

UNIVERSITEIT GENT

MASTER THESIS

Beeldoptimalisatie voor 3D Elektronenmicroscopie Beelden

Promotor:

dr. Yvan SAEYS

Co-promotor:

Auteur:

Prof. dr. ir. Wilfried Philips

Joris ROELS

Begeleiders:

dr. ir. Hiep Luong

ir. Jan Aelterman

lic. Jonas De Vylder

*Scriptie voorgelegd tot het behalen van de
graad van Master of Science in de Wiskundige Informatica*

30 mei 2014



“There are things known and things unknown and in between are the doors.”

Jim Morrison

Voorwoord

In een masterthesis wordt gestreefd naar concrete, originele en vernieuwende resultaten. Het is echter geen voor de hand liggende opdracht voor een student om dit tot een goed einde te brengen. Dit werk is het resultaat van negen maanden intensief onderzoek en samenwerking. Speciaal daarvoor verdienken enkele mensen een dankwoord.

In de eerste plaats wil ik mijn promotor dr. Yvan Saeys en co-promotor Prof. dr. ir. Wilfried Philips bedanken voor het mogelijk maken van dit onderzoek. Verder heb ik altijd kunnen rekenen op de uitstekende opvolging en feedback van mijn thesisbegeleiders: dr. ir. Hiep Luong, ir. Jan Aelterman en lic. Jonas De Vylder. Voor al mijn vragen en antwoorden kon ik steeds bij hen terecht. Vervolgens wil ik ook iedereen binnen de Bio Imaging Core (VIB Gent) bedanken voor hun uitstekende samenwerking. In het bijzonder verdienken dr. Chris Guerin, dr. Saskia Lippens, dr. Anna Kremer en Benjamin Pavie speciale aandacht voor hun creatieve ideeën en prachtige 3D elektronenmicroscopiebeelden.

Vervolgens wil ik mijn vrienden bedanken om me er aan te doen herinneren dat inspanning af en toe ook eens plaats moet maken voor ontspanning. Om af te sluiten wil ik ook nog mijn familie bedanken voor hun geapprecieerde steun de afgelopen maanden. In het bijzonder gaat mijn dank uit naar mijn ouders die altijd achter mij hebben gestaan, ook al hadden ze er geen idee van wat al die scripts en formules wouden zeggen. Zonder hen had ik deze kans tenslotte nooit gekregen.

Inhoudsopgave

Extended Abstract	i
Voorwoord	ii
Inhoudsopgave	iii
Lijst van Figuren	v
Lijst van Tabellen	vi
Acroniemen	vii
1 Inleiding	1
1.1 Inleiding tot de microscopie	2
1.1.1 Historiek	2
1.1.2 Soorten microscopen	3
1.1.2.1 Optische microscoop	3
1.1.2.2 Elektronenmicroscoop	4
1.2 Probleemstelling	7
1.2.1 Acquisitie van 3D SEM-beelden	7
1.2.2 Problematiek bij de acquisitie	8
1.2.2.1 Ruis	9
1.2.2.2 Blur	11
1.2.3 Doelstelling	12
1.2.4 Werkplan	12
2 Analyse van EM beeldvorming en -degradatie	14
2.1 Beschrijving van de data	14
2.2 Ruismodel	16
2.3 Ruisanalyse	18
2.3.1 Afzonderen van de ruis	18
2.3.2 Distributie	20
2.3.3 Stationariteit	22
2.3.4 Spatiale correlatie	23
2.3.5 Signaalafhankelijkheid	25
2.4 Lensblur	28
2.5 Conclusie	30

3 Beeldreconstructie	32
3.1 Voorgesteld ruismodel	32
3.2 Literatuurstudie	33
3.2.1 Ruisonderdrukking	33
3.2.1.1 Laagdoorlaatfilter	33
3.2.1.2 Anisotrope diffusie	34
3.2.1.3 Non-local means	37
3.2.1.4 Bayesiaanse schatters	40
3.2.1.5 Alternatieve ruisonderdrukkingstechnieken	42
3.2.2 Deconvolutie	43
3.2.2.1 Invers filter	44
3.2.2.2 Bayesiaans schatter als deconvolutiemethode	45
3.3 NLMS als MAP schatter	47
3.4 NLMS-SCB	49
4 Experimentele resultaten	54
4.1 Beeldoptimalisatie bij artificiële ruisbeelden	54
4.2 Beeldoptimalisatie bij EM-beelden	58
5 Conclusie en toekomstig werk	62
A Notaties	65
A.1 Convolutienotatie	65
A.2 Matrixproductnotatie	66
B Eigenschappen van convoluties	68
C De fouriertransformatie	70
D Het conjugate gradient algoritme	72
Bibliografie	74

Lijst van figuren

1.1	Zeiss Merlin 3D EM	2
1.2	De allereerste microscopen	2
1.3	Principe van beeldvorming bij optische microscopie [4, 5]	4
1.4	Principe van beeldvorming bij elektronenmicroscopie	5
1.5	Principe van horizontaal scannen	6
1.6	Verschil in scherptediepte [7]	6
1.7	Storingsfactoren bij 3D SEM	9
1.8	Ruis in een EM-beeld	10
1.9	Segmentatie bij EM ruisbeelden.	11
1.10	Wazig EM-beeld	11
2.1	Origineel EM-beeld	15
2.2	Herschaald EM-beeld	16
2.3	Waargenomen ruisbeeld in de afwezigheid van een elektronenstraal.	19
2.4	Distributie ruisintensiteiten	21
2.5	Stationariteit van de ruis	22
2.6	Spatiale correlatie van de ruis	24
2.7	Signaalafhankelijkheid van de ruis	27
2.8	Geblurred ‘ruisvrij’ rooster	29
3.1	Ruisonderdrukking met een laagdoorlaatfilter	34
3.2	Ruisonderdrukking met anisotrope diffusie	36
3.3	NLMS gewichtsfuncties	39
3.4	Ruisonderdrukking met NLMS	40
3.5	Ruisonderdrukking met een MAP schatter	42
3.6	Deconvolutie met het invers filter	45
3.7	Deconvolutie met een MAP schatter	46
4.1	Beeldreconstructie met pixelgebaseerde NLMS varianten bij gecorreleerde, signaalafhankelijke ruis.	55
4.2	Beeldreconstructie met vectorgebaseerde NLMS varianten bij gecorreleerde, signaalafhankelijke ruis.	56
4.3	Referentiebeelden	57
4.4	NLMS-SC vergelijking: PSNR i.f.v. referentiebeeld.	57
4.5	Beeldreconstructie met NLMS varianten bij gecorreleerde, signaalafhankelijke ruis en beeldvervaging.	58
4.6	NLMS-SCB vergelijking: PSNR i.f.v. referentiebeeld.	59
4.7	Beeldreconstructie van EM beelden met het NLMS-SC(B) algoritme	60
4.8	Beeldreconstructie van EM beelden met alternatieve algoritmen	61

Lijst van tabellen

2.1	<i>p</i> -waarden Chi-square goodness-of-fit test.	22
2.2	Stationariteit van de ruis i.f.v. de intensiteit	23

Acroniemen

OM	Optische microscopie
EM	Elektronenmicroscopie
TEM	Transmissie-elektronenmicroscopie
SEM	Scanning electron microscopy
STEM	Scanning transmission electron microscopy
SBF	Serial block-face
VIB	Vlaams Instituut voor Biotechnologie
MSE	Mean squared error
PSNR	Peak signal to noise ratio
DFT	Discrete fouriertransformatie
IDFT	Inverse discrete fouriertransformatie
PSD	Power spectral density
ACF	Autocorrelatiefunctie
KKM	Kleinste-kwadratenmethode
MAP	Maximum a posteriori
ML	Maximum likelihood
NLMS	Non-local means
NLMS-SC	Non-local means (signaalafhankelijk, gecorreleerd)
NLMS-SCB	Non-local means (signaalafhankelijk, gecorreleerd, blur)

Hoofdstuk 1

Inleiding

Een groot deel van het biologisch en medisch onderzoek heeft zijn resultaten te danken aan de microscopie. Dit onderwerp heeft in de afgelopen jaren dan ook een enorm grote vooruitgang geboekt. Tegenwoordig is het mogelijk om biologisch materiaal op nanoschaal waar te nemen dankzij de elektronenmicroscopie. Vrijwel alle elektronenmicroscopen zorgen echter voor ongewenste beeldartefacten in de waargenomen beelden. Dit is een grote hindernis voor toekomstig onderzoek en verdere ontwikkelingen.

De vraag naar beeldoptimalisatie bij 3D EM kwam vanuit het Vlaams Instituut voor Biotechnologie. Dit is een non-profit onderzoeksinstituut verspreid over campussen van vier Vlaamse universiteiten: UGent, KU Leuven, Universiteit Antwerpen en Vrije Universiteit Brussel. Hun onderzoek draait voornamelijk rond levenswetenschappen en het aanpassen van bestaande technieken naar industriële (vooral landbouw) en medische toepassingen.

Sinds 2012 heeft het VIB in Gent en Leuven twee sites voorzien die zich specialiseren in microscopische beeldopname van biologisch materiaal: de Bio Imaging Core. De site in Leuven houdt zich bezig met onderzoek naar dynamische (temporele) processen in cellen, terwijl in Gent de nadruk ligt op 3-dimensionaal (spatiaal) onderzoek van weefsel. Om dit laatste mogelijk te maken, is de Bio Imaging Core uitgerust met een state-of-the-art 3D EM: de Zeiss Merlin (Figuur 1.1). De (beschadigde¹) EM-beelden die hiermee opgenomen worden, bemoeilijken het onderzoek. De vraag naar beeldoptimalisatie is dus een logische stap, aangezien 3D EM een enorm duur proces is met veel potentieel.

We starten in Sectie 1.1 met een kort overzicht van de evolutie van de microscopie in de loop der jaren. Ook worden de verschillende soorten microscopen en hun eigenschappen hier kort toegelicht. In Sectie 1.2 wordt vervolgens het beeldoptimalisatieprobleem

¹Lees: EM-beelden die onderworpen zijn aan tal van beeldartefacten waaronder ruis, vervaging, etc.



Figuur 1.1: Zeiss Merlin 3D EM



(a) De Jansen microscoop (b) De Van Leeuwenhoek microscoop

Figuur 1.2: De allereerste microscopen

geformuleerd. We motiveren de relevantie van het probleem en onze doelstellingen met deze scriptie. We sluiten af met een overzicht van ons werkplan.

1.1 Inleiding tot de microscopie

1.1.1 Historiek

De allereerste microscoop (Figuur 1.2a) zou gebouwd zijn door Sacharias Jansen omstreeks 1590. Deze bestond uit drie beweegbare delen en twee lenzen en maakte vergrotingen van 3 tot 9 keer mogelijk, afhankelijk van de uitgerokken toestand. De eerste echte doorbraak binnen de microscopie kwam echter enkele jaren later dankzij Antoni van Leeuwenhoek. Van Leeuwenhoek was expert in de ontwikkeling van lenzen en het ontwerp van zijn microscoop (Figuur 1.2b) haalde daardoor vergrotingen tot 275 maal.

Dankzij deze vooruitgang wist Van Leeuwenhoek de eerste bacteriën en rode bloedcellen waar te nemen. Later kwam men zelfs tot een van de meest fundamentele eigenschappen binnen de biologie: organismen zijn opgebouwd uit cellen [1]. Steeds beter wordende

microscopen betekenden de start van een nieuw biologisch onderzoeks veld: de microscopische anatomie [2].

In de loop der jaren is enorm veel vooruitgang geboekt binnen de microscopie. Hedendaagse lichtmicroscopen zijn in staat tot vergrotingen van 2000 maal², andere geavanceerdere elektronenmicroscopen kunnen zelfs tot enkele miljoenen uitvergrooten, waardoor onderzoek op nanoschaal mogelijk gemaakt wordt. Hedendaagse elektronenmicroscopen zorgen echter voor een grote hoeveelheid beeldartefacten in waarnemingen, waardoor verder onderzoek sterk gehinderd wordt. Dit probleem zullen we trachten op te lossen in deze scriptie.

1.1.2 Soorten microscopen

Er bestaan verschillende soorten microscopen en afhankelijk van het type zal dit zijn voor- en nadelen hebben voor de toepassing. Het onderscheid in microscopen wordt bepaald door de manier van waarnemen. De meeste microscopen maken hiervoor gebruik van licht- of elektronenstralen. We bespreken kort de types die hiermee overeenstemmen.

