $\lambda$ . Når en approksimativ løsning er fundet øges  $\lambda$  og en ny løsning søges i omegnen af den først fundne. Dette gentages et antal gange indtil løsningen er stabil eller  $\lambda$  har nået en maksimal værdi. Processen svarer således til en *grov-til-fin-analyse* hvor der i starten fokuseres på glathedskravet og til slut på kravet om overensstemmelse mellem intensiteterne i korresponderende punkter. Dette giver større chance for at løsningen svarer til et globalt minimum af funktionen  $H$ .

### 9.3 Simuleret udglødning

Ovenstående metoder (og mange andre i billedanalyse) benytter at formulere den ønskede operation som løsningen til et optimeringsproblem. Hvis de optillede krav er tilstrækkeligt komplekse vil løsningsrummet ikke være konvekst. Dette betyder, at simple metoder til optimering kun vil kunne levere en lokalt bedste løsning og ikke den bedste løsning. Jo mere komplekse kravene er, jo stærkere - og ofte mere beregningskrævende - optimeringsmetoder må der tages i anvendelse. Simuleret udglødning er flagskibet i armadaen af optimeringsmetoder. Simuleret udglødning kan håndtere vilkårligt komplekse krav/formuleringer og samtidig garantere en globalt bedste løsning – når blot der afsættes tilstrækkeligt med beregningskraft.

Simuleret afkøling (også kaldt simuleret udglødning, simuleret nedkøling mv.) er en metode, der er inspireret af krystaldannelse. Energien af en kold krystal er defineret ved summen over alle molekyler i krystallen af deres potentielle energi. Ved opvarmning af krystallen tilføres termisk energi. Afhængig af mængden af termisk energi kan molekylerne bevæge sig mere eller mindre frit. Efter opvarmningen afkøles krystallen. Herved fjernes termisk energi. Hvis afkøling sker hurtigt, vil krystallen ofte have uregelmæssigheder. Hvis krystallen afkøles meget langsomt, er det muligt at krystallen bliver helt perfekt (molekylerne er arrangeret i et perfekt gitterværk). Energien for den perfekte krystal vil være minimal. I simuleret afkøling forsøges denne proces efterlignet. Dette kræver, at der til enhver løsning (molekylekonfiguration) kan associeres et energimål  $E$ .



For en given temperaturværdi  $T$  er den *termiske ligevægt* karakteriseret ved en konstant fordeling af energiværdi. For stor  $T$  er fordelingen meget bred, for lille  $T$  er fordelingen koncentreret om den aktuelle konfiguration. Fordelingen af energiværdi ændrer sig således under afkølingen. Hvis temperaturen holdes konstant opstår et termisk ligevægtstilstand. Fordelingen af energi  $E$  ved temperaturen  $T$  kan vises at være bestemt ved Gibbs-funktionen:

$$p_T(E) = \frac{1}{Z(T)} \exp\left(-\frac{E}{kT}\right) \quad (9.28)$$

hvor  $Z(T)$  er en normaliseringsfaktor (eng. partition function), og hvor  $k$  er en konstant (kaldet Boltzmann konstanten). Fordelingen selv kaldes ofte for *Boltzmann fordelingen*.

Inden for et kort tidsinterval  $dt$  bevæger molekylerne sig mere eller mindre tilfældigt med en hastighed, der er afhængig af temperaturen. Givet en molekylekonfiguration  $K_t$  til tiden  $t$  er det derfor rimeligt at antage, at mængden af opnåelige konfigurationer til tiden  $t + dt$  er parametriseret af temperaturen  $T$ . Vi vil kalde denne mængde for en *naboomegn* og betegne den ved  $\mathcal{N}(K_t, T)$ .

*Metropolis algoritmen* er en meget kendt og anvendt metode til at simulere udviklingen mod et termalt equilibrium. En skitse af algoritmen er angivet nedenstående. Her er energien  $E$  parametriseret af den konfiguration  $\eta$ , som vi ønsker at bestemme.

1. Lad  $\eta$  være en konfiguration med energi  $E(\eta)$ .
2. Gentag:
  - Lad  $\eta_{ny}$  være en ny konfiguration valgt tilfældigt blandt alle nabokonfigurationer i  $\mathcal{N}(K_t, T)$ . Lad  $\eta_{ny}$  have energien  $E(\eta_{ny})$ , og lad  $\Delta = E(\eta_{ny}) - E(\eta)$ .
  - Hvis  $\Delta \leq 0$  eller hvis  $\exp(-\frac{\Delta}{kT}) > \text{random}([0, 1])$ , så flyt konfiguration, dvs. lad  $\eta := \eta_{ny}$ .
3. Indtil equilibrium er fundet.

Så længe en ny konfiguration med lavere energi findes, flyttes den aktuelle konfiguration imod denne. Hvis den ny konfiguration har større energi, flyttes kun med en vis sandsynlighed beskrevet ved Boltzmann fordelingen. Dette muliggør, at optimeringen kan undslippe lokale minima. Efter tilstrækkeligt mange iterationer kan det vises, at en ligevægtstilstand indtræder. I denne flyttes lige mange gange til en konfiguration med højere energi, som til én med lavere energi. Det kan være vanskeligt at specificere klart hvornår ligevægten indtræffer. I mange algoritmer benyttes et fast (stort) antal iterationer.

En algoritme til simuleret afkøling kan nu konstrueres ved i trin langsomt at sænke temperaturen, og først fortsætte på næste trin når ligevægten er fundet:

1. Lad  $T = T_{start}$ .
2. Gentag:
  - Benyt Metropolis algoritme for temperatur lig  $T$ .
  - Sæk temperaturen  $T := g(T)$ .
3. Indtil stopkriterium nået.

Temperaturen sænkes ofte med en konstant faktor, dvs.  $g(T) = \alpha T$ , hvor  $\alpha$  er en konstant tæt på 1 (f.eks. 0.9). Stopkriteriet består typisk af en kombination af et loft over antallet af iterationer, og et krav om en gradvis formindskelse af energivineauet.

Der er, som det fremgår, intet krav til energifunktionen  $E$ , bortset fra, at denne skal være ikke-negativ. Metoder er derfor meget velegnet i situationer hvor  $E$  ikke er konveks. Metoden er meget generel, men også ekstremt beregningskrævende. Et problem ved anvendelse af metoden at definere naboområderne  $\mathcal{N}$ . Ved optimering over et kontinuert domæne er det naturligt at bestemme  $\mathcal{N}$  som en kugle (med centrum i den aktuelle konfiguration) og med radius bestemt ved en aftagende funktion af temperaturen  $r(T)$ . Denne tilgang er imidlertid farlig, fordi visse komponenter af parametervektoren ofte varierer meget mere end andre. En anden oplagt tilgang er løbende at registrere de  $N$  hidtil bedste konfigurationer, og at definere  $\mathcal{N}$  ud fra det konvekse hylster af disse konfigurationer. Vi skal ikke her diskutere de mange varianter af optimering ved simuleret udglødning yderligere. Interesserede studerende anbefales at læse [12], der er en klassisk værk om brug af stokastiske metoder i billedanalyse.

## 9.4 Diffusion

Diffusion er en proces hvor intensiteten behandles som en flydende størrelse, der kan diffundere mellem pixels. Ofte benyttes analogien 'varme' for intensitet. Den klassiske varmediffusion i et homogent materiale er bestemt af en diffusionskoefficient, der angiver hvor let varmen spredes. Ved at gøre denne konstant positionsafhængig (og eventuelt yderligere retningsafhængig) kan diffusionen (læs: filtreringen) foretages således at støj og mindre kontraster forsvinder, og således at større kontraster bevares eller fremhæves. Nedenstående beskrives en simpel filtrering ved diffusion.

### 9.4.1 Perona-Malik

Som tidligere nævnt består en skalarumsudvidelse  $I(x, y, t)$  af et billede  $I(x, y)$  af en repræsentation af kontinuert mange foldninger af billedet med gaussfunktioner med voksende varians  $t$ :

$$I(x, y, t) = G(0, t) \star I(x, y, 0) \quad (9.29)$$

hvor  $I(x, y, 0) = I(x, y)$  og hvor  $G(\eta, \sigma^2)$  er gaussfunktionen med middelværdi  $\eta$  og spredning  $\sigma$ . Ækvivalent kan vi betragte billedet  $I(x, y, t)$  som en repræsentation af varme til tiden  $t = 0$ ,

og skalarummet vil angive varmfordelingen som funktion af tiden. Denne analogi er holdbar fordi gaussfunktionen er løsning til varmelednings- eller diffusionsligningen:

$$I_t = c[I_{xx} + I_{yy}] \quad (9.30)$$

hvor  $c$  er en konstant, kaldet varmeledningskoefficienten. Den grundlæggende idé ved anisotrop diffusion er at løse antagelsen om konstant varmeledningskoefficient. I dette tilfælde bliver diffusionsligningen:

$$\begin{aligned} I_t &= \operatorname{div}(c(x, y, t) \nabla I) = c \Delta I + \nabla c \cdot \nabla I \\ &= c(x, y, t)[I_{xx}(x, y) + I_{yy}(x, y)] + c_x(x, y, t)I_x(x, y) + c_y(x, y, t)I_y(x, y) \end{aligned} \quad (9.31)$$

Hvis  $c$  tillades at være lille hvor kontrasten i billedet er stor (og iøvrigt vælges fornuftigt), da vil resultatet af diffusionen blive at:

- Intensiteten, som funktion af tiden  $t$ , vil blive mere og mere glat mellem de store kontraster i billedet. Til slut vil billedet være stykvis konstant. Alt støj mellem kanterne vil være filtreret bort.
- Kontrasterne i billedet bliver skarpere (svarende til en udfoldning af en eventuel forvrængning).

Problemet er at finde et fornuftigt udtryk for  $c(x, y, t)$ . Det kan vises at ovenstående kan opfyldes hvis  $c$  vælges således at:

1.  $c(x, y, t) = g(|\nabla I(x, y, t)|)$
2.  $g(0) = 1$
3.  $g(\cdot)$  er kontinuert, differentiabel, positiv, og aftagende

To eksempler på valg af  $g$  er:

$$g(s) = \frac{1}{1 + [\frac{s}{K}]^{1+\alpha}} \quad (9.32)$$

$$g(s) = e^{-(\frac{s}{K})^2} \quad (9.33)$$

hvor  $K$  og  $\alpha$  er parametre. Antag at vi har valgt  $c$  som skitseret ovenfor, og at vi kun ser på den en-dimensionale diffusion (langs gradienten) over en kant. Lad  $\phi(I_x) = cI_x = g(I_x)I_x$ . Diffusionsligningen kan nu skrives:

$$I_t = \frac{\partial}{\partial x} \phi(I_x) = \phi'(I_x) \cdot I_{xx} \quad (9.34)$$

Antag videre at  $I$  er 3 gange differentiabel, og at  $c > 0$ . Tidsgradienten af intensitetshældningen kan nu udregnes (fordi vi kan ombytte rækkefølgen af differentiationer):

$$\begin{aligned} \frac{\partial}{\partial t}(I_x) &= \frac{\partial}{\partial x}(I_t) \\ &= \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} \phi(I_x) \right) \\ &= \phi'' \cdot I_{xx}^2 + \phi' \cdot I_{xxx} \end{aligned} \quad (9.35)$$

Antag videre at kanten er orienteret således at  $I_x > 0$ . I vendetangent-punktet gælder at  $I_{xx} = 0$  og at  $I_{xxx} < 0$ . Derfor vil  $\frac{\partial}{\partial t}(I_x)$  have modsat fortegn af  $\phi'(I_x)$  i en omegn omkring vendetangent-punktet. Hvis  $\phi'(I_x) > 0$  vil hældningen af intensitetsfladen derfor mindskes med tiden  $t$ . Hvis  $\phi'(I_x) < 0$  vil intensitetsgradienten i kantpunktet vokse som funktion af  $t$ . Der eksisterer derfor en grænseværdi  $T$ , for hvilken kanter udglattes hhv. skærpes afhængig af størrelsen af intensitetsgradienten i det oprindelige billede  $I(x, y, 0)$ . Størrelsen  $T$  vil være parametriseret af  $K$ . Hvis  $g(s)$  vælges som gaussfunktionen i formel (9.33) er det let at vise, at  $T$  er givet ved  $T = K/\sqrt{2}$ . Værdien af  $K$  er således afgørende for, om en kontrast i billedet er tilstrækkelig stor til at forhindre udglatning over kontrasten. Typisk angiver brugeren værdien af  $K$ . Et godt valg er ofte afgørende for resultatet.

Anisotrop diffusion implementeres lettest ved diskretisering af diffusionsligningen, isolering af udtrykket for  $I(x, y, t)$  og konstruktion af et program, der genberegner udtrykket for  $I$  ved iteration. Den globale information om intensitetsfladen vil typisk sprede sig med en hastighed af 1 pixel per iteration. Diameteren af det største *segment* i billedet vil derfor bestemme det mindste antal iterationer, der er nødvendige. Ofte spredes den lavfrekvente komponent af intensitetsfunktionen meget langsommere, hvorfor mange flere iterationer må foretages før konvergens. Hvis et subjektivt tilfredsstillende resultat er tilstrækkeligt, kan man ofte nøjes med relativt få iterationer. En konkret implementation er beskrevet i [29, 36]. Her er en lang række forhold omkring teknikken diskuteret med større grundighed. En simpel implementation er beskrevet herunder:

Betragt grafen med fem knuder svarende til punktet  $(x, y)$  og dets fire naboer. Lad  $I_{x,y}^t$  angive intensiteten i punktet  $(x, y)$  til tiden  $t$ . Lad diffusionskoefficienten  $c_{x,y}^{i,j}$  angive evnen til varmeledning mellem punkterne  $(x, y)$  og  $(x + i, y + j)$ . Lad endvidere:

$$\begin{aligned} D_{x,y}^{-1,0} &= I_{x-1,y} - I_{x,y} \\ D_{x,y}^{1,0} &= I_{x+1,y} - I_{x,y} \\ D_{x,y}^{0,-1} &= I_{x,y-1} - I_{x,y} \\ D_{x,y}^{0,1} &= I_{x,y+1} - I_{x,y} \end{aligned} \quad (9.36)$$

Alternativt til ovenstående simple definitioner kan det første led i hver ligning erstattes med et lokalt gennemsnit af intensiteterne i den pågældende retning. Dette vil øge stabiliteten.

I det følgende approksimeres normen af gradienten af  $I$  med den absolutte værdi af projektionerne langs de fire kanter i grafen. Dette betyder at:

$$\begin{aligned} c_{x,y}^{-1,0} &= g(|D_{x,y}^{-1,0}|) \\ c_{x,y}^{1,0} &= g(|D_{x,y}^{1,0}|) \\ c_{x,y}^{0,-1} &= g(|D_{x,y}^{0,-1}|) \\ c_{x,y}^{0,1} &= g(|D_{x,y}^{0,1}|) \end{aligned} \quad (9.37)$$

Første led i diffusionsligningen (9.31) kan nu approksimeres ved:

$$\lambda[c_{x,y}^{-1,0} \cdot D_{x,y}^{-1,0} + c_{x,y}^{1,0} \cdot D_{x,y}^{1,0} + c_{x,y}^{0,-1} \cdot D_{x,y}^{0,-1} + c_{x,y}^{0,1} \cdot D_{x,y}^{0,1}] \quad (9.38)$$

hvor  $\lambda$  er en normaliseringskonstant. Perona og Malik ignorerer det andet led  $\nabla c \cdot \nabla I$  i diffusionsligningen. Begrundelsen herfor er, at  $\nabla I$  hurtigt bliver forsvindende i næsten alle punkter. Kun tæt på kanterne i billedet vil  $\nabla I$  vedblive at være stor. Dette betyder at  $c \approx 1$  næsten overalt. Derfor vil  $\nabla c \approx \vec{0}$  næsten overalt. Tæt på en kant vil  $\nabla c$  være parallel med  $\nabla I$ . Kun her vil sidste term i diffusionsligningen bidrage væsentligt. Argumentet er nu, at det område hvori  $\nabla c \cdot \nabla I$  bidrager væsentligt, meget hurtigt bliver forsvindende. Approksimationen til diffusionsligningen kan nu udtrykkes:

$$I_{x,y}^{t+1} = I_{x,y}^t + \lambda [c_{x,y}^{-1,0} \cdot D_{x,y}^{-1,0} + c_{x,y}^{1,0} \cdot D_{x,y}^{1,0} + c_{x,y}^{0,-1} \cdot D_{x,y}^{0,-1} + c_{x,y}^{0,1} \cdot D_{x,y}^{0,1}] \quad (9.39)$$

Diffusionen foregår ved iterativt at opdatere samtlige intensitetsværdier og diffusionskoefficienter. Det er ofte praktisk at lade diffusionskoefficienterne på randen af billedet være lig 0 (fast).



# Kapitel 10

## Detektion af kanter i billeder

Detektion af kanter i et billede er en meget hyppig operation i mange industrielle billedbehandlingsopgaver såvel som i metoder fra datamatsyn. Uformelt defineres billedkanterne ved en mængde af billedpunkter, hvor kontrasten er stor, dvs. hvor intensitetsfunktionen markant ændrer værdi. Kanterne angiver således, hvor i billedet “der sker noget”. Disse er derfor en god beskrivelse af billedet.

Kanterpunkterne i et billede vil typisk svare til projektioner af scenepunkter, hvor scenematerialet ændrer refleksionsegenskaber, hvor scenefladen markant ændrer orientering, eller hvor illuminationen af scenen varierer (skygger). Billedkanter kan overordnet klassificeres som forårsaget af topografiske eller fotometriske ændringer. Det er ikke umiddelbart muligt at klassificere de observerede kanter i et billede efter de “begivenheder”, der har forårsaget kanterne.

Detektion af kanter er et modelproblem i billedbehandling. Metoder til kantdetektion er blevet udviklet over de sidste 30 år, og nye metoder kommer stadig til. Kantdetektion er en svær opgave, fordi det ikke er veldefineret, hvad en kant er, samt fordi man ofte ønsker uforenelige egenskaber ved en kantdetektor. Ofte vil man kræve af en kantdetektor, at den kan markere positioner, der intuitivt opfattes som kanter, at kun disse positioner markeres, samt at lokaliseringen er nøjagtig. Detektoren ønskes altså såvel robust som nøjagtig. Det kan vises, at disse ønsker ikke kan opfyldes (optimalt) samtidig. Som vi senere skal se, er det imidlertid muligt at konstruere detektorer, hvor afvejningen mellem robusthed og nøjagtighed kan specificeres direkte ved en parameter.