1.1.2.1 Optische microscoop

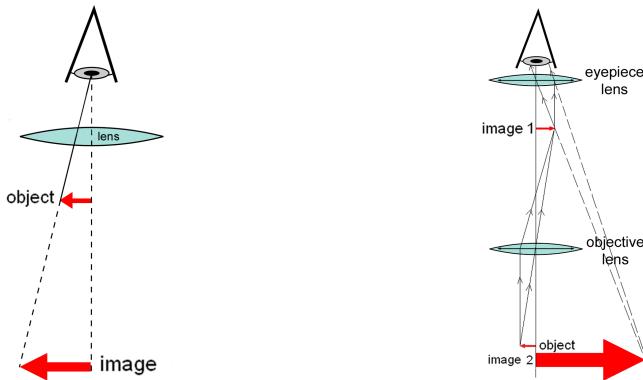
De meest bekende microscoop is de optische microscoop (lichtmicroscoop, OM) [3]. Deze maakt gebruik van lichtstralen en lenzen om zo tot een vergroot beeld te komen. Er bestaan twee soorten lichtmicroscopen:

- Enkelvoudige lichtmicroscoop: deze maakt gebruik van één lenssysteem (één of meerdere lenzen) om zo een vergroot beeld te vormen (Figuur 1.3a). In het geval van één lens in het systeem is dit niks anders dan een loep en zijn vergrotingen tot 20 maal mogelijk. Hoe meer lenzen gebruikt worden, hoe meer deze vergrotingsfactor kan oplopen.

Het eenvoudig concept geeft onmiddellijk een groot voordeel: de kostprijs ligt relatief laag. Ook zijn ze heel eenvoudig om mee te werken. Het grote nadeel aan deze microscopen is dat ze een lage vergrotingsfactor en dus minder toepassingsmogelijkheden hebben. Cellen en bacteriën kunnen hier bijvoorbeeld niet mee waargenomen worden.

- Samengestelde lichtmicroscoop: deze focussen eerst het licht tot een (licht vergroot) intern beeld binnen in de microscoop d.m.v. een objectief lenssysteem (één

²Er is geen gebrek aan technologie waardoor men slechts tot 2000 keer kan uitvergrooten met optische microscopen, maar een fysische limietgeving. Lichtstralen gedragen zich als golven en hebben een relatieve diffractielimiet waardoor de maximale vergrotingsfactor gelimiteerd is.



(a) De enkelvoudige microscoop: een object wordt door een lenssysteem omgevormd tot een groter virtueel beeld. (b) De samengestelde microscoop: een object wordt vergroot door het objectief en oculair lenssysteem.

Figuur 1.3: Principe van beeldvorming bij optische microscopie [4, 5]

of meerdere objectieflenzen). Een tweede oculair lenssysteem vergroot dit beeld uiteindelijk (Figuur 1.3b) tot het waargenomen beeld. De totale vergrotingsfactor is het product van de vergrotingsfactoren van de twee lenssystemen. Bij de meeste samengestelde lichtmicroscopen ligt dit tussen de 1000 en de 2000.

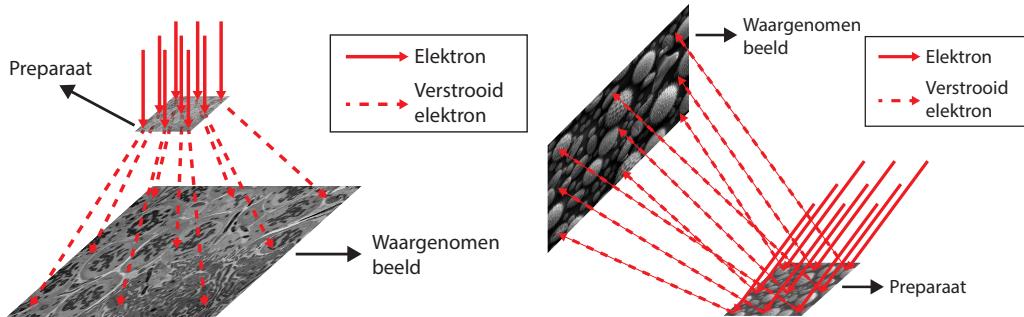
Het is duidelijk dat deze microscopen veel nuttiger zijn voor micro-onderzoek vergeleken met de enkelvoudige variant. Verder worden veel microscopen uitgerust met verschillende objectieflenzen zodat de vergroting makkelijk aangepast kan worden. De keerzijde van de medaille is echter dat de kostprijs hoger ligt.

Sommige lichtmicroscopen zijn uitgerust met CCD camera's waardoor er met digitale beelden kan gewerkt worden. Dit creëert tal van extra mogelijkheden op vlak van post-processing: optimalisatie, segmentatie, etc.

OM heeft tal van toepassingen binnen microbiologisch en farmaceutisch onderzoek, biotechnologie, medische diagnose, micro-elektronica, etc. OM is echter gelimiteerd tot micro-onderzoek, tegenover de hedendaagse geavanceerde elektronenmicroscopen die vergrotingen tot op nanoniveau halen.

1.1.2.2 Elektronenmicroscoop

OM is fysisch gelimiteerd tot een vergroting van ongeveer 2000 maal vanwege de relatief lage diffractielimiet van lichtgolven. Elektronenstralen gedragen zich echter eveneens als golven en hebben een veel hogere diffractielimiet. Bijgevolg zullen microscopen gebaseerd op elektronenstralen (zogenaamde elektronenmicroscopen, EM [6]) veel minder last hebben van deze fysische vergrotingslimiet. Een elektronenkanon en elektrostatische en -magnetische lenzen worden gebruikt om zo een object gericht met elektronen te bestralen. Dit kan op twee manieren gebeuren:



(a) Transmissie-elektronenmicroscoop: elektronen verplaatsen zich doorheen het oppervlak om een beeld te vormen.
 (b) Rasterelektronenmicroscoop: de verstrooide elektronen worden gebruikt om het waargenomen beeld op te stellen.

Figuur 1.4: Principe van beeldvorming bij elektronenmicroscopie

- Transmissie-elektronenmicroscoop (TEM): hierbij wordt een dun preparaat beschoten met elektronen. Afhankelijk van de structuur van het object, zullen er veel of weinig elektronen doorheen het preparaat geraken (Figuur 1.4a). De waargenomen elektronen zullen gedetecteerd worden en via een fotografische film of CCD camera voor het uiteindelijk beeld zorgen.

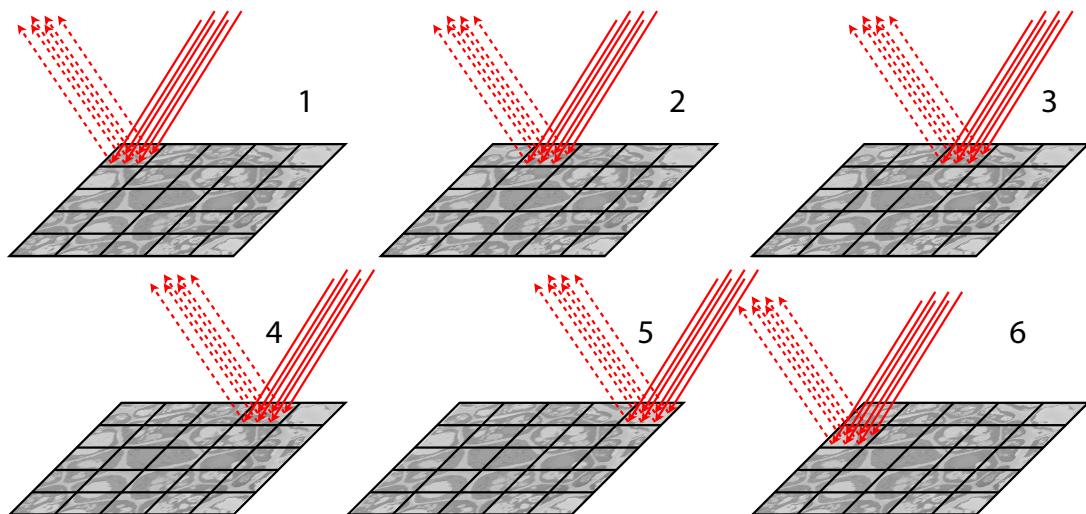
Deze EM's halen vergrotingen van 1 tot 50 miljoen keer en maken onderzoek op nanoschaal dus mogelijk. Een belangrijk nadeel is echter dat het preparaat heel dun ($\pm 100 \text{ nm}$) moet zijn om de elektronendoorgang te garanderen.

- Rasterelektronenmicroscoop (Scanning Electron Microscope, SEM): bij deze vorm van EM wordt een elektronenstraal gefocust. Deze elektronen zullen in een denkbeeldig rooster cel per cel³ het waar te nemen oppervlak bestralen. Bij impact van de elektronenstraal op het object wordt een deel van de elektronen opgenomen door het object en een ander deel verstrooid. De verstrooide elektronen worden gebruikt als informatiebron om een beeld op te bouwen (Figuur 1.4b).

Het scanning principe gebeurt meestal in horizontale richting. Het preparaat wordt denkbeeldig verdeeld in een fijn rooster. De cellen van het rooster zullen hierbij pixels in het uiteindelijk beeld voorstellen en kunnen tot $0,35 \text{ nm}^2$ klein zijn. De grootte van deze cellen bepaalt de resolutie van het resulterend beeld⁴. Vervolgens wordt een geconcentreerde elektronenstraal afgevuurd op een cel en zullen de verstrooide elektronen een pixelwaarde in die cel bepalen. De duur van het bestralen van een cel wordt ook wel de *dwell time* genoemd. Deze procedure wordt cel per cel, in horizontale volgorde, herhaald. Figuur 1.5 illustreert dit principe. Om ervoor te zorgen dat de elektronenstraal correct afgebogen wordt, moet deze via

³Met een cel bedoelen we een cel van het rooster, niet in de biologische betekenis.

⁴Op die manier kan van een preparaat van slechts $5 \mu\text{m}^2$ groot een 14 megapixel beeld opgenomen worden!



Figuur 1.5: Principe van horizontaal scannen

De elektronenstraal bestraalt elke cel van het rooster in horizontale richting om een beeld te vormen.



(a) Beeld met weinig scherptediepte: objecten veraf gelegen worden vager gezien dan objecten dichtbij.

(b) Beeld met veel scherptediepte: objecten veraf gelegen worden even scherp gezien als objecten dichtbij.

Figuur 1.6: Verschil in scherptediepte [7]

elektrostatische en -magnetische lenzen begeleid worden, net zoals optische lenzen lichtstralen afbuigen.

Tegenover TEM is het hierbij niet nodig om een dun preparaat te voorzien aangezien de teruggekaatste elektronen worden gedetecteerd. Het mechanisme zoals hierboven beschreven zorgt bovendien ook voor een grote scherptediepte. D.w.z. dat voorwerpen scherp gezien worden, of ze nu veraf of dichtbij gelegen zijn. Dit principe wordt geïllustreerd in Figuur 1.6a en 1.6b. De reden van deze grotere scherptediepte bij SEM is dat sommige elektronen dieper op het oppervlak binnendringen en vervolgens verstrooid worden.

- Raster- en transmissie-elektronenmicroscopie (Scanning Transmission Electron Microscopy, STEM): dit is een combinatie van de TEM en SEM waarbij informatie

bijgehouden wordt van zowel de doorgedrongen als gereflecteerde elektronen om een beeld te vormen.

Over het algemeen zijn er enkele nadelen aan EM: het is duur en vereist een uitgebreide opleiding om ermee aan de slag te gaan. Ook kan het aanmaken van preparaten enorm tijdsintensief zijn. De mogelijkheden zijn echter veel uitgebreider dan bij OM vanwege het extreme resolutieverschil. EM zal daarom meer gebruikt worden bij toepassingen op nanoschaal.

Een belangrijke toevoeging bij geavanceerde SEM is de mogelijkheid om beelden in 3D waar te nemen. De SEM maakt een opname van een gladgemaakt staal. Daarna wordt m.b.v. een diamantmes een dun slice van het staal gesneden. Vervolgens kan de SEM opnieuw een opname maken van het gladgemaakt staal. Deze vorm van 3D EM wordt ook wel *serial block-face* (SBF) SEM genoemd. Het resultaat is een stack van 2D EM-beelden die in 3D kan voorgesteld worden. Bij 3D EM moet men er echter wel rekening mee houden dat het preparaat vernietigd is na de beeldopname. Dit vormt meestal geen probleem, aangezien men op die manier een hoge resolutie voorstelling van het preparaat in 3D bekomt.

1.2 Probleemstelling

1.2.1 Acquisitie van 3D SEM-beelden

Een belangrijk gegeven is dat in 3D SEM de elektronenstraal en het ontwikkeld signaal aan tal van invloedsfactoren onderworpen zijn. Signaalversterking, hitte, magnetische lenzen, etc. zijn factoren die zorgen voor ongewenste effecten zoals ruis en wazigheid in het beeld.

Chronologisch ziet het opnameprocess van de 3D EM er als volgt uit:

1. Initieel wordt een gecontroleerde elektronenstraal afgevuurd vanuit het elektronenkanon en afgebogen a.d.h.v. elektrostatische en -magnetische lenzen om zo op het oppervlak van het preparaat in te vallen.
2. Bij inval op het preparaat zal de staat van de elektronen veranderen:
 - Een deel van de elektronen zal energie verliezen, dit wordt meestal gecomprimeerd door de afgave van hitte.
 - Een deel van de elektronen zal verstrooid worden. De richting van de verstrooiing is afhankelijk van de topologie van het oppervlak.

- Een deel van de elektronen zal interageren met het oppervlak en ervoor zorgen dat andere elektronen teruggestraald worden (zogenaamde *secundaire elektronen*).
3. De terugkerende elektronen worden terug afgebogen a.d.h.v. elektrostatische en -magnetische lenzen om door de detector waargenomen te worden.
 4. De detectoren verwerken de informatiestroom tot een signaal. Dit gebeurt d.m.v. het tellen van elektronen. Typisch zit hier een statistische fout op die gemodelleerd kan worden door een signaalafhankelijk Poisson proces. Afhankelijk van de dwell time moet dit signaal versterkt worden om het uiteindelijk beeld te vormen. Een grote dwell time vereist minder amplificatie dan een lage dwell time. Een toename in dwell time zorgt echter voor een toename in acquisitietijd. Dit kan tot artefacten zoals charging (het verschijnsel waarbij een opstapeling van elektronen vrij komt en op die manier voor vlekken in het beeld zorgt) leiden.
 5. Stappen 1 t.e.m. 4 worden voor elke cel (en resulterende pixelwaarde) herhaald in een horizontale scanrichting.
 6. Wanneer het volledig oppervlak waargenomen is, snijdt een diamantmes een nieuw slice van het preparaat dat opnieuw kan waargenomen worden.

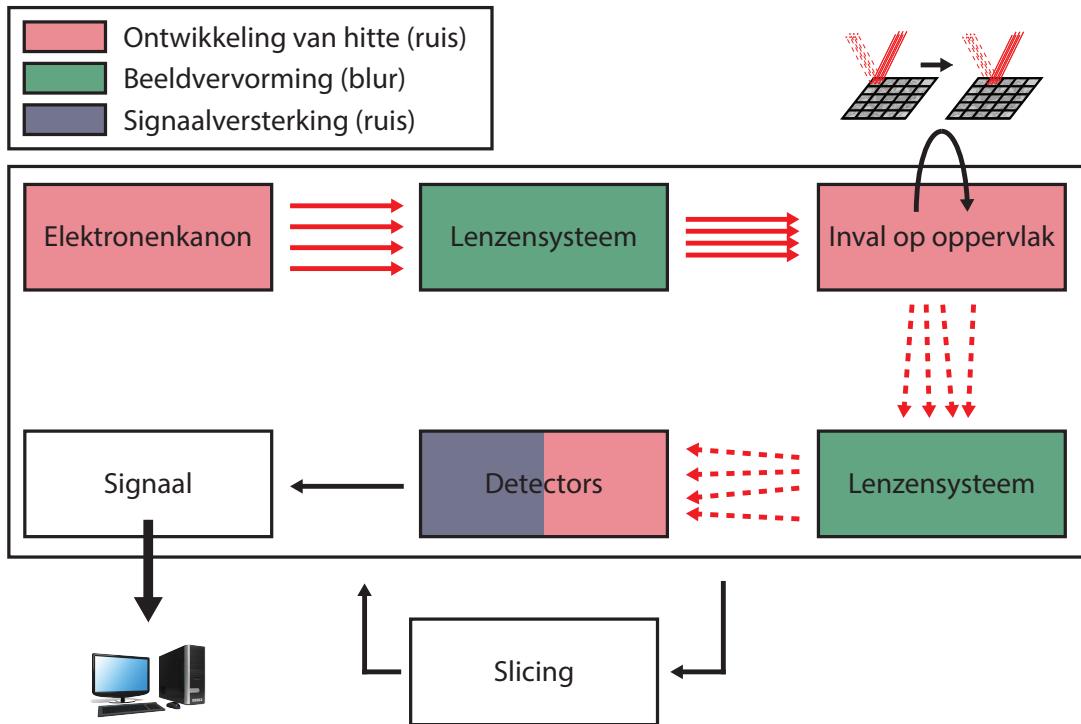
Het is duidelijk dat de reeds vernoemde storingsfactoren dus een invloed zullen hebben bij de acquisitie.

1.2.2 Problematiek bij de acquisitie

De beeldacquisitie van 3D SEM-beelden omvat tal van factoren die voor beeldartefacten kunnen zorgen. Twee van de meest belangrijke artefacten die we waarnemen in deze beelden zijn ruis en blur. Figuur 1.7 geeft een schematische weergave van de beeldacquisitie en zijn invloed op deze beeldartefacten. Dit kan bv. te wijten zijn aan ontwikkeling van hitte, signaalversterking, lenzen, etc.

Uiteraard is dit een sterk vereenvoudigd model van de realiteit. In praktijk zijn er vanwege o.a. de complexe elektronica en lenslimieten nog veel meer factoren die zorgen voor beeldartefacten:

- Ruis: beeldverwerkingsliteratuur leert ons dat deze elektronica en signaalversterking typisch voor (ev. signaalafhankelijke) ruis in het beeld zal zorgen.
- Blur: complexe lenssystemen hebben hun limieten op vlak van beeldvorming. Dit uit zich meestal in wazigere beelden (blur).



Figuur 1.7: Storingsfactoren bij 3D SEM

Dit zijn de twee belangrijkste beeldartefacten die we terugvinden in 3D SEM en zullen trachten te reduceren.

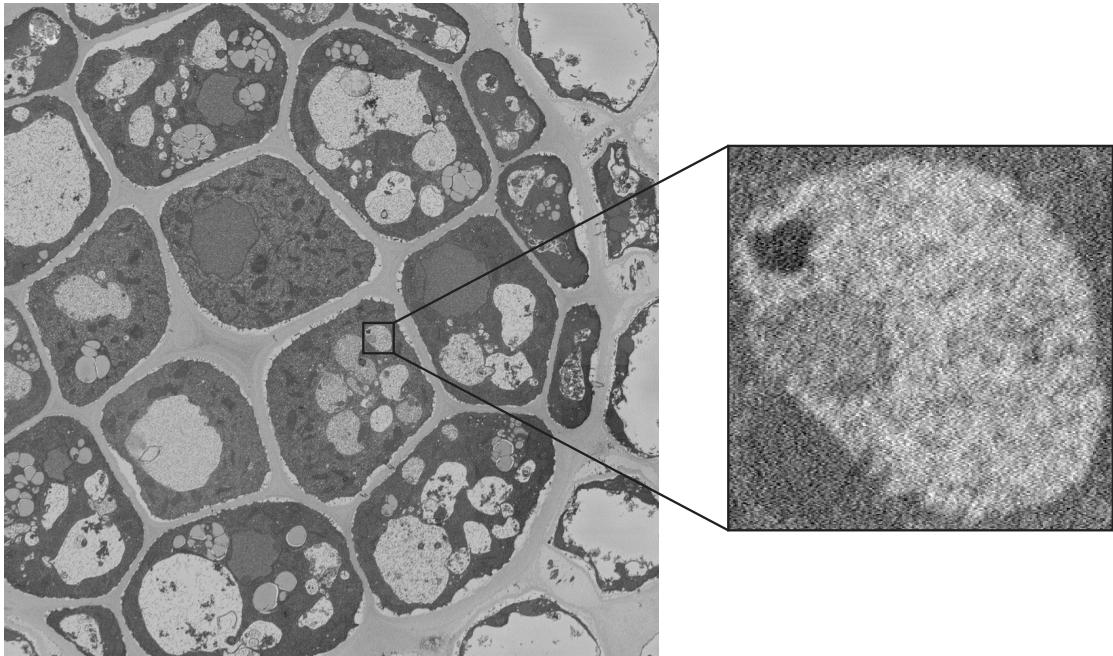
Opmerking 1.1. Men kan zich afvragen of het wel nodig is om deze beeldartefacten te verwijderen. De meeste geavanceerde EM systemen voorzien zelf een post-processing stap met als doel artefacten te verwijderen, dus uiteindelijk is er in de beelden misschien zelfs geen sprake van ruis of blur.

Dit is in de praktijk echter niet het geval. Als er nog een post-processing stap plaatsvindt, gebeurt dit meestal a.d.h.v. zeer eenvoudige methoden (laagdoorlaatfilter, Sectie 3.2.1.1) die bovendien extra nadelen hebben, zoals het introduceren van extra blur, terwijl de ruis nog niet allemaal verdwenen is. Kwalitatieve ruisonderdrukking en deblurring als post-processing is dus een noodzakelijke stap.

1.2.2.1 Ruis

In praktijk is er met de resulterende EM-beelden op het eerste zicht niks fout. Dit komt echter omdat we het beeld niet in volledige resolutie zien⁵. Als we close-ups van de beelden gaan bekijken, wordt het al moeilijker om structureel onderscheid te maken

⁵Vergelijk het met een kruisteken en een punt dat vanop verre afstand beide als een punt waargenomen worden. Op een gelijkaardige manier zien we de details niet in het uitgezoomd EM-beeld. Dit is het gevolg van een interpolatiestap die moet gebeuren wanneer een 100 megapixel beeld moet weergegeven worden op een 10 megapixel beeldscherm.



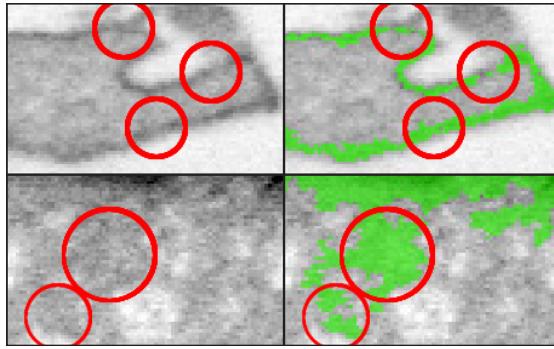
Figuur 1.8: Ruis in een EM-beeld

door de sterke aanwezigheid van ruis (Figuur 1.8⁶). De grote hoeveelheid ruis maakt het zelfs voor experts heel moeilijk om bv. bepaalde structuren in celkernen te analyseren.

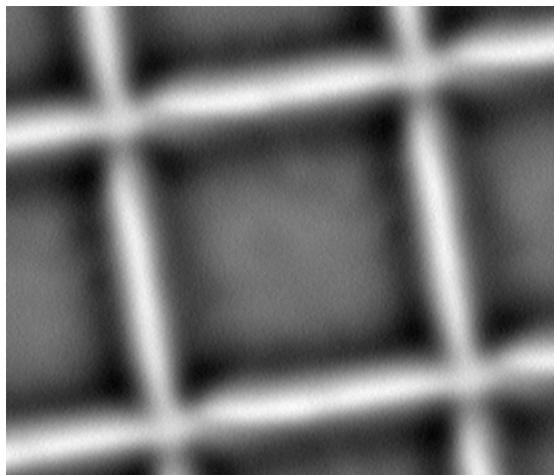
Een tweede probleem is de impact van ruis op verdere analyse en de foutieve resultaten en conclusies die hieruit volgen. EM-beelden worden voornamelijk gebruikt voor analyse en voor zover dit mogelijk is, probeert men dit zo veel mogelijk geautomatiseerd te doen. Echter, als de beelden onderhevig zijn aan een grote hoeveelheid ruis, zullen de meeste post-processingsalgoritmen dit niet in acht nemen en dus voor foutieve resultaten zorgen. Als voorbeeld nemen we een van de meest gebruikte beeldverwerkingsalgoritmen bij EM: segmentatie. Hierbij gaat men op zoek naar gelijkaardige structuren binnen een beeld (bv. het automatisch annoteren van alle celkernen). In praktijk zullen de randen van segmenten steeds langs randen van objecten liggen. Ruis kan deze randen echter sterk doen fluctueren waardoor onverwachte randen van segmenten gedetecteerd worden.