### 10.1 Kantmodeller

Når vi taler om kanter i et billede og vil udvikle en metode til detektion af disse, er det vigtigt eksplicit at formulere sin model af en kant. Typisk er modeller af kanter 1-dimensionale. Dette svarer til at betragte et snit af intensitetsfladen i intensitetsgradientens retning. Formålet med antagelsen er at simplificere modellen. Antagelsen implicerer at kanter i et billede er lokalt retlinede. Dette er naturligvis ikke altid tilfældet. Mange kantdetektorer har derfor også problemer med at lokalisere kanter, når disse krummer som ved et hjørne. I det følgende antages at kanterne er lokalt retlinede. Vi behøver derfor kun at formulere modeller af kanter for signaler. Problemet

er således at formulere en enkel men tilstrækkelig generel model for, hvorledes intensitetsværdien ændrer sig i en omegn af kanten. Antag at en endimensionel kant med amplitude  $A$  er lokaliseret i  $x_0$ . Intensitetsfladens opførsel i en lille omegn om  $x_0$  kan da modelleres ved:

$$E(x) = Au(x - x_0) + B \quad (10.1)$$

hvor funktionen  $u(x)$  angiver kantmodellen translateret således, at kanten er lokaliseret i  $x = 0$  og skaleret til intensitetsintervallet  $[0;1]$ . Den mest brugte model svarer til antagelsen at ændringen i intensitetsværdi sker diskontinuert som ved et trappetrin. Modellen kaldes derfor for trappetrinsmodellen

$$u(x) = \begin{cases} 0 & \text{hvis } x < 0 \\ 1 & \text{hvis } x \geq 0 \end{cases} \quad (10.2)$$

Typisk antages (som i ovenstående ligning) endvidere at intensitetsværdierne på hver side af kanten er lokalt konstante. Den primære fordel ved modellen er at den er simpel. Modellen har ingen parametre. Dette har betydning ved en teoretisk udledning af en optimal kantdetektor. I en anden kantmodel antages, at intensitetsændringen over kanten kan beskrives ved en rampe.

$$u(x) = \begin{cases} 0 & \text{hvis } x \leq -t \\ \frac{1}{2}(1 + \frac{x}{t}) & \text{hvis } -t < x < t \\ 1 & \text{hvis } x \geq t \end{cases} \quad (10.3)$$

Denne kantmodel er mere kompliceret idet dennes udstrækning ( $2t$ ) (og dermed rampens hældningen  $\frac{1}{2t}$ ) har betydning ved en detektion. En tredje kantmodel kan være en parametriseret arcustangens funktion. I denne model angiver parameteren  $\tau$  hældningen af kantprofilen i kantpunktet.

$$u(x) = \arctan(\tau x) \quad (10.4)$$

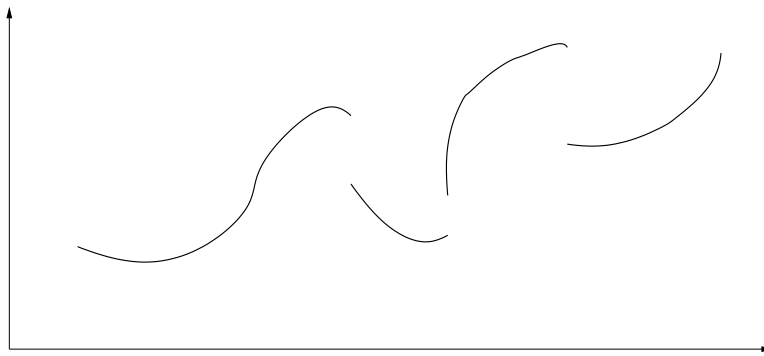
En fjerde model er:

$$u(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{t^2}{\sigma^2}\right) dt \quad (10.5)$$

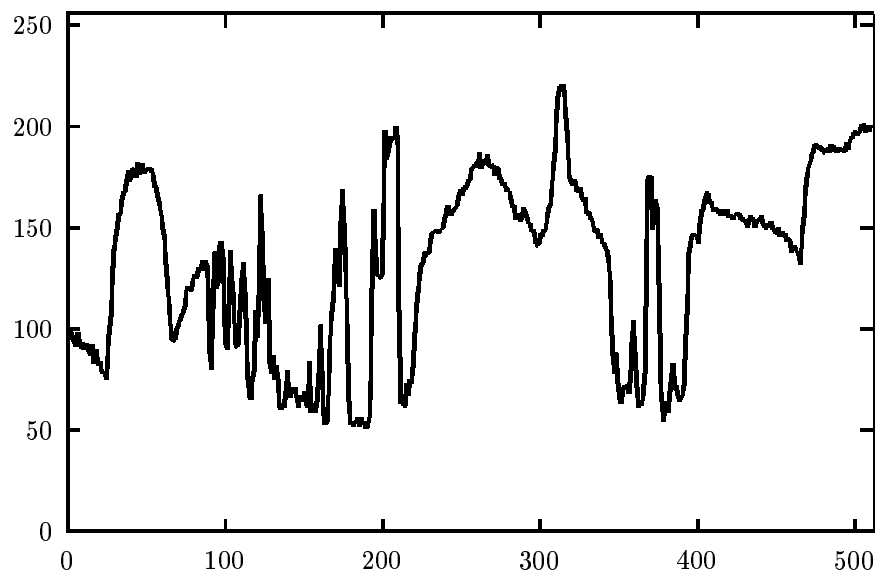
Som eksempel på en mere generel model af en kant er en metode at modellere intensitetsfladen på "højre" hhv. "venstre" side af kanten separat f.eks. ved brug af anden eller tredje ordens polynomier. I sidstnævnte tilfælde vil modellen af en kant således involvere 8 parametre, og vil kunne modellere kanter som i nedenstående illustration. Hvilken kantmodel man vælger må naturligvis afhænge af hvorledes kantprofilerne i de billeder man ønsker at behandle ser ud, samt af formålet med kantdetektionen. Generelt vil et billede ikke indeholde kanter med kun én profilform.

Det er som regel en fordel at vælge en forholdsvis generel model, der ikke indeholder for mange parametre. Problemet er, at hvis man opererer med flere kantmodeller, kan det være svært at afgøre hvilken model, der i en givet situation skal anvendes. En metode er at prøve dem alle og vælge den simpleste, der beskriver data bedst. Dette er oplagt meget beregningskrævende. Endelig kan man undersøge hvilke kantprofiler, der faktisk forekommer i billeder. Nedenstående billede viser intensitetsvariationen i skanlinie nummer 300 i billedet af *Lena*. Som det ses, kan antagelsen om en bestemt af de ovennævnte kantmodeller diskuteres. Større statistiske undersøgelser peger dog på at trappetrinsmodellen er en rimelig model.





Figur 10.1: Illustration af nogle af de mere komplekse kanter, som kan forekomme i billeder.



Figur 10.2: En skanlinie taget fra et portrætbillede

## 10.2 Støjproblemet

Nogle teoretisk specificerede kantoperatorer gør ikke klart, hvorvidt operatoren forudsættes anvendt på støjfri billeder, eller hvilke antagelser der gøres om en eventuel støj. Sædvanligvis indeholder de fleste billeder støj. Denne kan være tilført billedet under optagelsen eller ved transmission af billedet. Det er også muligt at støjen kan være tilført inden billedet er optaget, f.eks. ved at lysstrålerne har passeret gennem et inhomogent medie før de registreres af sensoren.

Uformelt beskriver kanterne i et billede de punkter, hvor intensitetsfunktionen markant ændrer værdi. I disse punkter må intensitetsgradienten derfor være numerisk stor. Dette indikerer at detektion af kanter til en vis grad afhænger af målinger af de partielle afledede af intensitetsfunktionen.

Betragt et (kontinuert) signal  $f(x)$ , og antag at det observerede signal  $g(x)$  fremkommer af  $f(x)$  ved addition af et støjled  $\eta(x)$ , der her approximeres ved en højfrekvent sinusbølge med lille amplitude.

$$g(x) = f(x) + \beta \sin(\omega x) \quad (10.6)$$

Ved differentiation af  $g(x)$  ses, at selv om støjens amplitude er lille, vil støjledet have en dominerende indflydelse på den afledede, fordi  $\omega$  er stor.

$$g'(x) = f'(x) + \beta\omega \cos(\omega x) \quad (10.7)$$

*Differentiation er således en ustabil operation.* Ved højere ordens afledede øges graden af ustabilitet. Da kantdetektion jvf. den foregående diskussion er knyttet til måling af partielle afledede ses, at også kantdetektion er en ustabil operation. Det er ofte nødvendigt at foretage en støjfiltrering før (eller samtidig med) anvendelsen af differentialoperatorer. Som vi tidligere har set kan støjfiltrering foretages ved foldning med et lavpas-filter, dvs. en integration. En sådan integration kan sikre stabiliteten.

Sædvanligvis antages at støjen er ukorreleret. I modsat fald vil det være nødvendigt at foretage en støjfiltrering, der er konstrueret til at fjerne netop den observerede støj. Det antages som regel også, at støjen har middelværdi nul. Fordelingsfunktionen for støjen er afhængig af den process, hvorved støjen er tilført billedet. En rimelig og generel antagelse er, at støjen er normalfordelt. Dette svarer til at støjen er tilført billedet ved addition af mange små uafhængige støjbidrag, hvis individuelle fordelinger er vilkårlig.

I forbindelse med differentiation skal man yderligere være opmærksom på, at der ved diskretisering af billedet er elimineret høje frekvenser i fourierspektret for det "oprindelige" kontinuerte billede. Yderligere er billedet kvantificeret til typisk 256 forskellige intensitetsværdier. Sidstnævnte forhold gør det problematisk at anvende en streng matematisk definition af en kant som en diskontinuitet (nemlig at grænseværdierne fra højre hhv. venstre er forskellige). Endelig kan der være tale om egentlige kvantificeringsfejl.

## 10.3 Numeriske metoder til måling af afledede

I det følgende skal vi betragte nogle simple metoder til måling af afledede for en diskrete endimensionale funktion  $f$ . Lad  $h$  være et positivt heltal. De forlæns, baglæns og centrale differens approximationer med støtte  $h$  til den første ordens afledede af  $f(x)$  er defineret ved:

$$f_x^{\text{for}}(x) = \frac{f(x+h) - f(x)}{h} \quad (10.8)$$

$$f_x^{\text{bag}}(x) = \frac{f(x) - f(x-h)}{h} \quad (10.9)$$

$$f_x^{\text{cen}}(x) = \frac{f(x+h) - f(x-h)}{2h} \quad (10.10)$$

hvilket fås af Taylor-rækkeudviklingerne for den til  $f(x)$  svarende kontinuerte funktion  $u(x)$ :

$$u(x+h) = u(x) + hu'(x) + \frac{h^2 u''(x)}{2} + \frac{h^3 u'''(x)}{6} + \dots \quad (10.11)$$

$$u(x-h) = u(x) - hu'(x) + \frac{h^2 u''(x)}{2} - \frac{h^3 u'''(x)}{6} + \dots \quad (10.12)$$

$$u(x+h) - u(x-h) = 2hu'(x) + 2\frac{h^3 u'''(x)}{6} + O(h^4) \quad (10.13)$$

Det ses, at forlæns og baglæns approximationerne tilnærmer de kontinuerte afledede med en fejl af orden  $O(h)$  og at fejlen ved den centrale approximation er af orden  $O(h^2)$ . Ligeledes ses, at den centrale approximation er lig middelværdien af den baglæns og den forlæns approximation. Sædvanligt benyttes den centrale approximation, fordi denne giver symmetrisk svar ved anvendelse på en ideel trappetrins-kant. Nøjagtigheden af approximationen er afhængig af støttestørrelsen  $h$ . Var  $f(x)$  kontinuert, ville approximationen konvergere mod den afledede for  $h$  gående mod 0. I det diskrete tilfælde kan vi ikke vælge  $h$  mindre end 1. Dette vil være det optimale valg under den antagelse, at vi faktisk ønsker af måle den afledede af det observerede signal  $f(x)$ .

I praksis er kontrastovergangen ved en kant sjældet diskontinuert. Ofte kan denne beskrives ved foldningen af en (ukendt) punktspredningsfunktion med en ideel trappertinskant (eksemplificeret ved et billede optaget med dårlig fokusering på scenen). Desuden vil signalet/billedet ofte indeholde støj. Endelig er billedet diskretiseret og billedværdierne kvantificeret. Det er derfor nødvendigt at forholde sig til den *skala*, hvorover intensitetsændringen sker, samt til mængden af støj. Begge forhold relaterer til opløsningen i gråtoneværdier.

Vi er som regel interesseret i at måle den afledede af det "oprindelige" signal, og ikke af det observerede signal. Ved direkte anvendelse af ovenstående differensapproximationer på signalet kan bestemmelsen blive unøjagtig eller helt forkert. En simpel metode til at øge robustheden (men ikke nøjagtigheden) af målingen er at øge støttestørrelsen  $h$ . En anden metode er at støjfiltrere signalet før eller evt. samtidig med anvendelsen af differensapproximationen. Anvendes eksempelvis den centrale differensapproximation med  $h$  lig 1, på en boxfiltrering med støtte 3 af signalet, fås den samlede approximation til:

$$f_x^{\text{boxC}} = \frac{f(x+2) + f(x+1) - f(x-1) - f(x-2)}{6} \quad (10.14)$$

Generelt vil filtret blive mere robust jo kraftigere de høje frekvenser i  $g(x)$  bliver dæmpet ved lavpassfiltreringen. Dette sker på bekostning af, at det lokale maksima i filtersvaret for en kant bliver mere afrundet (dårligere lokaliseret) og vil få mindre amplitude. Generelt vil det ikke være muligt at konstruere noget filter til måling af afledede, som både er robust og i stand til at lokalisere kanter nøjagtigt. Vi skal senere se nærmere på dette forhold. Det vel nok sværeste problem i kantdetektion er at bestemme det støjfilter, der er optimalt både mht. robusthed og lokaliseringssevnen. Udledningen af et sådant filter afhænger naturligvis af mere præcise definitioner for robusthed og lokaliseringssevne. I afsnit 10.7 skal vi se et eksempel på et sådant optimalt filter.

## 10.4 Beudet operatorer

I dette afsnit beskrives en mere generel metode til udledning af filtre for differensapproximationer til de partielle afledede af et billede. Filtrene er todimensionale og relativt enkle at udlede. I modsætning til differensapproximationerne er det muligt at angive et mål (støttestørrelsen) for robustheden af filtrene.

Beudet operatorene af  $N$ 'te orden med kvadratisk støtte  $(2M + 1) \times (2M + 1)$  er defineret som de lineære positionsinvariante operatorer, der, foldet med billedet, svarer til det bedste fit (i mindste kvadraters betydning) til intensitetsfladen med et  $n$ 'te ordens polynomie. Anderledes sagt: For hver punkt  $P$  i billedet indlægger vi et *lokalt koordinatsystem*  $(P, x, y)$ , og tilnærmer intensitetsfladen  $E(x, y)$  med et polynomie  $z(x, y)$  defineret ved:

$$\begin{aligned} z(x, y) &= \sum_{i,j=0}^{\infty} \frac{\partial^{i+j}}{\partial x^i \partial y^j} E(x, y) x^i y^j \\ &= a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 + \dots \end{aligned} \quad (10.15)$$

Den reelle værdi  $a_0$  tilnærmer således intensiteten i  $P$ . Til den  $i$ 'te koefficient  $a_i$  svarer en Beudet operator, der foldet med intensitetsfunktionen i en omegn om  $P$ , giver værdien  $a_i$ . Opgaven er at bestemme  $a$ 'erne i Taylor rækkeudviklingen for  $E(x, y)$ . Hvert  $a_i$  beskriver et filter  $A_i$ . Bestemmelsen af  $a_0$  vil eksempelvis give os et filter til estimation af intensitetsværdien i  $(x, y) = (0, 0)$

For  $N$ 'te ordens operatorer afskæres rækken efter det  $N$ 'te ordens led. Vi begår således en trunkeringsfejl. I det følgende ignoreres denne fejl. De søgte koefficienter bestemmes ved at betragte kvadratafgigelsen (kvadratfejlen)  $e^2$  givet ved:

$$e^2 = \sum_{x=-M}^M \sum_{y=-M}^M [E(x, y) - z(x, y)]^2 \quad (10.16)$$

De  $a$ 'er, der minimerer kvadratfejlen  $e^2$ , findes lettest ved at opskrive udtrykket på matriceform. For at lette notationen i opstillingerne nummereres hvert punkt i støtteområdet med et tal  $\in [1 : n]$  hvor  $n = [2M + 1]^2$ . For såvel koordinatværdier og intensiteter angiver indices en reference til et punkt. Nummereringen foretages søjle for søjle i støtteområdet (fra lille koordinatværdi mod større).

$$\mathbf{E} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_{n-1} \\ E_n \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{\text{coeff}(N)} \end{pmatrix} \quad (10.17)$$

hvor  $\text{coeff}(N) = \frac{N(N+3)}{2} + 1$  er antallet af koefficienter i et  $N$ 'te ordens polynomium. Søjlematricen

$\mathbf{E}$  indeholder alle intensitetsværdierne inden for støtteområdet.

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 & y_1 & x_1 y_1 & x_1^2 & y_1^2 & \cdots \\ 1 & x_2 & y_2 & x_2 y_2 & x_2^2 & y_2^2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1} & y_{n-1} & x_{n-1} y_{n-1} & x_{n-1}^2 & y_{n-1}^2 & \cdots \\ 1 & x_n & y_n & x_n y_n & x_n^2 & y_n^2 & \cdots \end{pmatrix} \quad (10.18)$$

Med ovenstående definitioner kan vi nu skrive formel (2.15) som en matrixligning:

$$e^2 = \|\mathbf{E} - \mathbf{X}\mathbf{A}\|^2 \quad (10.19)$$

Den vektor  $\mathbf{A}$ , der minimerer  $e^2$ , fås ved differentiation:

$$\frac{\partial e^2}{\partial \mathbf{A}} = -2\mathbf{X}^t (\mathbf{E} - \mathbf{X}\mathbf{A}) = 0 \quad (10.20)$$

hvor  $t$  angiver transponering. Videre fås:

$$\mathbf{X}^t \mathbf{E} = \mathbf{X}^t \mathbf{X} \mathbf{A} \quad (10.21)$$

og dermed

$$\mathbf{A} = \mathbf{X}^\# \mathbf{E} = (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{E} \quad (10.22)$$

hvor matricen  $\mathbf{X}^\#$  kaldes den *pseudoinverse* (eller den generaliserede Penrose-More inverse) til matricen  $\mathbf{X}$ . Bemærk at den pseudoinverse kun indeholder talværdier. Opskrives resultatet på sædvanlig form, ses at de aktuelle værdier af  $\mathbf{A}$  fås ved foldning af funktionen  $E(x,y)$  med den (de) funktion(er), hvis matrice(r) er givet ved rækkerne i  $\mathbf{X}^\#$ .

### Eksempel

Som eksempel betragtes her udledningen af Beaudet operatorene af første orden for  $M = 1$ . I dette tilfælde er modellen givet ved:  $z(x, y) = a_0 + a_1 x + a_2 y$ . Matricen  $\mathbf{X}$  får udseendet:

$$\mathbf{X} = \begin{pmatrix} 1 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & -1 \\ 1 & -1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (10.23)$$

hvor anden og tredje søjle angiver  $x$  hhv.  $y$ -koordinatværdier. Videre fås:

$$\mathbf{X}^t \mathbf{X} = \begin{pmatrix} 9 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{pmatrix} \quad (10.24)$$

Ved matriceinvertering -og multiplikation fås at:

$$\mathbf{X}^\sharp = \frac{1}{18} \begin{pmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ -3 & 0 & 3 & -3 & 0 & 3 & -3 & 0 & 3 \\ -3 & -3 & -3 & 0 & 0 & 0 & 3 & 3 & 3 \end{pmatrix} \quad (10.25)$$

Dekomponeres denne matrice svarende til  $a_0$ ,  $a_1$  og  $a_2$  får vi:

$$\mathbf{A}_0 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \mathbf{A}_1 = \frac{1}{6} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad \mathbf{A}_2 = \frac{1}{6} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad (10.26)$$

Jo større støtten for Beaudetoperatorene vælges, jo mere robuste vil disse angive de partielle afledede (ved foldning med intensitetsfladen). Tilsvarende vil nøjagtigheden af estimerterne blive mindre. Bemærk endvidere, at Beaudetoperatorer (med samme støtte) vil være forskellige afhængigt af ordenen af det approximerende polynomie. Således er masken svarende til middelværdi-operatoren  $A_0$  for de 2. ordens operatorer med støtte  $3 \times 3$  givet ved:

$$\mathbf{A}_0 = \frac{1}{9} \begin{pmatrix} -1 & 2 & -1 \\ 2 & 5 & 2 \\ -1 & 2 & -1 \end{pmatrix} \quad (10.27)$$

Læseren opfordres til at eftervise nedenstående 2. ordens operatorer med støtte  $5 \times 5$  (brug et “matematikprogram” som f.eks. **maple**, **matlab** eller **matematica**).

$$\mathbf{A}_0 = \frac{1}{175} \begin{pmatrix} -13 & 2 & 7 & 2 & -13 \\ 2 & 17 & 22 & 17 & 2 \\ 7 & 22 & 27 & 22 & 7 \\ 2 & 17 & 22 & 17 & 2 \\ -13 & 2 & 7 & 2 & -13 \end{pmatrix} \quad \mathbf{A}_1 = \frac{1}{50} \begin{pmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{pmatrix} \quad (10.28)$$

samt

$$\mathbf{A}_2 = \frac{1}{50} \begin{pmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{pmatrix} \quad \mathbf{A}_4 = \frac{1}{35} \begin{pmatrix} 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \end{pmatrix} \quad (10.29)$$

og

$$\mathbf{A}_5 = \frac{1}{35} \begin{pmatrix} 2 & 2 & 2 & 2 & 2 \\ -1 & -1 & -1 & -1 & -1 \\ -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 2 & 2 & 2 & 2 & 2 \end{pmatrix} \quad \mathbf{A}_3 = \frac{1}{100} \begin{pmatrix} 4 & 2 & 0 & -2 & -4 \\ 2 & 1 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 1 & 2 \\ -4 & -2 & 0 & 2 & 4 \end{pmatrix} \quad (10.30)$$

Beaudetoperatorerne er beskrevet i [4].