Figuur 1.9 illustreert de impact van ruis op segmentatie [8]. De groene regio's stellen de voorspelde segmenten voor. Als we de corresponderende regio's, aangeduid door de rode cirkels, echter met elkaar vergelijken, zien we dat de ruis zorgt voor ongewenste resultaten in de segmentatie. Enerzijds zorgt ruis in gebieden met veel textuur ervoor dat deze textuur niet gedetecteerd wordt. Bij het gesegmenteerd beeld bovenaan in Figuur 1.9 worden bv. bepaalde randen niet gedetecteerd door het algoritme vanwege de ruisverstoring. Anderzijds zorgt ruis in gebieden met weinig textuur ervoor dat

⁶Om een idee te geven van de omvang van deze beelden: moesten we het linkerbeeld in Figuur 1.8 op dezelfde schaal bekijken als het rechterbeeld, zouden we een beeld van één vierkante meter hebben.



Figuur 1.9: Segmentatie bij EM ruisbeelden.
De ruis zorgt voor incorrecte segmentatiegrenzen.



Figuur 1.10: Wazig EM-beeld

onverwachte textuur gedetecteerd wordt. Het gesegmenteerde beeld onderaan in Figuur 1.9 illustreert dit.

1.2.2.2 Blur

Zoals vermeld is er o.a. dankzij lenslimieten ook sprake van blur in de beelden. Dit valt sterk op wanneer we beelden waarnemen met een geometrische structuur. Een voorbeeld wordt gegeven in Figuur 1.10⁷ waar idealiter een scherp afgelijnd rooster zou moeten te zien zijn. In plaats daarvan zien we wazigere randen. Merk op dat deze blur, net zoals bij ruis, dezelfde gevolgen kan hebben voor automatische beeldanalyse. De wazigere randen maken het moeilijker om enerzijds manueel onderscheid te maken tussen bepaalde structuren. Anderzijds zorgt dit ook voor onverwachte resultaten bij post-processing beeldverwerkingsalgoritmen.

⁷Dit beeld werd bekomen door verschillende gelijkaardige opnames uit te middelen waardoor de hoeveelheid ruis afnam (zie Sectie 2.4). In praktijk is dit geen mogelijk werkwijze om beelden te reconstrueren, aangezien nooit meerder versies van eenzelfde beeld beschikbaar zijn.

1.2.3 Doelstelling

De bedoeling van deze scriptie is, zoals de titel zegt, het optimaliseren van 3D EM-beelden. Daarbij stoten we op de vraag: wat is een optimaal beeld? De meest succesvolle reconstructieproblemen binnen de beeldverwerking starten van een analyse van het probleem en simuleren dit vervolgens op referentiebeelden die overeenkomen met de realiteit. In het geval van 3D EM, echter, beschikken we niet over deze referentiebeelden. De enige informatiebron die beschikbaar is, zijn de beschadigde beeldversies. De analyse van de beeldartefacten zal dus van essentieel belang zijn om deze zo goed mogelijk te modelleren op referentiebeelden. Het is onze bedoeling om via deze grondige analyse, de huidige optimalisatietechnieken aan te passen, zodat een betere reconstructie kan uitgevoerd worden.

Voor de duidelijkheid dienen we nog op te merken dat optimale beelden zeker en vast niet hoeven overeen te komen met ‘mooiere’ beelden. De opgenomen EM-beelden komen (dankzij de toevoeging van ruis en blur) niet overeen met de werkelijkheid. De bedoeling van onze optimalisatiestap is om de opgenomen beelden meer te doen overeenstemmen met de realiteit, m.a.w. de beelden er meer doen uitzien zoals ze er zouden moeten uitzien. Hoe zo een beeld er moet uitzien, kan niet deterministisch vastgelegd worden, maar zal moeten bevestigd worden door experts.

In een EM-beeld kan het bv. zijn dat minuscule structuren ter grote van enkele pixels te zien zijn. Een slecht optimalisatie-algoritme zou dit beschouwen als ruis en bijgevolg verwijderen, iets wat we zeker en vast niet willen bereiken met onze methode.

Als laatste is het niet onze bedoeling om het optimalisatie-algoritme enkel en alleen af te stemmen op 3D EM. Bij de beeldreconstructie van ‘niet-EM’-beelden zou het dus enkel nodig moeten zijn om de parameters opnieuw in te stellen zodanig dat hier ook optimalisatie mogelijk is.

1.2.4 Werkplan

Zoals in voorgaande secties beschreven, is er een significante hoeveelheid ruis en blur aanwezig in de waargenomen EM-beelden. De analyse van deze ruis en blur zal van essentieel belang zijn om een optimale beeldreconstructie te garanderen. Zo is uit gerelateerd onderzoek reeds gebleken dat bv. gebruik makend van correlatie in de ruis een beter resultaat kan bekomen worden ([9–11]). In het volgende hoofdstuk zullen we karakterisatie-eigenschappen afleiden van de ruis en blur en die vervolgens gebruiken bij de beeldreconstructie.

Vervolgens zullen we, gebruik makend van de ruis- en blurkenmerken, de huidige state-of-the-art beeldoptimalisatietechnieken aanpassen om een kwalitatievere beeldreconstructie te realiseren.

Een directe aanpak zou bv. eerst de ruis en vervolgens de blur uit het beeld verwijderen. Dit zijn echter twee technieken die elkaar tegenwerken. Dikwijls zal het verwijderen van ruis het beeld vervagen⁸. Het compenseren van blur, daarentegen, benadrukt de aanwezigheid van ruis⁹.

Als leidraad kiezen we dus voor één enkele optimalisatietechniek die tegelijkertijd zowel ruis als blur verwijdert. Daarbij is het enerzijds de bedoeling om zo veel mogelijk ruis te verwijderen en de textuurregio's en randen intact te houden. Anderzijds willen we een blurcompensatie uitvoeren zonder ruis te versterken.

⁸De kunst ligt hem erin om het beeld te vervagen in regio's waar enkel en alleen ruis zal verwijderd worden. Het vervagen van textuur en randen zal resulteren in een zwakkere reconstructie.

⁹Zwakke randen worden bij deze technieken meestal scherper gemaakt, waardoor ruis dus extra benadrukt wordt.

Hoofdstuk 2

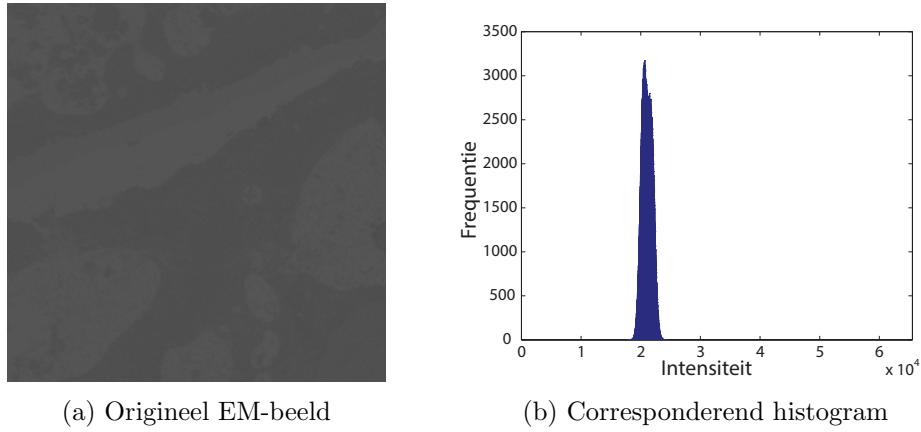
Analyse van EM beeldvorming en -degradatie

In Hoofdstuk 1 werd besproken hoe ruis en blur in EM-beelden verder onderzoek van deze beelden moeilijker maakt. Een uitgebreide analyse van deze beeldartefacten wordt voorgesteld om de reconstructie zo optimaal mogelijk te maken. In Sectie 2.1 beginnen we met een beschrijving van de 3D EM-beelden, aangezien deze niet overeenstemmen met dagdagelijkse afbeeldingen. Vervolgens stellen we in Sectie 2.2 een ruismodel op dat de beeldartefacten zo goed mogelijk modelleert. De parameters van dit model zullen enerzijds bepaald worden door de eigenschappen van de ruis. Deze worden uitvoerig besproken in Sectie 2.3. Anderzijds zal de lensblur eveneens in het ruismodel geïntegreerd zijn. De analyse hiervan gebeurt in Sectie 2.4. Uiteindelijk vatten we alles samen in Sectie 2.5 dat als startpunt zal dienen bij Hoofdstuk 3, waar onze reconstructietechniek wordt beschreven.

2.1 Beschrijving van de data

De meeste afbeeldingen die we in het dagelijks leven tegenkomen, zijn van het 8-bit formaat. D.w.z. dat voor elke pixel in dit beeld 8 bits per kleurkanaal voorzien worden. Een M bij N grijswaarde-afbeelding zal dus ongecomprimeerd MN bytes in beslag nemen. Vermits 8 bits vrijgehouden worden per pixel, zijn er dus $2^8 = 256$ verschillende intensiteiten toe te kennen aan een pixel¹. Een M bij N kleurenafbeeldingen zal ongecomprimeerd $3MN$ bytes in beslag nemen: MN bytes voor het rode, groene en blauwe kleurkanaal. Vermits 8 bits vrijgehouden worden per kanaal, per pixel, zijn er dus $(2^8)^3 = 16.777.216$ verschillende kleuren toe te kennen aan een pixel.

¹Doorgaans komt 0 overeen met zwart en 255 met wit.



Figuur 2.1: Origineel EM-beeld

EM-beelden, daarentegen, zijn van het 16-bit formaat². D.w.z. dat nu 16 bits per kanaal per pixel voorzien worden. Aangezien EM-beelden over het algemeen over één kanaal beschikken, zullen we ons in wat volgt enkel tot grijswaarde beelden beperken. Bijgevolg kan een pixel in een EM-beeld $2^{16} = 65536$ verschillende intensiteiten aannemen.

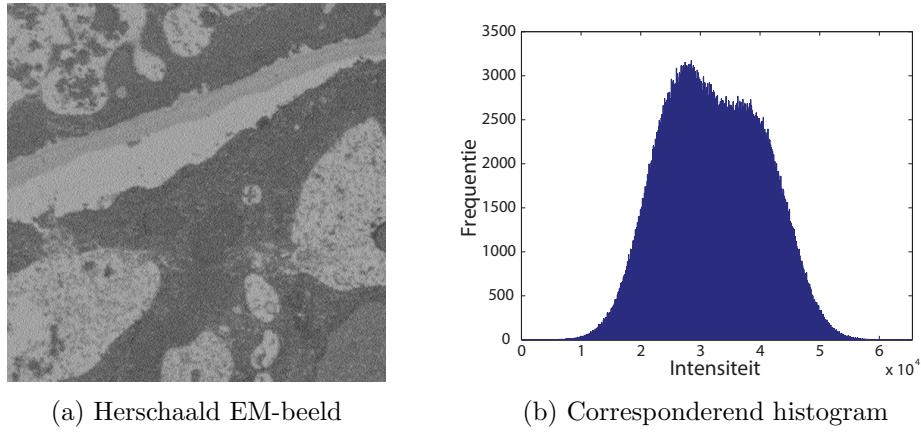
In Figuur 2.1a wordt een EM-beeld weergegeven, zoals het a.h.w. uit de microscoop komt. Op het eerste zicht is hier niet veel aan te zien. Als we naar het histogram (Figuur 2.1b) van dit beeld kijken, i.e. de verdeling van de pixelintensiteiten, merken we op dat de intensiteiten van het beeld weinig verschillen. Door het feit dat de beelden echter van 16-bit formaat zijn, wil dit zeggen dat de ze heel gedetailleerd zijn op intensiteitsniveau. Stellen we nu in een beeld \mathbf{x} (in matrixproductnotatie, Appendix A) $m = \min_i \mathbf{x}_i$ en $M = \max_i \mathbf{x}_i$, dan wordt $[m, M]$ het *dynamisch bereik* (*dynamic range*) van het beeld genoemd. Over het algemeen hebben EM-beelden een relatief klein dynamisch bereik. Om de beelden visueel duidelijker te maken, kunnen we het contrast doen toenemen door een lineaire transformatie op het beeld toe te passen waardoor het dynamisch bereik naar $[0, 2^{16} - 1]$ herschaald wordt:

$$\mathbf{x}' = \frac{\mathbf{x} - m}{M - m} (2^{16} - 1) \quad (2.1)$$

Deze herschaling is enkel en alleen voor voorstellingsdoeleinden en verandert op zich niks aan het beeld (de inverse transformatie kan eenvoudig terug uitgevoerd worden). Figuur 2.2a geeft de herschaalde versie van Figuur 2.1a weer. Het corresponderend histogram in Figuur 2.2b heeft duidelijk een breder dynamisch bereik.

Verder kan de x - en y -resolutie van een beeld voor opname ingesteld worden a.d.h.v. de celgrootte bij het scannen. De z -resolutie wordt bepaalt door de dikte van één slice. Dit is meestal van nanometerorde waardoor in de diepte ook aan hoge resolutie waargenomen

²Het doel hiervan is om over een grotere nauwkeurigheid te beschikken. Een 16 bit beeld beschikt over 2^8 meer intensiteiten dan een 8 bit afbeelding.



Figuur 2.2: Herschaald EM-beeld

kan worden. In praktijk varieert de grootte van de resulterende EM-beelden tussen 5000×5000 en 10000×10000 pixels, dit kan echter oplopen tot 64000×64000 pixels.

2.2 Ruismodel

Bij het analyseren van ruis of blur in een beeld is het van essentieel belang om dit principe goed te modeleren in het ruismodel. Het is immers de bedoeling dat we a.d.h.v. dit model het ‘ideaalbeeld’ kunnen onderscheiden van de ruis en blur zodat het gereconstrueerd beeld dit ideaalbeeld zo goed mogelijk benadert.

We zullen de ruis trachten te karakteriseren a.d.h.v. vier eigenschappen:

- Distributie: de verdeling van de ruis. Welke ruisintensiteiten komen veel of weinig voor? Ruis is meestal normaal verdeeld met gemiddelde 0 en constante variantie σ^2 , maar dit hoeft zeker en vast niet altijd zo te zijn. Zo hoeft de variantie bv. geen constante te zijn [12, 13].

Gaussiaanse ruis komt veel voor in bv. camerabeelden.

- Stationariteit: het is mogelijk dat de ruisstatistieken afhangen van de positie in het beeld [14]. Zo kan bv. langs de randen van het beeld meer ruis voorkomen vergeleken met de andere regio’s. In dat geval wordt de ruis *niet-stationair* genoemd. *Stationaire ruis* is onafhankelijk van de plaats in het beeld en gedraagt zich overal hetzelfde.

Niet-stationaire ruis komt dikwijls voor bij confocale microscopie.

- Spatiale correlatie: i.p.v. een spatiale afhankelijkheid vertonen ruispixels in veel gevallen een onderlinge afhankelijkheid [9–11, 15]. Dit is te herkennen aan bepaalde patronen die zich in de ruis herhalen. Dit type ruis wordt *gekleurde of gecorreleerde*

ruis genoemd. Als er geen onderlinge afhankelijkheid te bespeuren is, spreken we over *witte ruis*. Gecorreleerde ruis kan gemodelleerd worden d.m.v. een lineaire filter dat toegepast wordt op een wit ruissignaal.

Ruiscorrelatie is typisch te vinden in PAL televisiebeelden.

- Signaalafhankelijkheid: een andere mogelijkheid is dat de ruis afhankelijk is van het opgenomen signaal [16, 17]. Zo kan het bv. zijn dat in de donkere gebieden (lage intensiteit) er minder ruis aanwezig is, dan in heldere gebieden (hoge intensiteit). Microscopiebeelden gaan vaak gepaard met signaalafhankelijke ruis.

Een beeld vervagen (blurren) komt overeen met een laagdoorlaatfilter toepassen op het beeld. Deze filter karakteriseert de blur volledig.

We stellen het volgend ruismodel voor:

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \mathbf{H}\Sigma(\mathbf{x})\mathbf{n} \quad (2.2)$$

We werken hier met de matrixproductnotatie (Appendix A). De verschillende variabelen zijn als volgt gedefinieerd:

- $\mathbf{x} \in \mathbb{R}^{MN}$: vector die het ideaalbeeld voorstelt.
- $\mathbf{y} \in \mathbb{R}^{MN}$: vector die het waargenomen beeld voorstelt.
- $\mathbf{n} \in \mathbb{R}^{MN}$: dit zijn genormaliseerde, stochastische variabelen die de ruisdistributie modelleren. Met ‘genormaliseerd’ bedoelen we dat $\text{E}[\mathbf{n}] = \mathbf{0}$ en $\text{Var}(\mathbf{n}) = \mathbf{I}$. Meestal volgt dit een multivariate, normale verdeling $\mathcal{N}(\mathbf{0}, \mathbf{I})$.
- $\mathbf{H} \in \mathbb{R}^{MN \times MN}$: circulaire matrixvoorstelling van het filter corresponderend met de correlatie in de ruis. Zonder verlies van algemeenheid zorgen we er voor dat dit filter, genormaliseerd is, d.w.z. de som van alle elementen van het filter is 1.
- $\Sigma(\mathbf{x}) : \mathbb{R}^{MN} \rightarrow \mathbb{R}^{MN \times MN}$; deze functie modelleert de signaalafhankelijke standaardafwijking van de ruis. Diagonalelement $(\Sigma(\mathbf{x}))_{i,i}$ drukt de standaardafwijking van de ruis uit, gegeven dat de i -de pixel intensiteit \mathbf{x}_i heeft. In het geval van signaalafhankelijkheid is dit simpelweg $\sigma\mathbf{I}$ met σ de constante standaardafwijking.
- $\mathbf{G} \in \mathbb{R}^{MN \times MN}$: circulaire matrixvoorstelling van het laagdoorlaatfilter corresponderend met de blur. Zonder verlies van algemeenheid zorgen we er voor dat dit filter, genormaliseerd is, d.w.z. de som van alle elementen van het filter is 1.

Opmerking 2.1. In bovenstaand model wordt additieve ruis verondersteld. Dit is een assumptie die we moeten maken vanuit praktisch standpunt. Een model van de vorm $\mathbf{y} = f(\mathbf{x})$, met f een ruistoevoegende functie, zou te algemeen zijn om significante resultaten af te leiden. Bovendien dienen we ook op te merken dat signaalafhankelijke ruis geen additief proces is. Het voorgestelde model houdt echter de signaalafhankelijkheid in rekening door de ruisterm met een signaalafhankelijke matrix te vermenigvuldigen.

Opmerking 2.2. Merk op dat $\Sigma(\mathbf{x})$ een diagonalmatrix is. Signaalafhankelijke correlatie tussen de ruispixels wordt dus genegeerd en we beschouwen enkel de standaardafwijking van de ruis als ‘mogelijk signaalafhankelijk’.

Opmerking 2.3. De blur wordt rechtstreeks gemodelleerd op het ideaalbeeld \mathbf{x} , terwijl we ook zouden kunnen argumenteren dat de blur eveneens in het spel komt nadat de ruis toegevoegd is aan het ideaalbeeld. Een ruismodel dat dit principe volgt, zou er als volgt uitzien:

$$\mathbf{y} = \mathbf{G}'' (\mathbf{G}' \mathbf{x} + \mathbf{H}' \Sigma(\mathbf{x}) \mathbf{n}) \quad (2.3)$$

Hierin modelleren \mathbf{G}' en \mathbf{G}'' respectievelijk de blur voor en na ruistoevoeging en stelt \mathbf{H}' de ruiscorrelatie voor. Het model beschreven door Vergelijking 2.3 kan equivalent geschreven worden als:

$$\mathbf{y} = \mathbf{G}'' \mathbf{G}' \mathbf{x} + \mathbf{G}'' \mathbf{H}' \Sigma(\mathbf{x}) \mathbf{n} \quad (2.4)$$

Dit ruismodel is equivalent aan het voorgesteld model. Daarvoor stellen we $\mathbf{G} = \mathbf{G}'' \mathbf{G}'$ en $\mathbf{H} = \mathbf{G}'' \mathbf{H}'$ in Vergelijking 2.2.

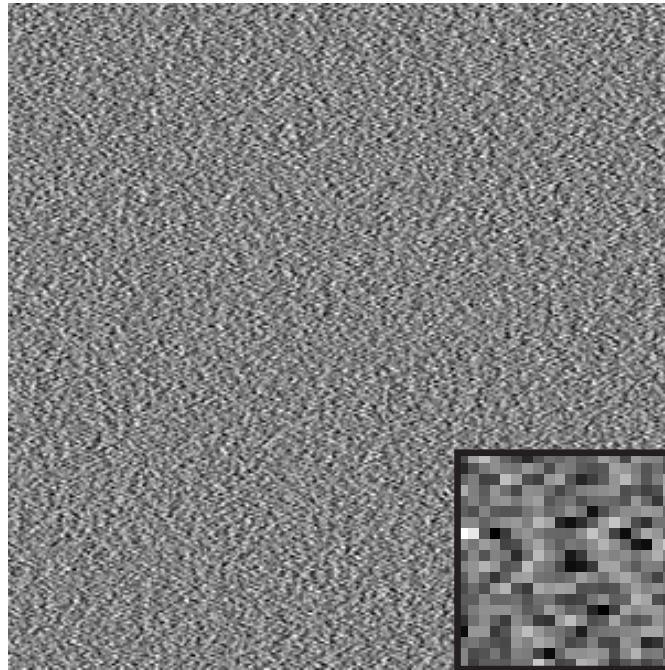
Opmerking 2.4. Tenslotte dienen we ook nog te benadrukken dat we beeldrestauratie uitvoeren op 2D beelden, terwijl de titel van deze thesis wel degelijk 3D beelden vernoemt. Bij de opname van een 2D EM-beeld is het mogelijk dat elektronen zich opstapelen in de onderliggende lagen van het preparaat. Dit kan zijn invloed hebben op volgende slices en dus ook op de 3D-structuur. Voor de eenvoud gaan we er echter van uit dat er geen afhankelijkheid is tussen opeenvolgende slices, wanneer die waargenomen worden.

2.3 Ruisanalyse

2.3.1 Afzonderen van de ruis

Om een kwalitatief onderzoek te kunnen uitvoeren op de ruis in de EM-beelden, dienen we dit op één of andere manier af te zonderen van de beelden. Dit hebben we experimenteel gedaan door de elektronenstraal in de EM uit te schakelen³. Op die manier

³Vergelijk dit met het uitschakelen van het licht in een kamer en vervolgens een foto te nemen.



Figuur 2.3: Waargenomen ruisbeeld in de afwezigheid van een elektronenstraal. Grijs correspondeert met intensiteit nul, rechtsonder is een close-up weergegeven van de ruis.

wordt geen signaal gedetecteerd en is dus $\mathbf{x} = \mathbf{0}$ in Vergelijking 2.2:

$$\mathbf{y} = \mathbf{H}\Sigma(\mathbf{0})\mathbf{n} \quad (2.5)$$

Het resultaat is een beeld (Figuur 2.3) dat enkel en alleen bestaat uit de ruiscomponent waar we de ruisparameters uit kunnen schatten. We dienen op te merken dat bij verder onderzoek op deze beelden automatisch signaalafhankelijkheid geldt. Volgens het ruismodel wordt dit immers gemodelleerd door een signaalafhankelijke $\Sigma(\mathbf{0})$.

Willen we de signaalafhankelijkheid mee betrekken in het schatten van de parameters, dan gaan we als volgt te werk:

- We zoeken een regio met verwachte intensiteit i en zo weinig mogelijk textuur of randen in een willekeurig EM-beeld, m.a.w. we zoeken een gebied waarvoor we kunnen aannemen dat \mathbf{x} constante intensiteit i heeft: $\mathbf{x} = \mathbf{i}$. Hierbij stelt \mathbf{i} een vector voor die uitsluitend bestaat uit i 's.