## 10.5 Detektionsmetoder

Som nævnt i et foregående afsnit er detektion af kanter knyttet til måling af de partielle afledede af intensitetsfunktionen. Lad  $\nabla E = (E_x, E_y)$  være intensitetsgradienten,  $\rho = \|\nabla E\|$  denne vektors størrelse, og lad  $\phi = \arctan(E_y/E_x)$  være intensitetsgradientens retning. Bemærk at intensitetsfladens stejlhed i en vilkårlig retning  $v$  er bestemt ved:

$$E_v = E_x \cos(v) + E_y \sin(v) \quad (10.31)$$

Den simplest tænkelige detektionsmetode er at definere kantpunkterne i et billede ved de punkter  $(x, y)$  hvor  $\rho$  overstiger en vis (foruddefineret) værdi  $T$ . Dette kaldes for detektion ved grænseværdiafskæring.

$$\text{edge}(x, y) = \begin{cases} 0 & \text{hvis } \rho < T \\ 1 & \text{hvis } \rho \geq T \end{cases} \quad (10.32)$$

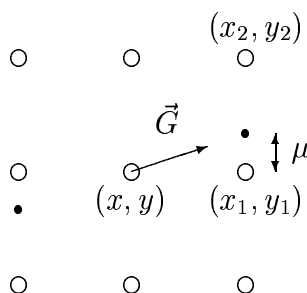
Metoden vil markere alle punkter hvor intensitetsfladen er stejlere end  $T$ . Disse punkter vil ofte udgøre fede kanter (afhængig af valget af  $T$ ), men ofte vil billedpunkter, der intuitivt klart ses som kanter, ikke markeres. Kvaliteten af metoden er derfor meget afhængig af et godt valg af  $T$ . Ofte vil det ikke være muligt at vælge  $T$ , sådan, at alle intuitivt observerede kanter markeres, og sådan at de markerede kanter er acceptabelt tynde.

Bemærk at vi ikke lagde os fast på nogen bestemt norm, da vi definerede  $\rho$ . Sædvanligvis benyttes den euclidiske norm, fordi denne er rotationssymmetrisk. I situationer hvor horisontale og vertikale kanter har særlig preference, kunne man i stedet bruge den beregningsmæssigt hurtigere maksimumsnorm. I det følgende antages at den euclidiske norm benyttes.

I forhold til ovennævnte detektionsmetode er en mere avanceret metode kun at markere kanter ved de punkter  $P$ , hvor  $\rho$  lokalt antager maksimal værdi (eng. non maximum suppression). Dette gøres ved at sammenligne  $\rho$ -værdien i  $P$  med  $\rho$ -værdierne i to punkter i en vis afstand (f.eks. 1 pixel) fra  $P$ . De to punkter skal vælges i intensitetsgradientens positive og negative retning. Punktet  $P$  markeres som kantpunkt, hvis  $\rho$  i  $P$  er større end  $\rho$  i begge de andre punkter. Dette er illustreret i figur 10.3.

Figur 10.4 illustrerer hvorledes non-maximum undertrykkelsen kan implementeres. I kodeskiten angiver  $G$  gradientstørrelsen  $\rho$ .  $G_x$  hhv.  $G_y$  angiver estimerede mål af de partielle afledede af intensiteten.

Bemærk at i ovenstående skitse angiver værdierne  $m1$  og  $m2$  interpolerede størrelser, der “måles” på randen af kvadratnettet, hvor billedpunkterne er knuder, sådan som det er illustreret i Figur 10.3. I metoden benyttes en simpel grænseværdiafskæring i afgørelsen om et kantpunkt er tilstrækkeligt markant til at skulle medtages. Denne metode resulterer ofte i, at en kant bliver opsplittet i mange korte kantstykker. I en alternativ metode, kaldet *grænseværdiafskæring med hysteresis*, er ideen at registrere alle kantpunkter, hvor gradientstørrelsen enten er over en vis nedre værdi, eller hvis kantpunktet ligger på en kant, hvor gradientstørrelsen i mindst et kantpunkt



Figur 10.3: Interpolation ved detektion af lokale maksima

```

/* do non maximum suppression in (x,y) */
#define round (1.0 - sqrt(2.0)/2.0)
if ((m = G[x][y]) > THreshold)
{
    gx = Gx[x][y]/m;
    gy = Gy[x][y]/m;

    /*point on axis*/
    x1 = (gx > 0.0) ? (int)(gx + round) : (int)(gx - round);
    y1 = (gy > 0.0) ? (int)(gy + round) : (int)(gy - round);

    /*point on diagonal*/
    x2 = (gx > 0.0) ? 1 : -1;
    y2 = (gy > 0.0) ? 1 : -1;

    gxm = fabs(gx);
    gym = fabs(gy);
    my = (gym < gxm) ? gym/gxm : ((gym > gxm) ? gxm/gym : 0.0);

    m1 = (1.0-my)*G[x+x1][y+y1] + my*G[x+x2][y+y2];
    m2 = (1.0-my)*G[x-x1][y-y1] + my*G[x-x2][y-y2];
    if ((m > m1) && (m > m2)) ``MARK LOCAL MAXIMA``
}

```

Figur 10.4: Kode til non-maximum undertrykkeslse.

er større en end øvre værdi. Metoden er overlegen i forhold til metoder, hvor der kun benyttes en grænseværdi, fordi metoden bevarer kanter, hvor gradientstørrelsen varierer meget. Ulemper ved metoden er, at den kræver angivelse af to værdier, og at den forudsætter en segmentering af kanterne i billedet. Segmenteringen kan foretages løbende, sådan at sammenhængende kantpunkter gennemløbes og markeres som besøgte. Under gennemløbet “slettes” punkter, hvor gradientstørrelsen er mindre end den nedre grænseværdi, og positionerne af de øvrige kantpunkter registreres. Desuden registreres den største gradientstørrelse. Ved afslutning af gennemløbet undersøges om denne værdi er større end den øvre grænse. I bekræftende fald markeres alle registrerede kantpunkter som accepterede. I modsat fald markeres de som forkastede.



En anden metode til detektion af kantpunkter er at betragte de anden ordens partielle afledede af  $E$ . For et signal vil ekstrema i den førsteafledede svare til en nulpunktsgennemskæring (eng. zero crossing, ofte forkortet  $zc$ ) i den andenaflædede. Problemet i det todimensionale tilfælde er primært hvilken kombination af de andenaflædede (for  $E$ ), der er fornuftigt at betragte. Mest simpelt kan kantpunkter defineres ved nulpunktsgennemskæringer af laplaceoperatoren:

$$\nabla^2 E = \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) E = M = E_{xx} + E_{yy} \quad (10.33)$$

Laplaceoperatoren er rotationssymmetrisk og lineær i de afledede, hvilket betyder at dens værdi på et billede kan beregnes ved foldning af dette med et filter. Man kunne også forestilles sig en ikke-lineære kombination af de andenaflædede, f.eks. ved måling af Gausskrumningen af intensitetsfladen  $E$ :

$$K = E_{xx}E_{yy} - E_{xy}^2 \quad (10.34)$$

En tredje mulighed kunne være at definere kantpunkter som nulpunktsgennemskæringer af den andenaflædede i retning givet ved intensitetsgradienten.

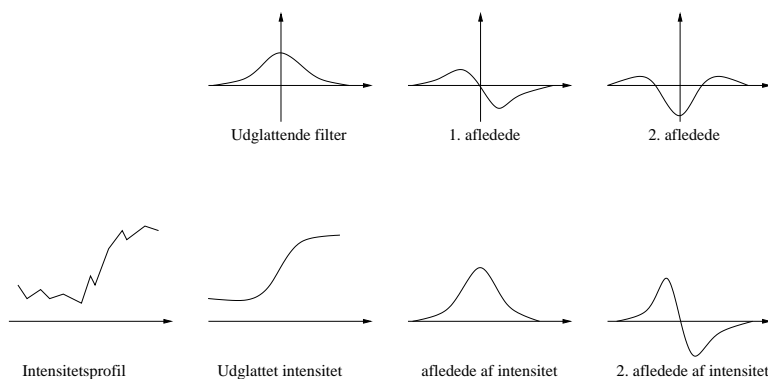
Detektion af nulpunktsgennemskæringer er relativt lette. Nulpunktsgennemskæringer i  $f(x,y)$  kan eksempelvis markeres hvis:

- $f(x-1,y) \cdot f(x+1,y) < 0$
- $f(x,y) = 0$  og  $[f(x-1,y) \neq 0 \text{ eller } f(x+1,y) \neq 0]$
- $f(x,y-1) \cdot f(x,y+1) < 0$
- $f(x,y) = 0$  og  $[f(x,y-1) \neq 0 \text{ eller } f(x,y+1) \neq 0]$

Valget mellem første- og anden ordens differentialoperatorer er tildels et spørgsmål om smag. Fordelen ved måling af førsteafledede er, at disse kan bestemmes mere stabilt end andenaflædede. Fordelen ved anvendelse af andenaflædede er, at nulpunktsgennemskæringer er lettere at detektere end lokale ekstrema. Der findes særdeles udmærkede kantdetektorer af begge typer.

For detektorer baseret på de førsteafledede benyttes filterværdien i kantpunktet som indikation af detektionsstyrken. For detektorer baseret på de andenaflædede kan stejlheden i nulpunktsgennemskæringen benyttes som mål for detektionsnøgagtigheden. Denne størrelse er klart relateret til den tredjeafledede af intensiteten. Sikkerheden af detektionen (robustheden) er derimod knyttet til arealet af den numeriske værdi af det filtrerede signal (i en omegn om kantpunktet). Dette er illustreret i Figur 10.5. Begge størrelser skal naturligvis fortolkes ud fra viden om størrelsen af det udglattende filter. En mere detaljeret diskussion af brugbarheden af disse mål falder uden for rammerne af disse noter.

Der findes kantdetektorer der ikke eksplicit involverer målinger af partielle afledede såvel som detektorer der involverer målinger af afledede af højere orden end 2. Vi skal ikke i disse noter beskrive sådanne metoder.



Figur 10.5: Detektion af kanter ved maxima hhv. nulpunktsgennemskæring af den første hhv. andenafledede af funktionen foldet med et udglattende filter.

### 10.5.1 Differentialgeometri

I ovennævnte afsnit blev to anden ordens differentialoperatorer, nemlig *Gausskrumningen*  $K$  og Laplaceoperatoren eller *Middelkrumningen*  $M$  fremhævet. Disse størrelser er ikke valgt tilfældigt, men er centrale i teorien for “kurver og flader i rummet”. Enhver flade kan i ethvert punkt  $P$  lokalt approksimeres med en paraboloid (den oskulerende paraboloid).

Antag at koordinatsystemet er lagt med begyndelsespunkt i  $P$  således at  $x$ - $y$  planet er indeholdt i tangentplanet til fladen (i  $P$ ). I dette system kan fladen fremstilles ved:

$$z = f(x, y) \quad (10.35)$$

Enhver plan indeholdende  $z$ -aksen vil skære fladen i et såkaldt *normalsnit*. Normalsnittets skæring med  $x$ - $y$  planet danner en ret linie  $(x, y) = (\lambda \cos \theta, \lambda \sin \theta)$ , hvor  $\theta$  er vinklen mellem  $x$ -aksen og linien. Normalsnittet kan derfor fremstilles ved ligningen:

$$z = G(\lambda) = f(\lambda \cos \theta, \lambda \sin \theta) \quad (10.36)$$

For at finde krumningen af fladen langs normalsnittet dannes  $G'(\lambda)$  og  $G''(\lambda)$ :

$$G'(\lambda) = f'_x(\lambda \cos \theta, \lambda \sin \theta) \cos \theta + f'_y(\lambda \cos \theta, \lambda \sin \theta) \sin \theta \quad (10.37)$$

$$G''(\lambda) = f''_{xx}(\lambda \cos \theta, \lambda \sin \theta) \cos^2 \theta + 2f''_{xy}(\lambda \cos \theta, \lambda \sin \theta) \cos \theta \sin \theta + f''_{yy}(\lambda \cos \theta, \lambda \sin \theta) \sin^2 \theta \quad (10.38)$$

Skrives

$$r = f''_{xx}(0, 0); \quad s = f''_{xy}(0, 0); \quad t = f''_{yy}(0, 0) \quad (10.39)$$

kan vi opskrive normalsnittets krumning i punktet  $P$ :

$$\kappa_\theta = G''(0) = r \cos^2 \theta + 2s \cos \theta \sin \theta + t \sin^2 \theta \quad (10.40)$$

Det ses let, at krumningen langs alle normalsnit vil stemme overens med krumningen af den oskulerende paraboloid:

$$z = \frac{1}{2}[rx^2 + 2sxy + ty^2] = \frac{1}{2}[x \ y] \begin{bmatrix} r & s \\ s & t \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2}[x \ y] \mathbf{H} \begin{bmatrix} x \\ y \end{bmatrix} \quad (10.41)$$

hvor  $\mathbf{H}$  kaldes *Hessian matricen*. Ved at rotere x-y koordinatsystemet omkring z-aksen kan man opnå at produktledet  $2sxy$  forsvinder. Herved opstår *Eulers formel*:

$$\kappa_\theta = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta \quad (10.42)$$

hvor  $\theta$  nu angiver vinklen mellem den ny x-akse og normalsnittet. Størrelserne  $\kappa_1$  og  $\kappa_2$  kaldes *hovedkrumningerne* i  $P$ , og er krumningerne for de til  $\theta = 0$  og  $\theta = \frac{\pi}{2}$  svarende normalsnit. Væsenligt er at tangentretningerne svarende til hovedkrumningerne er vinkelrette. Hovedkrumningerne bestemmes som rødderne i determinanten:  $\det(\mathbf{H} - \lambda \mathbf{I})$ .

$$\begin{vmatrix} r - \lambda & s \\ s & t - \lambda \end{vmatrix} \quad (10.43)$$

og tilfredsstiller ligningerne:

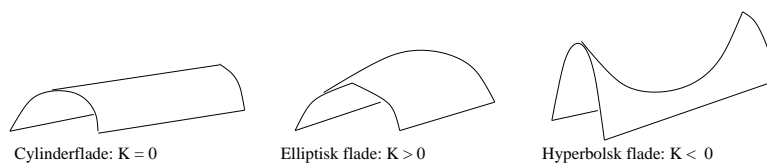
$$\kappa_1 + \kappa_2 = M = r + t, \text{ og } \kappa_1 \kappa_2 = K = rt - s^2 \quad (10.44)$$

Heraf ses, at størrelserne  $M$  og  $K$  er rotationsinvariante, dvs. at målene er uafhængige af, hvorledes det benyttede koordinatsystem er lagt. Dette kan have meget stor betydning for fortolkningen af intensitetsfladens lokale opførsel.

Antag at  $\kappa_1 \geq \kappa_2$  (ellers omdøb de to størrelser). Af Eulers formel fås:

$$\kappa_\theta = \kappa_2 + (\kappa_1 - \kappa_2) \cos^2 \theta \quad (10.45)$$

Heraf ses, at der for alle  $\theta$  gælder at:  $\kappa_1 \geq \kappa_\theta \geq \kappa_2$ . I det specielle tilfælde, hvor  $\kappa_1 = \kappa_2$  ses at  $r = t$  og  $s = 0$ , hvilket svarer til at fladen lokal har kugleform. Mere generelt kan vi, som illustreret i Figur 10.6, klassificere fladeformen efter gausskrumningen  $K$ .



Figur 10.6: Klassifikation af flader i 3D.

### Elliptisk flade: $K > 0$

Hovedkrumningerne  $\kappa_1$  og  $\kappa_2$  har samme fortegn. Den oskulerende paraboloid er derfor elliptisk, hvorfor punktet  $P$  kaldes elliptisk. Fladen ligger, ligesom paraboloiden, helt på den ene side af tangentplanet.

**Hyperbolsk flade:**  $K < 0$ 

Hovedkrumningerne  $\kappa_1$  og  $\kappa_2$  har modsat fortegn. Den oskulerende paraboloid er hyperbolsk, hvorfor punktet  $P$  kaldes hyperbolsk. Fladen ligger på begge sider af tangentplanet. Paraboloiden skærer tangentplanet langs asymptoteretningerne for paraboloiden.

**Parabolsk flade:**  $K = 0$ 

En af hovedkrumningerne (eller begge) er lig nul. Hvis  $\kappa_1 = \kappa_2 = 0$  er paraboloiden udartet til et plan (tangentplanet). Hvis kun en af hovedkrumningerne er lig nul, beskriver den oskulerende paraboloid en cylinderflade. I dette tilfælde kaldes punktet  $P$  for parabolsk.

En egenskab ved kanter defineret ved nulpunktsgennemskæringerne af gausskrumningen  $K$  er at kanterne vil adskille områder i intensitetsfladen, der er elliptiske ( $K > 0$ ) hhv. hyperbolske ( $K < 0$ ). Kanterne selv kaldes de parabolske kurver. Det har været diskuteret meget om denne klassifikation (segmentering) af intensitetsfladen er brugbar i datamatyn såvel som i en forståelse af det menneskelige syn. Diskussionen er ikke endt i noget klart svar.

## 10.6 Laplace-Gauss-filtret

Laplace-Gauss filtret er formodentlig et af de mest kendt og benyttede filtre i anvendelser, hvor 1 pixel tynde kanter af god kvalitet kræves. Filtret er givet ved laplaceoperatoren på en gauss-funktion. Kantpunkterne defineres som nulpunktsgennemskæringerne i foldningen af filtret med intensitetsbilledet.

$$LG(x, y) = \nabla^2(G_\sigma * E(x, y)) = (\nabla^2 G_\sigma) * E(x, y) \quad (10.46)$$

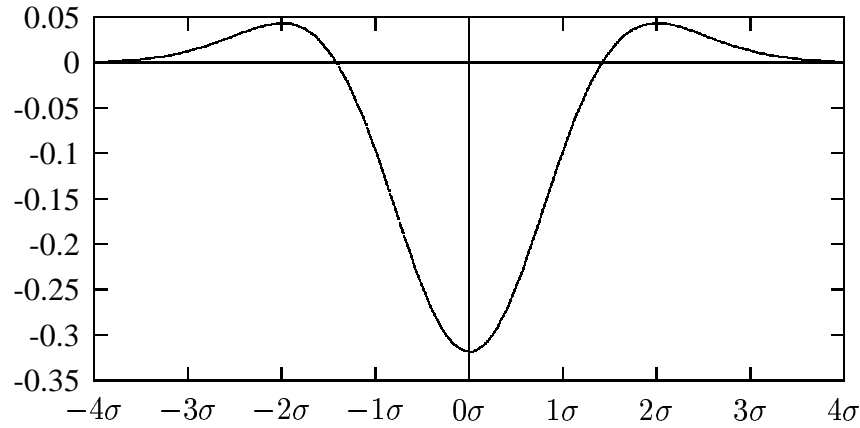
$$= \left( \frac{\partial^2}{\partial x^2} G_\sigma + \frac{\partial^2}{\partial y^2} G_\sigma \right) * E(x, y) \quad (10.47)$$

$$= \frac{1}{2\pi\sigma^2} \left[ \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} * E(x, y) \quad (10.48)$$

Filtret ses at være rotationssymmetrisk og bestemt af parametren  $\sigma$ , der er spredningen af gaussfiltret, men ofte kaldes *skalaparametren*. Den fouriertransformerede af filtret vil også være rotationssymmetrisk. Skrives  $s^2 = u^2 + v^2$  får vi den fouriertransformerede:

$$F \{ \nabla^2 G_\sigma \} (u, v) = H(s) = -s^2 e^{-\frac{\sigma^2 s^2}{2}} \quad (10.49)$$

Det ses at filtret er et båndpasfilter med ekstrema for  $s = \pm 2/\sigma^2$ . Jo større skalaparametren  $\sigma$  vælges, jo lavere frekvenser baseres kandedetektionen på. Mindre variationer i intensitetsfunktionen vil altså ikke give anledning til nogen kanter. Vælges skalaparametren meget lille baseres detektionen i stedet på de høje frekvenser i frekvensspektret. Små og meget lokale intensitetsændringer vil i dette tilfælde resultere i nulpunktsgennemskæringer i resultatet af foldningen.