- Voor deze regio geldt:

$$\begin{aligned}
 \mathbf{y} &= \mathbf{G}\mathbf{i} + \mathbf{H}\Sigma(i)\mathbf{n} \\
 &\Downarrow \\
 \mathbf{y} - \mathbf{i} &= \mathbf{G}\mathbf{i} - \mathbf{i} + \mathbf{H}\Sigma(i)\mathbf{n} \\
 &= \mathbf{i} - \mathbf{i} + \mathbf{H}\Sigma(i)\mathbf{n} \\
 &= \mathbf{H}\Sigma(i)\mathbf{n}
 \end{aligned}$$

Merk op dat $\mathbf{G}\mathbf{i} = \mathbf{i}$ vanwege de normalisering van \mathbf{G} en Eigenschap B.3 in de Appendix.

In praktijk is het niet evident om elke intensiteit i terug te vinden in testbeelden. Aan gezien we veronderstellen dat de ruis \mathbf{n} verwachtingswaarde $\mathbf{0}$ heeft, vinden we echter:

$$\begin{aligned}
 \mathbb{E}[\mathbf{y} - \mathbf{i}] &= \mathbb{E}[\mathbf{H}\Sigma(i)\mathbf{n}] \\
 &= \mathbf{H}\Sigma(i)\mathbb{E}[\mathbf{n}] \\
 &= \mathbf{0} \\
 &\Downarrow \\
 \mathbb{E}[\mathbf{y}] &= \mathbb{E}[\mathbf{i}] \\
 &= \mathbf{i}
 \end{aligned}$$

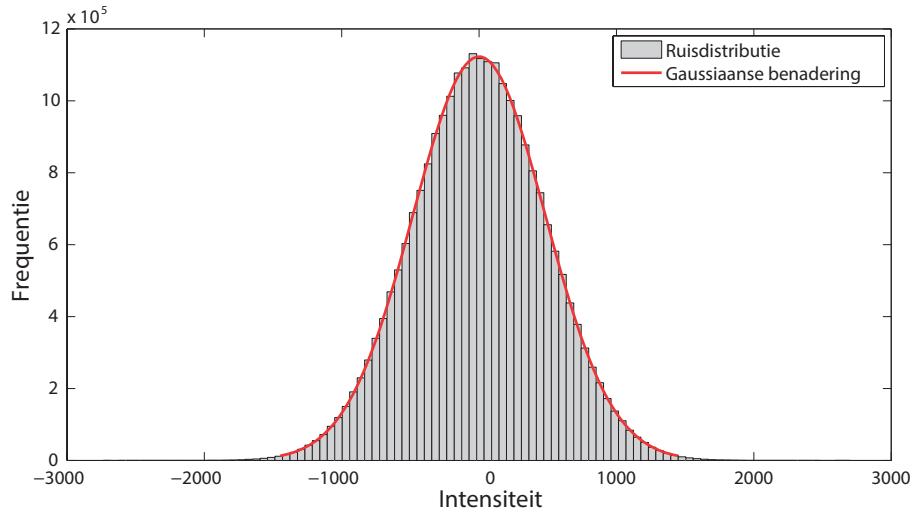
Voor verschillende patches \mathbf{y} zullen we de corresponderende intensiteit dus schatten d.m.v. het gemiddelde:

$$\hat{i} = \frac{\sum_{k=0}^{MN-1} (\mathbf{y})_k}{MN} \quad (2.6)$$

Om de signaalafhankelijkheid mee te betrekken in het systeem, zullen we dus zo veel mogelijk regio's van 'constante' intensiteit i zoeken. Deze intensiteit zal geschat worden door het gemiddelde te nemen van de waargenomen pixels \mathbf{y} . Als we deze waarde \hat{i} aftrekken van het waargenomen beeld, krijgen we het ruisbeeld dat onderzocht kan worden op de vernoemde ruiskarakteristieken.

2.3.2 Distributie

Zoals eerder vermeld, is de meest voorkomende ruisverdeling, de normale (Gaussiaanse) verdeling (afkomstig van de elektronica in opnameapparatuur). In de meeste gevallen kan dit zelfs nog vereenvoudigd worden naar witte, Gaussiaanse ruis. De meeste reconstructie-algoritmen, die als doel hebben om zo veel mogelijk ruis te verwijderen, vertrekken dan ook van deze assumptie [18–20].



Figuur 2.4: Distributie ruisintensiteiten

Vermits de ruis bij de EM voornamelijk veroorzaakt wordt door de complexe elektronica, verwachten we hier ook Gaussiaanse ruis. Dit wordt bevestigd als we de verdeling van de ruisintensiteiten bekijken in Figuur 2.4. Ook bij de aanwezigheid van signaal blijkt de ruis zich volgens een normale verdeling te gedragen.

Om een idee te geven hoe dicht de verdeling bij de Gaussiaanse verdeling aansluit, zullen we een Chi-square goodness-of-fit test [21] uitvoeren. In deze test zullen we uitgaan van de nulhypothese H_0 : elke ruispixel gedraagt zich volgens een normale verdeling. Vervolgens heeft men voor de i -de ruispixel enerzijds een werkelijke waarde \mathbf{n}_i (de waargenomen ruisintensiteit), anderzijds kan op basis van de nulhypothese een verwachte waarde $\hat{\mathbf{n}}_i$ bepaald worden voor deze ruispixel. Het kwadratisch verschil tussen deze twee waarden zou klein moeten zijn als de nulhypothese waar is. Dit verschil wordt relatief bekeken t.o.v. de verwachte waarde. Vervolgens worden deze relatieve verschillen gesommeerd over alle ruispixels. De Chi-square goodness-of-fit test berekent dus volgende teststatistiek:

$$\chi^2 = \sum_{i=0}^{MN-1} \frac{(\mathbf{n}_i - \hat{\mathbf{n}}_i)^2}{\hat{\mathbf{n}}_i}$$

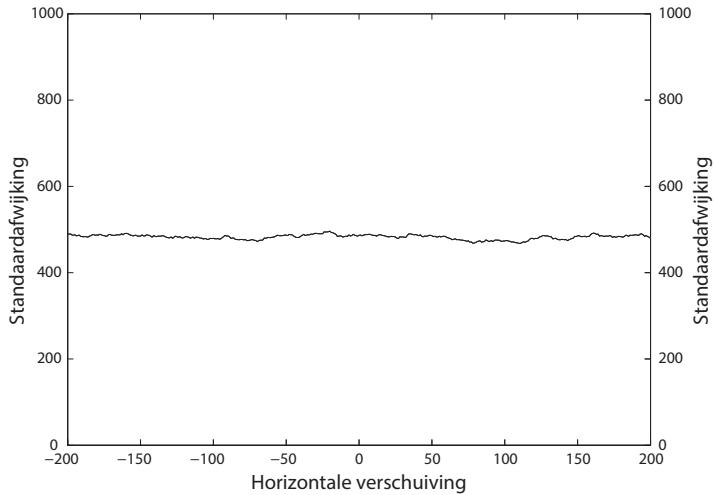
Als de nulhypothese waar is, zou dit een χ^2 -distributie moeten volgen. De berekende waarde χ^2 zou bovendien klein moeten zijn. Verder kan deze test ook bepalen hoe groot de kans is om een grotere waarde dan χ^2 te verkrijgen (onder de nulhypothese) bij een herhaling van het experiment. Dit wordt ook wel de *p-waarde* genoemd. Een kleine *p-waarde* betekent dus dat de nulhypothese hoogstwaarschijnlijk een foutieve aanname is.

Tabel 2.1 geeft de *p-waarden* die corresponderen met een Chi-square goodness-of-fit test uitgevoerd op ruispatches met verschillende achtergrondintensiteit. We zien dat voor

Intensiteit	0	21186	21426	21614
<i>p</i> -waarde	0,2791	0,3076	0,2639	0,3816
Intensiteit	22757	23013	23252	23799
<i>p</i> -waarde	0,9510	0,6891	0,5135	0,4775

Tabel 2.1: *p*-waarden Chi-square goodness-of-fit test.

Hoge *p*-waarden duiden op het feit dat de nulhypothese waarschijnlijk waar is.



Figuur 2.5: Stationariteit van de ruis

geen enkele intensiteit de *p*-waarde aan de lage kant (i.e. kleiner dan een significantieniveau $\alpha = 0,05$) is. We kunnen dus nergens besluiten dat de nulhypothese foutief is. Bijgevolg kunnen we vaststellen dat de ruis (veroorzaakt door de elektronica in het systeem) Gaussiaans verdeeld is, onafhankelijk van het waargenomen signaal. We kunnen dus in ons ruismodel (Vergelijking 2.2) $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ stellen.

2.3.3 Stationariteit

In deze sectie zullen we nagaan of er een spatiale afhankelijkheid is in de ruis. Hiervoor zullen we verschillende ruisblokken beschouwen en de standaardafwijking van de ruis in dit blok meten. Op die manier kunnen we de hoeveelheid ruis in functie van de positie in het beeld weergeven en zijn mogelijke afhankelijkheden eenvoudig te detecteren.

Spatiale onafhankelijkheid zou willen zeggen dat deze berekende standaardafwijkingen $\tilde{\sigma}(m, n)$ constant zijn in functie van de positie. Dit is het geval als de standaardafwijking $\text{std}(\tilde{\sigma})$ gelijk is aan 0. Hoe dichter dit bij 0 ligt, hoe constanter de functie $\tilde{\sigma}(m, n)$ zich gedraagt. Figuur 2.5 geeft de functie $\tilde{\sigma}$ weer voor een constante n_0 en variabele m : een 50×50 blok werd geselecteerd in een ruisbeeld en vervolgens naar links en rechts

Intensiteit	0	21186	21426	21614
std($\tilde{\sigma}$)	3,072	3,442	4,236	3,718
Intensiteit	22757	23013	23252	23799
std($\tilde{\sigma}$)	6,373	3,578	3,438	5,272

Tabel 2.2: Stationariteit van de ruis i.f.v. de intensiteit

verschoven. Wetende dat de standaardafwijking van de ruis hoogstens een 10 à 20-tal intensiteitseenheden verschilt in functie van de verschuiving en rekening houdend met het feit dat de maximale intensiteit 65535 is, mogen we deze grafiek als een constante beschouwen.

In de horizontale richting is bijgevolg geen afhankelijkheid te bespeuren. Aangezien het resultaat moeilijker weer te geven is voor 2D verschuivingen en voor verschillende waargenomen intensiteiten, worden in Tabel 2.2 de standaardafwijkingen std($\tilde{\sigma}$) gegeven voor variabele intensiteit. We merken op dat deze waarden inderdaad dicht bij 0 liggen. We kunnen uit deze cijfers afleiden dat de standaardafwijking, ongeacht de signaalintensiteit, nooit meer dan 7 intensiteitseenheden is, wat verwaarloosbaar is tegenover een typisch dynamisch bereik [20000, 25000]. We mogen dus besluiten dat de ruis stationair is, onafhankelijk van het waargenomen signaal.

2.3.4 Spatiale correlatie

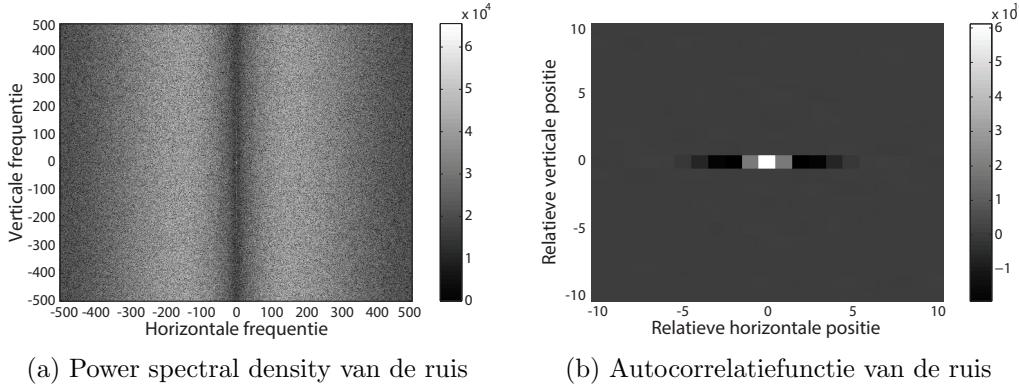
In deze sectie willen we \mathbf{H} uit het ruismodel (Vergelijking 2.2) schatten. Deze modelleert de onderlinge afhankelijkheden tussen ruispixels. Daarvoor zullen we gebruik maken van de power spectral density en de autocorrelatiefunctie:

Definitie 2.5. Zij $x \in \mathbb{R}^{M \times N}$ een beeld in convolutienotatie (Appendix A). Dan is de *power spectral density* PSD(x) van x gedefinieerd als:

$$(\text{PSD}(x))_{\omega_i, \omega_j} = |X_{\omega_i, \omega_j}|^2 \quad (2.7)$$

waarbij X de discrete fouriertransformatie (DFT) van x is.