Figur 10.7: Et tværsnit af det to-dimensionale Laplacian-Gaussian filter

Et tværsnit af Laplace-Gaussfiltret er vist i Figur 10.7. Filteret har en form, der kan minde om en mexikansk hat, hvorfor filtret ofte går under dette navn. Den centrale (negative) område af filtret har diameter  $w = 2\sqrt{2}\sigma$ . Filtret implementeres ofte ved en kvadratisk støtte, hvis sidelængde  $l = kw$ , hvor  $k$  er et lille tal mellem 3 og 5. Størrelsen af filtrets støtte er således afhængig af  $\sigma$ . Filtret giver som regel en god lokalisering af kanterne i et billede, hvis disse er lineære og adskilder to intensitetsflader af konstant intensitet (trappetrinsmodellen). Ved krumme kanter (f.eks. hjørner) kan afvigelsen mellem den detekterede og den korrekte lokalisering dog blive stor (op mod  $\frac{1}{2}\sigma$ ). Ukorrekt lokalisering forekommer også ved tynde aflange områder, hvis intensitet afviger fra omgivelsernes intensitet. Hvis  $w$  er større end områdets bredde  $n$ , vil afvigelsen være givet ved  $(w - n)/2$ . Disse tilfælde på fejllokalisering er illustreret i Figur 10.8. Videre diskussion kan f.eks. findes i [5].

Bemærk at Laplace-Gaussfiltret ikke er seperabelt. Filtret kan imidlertid skrives som en sum af to seperable filtre:

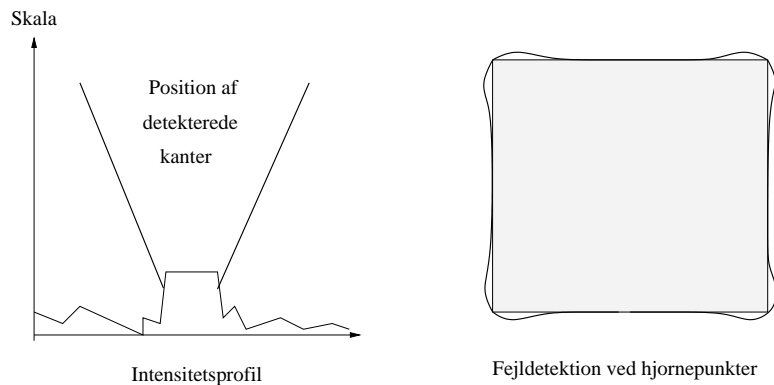
$$h_1(t) = \sqrt{K} \left[ 1 - \frac{t^2}{\sigma^2} \right] e^{-\frac{t^2}{2\sigma^2}} \quad (10.50)$$

$$h_2(t) = \sqrt{K} e^{-\frac{t^2}{2\sigma^2}} \quad (10.51)$$

$$\nabla^2 G_\sigma(x, y) = h_1(x)h_2(y) + h_2(x)h_1(y) \quad (10.52)$$

hvor konstanten  $K$  for det uskalerede filter er lig  $\frac{1}{2\pi\sigma^4}$ . Idet filtret ofte kun benyttes som forbehandling til en detektion af nulpunktsgennemskæringer, kan konstanten  $K$  imidlertid ofte vælges vilkårligt.

Kanter detekteret ved nulpunktsgennemskæringerne af foldningen med Laplace-Gauss-filtret har en række særlige egenskaber, der stammer fra de tilsvarende unikke egenskaber for Gaussfiltret og den tilhørende skalarumsteori. Opfattet som funktion af skala kan det vises at kanterne



Figur 10.8: Til venstre ses at afstanden mellem de detekterede kanter af en mørk/lys streg på en lys/mørk baggrund er lineært afhængig af  $\sigma$ . Til højre ses at Laplace-Gauss operatoren afrunder hjørner.

i et en-dimensionalt signal gradvist forsvinder ved voksende skala. Således vil der aldrig opstå nye kanter på en højere skala. Anderledes sagt vil enhver (nok så dårligt lokaliseret) kant detekteret på høj skala kunne spores ned gennem skalarummet. Herved kan præcisionen af kanten forbedres. Denne egenskab er imidlertid ikke bevaret i 2D, eller i højere dimensioner. Siden Witkin [37] i 1983 publicerede en banebrydende artikel om skalarum, er der opbygget stor viden om, hvorledes flerdimensionale funktioner opfører sig i skalarummet. Det vil imidlertid føre for vidt her at diskutere denne skalarumsteori.

## 10.7 Canny's kantoperator

Canny [8] analyserede i 1986 de krav og ønsker, som man naturligt kan stille til en kantdetektor, og opstillede en funktional, hvorved kvaliteten af enhver detektorprofil kan evalueres. Ved maksimering af funktionalen (under brug af variationsregning) kan en optimal detektor udledes. Tilsvarende analyse kan anvendes for andre detektorer (f.eks. liniedetektorer). Metoden er derfor særdeles anvendelig. Desværre er det ikke muligt i dette kursus at udnytte variationsregningens velsignelser. Vi vil derfor nøjes med at formulere Cannys antagelser og gengive hans resultater. Interesserede læsere henvises til [8, 9] for grundigere studier.

Som mange andre metoder analyserer Canny detektion af trappetrinskanter i en-dimensionale signaler. Udvidelse til to dimensioner opnås ved multiplikation med et ortogonalt udglatningsfilter. Ud over trappetrinsmodellen antager Canny, at kanten er isoleret, dvs. vel separeret fra andre kanter. Det antages, at en observerede kant kan beskrives ved summen af en ideel trappetrin af højde  $A$  og en normalfordelt hvid støj med varians  $n_0$ . Canny ønsker at detektere det lineære positionsinvariante en-dimensionale filter  $f(x)$ , der maksimerer:

1. Signal/Støjforholdet.
2. Lokaliseringen af trappetrins-kanten.

## 3. Entydigheden af kantmarkeringen.

Om filteret  $f(x)$  antages at det er kontinuert og passende mange gange differentiabelt. Ved at opstille udtryk for foldningen af  $f(x)$  med signalet, og ved opløsning i udtryk for hvorledes den ideelle trappetrinskant hhv. støjen indvirker på filtersvaret, opstiller Canny først udtryk for filterets signal-støj-forhold (S.N.R.), og for filterets lokaliseringsevne.

$$\text{S.N.R.} = \frac{A \int_{-\infty}^0 f(x) dx}{n_0 \sqrt{\int_{-\infty}^{\infty} f^2(x) dx}} = \frac{A}{n_0} \Sigma \quad (10.53)$$

hvor  $\Sigma$  angiver et signaluafhængigt mål for filterets signal-støj-forhold. Lokaliseringsevnen for filteret  $f(x)$  kan derfor måles ved:

$$\text{Lokalisering} = \frac{A}{n_0} \frac{|f'(0)|}{\sqrt{\int_{-\infty}^{\infty} f'^2(x) dx}} = \frac{A}{n_0} \Lambda \quad (10.54)$$

Her angiver  $\Lambda$  et signaluafhængigt mål for filterets evne til lokalisering af en kant. Da kvotienten  $A/n_0$  dataafhængig og uden betydning for bestemmelsen af det optimale filter  $f(x)$ , er dette derfor karakteriseret ved  $\Sigma$  og  $\Lambda$ .

Canny ønsker at optimere signal-støj-forholdet og lokaliseringsevnen samtidig og betragter derfor produktet

$$\Theta(f) = \Sigma(f) \Lambda(f) \quad (10.55)$$

Dette udtryk kaldes *performance målet for en kantoperator*. En vigtig observation er, at en skalering  $f(x/w)$  ingen indflydelse har på performance målet  $\Theta$ .

$$\Theta(x/w) = \Sigma(x/w) \Lambda(x/w) = [\sqrt{w} \Sigma(f)] \cdot \left[ \frac{1}{\sqrt{w}} \Lambda(f) \right] = \Sigma(f) \Lambda(f) = \Theta(f) \quad (10.56)$$

Heraf ses at hvis  $f(x)$  optimerer  $\Theta$ , da vil  $f(x/w)$  også optimere  $\Theta$ . Der eksisterer, med andre ord, en hel klasse af optimale filtre. Disse er karakteriseret ved deres robusthed (S.N.R.-forholdet) og ved lokaliseringsevnen. Vælges  $w$  stor øges robustheden, hvorimod lokaliseringsevnen forringes. Vi kan således opnå vilkårlig god robusthed og lokaliseringsevne blot ikke samtidig. Denne observation er af fundamental betydning, ikke kun for Canny's kantoperator, men langt mere generelt.

Maksimering af ovenstående funktional leder til et filter, der kan beskrives ved en differens af to box-filtre:

$$f_{\infty}(x) = \begin{cases} \beta & \text{for } -W < x < 0 \\ -\beta & \text{for } 0 < x < W \\ 0 & \text{ellers} \end{cases} \quad (10.57)$$

Som diskuteret tidligere er problemet ved anvendelse af dette filter, at dette har meget stor båndbredde. Hvis signalet  $I(x)$  indeholder støj, betyder dette, at filteret ofte vil give mange lokale maksima. Disse vil ofte ligge tæt, og det vil ikke være muligt at vælge et maksima (oprindende

fra kanten) frem for andre (oprindende fra støjkomponenten). For at undgå dette problem (multiple lokale maksima) er det nødvendigt at tilføje endnu et led til funktionalen.

Canny ønsker at begrænse antallet af lokale maksima, sådan at der inden for bredden af impulssvaret for en ideel trappetrinskant kun forekommer et maksimum. Vil tilføjelse af dette krav fremkommer funktionalen:

$$\Theta(f) = \frac{\int_{-\infty}^0 f(x) dx}{\sqrt{\int_{-\infty}^{\infty} f^2(x) dx}} \cdot \frac{|f'(0)|}{\sqrt{\int_{-\infty}^{\infty} f'^2(x) dx}} \cdot \frac{\sqrt{\int_{-\infty}^{\infty} f'^2(x) dx}}{\sqrt{\int_{-\infty}^{\infty} f''^2(x) dx}} \quad (10.58)$$

Ved brug af (ikke triviell) variationsregning finder Canny, at det søgte filter har formen:

$$f(x) = e^{\alpha x} [a_1 \sin \omega x + a_2 \cos \omega x] + e^{-\alpha x} [a_3 \sin \omega x + a_4 \cos \omega x] + c \quad (10.59)$$

hvor de fire konstanter  $a_1$  til  $a_4$  er bestemt af bibetingelsen  $f'(0) = s$  og af grænsebetingelserne:

$$f(0) = 0 \quad f(-W) = 0 \quad f'(-W) = 0 \quad (10.60)$$

og hvor  $W$  er støtteradius for filteret. Filteret er således ikke bestemt fuldstændigt, men parametriseret af  $\alpha$ ,  $\omega$ ,  $c$  og  $s$ . Disse parametre bør bestemmes således, at  $f(x)$  ekstremisere funktionalen (betragtet som funktion af disse fire parametre). Canny opgiver at løse dette problem analytisk og bestemmer i stedet det optimale filter ved stokastisk optimering.

Et væsentligt resultat ved Canny's analyse er, at der, som vi så tidligere, findes en hel klasse af optimale filtre. Disse er bestemt ved at have konstant produkt af lokaliseringsevne  $\Lambda$  og robusthed  $\Sigma$ . Det være muligt at opnå vilkårlig god lokaliseringsevne eller vilkårlig god robusthed, kun kan dette ikke opnås samtidig. *Nøgletallene*  $\Sigma\Lambda$  hhv.  $\xi$  for et kantfilter vil ligge i intervallerne  $[0 : \infty]$  hhv.  $[0 : 1]$ . For Canny's optimale filter (bestemt ved stokastisk optimering) er nøgletallene:

$$\text{Canny:} \quad \Sigma \cdot \Lambda = 1.12 \quad \text{og} \quad \xi = 0.58 \quad (10.61)$$

Canny observerer, at det optimale filter har en form, der er meget lig formen af den førsteafledede af et gaussfilter.

$$G'(x) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (10.62)$$

Canny vælger på dette løse grundlag at erstatte sit optimalt udledte filter med den førsteafledede af gaussfiltret, men bemærker dog at gaussfiltretes performance mål  $\Lambda\Sigma$  og dets evne  $\xi$  til separering af nabomaksima er dårligere end for det optimale filter. *Nøgletallene* for den afledede af gaussfunktionen er:

$$G': \quad \Sigma \cdot \Lambda = 0.92 \quad \text{og} \quad \xi = 0.51 \quad (10.63)$$

Det er let at beregne *nøgletallene* for en række andre mulige kantdetektorer, f.eks.:

$$\frac{\partial}{\partial x} \left( \frac{1}{1+\alpha x^2} \right) \quad \Sigma \cdot \Lambda = 1.47 \quad \text{og} \quad \xi = 0.41 \quad (10.64)$$

$$\frac{\partial}{\partial x} \left( \frac{1}{\cosh(\alpha x)} \right) \quad \Sigma \cdot \Lambda = 1.27 \quad \text{og} \quad \xi = 0.48 \quad (10.65)$$



Den sædvanlige implementation af Canny's kantoperator er at benytte  $G'(x) \cdot G(y)$  hhv.  $G(x) \cdot G'(y)$  til beregning af kantgraden horisontalt hhv. vertikalt. Ud fra disse størrelser beregnes "gradientstørrelse" -og orientering, og kantpunkter detekteres som lokale ekstrema af "gradientstørrelsen" i "gradientretningen" (Se tidligere afsnit om non-maximum-suppression). Visse implementationer benytter derefter en grænseværdiafskæring med hysteresese, mens andre benytter en mere simpel grænseværdiafskæring.

## 10.8 Deriche's kantoperator

Deriche [9] tager i sin videreudvikling af Canny's kantoperator udgangspunkt i filterformlen (10.59). Canny opgav at finde et analytisk udtryk for det optimale filter. Hovedideen i Deriche's arbejde er at erstatte visse af disse randbetingelserne svarende til antagelsen om et filter med uendelig støtte, altså:

$$f(0) = 0 \quad f(-\infty) = 0 \quad f'(-\infty) = 0 \quad |f'(0)| = s \quad (10.66)$$

Ved indsættelse kan det vises at disse betingelser (i hele det reelle område) leder til filtret:

$$f(x) = -ce^{-\alpha \cdot |x|} \cdot \sin \omega x \quad (10.67)$$

hvor  $c$ ,  $\alpha$  og  $\omega$  er reelle positive konstanter. Ved udregning af de integraler, som indgår i Canny's *performance index* fås:

$$\Lambda = (2\alpha)^{\frac{1}{2}} \quad (10.68)$$

$$\Sigma = \left[ \frac{2\alpha}{\alpha^2 + \omega^2} \right]^{\frac{1}{2}} \quad (10.69)$$

$$\xi = \left[ \frac{\alpha^2 + \omega^2}{5\alpha^2 + \omega^2} \right]^{\frac{1}{2}} \quad (10.70)$$

$$\Lambda \cdot \Sigma = \frac{2\alpha}{\sqrt{\alpha^2 + \omega^2}} \quad (10.71)$$

Lad nu  $\alpha = m \cdot \omega$ , og lad os undersøge performance for Deriche's kantoperator ved forskellige valg af  $m$ . Betragt først valget  $m = 1$ . I dette tilfælde fås nøgletallene:

$$m = 1 : \quad \Lambda \Sigma = \sqrt{2} \approx 1.414 \quad \xi = 0.58 \quad (10.72)$$

Det ses, at for  $m = 1$  er separationen af lokale maksima den samme som for Canny's filter. Derimod har Deriche's filter en performance, som er ca. 25 % bedre end Canny's filter. Vælges  $m \ll 1$  fås et filter, hvor separationen af lokale maksima er størst (nemlig 1.0). her er performancemålet  $\Sigma \Lambda$  imidlertid dårligt (nemlig  $2m$ , der er lille da  $m$  er lille). Den maksimale performance fås ved valg af  $m$  meget større end 1:

$$m \gg 1 : \quad \Lambda \Sigma \approx 2.0 \quad \xi = \frac{\sqrt{5}}{5} \approx 0.45 \quad (10.73)$$

Det ses, at performance for dette valg af  $m$  er omkring 79 % bedre end Canny's operator, hvorimod filtrets evne til separation af lokale maksima er lidt ringere end for Canny's operator. Når vi i det følgende refererer til Deriche's kantoperator, forstås det filter hvor konstanten  $m$  er valgt meget stor. Før vi går videre med at beskrive, hvorledes filtret kan implementeres, skal vi dog først se en simplifikation af filtret. Bemærk at for stort  $m$  vil  $\sin \omega x \approx \omega x$ . Dette leder til filtret:

$$g(x) = -cx e^{-\alpha|x|} \quad (10.74)$$

Det simplificerede filter har naturligvis samme performance som det forrige udledte filter. Deriche's filter er derfor essentielt parametriseret ved kun en parameter  $\alpha$ . Denne parameter har omtrent samme funktion som  $1/\sigma$  i den førsteafledede af gaussfiltret. Øges/mindskes  $\alpha$  øges/mindskes lokaliseringsevnen og robustheden mindskes/øges.

Deriche's kantfilter er et af de bedste filtre til detektion af de kanter, der opfylder de grundlæggende antagelser, specielt at disse er vel separerede. Årsagen til filterets gode performance er dets lange hale, der tillader effektiv støjfiltrering. Dette er imidlertid også filterets svaghed. Hvis kanterne ligger tæt i et billede vil filtersvaret i et kantpunkt være kraftigt påvirket af de omkringliggende strukturer. Dette kan resultere i en dårligere detektion end man ellers kunne forvente.

Da Deriche's filter har uendelig støtte, implementeres dette normalt ikke ved FIR-filtrering (foldning), men ved IIR-filtrering (rekursiv filtrering). Denne metode introduceres ultrakort nedenstående. Desværre kræver beregning af filterkoefficienterne et grundigt kendskab til z-transformationen og dens anvendelser, som ikke er omfattet i disse noter. For en beskrivelse af hvorledes Deriche's filter er implementeret i praksis henvises derfor til [9].

## IIR-filtrering

Ved IIR-filtrering (Infinite Impulse Response filtering) benyttes *kausale filtre*. Vi opfatter hver række i et billede som en sekvens af målinger opsamlet en efter en, således at vi til et givet tidspunkt (en given pixel) kun har kendskab til de forudgående målinger (pixels med mindre koordinat). Ved en løbende filtrering vil alle pixels med koordinat mindre end den aktuelle kunne indgå i beregningen. Anderledes sagt er filterkoefficienterne  $f^+(n)$  lig 0 for  $n > 0$ . Ønsker vi at "simulere" filtrering med et ikke-kausalt filter  $f(x)$ , opdeles dette i to kausale filtre  $f^+(n)$  og  $f^-(n)$  sådan at deres sum giver  $f(x)$ .

Fordelen ved opdeling af et filter i kausale delfiltre er irrelevant, hvis filtret vides at have lille støtte, idet en traditionel implementering da vil være hurtig. Hvis filterstøtten er stor, er det i visse tilfælde muligt at implementere hvert af de kausale delfiltre sådan, at filtreringen vil være *meget* hurtig. Vi skal ikke her beskrive kravene til, at implementationen kan foretages. Givet at betingelserne er opfyldt vil filtreringen kunne foretages efter følgende skema :

$$y^+(t) = a_{11} * x(t-1) + a_{21} * x(t-2) \cdots a_{n1} * x(t-n) \quad (10.75)$$

$$+b_{11} * y(t-1) + b_{21} * y(t-2) \cdots a_{m1} * y(t-m) \quad (10.76)$$

$$y^-(t) = a_{12} * x(t+1) + a_{22} * x(t+2) \cdots a_{n2} * x(t+n) \quad (10.77)$$

$$+b_{12} * y(t+1) + b_{22} * y(t+2) \cdots a_{m2} * y(t+m) \quad (10.78)$$

$$y(t) = y^+(t) + y^-(t) \quad (10.79)$$

hvor  $a$ 'erne og  $b$ 'erne er konstanter udledt fra formen af de to kausale filtre,  $n$  og  $m$  er to små heltallige konstanter. Funktionen  $x(t)$  er inddata-signalet og  $y(t)$  er uddata-signalet. Generelt gælder, at jo mere kompliceret filterformen er, jo større skal  $n$  og  $m$  være. For de filtre vi taler om er det som regel tilstrækkeligt at lade  $n = m = 2$ . Foretages filtreringen som i ovenstående skema kaldes denne for *rekursiv*. Af ovenstående rekursive filtrering ses, at IIR-filtreringen foretages med en filterstøtte på  $n$  ( $m$ ) pixels. Til gengæld udnyttes information om samtlige tidligere "besøgte" pixels ved reference til filtersvaret  $y$  for  $t-1$ ,  $t-2$  osv. Det er oplagt, at for små  $n$  og  $m$  vil en filtrering efter ovenstående skema være overordentlig hurtig.

## 10.9 Detektion af linier

Svarende til detektion af kanter kan man også ønske at detektere linier (eng. ridges). En linie er en lokalt retlignet/parabolsk struktur, der er lysere eller mørkere end baggrunden. Betragt et snit af intensitetsfunktionen vinkelret på linieretningen og et punkt  $P$  midt på snittet. Lad  $P$  udgøre origo for et lokalt koordinatsystem. Svarende til trappetrinsmodellen for en kant benytter vi her modellen:

$$l(x) = \begin{cases} 1 & \text{hvis } |x| \leq W \\ 0 & \text{ellers} \end{cases} \quad (10.80)$$

hvor  $2W$  er bredden af linien. En prototype på et filter til markering af punktet  $P$  som et liniepunkt er den andenafledede af en Gaussfunktion:

$$\frac{\partial^2 G_\sigma}{\partial x^2} = \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{\sigma^4} (x^2 - \sigma^2) e^{-\frac{x^2}{2\sigma^2}} \quad (10.81)$$

Denne funktion har et centralt negativt område og er positiv for  $|x| > \sigma$ . Vælges  $\sigma = W$  vil (den en-dimensionale) foldning af filteret med linieprofilen give et ekstremum i punktet  $P$ . På begge sider af  $P$  vil der i afstanden  $2W$  være ekstrema af modsat fortegn. Disse svarer til de positive maksima af filteret og vil have væsentlig mindre numerisk foldningsværdi. For at undgå fejldetektion af disse (side-lobes) som liniepunkter er det nødvendigt at foretage ekstremumsdetektionen over et område med radius  $> 2W$ .

Der eksisterer naturligvis en lang række andre muligheder for liniefiltre. Svarende til Canny's kantoperator har Canny udviklet et optimalt liniefilter. Dette ligner - men er langt fra identisk - med ovenstående. Et ofte benyttet alternativ til ovenstående filter er DOG-filteret (Difference Of Gaussians):

$$\text{DOG}(x) = G_{r\sigma} - G_\sigma \quad (10.82)$$

hvor  $r > 1$ . Hvis DOG-filteret skaleres passende vil det konvergere mod ovennævnte filter for  $r$  gående mod 1.

Uanset valget af profil-filter er det nødvendigt at udvide dette til to dimensioner ved praktisk anvendelse. Dette gøres ved at multiplicere filteret med et ortogonalt lav-pas filter, f.eks. et Gauss-filter. I det lokale koordinatsystem får vi derfor filteret:

$$L_{\sigma}(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{\sigma^4} (x^2 - \sigma^2) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (10.83)$$

hvor vi for nemhed har valgt samme spredning  $\sigma$  langs begge akser.

I praksis vil linierne i et billede ikke være orienteret efter de to koordinatakser, men kan have vilkårlig orientering  $\theta$ . Denne vinkel kan for hver pixel estimeres som beskrevet i afsnit 3.4.1 om anisotrop filtrering) ved at analysere kovariansmatricen for gradienterne i en rimelig stor omegn om punktet. Det er vigtigt her at gradientretningerne beregnes modulo  $\pi$ , således at modsat rettede gradienter indgår som ens.

Givet den lokale orientering kan filteret roteres og filterværdien beregnes. I praksis er det ofte tilstrækkeligt at operere med en lille bank af (f.eks. 8) forudberegnete filtre. Banken indiceres ved kvantificering af  $\theta$ .

Som det fremgår er liniedetektion væsentlig vanskeligere end kantdetektion, fordi det ikke er tilstrækkeligt at filtrere langs koordinatakserne. Væsentlig er også, at hvis linier med meget forskellig tykkelse  $W$  skal kunne detekteres, da er det nødvendigt at benytte flere filtre. Kombination af flere af sådanne filtersvar er ikke enkel. Foretages filtreringen i (et gaussisk) skalarum er det dog muligt at normalisere de enkelte filtre således at deres filtersvar kan sammenlignes fornuftigt. Dette ligger dog uden for rammerne af disse noter.

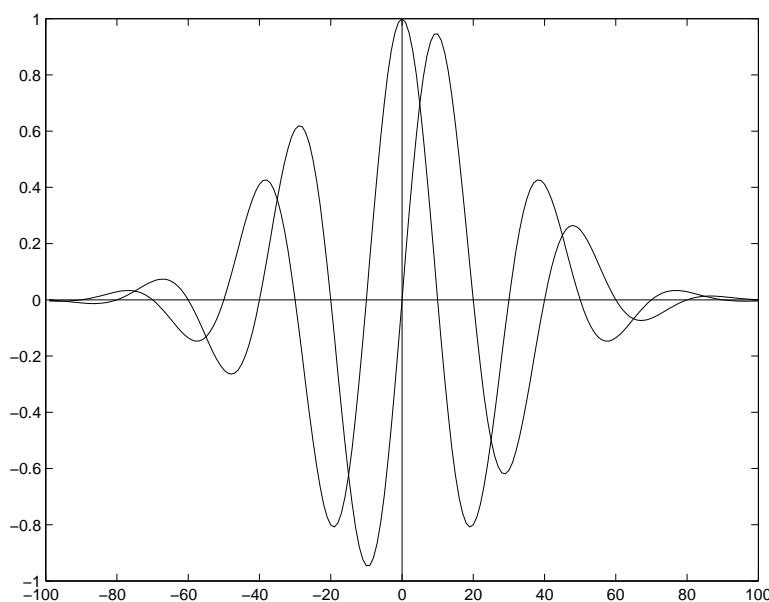
## 10.10 Kvadraturfiltre

Filtre  $f_e$  til kantdetektion hhv.  $f_l$  til liniedetektion er ulige hhv. lige. Det er nærliggende at betragte den klasse af sådanne filterpar, hvor de to filtre adskiller sig ved et 90 graders faseskift. Sådanne par vil oplagt have ens energi, hvilket betyder, at de to filtersvar kan sammenlignes. Et sådant filterpar siges at være i kvadratur.

Lad  $f_e$  og  $f_l$  være i kvadratur og lad  $f = f_l + i * f_e$  være det kombinerede (komplekse) filter. Den fouriertransformerede bliver  $F = F_l + i * F_e$ . Da  $F_l$  vil være lige og reel mens  $F_e$  vil være ulige og rent imaginær ses, at  $F$  vil være reel. Da filtrene er i kvadratur, vil  $F$  være 0 (nul) i hele det ene halvplan.

Vi kan foretage en kombineret foldning med begge filtre ved at fouriertransformere billedet, multiplicere med det kombinerede frekvensfilter, invers fouriertransformere resultatet og dekomponere dette i de to komponenter. Hvis begge foldninger er relevante, er denne metode oplagt hurtigere end separate foldninger. Ved i hver pixel at beregne fasevinklen, fås et udtryk for kantgraden versus liniegraden i punktet. Tilsvarende vil størrelsen af det komplekse kombinerede foldningsresultat angive et mål for "aktiviteten" i punktet.

Det klassiske eksempel på et kvadraturfilterpar er cosinus hhv. sinus modulerede gaussfiltre.



Figur 10.9: Et kvadratur filter par.

Sådanne filtre kaldes også *Gabor filtre*.

$$\text{gabor}_t(x) = \cos(2\pi x\omega + \pi * t) \cdot e^{-\frac{1}{2}\frac{x^2}{\sigma^2}} \quad (10.84)$$

hvor  $t = [0, 1]$  angiver fasen,  $\omega$  angiver frekvensen og  $\sigma$  angiver lav-pas karakteristikken (skalaen). Ved brug af foldningssætningen ses, at de to filtre i frekvensområdet fremkommer ved foldning af en gauss med  $(\delta(u+w) + \delta(u-w))$  hhv.  $(\delta(u+w) - \delta(u-w))$ . For  $u > 0$  er de to frekvensfiltre altså ens. For  $u < 0$  har de to filtre modsat fortegn. Det kombinerede frekvensfilter vil derfor være 0 for  $u < 0$ .

Gabor filtre er ikke optimale til detektion af linier og kanter. Dette skyldes bl.a. svingningerne i filtrene. Derimod har de med stor succes være brugt til måling af tekstur ved segmentering af billeder i områder med ens tekstur.

To meget anvendte kant og liniefiltre er den første hhv. anden afledede af et gaussfilter. Disse filtre er imidlertid ikke i kvadratur. Det er let at konstruere kvadraturfiltre i frekvensområdet ved at sætte filterværdierne i det ene halvplan til nul. En approksimation til brugen af afledede af gausser kan fås f.eks. ved i det andet halvplan at summere de to frekvensfiltre. Sådanne approksimationer er desværre ikke altid gode mht. deres detektionsegenskaber. Problemet er knyttet til, hvorledes filtrene opfører sig nær nul i frekvensdomænet. Et godt liniefilter skal nærme sig nul glat som  $u^2$ , hvorimod et godt kantfilter skal nærme sig nul skarpt som  $u$ . Intet filter kan gøre begge dele lige godt.



# Kapitel 11

## Detektion af hjørner og andre interessepunkter

Detektion af stabile og vellokaliserede billedpunkter er en væsentlig operation i mange anvendelser, f.eks. sporing af punkter over tid (videoanalyse), fokusering af beregningskraften i billedsnit med megen struktur, genkendelse af objekter, navigation etc. Bemærk at kantpunkter i sig selv ikke er gode interessepunkter, fordi de generelt kun er vellokaliserede i gradientretningen. Generelt ønsker vi punkterne vellokaliserede, gerne med subpixel præcision, robuste mht. støj, ændringer i rotation i billedplan såvel som i dybden, beherskede perspektiviske projektioner, skalering, etc. Som for kantdetektion er det vanskeligt at få opfyldt disse krav samtidig. De fleste metoder giver ofte enten for mange eller for få (eller helt andre) markeringer end man intuitivt ville forvente. Dette skyldes ikke at metoderne er dårlige, men at det er svært at formulere præcise krav til en detektion, som passer på alle billeder. Der findes et meget stort antal detektionsmetoder og varianter. Nedenstående gennemgås nogle få af de mere populære.

### 11.1 Simple interessepunktsdetektorer

Moravec's interessepunktsdetektor (fra 1979) er formodentlig den tidligst alment kendte metode. Betragt et punkt og et afgrænset snit af intensitetsfladen i gradientens retning. I et kantpunkt vil variationen af intensiteten langs snittet være stor. I den ortogonale retning (langs kanten) vil variationen være lille. I et interessepunkt vil et naturligt krav være at variansen af intensiteten i begge retninger er stor. I praksis måles variansen i typisk 4 retninger (langs de to koordinataksler og langs diagonalerne), og den mindste af de fire størrelser bestemmes for hver pixel. Dernæst detekteres de punkter, hvor dette mål er lokalt maksimalt. For at begrænse antallet af markeringer foretages ofte en grænseværdiafskæring. Ud over en sådan parameter kræver metoden kun angivelse af støtteradius for variansberegningen. Visse implementationer benytter at vægte punkter langt fra det potentielle interessepunkt lavere end punkter tæt på (f.eks. ved en gaussisk vægtfunktion). Dette kan øge robustheden af detektionen. Metoden er simpel og hurtig, men giver ofte mange flere markeringer end forventet. Lokaliseringen af de markerede punkter er endvidere ofte mindre god.

En lignende, men måske teoretisk bedre funderet metode er basert på sammenligninger af korrelation. Lad  $f(x, y) = I(x, y) \circ I(x, y)$  være autokorrelation af intensiteten i en lille omegn omkring  $(x, y)$ , lad  $f_{u,v}(x, y) = I(x, y) \circ I(x + u, y + v)$  være krydkorrelationen mellem tilsvarende udsnit omkring  $(x, y)$  og  $(x + u, y + v)$ , og lad  $g(u, v) = f_{u,v}(x, y)/f(x, y)$ . Hvis  $(x, y)$  skal være et markant interessepunkt må  $g(\pm 1, \pm 1)$  være væsentlig mindre end  $g(0, 0) = 1$  (f.eks. 0.5). Hvis punktet også skal være unikt inden for støtteområdet bestemt af  $(u, v)$ , må  $g(u, v) < T$  hvor  $T$  er en lille grænseværdi (f.eks. 0.3). Metoden er interessant, fordi den bygger på et eksplicit sammenligningsmål, der også kan benyttes ved genfindning af korreponderende punkter i andre billeder (f.eks. ved tracking). Såvel støttestørrelsen som de to grænseværdier angiver størrelser, der er umiddelbart meningsfulde, hvilke gør det lettere at angive disse. Som for Moravec's operator er kvaliteten af de detekterede punkter ofte af behersket kvalitet.

## 11.2 Hjørnedetektion

Ovenstående metoder fokuserer generelt på detektion af punkter hvor intensitetsfladen udviser stor variation. I områder med kaotisk varierende intensitet er der derfor stor chance for at mange punkter markeres. Som alternativ til sådanne metoder bygger nedenstående på egenskaber, der er mere direkte knyttet til strukturen i data, i modsætning til tekstur og støj. Metoderne er oprindeligt udviklet til detektion af hjørnepunkter.

### 11.2.1 Krumningsmål

En simpel definition på et hjørnepunkt er et punkt hvor konturen gennem punktet krummer meget. Ideelt vil hjørnepunkterne være en delmængde af mængden af kantpunkter. Betragt et billedpunkt  $P$  og lad  $(P, x, y)$  være et koordinatsystem med origo i  $P$ . Lad  $f(x, y)$  være niveaukurven, der passerer gennem  $P$ , dvs kurven:  $f(x, y) = I(x, y) - I(0, 0) = 0$ . Ved brug af implicit differentiation (se afsnittet om differentialgeometri i Kapitel 10) kan vi bestemme de partielle afledede af  $y$  mht.  $x$  i termer af de partielle afledede af  $f$ . Ved substitution fås at krumningen af isointensitetskurven er:

$$\kappa = \frac{d^2y/dx^2}{(1 + (dy/dx)^2)^{\frac{3}{2}}} = \frac{f_{xx}f_y^2 - 2f_{xy}f_xf_y + f_{yy}f_x^2}{(f_x^2 + f_y^2)^{\frac{3}{2}}} \quad (11.1)$$

En anden måde at opnå samme udtryk er ved at betragte gradientretningen  $\theta$  defineret ved:  $\tan(\theta) = f_y/f_x$ . De partielle afledede af  $\theta$  er givet ved:

$$\theta_x = \frac{f_{xy}f_x - f_{xx}f_y}{f_x^2 + f_y^2} \quad (11.2)$$

$$\theta_y = \frac{f_{yy}f_x - f_{xy}f_y}{f_x^2 + f_y^2} \quad (11.3)$$

$$(11.4)$$

Ved at projicere ændringen  $(\theta_x, \theta_y)$  af gradientretningen ind på vektoren  $(-f_y, f_x)$  vinkelret på gradientretningen fås samme udtryk.



I praksis ønskes, at krumningen  $\kappa$  er numerisk stor. Punkter med lille kontrast ønskes sædvanligvis ikke markeret. Det er derfor sædvane at multiplicere det udledte udtryk med gradientstørrelsen. Herved fås hjørnemålet:

$$K = \frac{|f_{xx}f_y^2 - 2f_{xy}f_xf_y + f_{yy}f_x^2|}{f_x^2 + f_y^2} \quad (11.5)$$

Metoder, der bygger på ovenstående udtryk går under navnet *Kitchen-operatoren* [21]. I den oprindelige version benytter fortatterne Beaudet-operatorer til beregning af de partielle afledede af intensiteten. Andre metoder, f.eks. foldning med afledede af en Gauss, er naturligvis også mulige. Dette muliggør såvel detektion som verifikation af hjørnepunkter i skalarum. Efter beregning af  $K$  i alle punkter foretages detektion af lokale maxima. Det vil ofte være nødvendigt også at benytte en grænseværdiafskæring. En rimelig grænseværdi kan være vanskelig af specificere uden godt kendskab til såvel metoden som til billedet.

### 11.2.2 Egenværdianalyse

En tredje metode til hjørnedetektion bygger på en statistisk analyse af gradientvektorerne i en omegn af hvert potentielt hjørnepunkt. Som vi tidligere har set (i kapitel 3 om positionsinvariant filtrering) indeholder kovariansmatricen (se ligning (3.27)) for gradienterne information om de dominerende gradientretninger. Lad  $f(x, y)$  være intensiteten og antag at gradienterne  $(f_x(x, y), f_y(x, y))$  er beregnet. Vi vil her undlade normalisering mht. middelgradienten og betragte korrelationsmatricen (struktur tensoren):

$$C = \frac{1}{N} \begin{bmatrix} \sum f_x^2 & \sum f_x f_y \\ \sum f_x f_y & \sum f_y^2 \end{bmatrix} \quad (11.6)$$

hvor summationerne foregår over et lokalt billedusnit med  $N$  pixel. Som i kapitel 3 betragter vi nu egenværdierne  $\lambda_1 \geq \lambda_2$  for matricen  $C$ . For et område med konstant intensitet er begge egenværdier lig nul. For et område hvor intensiteten er lokalt plan, vil  $\lambda_2 = 0$ , og  $\lambda_1$  vil være lig kvadratet på gradientstørrelsen. Tilsvarende vil  $\lambda_2 = 0$  og  $\lambda_1$  være stor i et punkt der ligger på en retlinet kant med stor kontrast. I et hjørnepunkt findes to dominerende gradientretninger. Her vil  $\lambda_1 \geq \lambda_2 \gg 0$ .

Metoden er derfor at detekterer hjørnepunkter som pixels, hvor  $\lambda_2(x, y)$  er lokalt maksimal, og hvor  $\lambda_2(x, y) > T$ , hvor  $T$  er en grænseværdi. Ud over eventuelle parametre til beregning af de partielle afledede og støttestørrelsen for summationerne skal  $T$  specificeres. Dette kan helt som for de fleste andre metoder - være vanskelig. En mulighed er at analysere et histogram over de beregnede  $\lambda_2$ -værdier. Hvis brugeren har angivet et omtrentligt antal (eller procentdel) af kantpunkter, kan  $T$  findes ved et lokalt minimum eller lokalt knæk i histogrammet.

Metoden giver ofte rimelig robuste markeringer. Disse er imidlertid ikke altid godt lokaliseret. Dette afhænger dog af en lang række af implementationsdetaljer. Metoden findes i et stort antal varianter. En oplagt forbedring er at beregne  $C$  ved (f.eks. Gauss-) vægtede summationer. Pixels tæt ved  $(x, y)$  kan ekskluderes fra summationerne da gradienten er dårligt defineret i hjørnepunkter. Hastigheds-mæssigt er nedenstående variant (Harris-detektoren) af særlig interesse.