De power spectral density PSD(x) van x drukt dus uit hoe sterk een bepaalde frequentie (ω_i, ω_j) vertegenwoordigd is in het beeld. De PSD van een ruisbeeld zal ons dus vertellen welke ruisovergangen het meest of minst aan bod komen in de ruis.



Figuur 2.6: Spatiale correlatie van de ruis
We stellen correlatie in de horizontale richting vast.

Definitie 2.6. Zij $x \in \mathbb{R}^{M \times N}$ een beeld in convolutienotatie (Appendix A). Dan is de *autocorrelatiefunctie* $\text{ACF}(x)$ van x gedefinieerd als:

$$(\text{ACF}(x))_{i,j} = (x * x')_{i,j} = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} x_{k,l} x_{i-k,j-l} \quad (2.8)$$

waarbij $x'_{i,j} = x_{-i,-j}$ en we veronderstellen dat het beeld x periodiek uitgebreid wordt.

De autocorrelatiefunctie $\text{ACF}(x)$ van x filtert dus x met een gespiegelde versie van zichzelf. Op die manier drukt $(\text{ACF}(x))_{i,j}$ de correlatie uit tussen een willekeurige pixel op positie (k, l) en de pixel op positie $(k - i, l - j)$.

De PSD en ACF van een ruisbeeld zijn weergegeven in respectievelijk Figuren 2.6a en 2.6b. Om illustratiedenen is de PSD logaritmisch herschaald. We merken op dat de ACF enkel weergegeven is in een centraal 10×10 venster, aangezien de rest van de ACF bestaat uit nullen.

Aan het PSD kunnen we zien dat er geen significante veranderingen optreden in de verticale richting, d.w.z. elke verticale frequentie (horizontale randen) is even sterk vertegenwoordigd in de ruis. In de horizontale richting is dit echter niet het geval. Hier zien we een sterkere vertegenwoordiging van middelhoge horizontale frequenties tegenover de lage en hoge horizontale frequenties. Het gevolg is dat we een horizontaal streepjespatroon herkennen in de EM ruis (Figuur 2.3).

De ACF bevestigt dit streepjespatroon door een horizontale correlatie tussen nabijgelegen pixels.

Merk op dat deze horizontale correlatie consistent is met de horizontale scanrichting van de EM.

Ook bij aanwezigheid van signaal merken we dezelfde correlatie op. We kunnen dus besluiten dat de ruis gecorreleerd is. Meer specifiek, is de ruis horizontaal, maar niet

verticaal gecorreleerd. Deze correlatie is bovendien onafhankelijk van het waargenomen signaal. De ACF geeft de correlatie weer tussen nabijgelegen ruispixels. Bijgevolg zal de genormaliseerde ACF (d.i. een herschaling zodat de som van de elementen gelijk is aan 1) een goede schatting zijn voor het filter dat correspondeert met de matrix \mathbf{H} .

2.3.5 Signaalafhankelijkheid

De signaalafhankelijkheid van de ruis wordt gemodelleerd via de diagonalmatrix $\Sigma(\mathbf{x})$. Het diagonalelement $(\Sigma(\mathbf{x}))_{i,i}$ drukt de standaardafwijking van de ruis uit, gegeven dat pixelintensiteit x_i waargenomen werd. Het is dus onze bedoeling om een functie $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ te vinden die de standaardafwijking zo goed mogelijk benadert in functie van de waargenomen intensiteit. Vervolgens kunnen we dan eenvoudig $(\Sigma(\mathbf{x}))_{i,i} = \sigma(x_i)$ stellen om $\Sigma(\mathbf{x})$ te schatten. De gezochte σ zullen we bepalen a.d.h.v. een eenvoudiger model.

Veronderstel dat we signaalafhankelijke, witte ruis modelleren:

$$\begin{aligned} y &= x + \eta(x) \\ &= x + \sigma(x)n \end{aligned}$$

waarbij x en y respectievelijk het ideaalbeeld en waargenomen beeld voorstellen en η het ruissignaal dat opgesplitst werd in een signaalafhankelijkheidsfunctie σ en ruisverdelingsfunctie n die verwachtingswaarde 0 en standaardafwijking 1 heeft. Merk op dat we met de convolutienotatie werken (Appendix A).

Zoals eerder opgemerkt, valt ruis (veroorzaakt door elektronica) meestal te modelleren a.d.h.v. een Gaussiaans proces. Om signaalafhankelijkheid mee in dit model op te nemen, wordt echter meestal een extra Poissonverdeling (zie Sectie 1.2.2) mee in acht genomen [12, 22]. Dit volgt uit het feit dat bij het tellen van de elektronen statistische fouten optreden die gemodelleerd kunnen worden via een Poisson verdeling:

$$\eta(x_{i,j}) = \eta_p(x_{i,j}) + \eta_g \quad (2.9)$$

waarbij η_p de signaalafhankelijkheid modelleert en η_g de oorspronkelijke Gaussiaanse ruis. De twee ingevoerde verdelingen karakteriseren zich als volgt:

$$\begin{aligned} \chi(x_{i,j} + \eta_p(x_{i,j})) &\sim \mathcal{P}(\chi x_{i,j}) \\ \chi(\eta_g) &\sim \mathcal{N}(0, \sigma_0^2) \end{aligned}$$

waarbij $\chi > 0, \sigma_0 \geq 0$ constanten zijn en \mathcal{P} en \mathcal{N} respectievelijk de Poisson en Gaussiaanse verdeling voorstellen. Aangezien voor een Poissonverdeling $\mathcal{P}(\lambda)$ geldt dat de

variantie en verwachtingswaarde gelijk zijn aan λ , vinden we:

$$\mathrm{E}[\chi(x_{i,j} + \eta_p(x_{i,j}))] = \mathrm{Var}(\chi(x_{i,j} + \eta_p(x_{i,j}))) = \chi x_{i,j} \quad (2.10)$$

Aangezien

$$\mathrm{E}[\chi(x_{i,j} + \eta_p(x_{i,j}))] = \chi x_{i,j} + \chi \mathrm{E}[\eta_p(x_{i,j})] \quad (2.11)$$

en

$$\mathrm{Var}(\chi(x_{i,j} + \eta_p(x_{i,j}))) = \chi^2 \mathrm{Var}(\eta_p(x_{i,j})) \quad (2.12)$$

vinden we door Vergelijking 2.10 en 2.11 dat $\mathrm{E}[\eta_p(x_{i,j})] = 0$ en door Vergelijking 2.10 en 2.12 dat $\mathrm{Var}(\eta_p(x_{i,j})) = \frac{x_{i,j}}{\chi} = \alpha x_{i,j}$ als we $\alpha = \chi^{-1}$ stellen. Bijgevolg is enerzijds

$$\begin{aligned} \mathrm{Var}(y_{i,j}) &= \mathrm{Var}(\sigma(x_{i,j})n_{i,j}) \\ &= \sigma(x_{i,j})^2 \mathrm{Var}(n_{i,j}) \\ &= \sigma(x_{i,j})^2 \end{aligned}$$

en anderzijds

$$\begin{aligned} \mathrm{Var}(y_{i,j}) &= \mathrm{Var}(\eta_p(x_{i,j})) + \mathrm{Var}(\eta_g) \\ &= \alpha x_{i,j} + \sigma_0^2 \end{aligned}$$

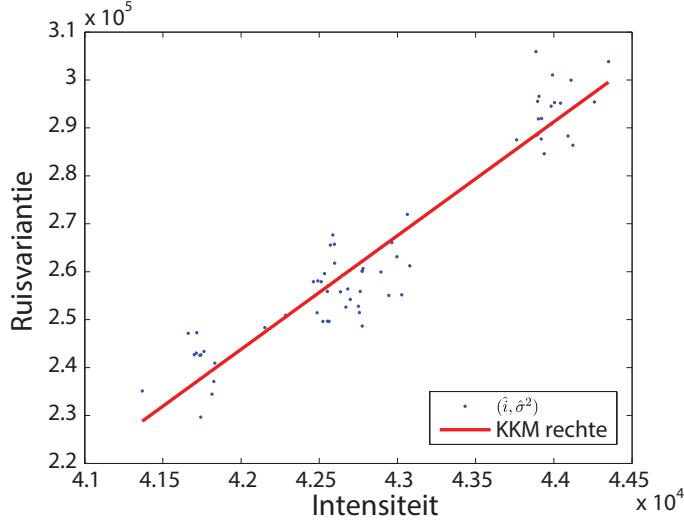
We vinden dus

$$\sigma(x_{i,j}) = \sqrt{\alpha x_{i,j} + \sigma_0^2} \quad (2.13)$$

Opmerking 2.7. De variabelen α en σ_0 hebben een specifieke betekenis in deze context. α illustreert de mate van signaalafhankelijkheid: $\alpha = 0$ resulteert in een signaalonafhankelijkheid, terwijl $\alpha \neq 0$ er voor zorgt dat per intensiteitseenheid de variantie van de ruis met α toeneemt. σ_0 stelt de standaardafwijking van de ruis voor, in afwezigheid van signaal. In het geval $\alpha = 0$ is dit eenvoudigweg de standaardafwijking van de ruis, onafhankelijk van het waargenomen signaal.

Via het vereenvoudigd model, zullen we dus veronderstellen dat de signaalafhankelijkheidsmatrix van de vorm $(\Sigma(\mathbf{x}))_{i,i} = \sigma(\mathbf{x}_i) = \sqrt{\alpha \mathbf{x}_i + \sigma_0^2}$ is. Er rest ons dus enkel nog een afschatting van de parameters α en σ_0^2 .

Deze schatting zal gebeuren door patches te beschouwen in arbitraire beelden met zo weinig mogelijk textuur en randen (net zoals in Sectie 2.3.1). We zullen van deze patches \mathbf{y} veronderstellen dat ze een constante intensiteit i hebben, die we schatten a.d.h.v. Vergelijking 2.6. Deze intensiteit wordt afgetrokken van het patch \mathbf{y} zodat we een schatting van het ruisbeeld bekomen. Vervolgens wordt hiervan de standaardafwijking $\hat{\sigma}$ berekend. Op die manier vinden we voor verschillende intensiteiten \hat{i} de corresponderende



Figuur 2.7: Signaalafhankelijkheid van de ruis

standaardafwijking $\hat{\sigma}$ van de ruis die het signaal met zich mee brengt. De punten $(\hat{i}, \hat{\sigma}^2)$ zouden dan volgens Vergelijking 2.13 een linear verloop moeten vertonen, waardoor we via de kleinste-kwadratenmethode (KKM) een schatting kunnen maken voor de gezochte parameters:

$$(\hat{\alpha}, \hat{\sigma}_0^2) = \arg \min_{(\alpha, \sigma_0^2)} \sum_{\hat{i}} \left(\hat{\sigma}^2 - (\alpha \hat{i} + \sigma_0^2) \right)^2 \quad (2.14)$$

Figuur 2.7 geeft de punten $(\hat{i}, \hat{\sigma}^2)$ weer en de corresponderende oplossingsrechte uit de kleinste-kwadratenmethode. De oplossing van Vergelijking 2.14 is in dit geval $(\hat{\alpha}, \hat{\sigma}_0^2) = (23.73, -75290)$. Merk op dat $\hat{\sigma}_0^2 < 0$ ten gevolge van twee zaken:

- De beelden die uit de EM komen, zijn geïnverteerd.
- Vanwege het klein dynamisch bereik van de EM-beelden, wordt door de EM een voor ons onbekend achtergrondsignaal T bij de beelden opgeteld.