### 11.2.3 Harris-detektoren

Harris-detektoren (også kaldet Harris-Stephen- eller Plessey-detektoren) tager udgangspunkt i ovendefinerede korrelationsmatrice  $C$ . I stedet for at betragte den mindste egen værdi  $\lambda_2$  betragtes forholdet  $\lambda_2/\lambda_1$  mellem egen værdierne. Idet vi ignorerer ordningen af de to egen værdier kan vi definere en familie af hyperbler :

$$(\lambda_1 - a\lambda_2) \cdot (\lambda_1 + a\lambda_2) = b \quad (11.7)$$

med asymptoter  $\lambda_1 = a\lambda_2$  og  $\lambda_2 = -a\lambda_1$ , hvor  $0 < a < 1$  implicit specificerer afgrænsningen af områder hvor gradientorienteringen er homogen ( $b < 0$ ) hhv. inhomogen ( $b > 0$ ). Ovenstående ligning kan let omskrives til:

$$\lambda_1 \cdot \lambda_2 - \frac{a}{(1+a)^2} \cdot (\lambda_1 + \lambda_2)^2 = \frac{b}{(1+a)^2} \quad (11.8)$$

Nu udnyttes at

$$\det(C) = \lambda_1 \cdot \lambda_2 \quad (11.9)$$

$$\text{tr}(C) = \lambda_1 + \lambda_2 \quad (11.10)$$

og vi får udtrykket

$$R = \det(C) - k \cdot \text{tr}(C)^2 \quad (11.11)$$

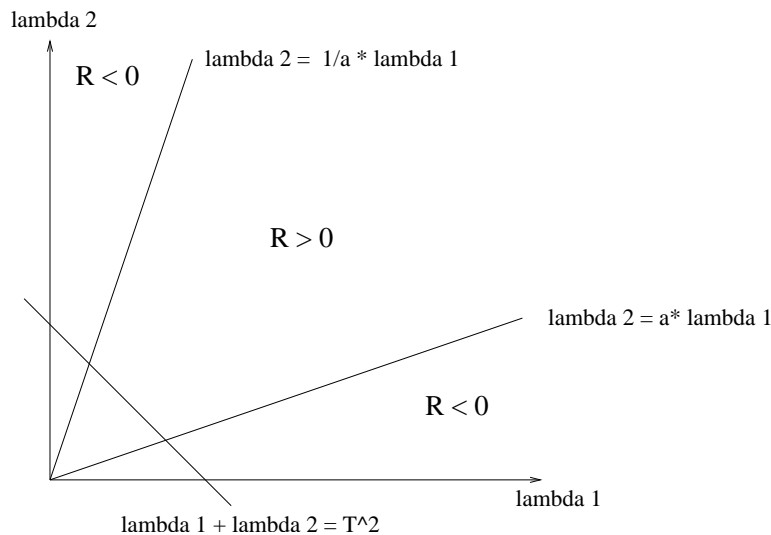
hvor  $k = \frac{a}{(1+a)^2}$ . Ved valg af  $b = 0$  har vi opnået, at  $R$  er positiv i det inhomogene område (med potentielle hjørner) og negativ i det homogene område. Dette er illustreret i figur 11.1. Fordelen ved ovenstående udtryk er, at eksplicit beregning af egen værdierne undgås. I stedet skal kun determinanten og sporet af  $C$  beregnes.

Hjørnepunkter detekteres (givet parameteren  $k$ ) som de punkter hvor  $R$  er positiv, og har lokalt maksimal værdi. Svarende til et krav om en minimal (kvadreret) gradientstørrelse kræves også at  $\text{tr}(C) > T^2$ , hvor  $T$  er en grænseværdi.

Ovenstående metode er særdeles udbredt og kan give udmærkede detektioner, hvis parametrene  $k$  og  $T$ , samt parametrene for beregning af  $C$  er valgt hensigtsmæssigt. Dette er imidlertid ikke let.

## 11.3 Junction detektion

Junctions er billedpunkter, hvor tre områder med forskellig intensitet mødes. Tæt på en junction varierer gradientretningen voldsomt. Flere af ovenstående metoder til hjørnetedetektion vil derfor også kunne detektere sådanne punkter. Det bemærkes, at tæt ved en junction vil gradienten være dårligt bestemt og detektion af kantpunkter vil ofte være fejlagtig. Der findes kun meget få metoder til detektion af junctions. En af de mere velkendte bygger på detektion af *End-Stops*. Et end-stop er et punkt, hvori en eller flere retlinede strukturer ender. Vi skal her kun betragte kanter, og kun skitsere en simpel detektionsmetode.



Figur 11.1: Eigenrådområde for Harris-detektion.

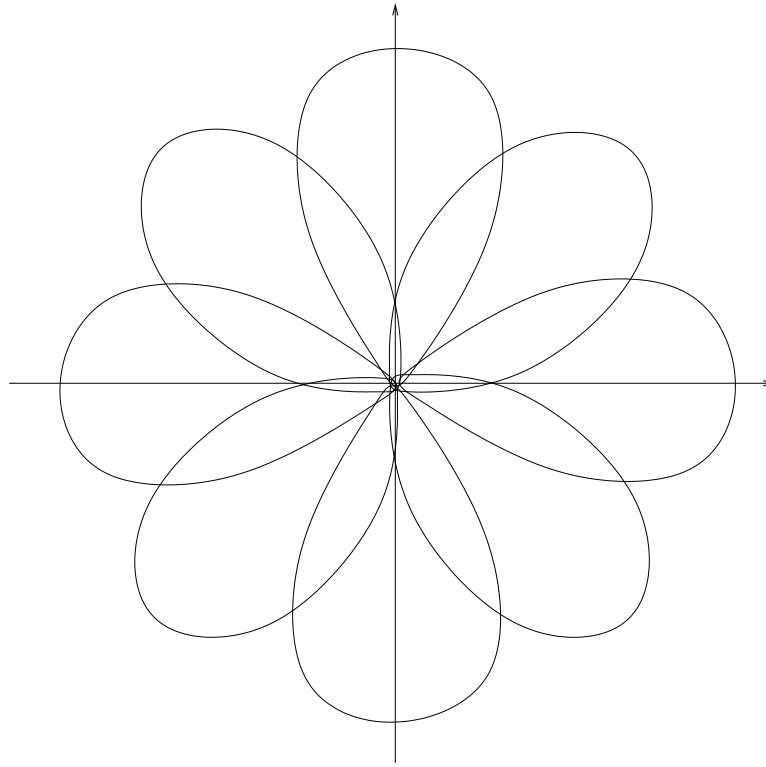
Antag at kantpunkterne i et billede er detekteret og at hver kantpunkt har tilknyttet såvel gradientretning som gradientstørrelse. Opdel de mulige retninger i  $N$  grupper (f.eks. 8 eller 16 som illustreret i figur 11.2) og konstruer tilsvarende  $N$  billeder  $F$ . Lad  $(x, y)$  være et kantpunkt med gradientstørrelse  $m$  og kvantificeret gradientretning  $\tilde{v}$ . Lad  $v$  være den kvantificerede kantretning (vinkelret på gradientretningen). Konstruer  $F$  ved  $F(i, x, y) = w(i, v) \cdot m$ . Et simpelt eksempel på vægtfunktionen  $w(i, v)$  kan f.eks. være:

$$w(i, v) = \begin{cases} 1 & \text{hvis } i = v \\ \frac{1}{2} & \text{hvis } |i - v| \text{ modulo } N = 1 \\ 0 & \text{ellers} \end{cases} \quad (11.12)$$

Konstruktionen af  $F$  tjener til opsplitning af gradientstørrelserne efter omtrentlig gradientretning. Denne opsplitning behøves ikke foretaget konkret, men letter forståelsen af de følgende operationer.

I et end-stop, der markerer et hjørne hhv. en junction ender 2 hhv. 3 lokalt lineære kanter. For hver pixel  $(u, v)$  opdeles en cirkelskive polært med centrum i pixelen og med (stor) støtteradius  $R$  i  $N$  segmenter svarende til de  $N$  orienteringer. I praksis bør segmenterne være overlappende og punkter tæt på centerpixelen bør ikke indgå i noget segment. Svarende til de  $N$  segmenter kan vi danne  $N$  binære billeder  $G(i, x, y)$  således at  $G(i, x, y) = 1$ , hvis  $(x, y)$  tilhører segment  $i$ .

Hvis der findes en retlinet kant i segment  $i$ , da vil summen  $s(i)$  af gradientstyrkerne i de tilsvarende udsnit af  $F(i)$  og  $F(i + N/2)$  være stor. Årsagen til, at der er to elementer i summen er, at vi ikke ønsker, at beregningen afhænger af kontrastfortegnet på kanten. Idet vi ignorerer



Figur 11.2: Opdeling af naboområde i 8 overlappende segmenter.

modulo-operationen på retningsangivelserne fås:

$$s(i) = \sum_x \sum_y [F(i, x + u, y + v) + F(i + N/2, x, y)] \cdot G(i, x, y) dx dy \quad (11.13)$$

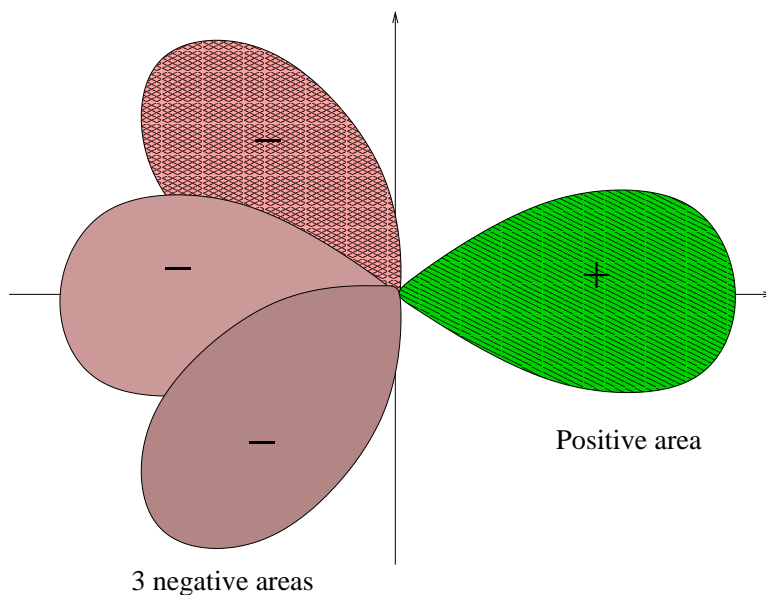
hvor summationen foretages over et (stort) støtteområde. Funktionen  $s(i)$  angiver altså hvor mange kantpunkter (talt med gradientstyrke), hvis fortegnsnormerede gradientretning er vinkelret på segmentretningen.

Antag at  $(u, v)$  ligger midt på en retlinet kant med kvantificeret retning  $i$ , og lad  $i' = (i + N/2)$ . Da vil både  $s(i)$  og  $s(i')$  være store. Hvis kanten ikke er retlinet, men krummer svagt, vil  $s(i)$  og enten  $s(i' + 1)$  eller  $s(i' - 1)$  være store. For at detektere en kant, der ender, beregnes derfor:

$$t(i) = \max \{0, s(i) - s(i') - s(i' + 1) - s(i' - 1)\} \quad (11.14)$$

Ved at summere  $E = \sum_{i=1}^N t(i)$  kan vi derfor få et enkelt mål  $E$ , der afspejler antal og styrke af kanter der ender i  $(u, v)$ .

Ved at foretage non-maximum suppression og grænseværdiafskæring på  $E(u, v)$ , kan vi sluttelig detektere junctions. Eventuelt kan vektoren  $t(i)$  analyseret mht. verifikation af antallet af kanter der ender i  $(u, v)$ . En sådan analyse (detektion af lokale maksimum) vil også kunne give



Figur 11.3: End-stop detektion for positiv horisontal kant.

kanternes orientering.

Det kan bemærkes, at der kan tænkes et stort antal variationer af metoden. Eksempelvis behøves kantpunkterne ikke detekteres eksplicit, omend dette formodentlig vil øge robustheden. Vægtfunktionen  $w$  kan vælges på mange måder. Beregningen af  $E$  ved summation af  $t(i)$  kan modificeres til også at omfatte en lille brøkdel af  $s(i)$ . Herved kan især lokaliseringen af afrundede hjørner forbedres.

I forhold til de tidligere beskrevne metoder til hjørnedetektion er ovenstående metode omstændelig og beregningstung. Typisk kræves et stort støtteområde for beregning af  $s(i)$  (f.eks. radius på 20 pixel). Dette gør detektionen robust, men også at kun junctions på stor skala detekteres.

## 11.4 Blobs

Et blob er en lys/mørk plet på en mørk/lys baggrund. Formen af et blob modelleres ofte ved en cirkel eller en ellipse. Blobdetektion omhandler bestemmelse af centeret af pletten. Den klassiske metode til blobdetektion er filtrering med en todimensional rotationssymmetrisk Laplace-Gauss-funktion.

$$\nabla G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \frac{1}{\sigma^4} (x^2 + y^2 - 2\sigma^2) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (11.15)$$

Svarende til den en-dimensionale version, som vi anvendte til detektion af linieprofiler, vil den to-dimensionale version give et ekstremum i blobcenteret for blobs med radius omkring  $\sqrt{2}\sigma$ . Ved

brug af et antal filtre med voksende  $\sigma$  og ved såkaldt skalarumsnormalisering af filtersvarende er det muligt ud over ekstremumsdetektionen også at estimere blobstørrelsen. Skalarumsnormalisering [22] består i at skalere således at filtersvaret kun er afhængig af inddata, og ikke af det anvendte filter.

Fordelen ved ovenstående filter er, at foldningen kan foretages hurtigt (se afsnit 10.6). Hvis man ønsker at generalisere filteret således, at aflange blobs kan detekteres, kræver dette dels estimation af kovariansmatricen for gradienten i hver pixel, og dels anvendelse af et filter fremkommet af laplaceoperatoren af den generelle todimensionale gaussfunktion (se afsnit 3.4). Filteret skal således estimeres og beregnes for hver pixel. Dette vil naturligvis gøre filtreringen meget langsommere end den simple rotationssymmetriske filtrering.

Efter filtreringen detekteres ekstremumpunkter. Som for linedetektion er det vigtigt, at det område, hvori der foretages ekstremumsanalyse, er tilstrækkeligt stort til at sikre mod fejlmarkeringer på grund af filtrets side-lobes.

## 11.5 Symmeteripunkter

Der findes et væld af symmetridetektorer. Visse opererer på intensiteter, på gradienter eller på kantpunkter, visse detektorer antager symmetri ved spejling om et punkt, andre kan håndtere spejling om en akse. Fælles er definitionen af en (som regel cirkulært) støtteområde  $\Omega$ , en vægtfunktion  $w$  (som regel en gauss), samt detektion ved non-maksimum undertrykkelse af et beregnet symmetrimål  $S$ . Vi skal her se en simpel metode, der opererer på gradienter og antager spejling om et punkt.

Betragt et punkt  $P$  og et lokalt koordinatsystem med origo i  $P$ . Lad  $r = \sqrt{x^2 + y^2}$ . Lad  $\Omega$  være halvcirkelskiven  $\{0 < r < R\} \cap (\{x > 0\} \cup \{x = 0 \wedge y > 0\})$ . Lad  $G(x, y)$  være gradienten i  $(x, y)$ . For perfekt symmetri gælder at  $G(x, y) = -G(-x, -y)$ . Et symmetrimål er derfor oplagt:

$$S = \sum_{(x,y) \in \Omega} w(r) \cdot |G(x, y) + G(-x, -y)|^2 \quad (11.16)$$

hvor  $w(r) = e^{-\frac{r^2}{2\sigma^2}}$ , med f.eks.  $\sigma = R/2$ . Symmetripunkter detekteres som lokale minima af  $S$ . I praksis kan der indvendes en del ved ovenstående simple metode. Eksempelvis vil punkter i områder uden struktur blive markeret. En tænkelig forbedring kunne være at betragte gradienterne i polære koordinater  $(r_+, \theta_+)$  hhv.  $(r_-, \theta_-)$ , at bestemme bidragene ved den gennemsnitlige gradientstørrelse og cosinus til vinklen mellem vektorerne:

$$S = \sum_{(x,y) \in \Omega} w(r) \cdot \frac{r_+ + r_-}{2} \cdot \cos^4(\theta_+ - \theta_- + \pi) \quad (11.17)$$

Et eksempel på resultatet af sådan beregning er vist nedenstående i figur 11.4. Bemærk den nydelige markering af krydset mellem de to søm, og af hullerne i saksen. Bemærk også at mange punkter som ikke intuitivt forventes at have stor symmetrigrad har et højt mål ved ovenstående simple metode.



Figur 11.4: Et billede og et symmetrimål.

## 11.6 Hough-transformationen

Hough-transformationen er egentlig en metode til lokalisering af parametriserede kurver. Metoden benytter ofte en indledende kantdetektion. Forudsætningen for at benytte metoden er, at der søges efter kurver, som er simpelt parametriseret. Det klassiske eksempel er søgning efter rette liniestykker. Ideen er at foretage en (ikke reversibel) transformation fra billedplanet til et parameterrum. Det antages at de enkelte parametre varierer diskret mellem en kendt minimal og maksimal værdi. For en ret linie vil man ofte benytte parametriseringen:

$$y = ax + b \quad (11.18)$$

Parameterrummet er to-dimensionalt og udspændt af  $a$  og  $b$ . Hough-transformationen består i korthed af for hvert detekteret kantpunkt  $(x_0, y_0)$  at tælle *Hough-accumulatoren*  $A[a][b]$  en op for samtlige par  $(a, b)$ , der opfylder overstående ligning, dvs:

$$A[a][b] := A[a][b] + 1 \quad (11.19)$$

$$\text{for } a_{\min} < a < a_{\max} \text{ og} \quad (11.20)$$

$$b = y_0 - ax_0 \quad (11.21)$$

Når transformationen er foretaget, kan linierne bestemmes ved de  $(a, b)$ , hvor accumulatoren  $A$  har lokalt maksimum. Problemet ved ovenstående transformation er, at grænserne for  $a$  er  $\pm\infty$  for lodrette linier. Problemet kan let løses ved at benytte en anden parametrisering:

$$x \cos \theta + y \sin \theta = \rho \quad (11.22)$$

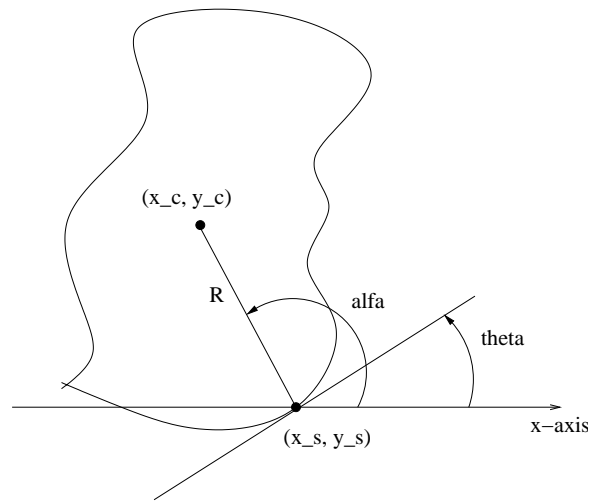
Det skal bemærkes at detektionen kun vil angive lokalisering af linier, og ikke udstrækningen af liniestykkerne.

Hough-transformationen kan (eksempelvis) også anvendes ved detektion/lokalisering af cirkler:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (11.23)$$

Her er parameterrummet tre-dimensionalt. Eksemplet illustrerer at anvendeligheden af Hough-transformationen ligger i parameterrumets størrelse. Transformationen vil være beregningsmæssig kostbar, hvis antallet af parametre er stort, eller hvis antal diskrete værdier som hver parameter kan antage er stort. Hough-transformationen kan også benyttes ved detektion af mere komplicerede parametriserede former.

Det er muligt at generalisere Hough-transformationen til geometriske figurer, der ikke er parametriserede men tabellerede. Tabelleringen af en silhouet af et kendt "objekt" foregår ved at vælge et referencepunkt  $(x_c, y_c)$  inden for dette. Derefter beregnes afstanden  $R$  for hvert punkt  $(x, y)$  på silhouetten til referencepunktet, samt tangentretningen  $\theta$  til silhouetten i  $(x, y)$  og vinklen  $\alpha$  mellem x-aksen og linien, der forbinder  $(x_c, y_c)$  og  $(x, y)$ . Dette er illustreret i figur 11.5.



Figur 11.5: Hough-transformationen for tabbelerede figurer.

Tabeller nu de to funktioner  $\alpha(\theta)$  og  $R(\theta)$ . Antag nu at de to tabeller er givet, og lad  $(x_s, y_s)$  være et kantpunkt med tangentretning  $\theta_s$ . Da vil de mulige lokaliseringer af referencepunktet være givet ved:

$$x_c = x_s + R(\theta_s) \cos \alpha(\theta_s) \quad (11.24)$$

$$y_c = y_s + R(\theta_s) \sin \alpha(\theta_s) \quad (11.25)$$

Hough-akkumulatoren  $A$  tælles, som tidligere beskrevet, 1 op for alle punkter  $(x_c, y_c)$ , der opfylder ovenstående ligning, og lokaliseringen af referencepunkterne for figurer af den tabellerede type findes ved bestemmelse af lokale maksima i akkumulatoren. Ovenstående skitse tager ikke højde for at en aktuel silhouet kan være roteret (eller skaleret) i forhold til prototypen. Det er dog relativt let at indføre dette ved at øge dimensionen af akkumulatoren. Tilføjelse af en rotationsdimension  $\phi$  og en skaleringsdimension  $S$  vil ændre ovenstående ligninger til:

$$x_c = x_s + SR(\theta) \cos (\alpha(\theta) + \phi) \quad (11.26)$$



$$y_c = y_s + SR(\theta) \sin(\alpha(\theta) + \phi) \quad (11.27)$$

Ovenstående modifikation øger altså akkumulatorens dimension fra to til fire. Den beregningsmæssige omkostning ved transformationen stiger derfor svarende til at tre (mod før en) parametre skal gennemløbes for hvert kantpunkt. Den generelle Hough-transformation er beskrevet i [1].

Der eksisterer en lang række varianter af Hough-transformationen. Eksempelvis kan estimationen foretages adaptivt, således at kvantificeringen i starten er meget grov, men for hver iteration forfines omkring de tidligere fundne lokale maksima i akkumulatoren. Dette øger beregningshastigheden væsentligt. I andre varianter søges metodens påvirkelighed af støj begrænset ved mere avancerede opdateringsmetoder.

Hough-transformationen er særlig anvendelig, hvis antallet af kantpunkter hørende til søgte strukturer er meget større end antallet af irrelevante kantpunkter. Hvis billedet, eksempelvis, er meget tekstureret vil det/de korrekte maksima i akkumulatoren ofte drukne i støj. Metoden skal derfor anvendes med varsomhed.



# Kapitel 12

## Segmentering

Segmentering af et billede er en proces, hvorved billedet opdeles i områder, sådan at hvert område beskriver et ”objekt”. Segmentering er i en vis forstand en komplementær proces til kantdetektion, på den måde at kanterne antages at angive grænserne mellem områderne. Ofte vil der imidlertid ikke være overensstemmelse mellem kanterne detekteret ved nogen kantdetektionsmetode og de fundne segmenter. F.eks. vil mange kanter ofte udgøre åbne konturer, mens segmenter altid har lukkede kanter. I dette kapitel diskuteres nogle klassiske og hyppigt anvendte metoder til segmentering af billeder.

Segmentering er en fundamental billedbehandlingsoperation. Dels er en segmentering nødvendig i mange billedbehandlingsopgaver, dels er segmenteringsoperationen teoretisk interessant, fordi den ligesom kantdetektionen er meget svær at foretage robust og nøjagtig. Problemet er, hvorvidt en intensitetsændring er tilstrækkelig kraftig til at markere overgangen til et nyt segment, eller om ændringen skal betragtes som støj eller segmentintern intensitetsvariation. Som ved kantdetektion lokaliseres segmenternes grænser ikke altid korrekt, og nogle intuitivt sammenhængende/adskildte segmenter detekteres som adskildte/sammenhængende.

### 12.1 Segmentering ved grænseværdiafskæring

Den simpleste form for segmentering af et billede vil være at bestemme segmenterne ved de billedområder, der fremkommer ved binarisering efter en grænseværdi. Problemet med metoden er oplagt, at det kan være umuligt at bestemme en fornuftig grænseværdi. Fordelen er at disse er meget hurtige. I industrielle situationer, hvor man kan kontrollere baggrunden, belysningen mv., vil disse metoder ofte være brugbare.

En meget enkelt metode til bestemmelse af en global hhv. lokal grænseværdi er at finde bunden af en dal i histogrammet over intensitetsværdierne for billedet hhv. et område i billedet. Ofte vil histogrammet imidlertid have mange lokale minima. En udglatning af histogrammet vil derfor være oplagt inden detektion af dalen. En anden mulighed er at detektere samtlige dale, og foretage segmenteringen ud fra sættet af grænseværdier. Denne metode, der svarer til en ikke-uniform kvantificering (se Kapitel 6), kan i visse tilfælde producere rimelige resultater. Forudsætningen er, at intensitetsvariationen inden for hvert ”objekt” er forholdsvis konstant og afvigende

fra ”nabo-objekternes” intensitet.

## 12.2 Simpel segmentering ved nabosammenligninger

En lidt mere avanceret metode er at bestemme tilhørsforholdet for de enkelte pixels ved sammenligning af intensiteten i nabo billedelementerne. Ved sådanne metoder benyttes en grænseværdi til afgørelse af hvorvidt to pixels tilhører samme segment. Denne værdi er ofte væsentlig lettere at angive som parameter til segmenteringsrutinen.

Segmentering ved nabosammenligninger er hurtige metoder, fordi de ofte kun kræver et gennemløb af billedelementerne. Tilsvarende er kvaliteten af resultaterne ofte mindre gode. Som vi skal se senere, kan det ofte være en fordel først at foretage en initial segmentering og dernæst at forbedre denne. Som initial segmentering er metoder, der benytter nabosammenligninger ofte udmærkede. Kodeskiten i figur 12.1 eksemplificerer segmenteringsmetoden.

```

1  procedure segment(i,j)
2      integer segnr;

3      if (ens(I[x][y],I[x-1][y])) begin
4          if (ens(I[x][y],I[x][y-1]))
5              segnr := merge(x,y);
6          else segnr := Seg[x-1][y];
7      end else if (ens(I[x][y],I[x][y-1]))
8          segnr := Seg[x][y-1];
9      else segnr := newseg();

10     Seg[x][y] := segnr;
11 end

```

Figur 12.1: Kode til simpel segmentering ved nabosammenligninger

hvor  $I$  er intensitetsbilledet,  $Seg$  er et billede angivende billedelementernes segmentnummer,  $newseg()$  er en funktion der returnerer et ubrugt segmentnummer, og  $ens(a,b)$  er en funktion, der returnerer sand eller falsk afhængig af et prædikat. Dette kan eksempelvis angive sand, hvis den numeriske differens mellem de to intensitetsværdier er mindre end en grænseværdi, dvs. hvis  $|a - b| < T$ . Proceduren  $segment(i,j)$  tænkes aktiveret en gang for alle billedpunkter, startende fra billedpunkter med de mindste koordinatværdier. Det er naturligvis nødvendigt af foretage segmenteringen af første billedrække inden proceduren  $segment(i,j)$  anvendes.

Funktionen  $merge(x,y)$  er mere kompliceret, idet det meget vel kan tænkes, at den kaldes i en situation hvor  $Seg[x-1][y] \neq Seg[x][y-1]$ . Dette svarer til, at to forskellige segmenter mødes og forenes i et. Det er derfor nødvendigt at gå tilbage i segmenteringen og rette allerede tildelte segmentnumre. En måde at styre dette problem på er, at opretholde en liste over samtlige segmenter og at benytte renummerering. Definition af og operation på en sådan liste er illustreret i figur 12.2. Ved anvendelse af skitsen antages det at funktionen  $newseg()$  initialiserer segmenttabellens indgange nr hhv.  $next$  med det ny segmentnummer hhv. konstanten -1.

```

1  typedef struct { int nr, next; } elem;
2  elem          *segtable;

3  int merge(x,y)  int x,y;  begin
4      int seg1, seg2, temp, index;

5      seg1 = segtable[seg[x-1][y]].nr;
6      seg2 = segtable[seg[x][y-1]].nr;

7      if (seg1 != seg2) begin
8          if (seg1 > seg2) begin
9              temp = seg1; seg1 = seg2; seg2 = temp;
10         end
11         segtable[seg2].nr = seg1;
12         index = seg2;
13         while (segtable[index].next != -1) begin
14             index = segtable[index].next;
15             segtable[index].nr = seg1;
16         end
17         segtable[index].next = segtable[seg1].next;
18         segtable[seg1].next = seg2;
19     end
20     return(seg1);
21 end

```

Figur 12.2: Kode til fletning af segmenter

## 12.3 Segmentering ved Split-metoder

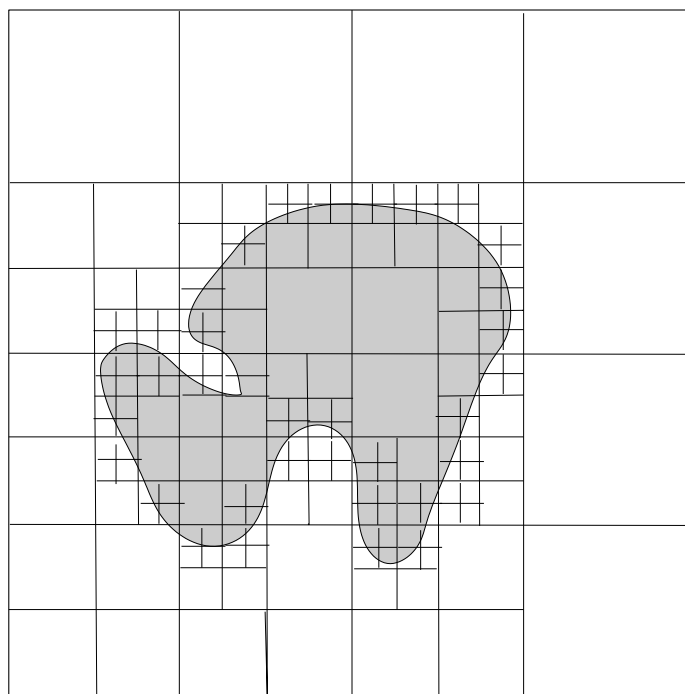
Segmentering ved split-metoder består i en rekursiv opdeling af billedet indtil et prædikat er opfyldt for alle segmenter. Prædikatet kan eksempelvis afgøre, hvorvidt intensiteterne i området er tilstrækkelig ens, eller om området indeholder kanter. Oplagte mål er variansen eller spandet ( $\max - \min$ ) af intensiteten. Opdelingen foretages typisk i fire lige store rektangulære delbilleder. Derfor egner split-metoder sig overordentlig godt til implementation ved en *quad-tree* repræsentation af billedet som illustreret i figur 12.3. Split-metoden genererer faktisk en sådan repræsentation.

Et væsentligt problem ved split-metoderne er, at nabopixels, der faktisk tilhører samme segment, ofte vil blive detekteret som tilhørende forskellige segmenter. Dette er afhængig af billedelementernes lokalisering i forhold til opdelingen.

Både split-segmenteringer og segmenteringer ved nabosammelingninger vil normalt resultere i en oversegmentering. Især områder, hvor gradienten er stor, vil ofte blive segmenteret i mange meget små segmenter. Der er derfor sædvanligt at kombinere sådanne metoder med metoder til fletning af segmenter.

## 12.4 Segmentering ved merge-metoder

I merge-segmenteringsmetoder startes med en initial segmentering, hvorefter segmenterne grupperes rekursivt indtil et *prædikat* er falsk for samtlige nabo-segmenter. Det er ofte overordentligt nyttigt at opretholde information om naboskabel mellem segmenterne. Hertil benyttes en *nabo-*



Figur 12.3: Split-segmentering vha. et quad-træ.

*graf*, der er en ikke-orienteret graf der forbinder hvert par af nabosegmenter. Hver kant i grafen er tilknyttet information om hvor godt de to segmenter passer sammen. Prædikatet kan opfattes som en operation på grafen hvorved kantinformationen binariseres.

Givet en klassifikation (efter prædikatet) af samtlige nabosegmenter er spørgsmålet, hvilken rækkefølge som segmenterne skal forenes i. Oplagt er at vælge det segmentpar, der opfylder prædikatet bedst. Dette vil ofte resultere i en suboptimal segmentering, der dog ofte er nær den optimale. Kunsten er at finde et omkostningsmål, der afspejler en fornuftig fletningsrækkefølge. Afhængig af billedtypen og formålet med segmenteringen kan dette eksempelvis omfatte:

- Forskellen i middelintensitet for de to segmenter.
- Variansen af intensitetsværdierne i foreningssegmentet.
- (Maksimum - Minimum) af intensiteten i foreningssegmentet.
- Summen af intensitetsgradientens størrelse langs segmentgrænsen.
- Segmenternes størrelse.
- Polynomielle fit af intensiteten i segmenterne.
- Hvorvidt en kant er detekteret i grænseområdet mellem segmenterne.

En ofte benyttet teknik er at opdele fletningen i flere faser, hver konstrueret til at udføre bestemte typer af segmentforeninger. Omkostningsmålet kan varieres lidt mellem faserne. Typisk er først at forene de store segmenter. Dette kan øge robustheden. Dernæst flettes middelstore segmenter. Til slut flettes de ofte mange små segmenter. Oprydningen i sidste fase er nødvendig da grænserne mellem de (approksimativt korrekt segmenterede) større områder vil ofte være fyldt med mange små (1 til 10 pixels store) segmenter. Disse er ofte forårsagede af rampeformede kanter i intensitetsfunktionen. Sådanne segmenter kan f.eks. forenes med de nabosegmenter, hvor omkostningsfunktionen er mindst, uanset hvor stor den så måtte være. Dette kan imidlertid resultere i ”flosede”segmentgrænser. En anden mulighed er at bestemme de små segmenters tilhørsforhold ud fra kantinformation.

Som tidligere nævnt er der en dualitet mellem kanter og segmenter i et billede. Nyere algoritmer til segmentering har forsøgt at udnytte kantinformationen i såvel split- som merge-faserne. Implicit kan kantinformationen udnyttes ved at måle middelværdien af intensitetsgradientens størrelse i grænseområdet mellem to segmenter, der er kandidater for en fletning. Jo mindre denne værdi er, jo lettere bliver det at forene segmenterne. Eksplicit kan kantinformationen udnyttes i en split-fase ved at kræve, at ingen segmenter må indeholde (eller være gennemskåret af) nogen kant. Lignende krav kan udnyttes i merge-fasen.

## 12.5 Seed growing segmentering

Simple seed growing teknikker benytter nabosammenligninger, som i den tidligere beskrevne metode. Forskellen er at der startes fra et antal punkter, der kan være valgt tilfældigt eller f.eks. sådan at gradienten i punkterne er minimal. Til hver segment (initielt en pixel) knyttes en liste af usegmenterede nabopixels. Iterativt flettes (for hvert segment) den bedst lignende nabo, indtil alle pixels er segmenteret. Sammenligningskriteriet kan f.eks. omfatte middelværdien i hvert segment. Det er klart at metoden er meget følsom over for antal og position af startpunkter. Kvaliteten af sådanne segmenteringer er ofte ringe. Dog kan metoden modificeres, som i nedenstående watershed algoritme, sådan at den kan producere gode resultater.

## 12.6 Watershed segmentering

Et vandskæl er en topografisk linie der adskiller områder, hvor nedbør vil flyde mod forskellige bassiner. Ved segmentering vil vi gerne opnå overensstemmelse mellem kanter og segmentgrænser. Opfat derfor gradientstørrelsen af intensitetsfunktionen som et bjerglandskab. De lokale minima vil udgøre bunden af bassinerne. Vandskællene er bestemt af de punkter, hvor gradientstørrelsen er lokalt maksimal, dvs. i kantpunkterne. Mellem disse punkter vil nedbør flyde i gradientens retning mod et bassin. Hvert segment er bestemt af det punkt, der udgør bassinbunden, samt de pixels, hvor nedbør vil flyde mod bassinbunden.

Der kan tænkes to simple metoder til segmenteringen. Den første klassificerer alle pixels ved at simulere, hvorledes vandet flyder, indtil dette har nået et punkt med gradientstørrelse

nul. Denne metode er ineffektiv. Den anden metode fungerer analogt til en seed-growing algoritme, hvor bassinbundene udgør startpunkterne. Disse detekteres først f.eks. som de hvor  $\|\nabla G_\sigma I\| = 0$ . Kriteriet for fletning af nabopixels er, at gradienten i naboen er rettet mod segmentet.

Væsentlig i Watershed-algoritmen er den initielle lav-pas filtrering, der dels sikrer at bassinbundene kan findes, dels at støj ikke forårsager en uhæmmet oversegmentering. Valget af  $\sigma$  specificerer som ved kantdetektion direkte afvejningen mellem under- og oversegmentering. Der findes forslag til hvorledes  $\sigma$  kan bestemmes lokalt. Sådanne benytter ofte at segmentere billedet for et sæt af  $\sigma$ -værdier og at bestemme den endelige segmentering ud fra stabiliteten af de enkelte segmenter som fkt. af skala. Et alternativ er at benytte watershed-algoritmen som initial (over-) segmentering og at flette segmenter efterfølgende.

## 12.7 Segmentering ved K-means algoritmen

K-means algoritmen er, som beskrevet i afsnit 6.4 en metode til klassifitation. Denne er især brugbar når data er flerdimensionale, som for farve- og multispektrale billeder, og når antallet af klasser/segmenter er kendt. Meget vigtigt er naturligvis også, at segmenteringen ikke umiddelbart omfatter spatiale naboegenskaber. De resulterende segmenter er således normalt ikke sammenhængende. Spatiale relationer kan dog til en vis grad indbygges ved at udvide datavektoren med de spatiale koordinater. For RGB-farvebilleder bliver denne således  $(R, G, B, x, y)$ . Hvis de enkelte elementer i datavektoren, som i dette eksempel, er af forskellig oprindelse, er det yderligere vigtigt at benytte et velovervejet afstandsmål. Den euklidiske afstand mellem punkter afspejler ofte ikke de egenskaber, som vi ønsker af segmenteringen. En oplagt mulighed er at normalisere alle koordinater inden anvendelse af k-means-algoritmen. En standard normalisering er at subtrahere middelværdien og dividere med spredningen:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - \bar{\mathbf{x}}}{C} \quad (12.1)$$

hvor  $C$  er diagonalmatricen af spredninger langs de forskellige elementer i featurevektoren  $\mathbf{x}$ . Vi skal vende tilbage til denne problemstilling i næste afsnit.

Som beskrevet i afsnit 6.4 er det et krav, at antallet af klasser/segmenter er kendt. Dette er et strengt krav ved anvendelse i segmentering, da selv trænede personer ofte har svært ved at angive et konsistent antal segmenter for mange billeder. En mulig løsning er at plote middelfstanden til klyngecentrene som funktion af antallet af klasser. Funktionen vil være stengt aftagende ofte meget indtil det korrekte antal klasser er nået, og derefter meget langsommere. Et bud på et godt antal klynger er derfor dette brydningspunkt.



## 12.8 EM algoritmen

En grundlæggende implicit antagelse ved brug af k-means algoritmen er at punkterne i de enkelte klynger kan opfattes som trukket af fordelinger, der bortset fra deres middelværdi er identiske og symmetriske. Dette er naturligvis sjældent tilfældet. I visse tilfælde kan man opnå en god approksimation ved at normalisere punktkoordinaterne, som beskrevet i ovenstående afsnit. Dette giver dog ikke altid gode resultater. Alternativt kan man løbende (i den iterative proces) søge at estimere fordelingerne for hver klynge separat. Den samlede fordeling af punkter kan derfor beskrives ved en *mixture-model*. En nyere, men allerede klassisk, metode er *Expectation-Maximization-algoritmen*, ofte forkortet til *EM-algoritmen*.

Før denne beskrives skal det dog bemærkes, at metoden kræver, at vi kan estimere (fitte) de enkelte fordelinger. Dette betyder at man skal vælge en klasse af fordelinger, hvis parametre (let) kan estimeres. Empiriske fordelinger, som ikke kan modelleres godt ved de valgte klasse, kan naturligvis heller ikke bestemmes korrekt. Et klassisk eksempel er to datasæt, der som spiraler snor sig ind i hinanden (uden at røre). Hvis modellen er en normalfordeling kan segmenteringen naturligvis ikke foretages fornuftigt. Sådanne datasæt er dog ofte kunstige, og i mange praktiske situationer vil man kunne anvende multivariate (flerdimensionale normale) modeller.

EM-algoritmen er solidt statistisk funderet, og den bagvedliggende teori fortjener væsentlig grundigere gennemgang end den, som præsenteres nedenfor. Her skal vi kun fokusere på princippet for metoden samt for de praktiske aspekter i.f.m. anvendelse i billedsegmentering. Det skal imidlertid understreges at metoden har et meget videre anvendelsesområde.

EM-algoritmen benytter, som k-means-algoritmen, en iterativ opdatering med to trin: Expectation eller E-trinet og Maximaization eller M-trinet. I modsætning til k-means-algoritmen klassificeres datapunkter ikke til kun een klasse. Derimod beregnes sandsynligheden (i hvert punkt i hver iteration) for at punktet stammer fra alle de  $K$  mulige klasser/segmenter.

**E-trin** Givet samtlige parametre, som beskriver fordelingerne af de  $K$  klasser, estimeres de forventede sandsynligheder for at hvert datapunkt/observation (her featurevektoren for hver pixel) stammer fra hver klasse.

**M-trin** Givet  $N$  observationer og deres sandsynlighed for tilhørsforhold til de  $K$  klasser estimeres parametrene for hver af de  $K$  fordelinger ved maksimum-likelihood estimation.

I praksis er E-trinet ofte let, hvorimod M-trinet kan være drilsk. Dette afhænger dog af den parametriske form af de valgte fordelinger. Maksimum-likelihood-estimationen angiver det sæt af parametre, der maksimere sandsynligheden for at se de faktiske observationer, dvs. de parametre, der bedst muligt beskriver data. For ikke-simple fordelinger kan den teoretiske udledning af udtryk for disse parameterværdier være kompliceret. Vi skal ikke her gennemgå teorien for maksimum-likelihood estimation (tag et statistikkursus), men antage at fordelingerne er normale.

Normale fordelinger er fuldstændigt beskrevet ved deres middelværdivektor  $\mu$  og deres kovariansmatrice  $\mathbf{C}$ :

$$G(\vec{x}) = \frac{1}{(2\pi)^{n/2} |\mathbf{C}|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\mu)^t \mathbf{C}^{-1}(\vec{x}-\mu)} \quad (12.2)$$

hvor  $n$  er dimensionen af observations/feature-vektoren (f.eks. 3 for RGB-værdier). Som beskrevet i Kapitel 3 er nivåkurverne for disse fordelinger ellipser. For  $K$  sådanne fordelinger er mixture modellen

$$p(\vec{x}|\Theta) = \sum_{k=1}^K p_k(\vec{x}|\theta_k)p(k) \quad (12.3)$$

hvor  $p(k)$  er vægten hvormed den  $k$ 'te fordeling indgår i den samlede fordeling,  $\theta_k = \{\mu_k, \mathbf{C}_k\}$  er parametervektoren for den  $k$ 'te fordeling, og  $\Theta = \{\theta_1, \dots, \theta_K, p(1) \dots p(K)\}$  er den samlede parametervektor. Den likelihood-funktion, som vi ønsker maksimeret i M-trinet er altså:

$$\prod_{\text{observation } j} \left( \sum_{k=1}^K p_k(\vec{x}_j|\theta_k)p(k) \right)$$

Vi er nu klar til at give første trin i algoritmen. Lad  $s$  angive iterationstallet, og antag initiale værdier  $\Theta^1$ .

### E-trin:

1. For alle pixels  $j$  og for alle klasser  $k$ :

$$p_{jk}^s = p_k^s(\vec{x}_j|\theta_k^s)p(k)$$

2. For alle pixels  $j$ :

$$p^s(k|\vec{x}_j) = \frac{p_{jk}^s}{\sum_{k=1}^K p_{jk}^s}$$

Den observante student vil bemærke at ovenstående blot er en formulering af Bayes regel:  $p(k|x) = \frac{p(x|k)p(k)}{p(x)}$ . I M-trinet skal nye værdier for parametrene beregnes. Disse udledes på sædvanlig måde ved at differentiere (logaritmen til) likelihood-funktionen mht. de ønskede variable og sætte resultatet lig nul. Vi skal ikke her gennemgå dette, men blot referere resultaterne.

### M-trin:

For alle klasser  $k$ :

$$1. p^{s+1}(k) = \frac{1}{N} \sum_{j=1}^N p^s(k|\vec{x}_j)$$

$$2. \mu_k^{s+1} = \frac{\sum_{j=1}^N \vec{x}_j p^s(k|\vec{x}_j)}{\sum_{j=1}^N p^s(k|\vec{x}_j)}$$

$$3. \mathbf{C}_k^{s+1} = \frac{\sum_{j=1}^N [(\vec{x}_j - \mu_k^s)(\vec{x}_j - \mu_k^s)^t] p^s(k|\vec{x}_j)}{\sum_{j=1}^N p^s(k|\vec{x}_j)}$$

EM-algoritmen itereres indtil konvergens, og hver pixels tildeles den klasse/segment, der maksimerer  $p(k|\vec{x}_j)$ . Denne værdi vil desuden angive sikkerheden på pixel-klassifikationen. Det kræves naturligvis at algoritmen, ud over at tallet  $K$ , forsynes med et initielt bud på parametervektoren  $\Theta$ . Dette er også et af de punkter der volder de største problemer ved praktisk anvendelse. Ofte er det acceptabelt at sætte  $p^1(k) = 1/K$ ,  $C^1 = \text{diag}(\sigma^2)$ , hvor  $\sigma$  er en brøkdel af den maksimale variation i koordinatværdier for featurevektoren  $\vec{x}$ . Derimod er der kun få gode råd mht. angivelse af middelværdierne (klyngecentre)  $\mu_k^1$ . Randomiseret valg vil ofte lede til forskellige segmenteringer. En mulig tilgang er at foretage et antal segmenteringer ud fra randomiseret valg af  $\mu_k^1$  og at vælge den, der samlet beskrive data bedst. Denne strategi er naturligvis beregningsmæssigt dyr. Afhængig af hvorledes data fordeler sig i feature rummet kan der også tænkes et antal (robuste men unøjagtige) metoder til en initial bestemmelse af  $\mu_k^1$ .

## 12.9 Andre metoder

I ovenstående afsnit har vi gennemgået en række af de klassiske metoder til segmentering. Der findes naturligvis et stor antal andre tilgange. Eksempelvis er metoder inspireret af Mumford-Shah funktionalen (se Kapitel 9) og videreudviklinger af dette, samt anvendelse af variationsregning med numerisk løsning af de tilhørende partielle differentiaalligninger, vundet stor udbredelse. Mange implementationer udnytter teknikker fra såkaldte *level-set metoder*, hvorved segmenteringshastigheden kan øges markant. Vi skal ikke her diskutere sådanne metoder.



# Litteratur

- [1] D. H. Ballard: *Generalizing the Hough transform to arbitrary shapes*, Pattern Recognition 13,2; 1981.
- [2] Ballard, Brown: *Computer Vision*; Prentice Hall 1982.
- [3] S. Banks: *Signal processing, Image Processing, and Pattern Recognition*; Prentice Hall 1990.
- [4] P. R. Beaudet: *Rotationally Invariant Image Operators*; Int. joint Conf. on Pattern Recognition, 1978.
- [5] V. Berzins: *Accuracy of Laplacian Edge Detectors*, CVGIP vol 27, 1984.
- [6] O. Brigham: *The Fast Fourier Transform and its Applications*, Prentice-Hall 1988.
- [7] P. J. Burt and E. H. Adelson: *The Laplacian Pyramid as a Compact Image Code*; IEEE trans. on communications vol. 31 no. 4 1983.
- [8] J. F. Canny: *A Computational Approach to Edge Detection*; IEEE PAMI vol. 8 no. 6 1986
- [9] R. Deriche: *Using Canny's Criteria to derive a Recursively Implemented Optimal Edge Detector*; Int. Jour. on Computer Vision, 1987
- [10] FR. Fabricius-Bjerre: *Lærebog i Geometri*; Polyteknisk forlag, 1982.
- [11] D.A. Forsyth, J. Ponce: *Computer Vision, A modern approach*, Prentice Hall, 2003.
- [12] Geman og Geman: *Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images*, IEEE PAMI vol. 6 no. 6 pp. 721-741, 1984.
- [13] R.C. Gonzales, Wintz: *Digital Image Processing*, Addison Wesley 1987.
- [14] R.C. Gonzales, R.E. Woods: *Digital Image Processing*, Addison Wesley 1992.
- [15] R.C. Gonzales, R.E. Woods: *Digital Image Processing*, 2ed., Addison Wesley 2001.
- [16] Haralick et.al.: *Image Analysis Using Mathematical Morphology*, IEEE Pattern Analysis and Machine Intelligence vol. 9, no. 4, 1987.

- [17] R. Hartley, A. Zisserman: *Multiple View Geometry in computer vision*; 2. ed. Cambridge Press 2003.
- [18] B.K.P. Horn *Robot Vision*, MIT Press 1986.
- [19] A. Hunt: *Digital Image Restoration*, Prentice Hall, 1977.
- [20] A. Jain: *A Fast Carhuen-Loeve Transform for a Class of Random Processes*; IEEE trans. on communication vol. 24.
- [21] Kitchen, Rosenfeld: *Gray-level corner detection*; Pattern Recognition Letters; 1, 1982.
- [22] T. Lindeberg: *Scale-Space Theory in Computer Vision* Kluwer, 1994.
- [23] J.R. Magnus, H. Neudecker: *Matrix differential calculus with applications in Statistics and Econometrics*, Wiley 2002.
- [24] A. Marion: *An Introduction to Image Processing*, Chapman and Hall 1987.
- [25] J.H. McClellan, R.W. Schafer, M.A. Yoder: *DSP First- A Multimedia Approach*, Prentice Hall 1999
- [26] J.H. McClellan, R.W. Schafer, M.A. Yoder: *Signal Processing First*, Pearson Education, 2003.
- [27] N. Nagao and T. Matsuyama: *A Structural Analysis of Complex Aerial Photographs*, 1980.
- [28] S. Olsen: *Ekstrakt af gamle noter i Mønstergenkendelse*, Se kursushjemmeside.
- [29] P. Perona, J. Malik: *Scale-Space and Edge Detection Using Anisotropic Diffusion*, IEEE Pattern Analysis and Machine Intelligence vol. 12, no. 7, 1990.
- [30] W. H. Press et.al.: *Numerical Recipes in C*, 2. ed; Cambridge Press 1992.
- [31] A. Rosenfeld, A.C. Kak: *Digital Picture Processing*, 2.ed. Academic Press 1992.
- [32] J. Serra and P. Soille: *Mathematical Morphology and Its Application to Image Processing*, Kluwer Academic Publishers, 1994.
- [33] L.G. Shapiro, G.C. Stockman: *Computer Vision*, Prentice Hall, 2001.
- [34] M. Sonka, V. Hlavac, R. Boyle: *Image Processing, Analysis and Machine Vision*, Brooks/Cole Publishing, 2.ed, 1999.
- [35] J. Teuber: *Digital Billedbehandling*, Teknisk Forlag, 1989.
- [36] J. Weickert: *Anisotropic Diffusion in Image Processing*, B.G. Teubner, Stuttgart 1998.
- [37] Witkin: *Scale Space Filtering*, Proceedings of IJCAI 1983.

# Indeks

Åbning, 109

A posteriori sandlynlighed, 65

A/D-konvertering, 13

Affin projektion, 102

Affoldning, 16

Albedo, 12

Aliasing, 29

Analyseligningen, 69

Anisotrop diffusion, 124

Autokorrelationen, 26

Båndbegrænset funktion, 27, 31

Båndbredde, 27

Basisbillede, 70, 75

Bayes Restoration, 65

Bayes sætning, 65

Beaudet operator, 134

Billedelement, 10

Billeder

    Basis, 70

    Binære, 10

    Dekomposition, 70

    Digitale, 10

    Gråtone, 10

    Multispektrale, 11

    Pyramider, 79

Billedforringelse, 57

Billedkvalitet, 13

Billedplan, 11

Billedprojektion, 70

Billedsyntese, 9

Billedtransformation, 69

Bit spejling, 35

Blob-detektion, 159

Boltzmann fordeling, 123

Box-filter, 45

Brændvidde, 11

Burt pyramide, 81

Butterworthfilter, 41, 43

Canny's kantoperator, 144

CCD-kamera, 11

Cirkeldetektion, 161

Cosinus transformation, 71

Cosinustransformationen, 14

Cut-off frekvens, 40

Datamatsyn, 9

De store tals lov, 46

Dekomposition, 70

    Ortogonal, 70

Den svage streng, 117

Deriche kantoperator, 147

Det inverse filter, 60

Diagonalisering, 74

Differens approksimation, 132

Diffusionsligning, 125

Egenværdi, 52

Egenvektor, 52

Ekspansion, 108

Elliptisk flade, 141

EM algoritmen, 171

EM-algoritmen, 94

End-Stops, 156

Energi, 25, 59

Energifunktion, 116

Erosion, 108

Eulers formel, 20, 141

Expectation-Maximization, 171

Farvebånd, 11

- Farvebilleder, 11
- Fasevinkel, 20
- FFT algoritmen, 31
- Filterfunktion, 14
- Filterkoefficienter, 13
- Filtrering, 13
  - Bånd-pas, 44
  - Båndafvisning, 44
  - Box-filter, 45
  - Butterworth, 41
  - Filterkoefficienter, 13
  - FIR, 13
  - Foldning, 14
  - Frekvensbaseret, 39
  - Gauss-filter, 43, 45
  - Global, 13
  - Høj-pas, 43
  - Hanning, 42
  - Homomorfisk, 48
  - Idealfilter, 40
  - IIR, 148
  - Ikke-lineær, 47
  - Intensitetsområde, 44
  - Invers, 60
  - Kausal, 148
  - Lav-pas, 40
  - Lineær, 14
  - Lineære, 39
  - Lokal, 13
  - Medianfilter, 47
  - Nagao-filter, 47
  - Normaliseret foldning, 54
  - positionsafhængig, 49
  - Positionsinvariant, 14
  - Rekursiv, 149
  - Wiener, 60
- Focus of attention, 80
- Foldning, 15
- Foldningssætningen, 25, 39
- Forvrængning, 57, 58
- Fourier
  - Baglængs transformation, 22
  - Diskret transformation, 21
  - Forlængs transformation, 22
  - Invers transformation, 19
  - Kontinuert transformation, 19
  - Origo, 23
  - Periode, 23
  - Periodicitet, 31
  - Transformationen, 18, 71
  - Translation, 23
- Fourierspektrum, 19
- Fouriertransformationen, 14
- Frekvensrepræsentationen, 69
- Frekvensbånd, 44
- Frekvensområde, 14
- Frekvensvariabel, 19
- Gabor filter, 151
- Gauss filteret, 43
- Gauss pyramide, 81
- Gauss-filteret, 45
- Gaussfunktion, 124
- Gausskrumning, 139, 140
- Gendannelsesproblemet, 58, 60
- Geometrisk middelværdifilter, 65
- Geometrisk opretning, 97
- Grænseværdiafskæring, 137, 165
  - Med hysteres, 137
- Gråtone-morfologi, 112
- Grov til fin analyse, 80
- Hanning filtret, 42
- Hardamard transformation, 72
- Harris-detektor, 156
- Hessian matrice, 141
- High frequency emphasis, 43
- Histogram Linearisering, 85
- Histogrambaserede metoder, 85
- Hit-or-Miss transformationen, 111
- Hjørnedetektion, 153
  - Egenværdianalyse, 155
  - Harris-detektor, 156
  - Kitchen-detektor, 154
  - Moravec, 153
- Homeografi, 102



- Homogene koordinater, 102
- Homomorfisk filtrering, 48
- Hotelling transformationen, 73
- Hough-accumulatoren, 161
- Hough-transformationen, 161
- Hovedkrumninger, 141
- Hyperbolsk flade, 142
- ICA, 77
- Idealfilter, 40, 43
- IIR-filtrering, 148
- Ikke-lineære filtre, 47
- Impulsfunktionen, 17
- Impulsrespons, 57
- Impulstog, 17, 29
- Independent Component Analysis, 77
- Intensitetsgradient, 131
- Interessepunkter, 153
- Interpolation, 99
  - Bilineær, 99
  - Kubisk, 99
  - Nærmeste nabo, 99
- Invers FFT, 35
- Junction-detektion, 156
- K-means algoritmen, 93
- K-means segmentering, 170
- Kantdetektion, 129
- Kantmodel, 129
  - Rampe, 130
  - Trappetrin, 130
- Kantoperator
  - Beaudet, 134
  - Canny, 144
  - Deriche, 147
  - Laplace Gauss, 142
  - Lokaliseringsevne, 145
  - Nøgletal, 146
  - Performance, 145
  - Robusthed, 145
- Kantprofil, 130
- Karhuen-Loeve transformationen, 73
- Kausale filtre, 148
- Kitchen-detektor, 154
- Klyngedetektion, 93
- Kontrast, 129
- Kontrastfremhævnning, 43
- Koordinatsystem, 10
- Korrelation, 26, 63
- Korrelationsmatrice, 63
- Korrelationssætningen, 26
- Kovariansmatrice, 74
- Kovariansmatrix, 50
- Krydskorrelation, 26
- Kvanter, 91
- Kvantificering, 91
- Kvantificeringsfejl, 91, 132
- Laplace pyramide, 81
- Laplace-Gauss filter, 142
- Laplaceoperator, 139
- likelihood-funktion, 65
- Lokal histogramlinearisering, 88
- Lokaliseringsevne, 133, 145
- Lukning, 109
- MAP estimat, 65
- Markov proces, 119
- Markov Random Fields, 118
- Markov random fields, 119
- Matematisk morfologi, 107
- Medianfiltret, 47
- Merge-prædikat, 167
- Merge-segmentering, 167
- Metropolis algoritmen, 122, 123
- Middelkrumning, 140
- Mindste kvadraters metode, 61, 98, 134
- Mixture-modeller, 171
- Moravec, 153
- Morfologi, 107
- Mumford-Shah, 117
- Nøjagtighed, 133
- Nabografer, 168
- Nagaofiltret, 47
- Non maximum suppression, 137
- Normaliseret foldning, 54

- Normalsnit, 140
- Opløsning, 10
- Optimal kvantificering, 92
- Optisk center, 11
- Oskulerende paraboloide, 140
- Parabolsk flade, 142
- Parabolske kurver, 142
- Parameterrum, 161
- Parsevals sætning, 25
- Paspunkter, 97
- PCA, 73
- Performancemål, 145
- Perona-Malik diffusion, 124
- Perspektivisk afbildning, 11
- Perspektivisk projektion, 102
- Pin-hole kamera, 11
- Pixel, 10
- Positionsinvariants, 57
- Principalkomponentanalyse, 73
- prior, 65
- Projektionslinier, 11
- Pseudo-invers, 61, 99, 135
- PSNR, 59
- Punktspredningsfunktion, 57, 133
- Pyramiderepræsentationer, 79
  - Burt pyramide, 81
  - Gauss pyramide, 81
  - Laplace pyramide, 81
- Quad-tree, 80, 167
- Reflektionsfordelingsfunktion, 12
- Registrering, 97
- Regularisering, 63, 115
  - Diskontinuert, 117
- Regulariseringsparameter, 116
- Rekursiv filtrering, 149
- Ringningseffekt, 40
- Robusthed, 133, 145
- Sampling, 17, 22, 79
- Samplingafstand, 31
- Samplingsætningen, 27
- Seed growing segmentering, 169
- Segmentering, 165
  - Grænseværdiafskæring, 165
  - Merge, 167
  - Nabosammenligninger, 166
  - Split, 167
- Segmentoprydning, 169
- Sensorer, 11
- Seperabilitet, 23, 69
- Signal-støjforhold, 59, 64, 145
- signal-støjforhold, 45
- Simuleret udglødning, 122
- Skala, 133
- Skalaparameter, 46, 142
- Skalarum, 46, 124
- Split-segmentering, 167
- Støj, 58, 131
  - Hvid, 59
  - Normalfordelt, 132
  - Stationær, 59
  - Ukorreleret, 132
- Støjfiltrering, 40, 58
- Støjfunktion, 57
- Støtte, 13
- Stationær proces, 119
- Stokastisk felt, 118
- Strukturelement, 109, 112
- Strukturtensor, 53
- Styrkespektrum, 19, 64
- Symmetri-detektion, 160
- Synteseligningen, 69
- Tæthedssfunktion, 85
- Taylorrække, 132
- Template matching, 27
- Transformationskerne, 69
- Udvidede funktioner, 18
- Umbra, 112
- Variationsregning, 116
- Varmeledningskoefficienten, 125

Walsh transformation, 72  
Warp-around error, 18, 25  
Warping, 100  
Watershed segmentering, 169  
Wavelet-transformationen, 77  
Wienerfilteret, 60, 62, 63  
    Parametrisk, 64  
  
Z-transformationen, 73  
Zero crossing, 139