Deze twee feiten bij elkaar zorgen er voor dat voor een waargenomen beeld \mathbf{x} , dit in werkelijkheid overeenstemt met $2^{16} - \mathbf{x} - \mathbf{T}$. De punten $(\hat{i}, \hat{\sigma}^2)$ komen dus niet overeen met de waargenomen intensiteit \hat{i} , maar met een verschoven versie ervan ($2^{16} - \hat{i} - T$). Aangezien T voor ons onbekend is, kunnen we σ_0^2 niet vastleggen en zal dit dus een parameter van het model worden.

Het is duidelijk dat er sprake is van signaalafhankelijkheid, aangezien $\alpha \neq 0$ en voor significante verschillen in ruisvariantie zorgt⁴.

2.4 Lensblur

De vervaging in EM-beelden wordt (zoals in vrijwel alle opname apparatuur) veroorzaakt door een complex lenzensysteem. De modellering hiervan kan gebeuren a.d.h.v. een laagdoorlaatfilter h . Om dit laagdoorlaatfilter te schatten, hebben we gekozen voor een eenvoudige aanpak. Een aantal beelden y_m ($m = 0, 1, \dots, K-1$) met duidelijke, repetitieve, geometrische eigenschappen (bv. een rooster) werden opgenomen door de EM:

$$y_m = h * x + n'_m$$

We werken opnieuw in convolutienotatie (Appendix A). Merk op dat we verwachten dat het ideaalbeeld x en de blurkernel h hetzelfde blijven bij de verschillende opnames. De waargenomen beelden zouden dus enkel mogen verschillen door het toegevoegd ruis-signal n'_m . Om de grote hoeveelheid ruis te verwijderen, hebben we de beelden y_m geregistreerd t.o.v. elkaar⁵ en vervolgens uitgemiddeld:

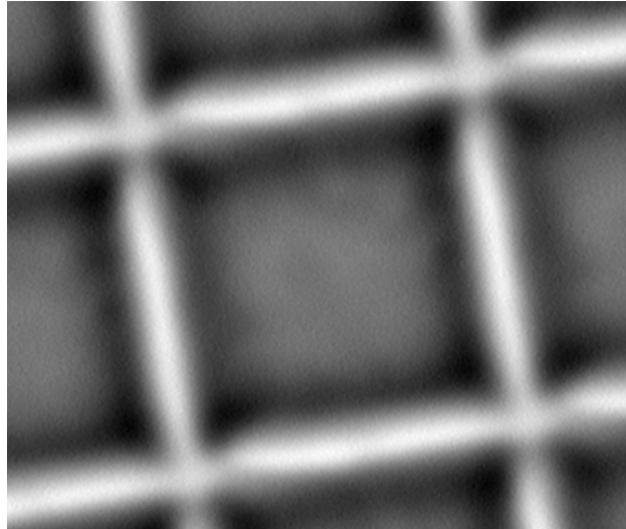
$$\begin{aligned} y &= \frac{1}{K} \sum_{m=0}^{K-1} y_m = \frac{1}{K} \sum_{m=0}^{K-1} (h * x + n'_m) \\ &= h * x + \frac{1}{K} \sum_{m=0}^{K-1} n'_m \end{aligned}$$

Als we voor de eenvoud veronderstellen dat de variantie van de ruis σ^2 is, dan zal de variantie van de uitgemiddelde ruis gelijk zijn aan:

$$\begin{aligned} \text{Var} \left(\frac{1}{K} \sum_{m=0}^{K-1} n'_m \right) &= \frac{1}{K^2} \sum_{m=0}^{K-1} \text{Var}(n'_m) \\ &= \frac{K\sigma^2}{K^2} \\ &= \frac{\sigma^2}{K} \end{aligned}$$

⁴Het grootste verschil in standaardafwijking komt overeen met de standaardafwijking, gemeten bij maximale, vergeleken met die bij minimale intensiteit. Dit is gegeven door: $\sqrt{\alpha(2^{16}-1)+\sigma_0^2} - \sqrt{\alpha 0 + \sigma_0^2} = 843$ met $\alpha = 23.73$ en $\sigma_0 = 500$. D.w.z. dat de ruis een standaardafwijking heeft die varieert tussen de 500 en 1343, afhankelijk van de waargenomen intensiteit.

⁵Bij het meermaals opnemen van eenzelfde beeld, kan het gebeuren dat dit verschoven en/of geroeteerde versies van elkaar zijn. Registratie-algoritmen zullen deze verschuivingen en rotaties ongedaan maken zodat de beelden overlappen.



Figuur 2.8: Geblurred ‘ruisvrij’ rooster

De standaardafwijking van de ruis zal dus met een factor \sqrt{K} afnemen. In ons geval was $K = 1000$, hetgeen de hoeveelheid ruis met een factor 31 doet dalen.

Op deze manier werd dus een nagenoeg ruisvrij beeld (Figuur 2.8) geconstrueerd dat enkel vervaging vertoont.

Vervolgens zouden we het verwacht geometrisch beeld x artificieel construeren en uit x en y het laagdoorlaatfilter h schatten op de volgende manier. In het fourierdomein komt de blurveroorzakende convolutie overeen met een pixelgewijs product (Eigenschap C.2 in de Appendix). Bijgevolg kan het filter h in het fourierdomein gevonden worden door:

$$H = \frac{Y}{\max(\varepsilon, X)}$$

waarbij X , Y en H respectievelijk de discrete fouriertransformatie van x , y en h zijn en de parameter ε een kleine constante is die eventuele numerieke instabiliteit vermindert. Omwille van deze gewenste stabiliteit hebben we het liefst dat X nagenoeg overal constante intensiteit heeft, hoewel niet-nul elementen in theorie eigenlijk voldoende zijn. Het perfecte ideaalbeeld x zou bestaan uit één enkele puntbron⁶. In dat geval zou $X = 1$ en bijgevolg $H = Y$.

Opmerking 2.8. Zoals te zien is in Figuur 2.8 wordt geen biologisch preparaat gebruikt om de blurkernel te schatten. In plaats daarvan wordt een metalen draad (gelegen op een waarnemingsvlak) waargenomen. Merk op dat dit zorgt voor een schaduweffect dat langs de randen te zien is. Van deze draad kan niet op dezelfde manier als bij biologische preparaten een 3D opname gemaakt worden, aangezien het diamantmes nu geen slices kan maken. Bijgevolg wordt op een andere manier waargenomen dan bij biologische

⁶Strikt gesproken kan zo een beeld in realiteit niet opgenomen worden aangezien beelden discreet gesampled worden. In de continue zin zou dit willen zeggen dat $x = \delta(x_1, y_2)$ met δ de diracdelta distributie en x_1 en x_2 continue reële variabelen.

samples en wordt de blurkernel niet volledig correct geschat. Het geeft echter wel al een idee hoe deze er uit zal zien. Een ideaal experiment zou deze draad inwerken in een preparaat zodanig dat hier op dezelfde manier als bij biologische samples een 3D opname van gemaakt kan worden. Dit is echter niet mogelijk, aangezien het diamantmes niet gemaakt is om door metaal te snijden.

Opmerking 2.9. We dienen bij deze methode op te merken dat overblijvende ruis in het beeld een zeer grote invloed heeft. Het viel ons eveneens op dat het beeld in Figuur 2.8 niet volledig ruisvrij is. Om een beter resultaat te krijgen, ging de dwell time maximaal ingesteld worden zodat reeds bij opname de ruis beperkt bleef⁷. Jammer genoeg werd op dit ogenblik een technisch defect bij de 3D EM vastgesteld en konden de beelden dus nog niet opgenomen worden.

Voor de algemeenheid zullen we dus in wat volgt er van uit gaan dat h niet gekend is, maar een parameter is in het model.

2.5 Conclusie

In Hoofdstuk 1 werd besproken welke beeldartefacten aanwezig waren in de EM-beelden. Hoofdzakelijk ging het daarbij om twee factoren: ruis en blur. Beide factoren kunnen beschreven worden door een aantal eigenschappen die opgenomen werden in het ruismodel (Vergelijking 2.2). Gebruik makend van deze eigenschappen, zal de ruis en blur op een zo efficiënt mogelijke manier verwijderd worden om een optimale beeldreconstructie te garanderen.

Om ruis te karakteriseren hebben we gebruik gemaakt van vier typerende eigenschappen: distributie, stationariteit, spatiale correlatie en signaalafhankelijkheid.

De distributie van de ruis bleek Gaussiaans te zijn, onafhankelijk van de waargenomen intensiteit. Dit is mooi meegenomen, aangezien de meeste beeldreconstructiealgoritmen van de veronderstelling van Gaussiaanse ruis uitgaan.

De ruis vertoont geen afhankelijkheden op basis van zijn positie in het beeld. We kunnen dus van stationaire ruis spreken, onafhankelijk van de waargenomen intensiteit.

De ruis bleek echter wel spatiaal gecorreleerd te zijn. Ze vertoont een duidelijk horizontaal streepjespatroon wat duidt op een correlatie in de horizontale richting. Onze analyse heeft dit bevestigd, onafhankelijk van de waargenomen intensiteit.

Tenslotte vonden we ook signaalafhankelijkheid in de ruis. Bij hoge intensiteiten blijkt de ruis sterker te variëren dan bij lage intensiteiten.

⁷Hoe meer elektronen kunnen geteld worden per pixel, hoe beter de intensiteitsvoorstelling en dus hoe minder ruis t.o.v. signaal.

Op vlak van de blur in de beelden kunnen we nog geen sluitende conclusies trekken. We zullen de blurkernel dan ook als parameter voorstellen in het model. In praktijk zullen we meestal gebruik maken van een Gaussiaanse blurkernel aangezien dit een van de meest courante modellen is.

De uitdaging zit hem nu in het feit dat de ruis niet alleen gecorreleerd is, maar ook signaalafhankelijk. Bovendien willen we hier ook nog eens het blurkenmerk aan toevoegen. Een reconstructie-algoritme dat deze drie eigenschappen in rekening neemt, is in de literatuur niet te vinden. We zullen in het volgende hoofdstuk dit probleem echter aanpakken.

Hoofdstuk 3

Beeldreconstructie

In Hoofdstuk 2 werd een analyse uitgevoerd rond de ruis en blur in EM-beelden. Dit was noodzakelijk om het ruismodel (Vergelijking 2.2) exact te kunnen beschrijven en het reconstructiealgoritme zo optimaal mogelijk te doen werken. We zullen de afgeleide eigenschappen gebruiken vertrekende van hedendaagse technieken. We beginnen met een kadering van ons ruismodel binnen de literatuur in Sectie 3.1. In Sectie 3.2 geven we een kort overzicht van enkele technieken die vandaag de dag gebruikt worden (niet alleen binnen EM) op vlak van ruisonderdrukking en het wegwerken van blur. Sectie 3.3 dient vervolgens als een aanloop naar het ontwikkeld algoritme. Uiteindelijk wordt in Sectie 3.4 het voorgesteld algoritme besproken.

3.1 Voorgesteld ruismodel

In het vorige hoofdstuk hebben we voorgesteld om met het volgende ruismodel verder te werken:

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \mathbf{H}\Sigma(\mathbf{x})\mathbf{n} \quad (3.1)$$

met \mathbf{x} en \mathbf{y} respectievelijk het ideaalbeeld en waargenomen beeld, \mathbf{G} , \mathbf{H} en $\Sigma(\mathbf{x})$ modelleren respectievelijk de blur, spatiale ruiscorrelatie en -signaalafhankelijkheid. Uit een gegeven beeld \mathbf{y} zullen we dus een schatting $\hat{\mathbf{x}}$ proberen maken die zo dicht mogelijk bij het ideaalbeeld \mathbf{x} gelegen is.

In de volgende sectie zullen we enkele reconstructiealgoritmen bespreken. Bemerk dat geen van deze technieken alle aspecten van ons beeld- en degradatiemodel beschouwt. We vinden in de literatuur geen enkel ruismodel zoals hetgeen wij voorstellen, in plaats daarvan wordt enkel van gecorreleerde ruis uitgegaan, of enkel van blur, etc.

3.2 Literatuurstudie

Binnen de beeldverwerking is beeldreconstructie een van de meest besproken onderwerpen in de literatuur. Het is dan ook vanzelfsprekend dat we beginnen met een literatuurstudie en een overzicht van de technieken die vandaag de dag beschikbaar zijn om ruis en/of blur te verwijderen uit beschadigde beelden. Deze technieken zullen bovendien ook een startpunt zijn om het uiteindelijk algoritme uit af te leiden.

3.2.1 Ruisonderdrukking

Ruisonderdrukking is een veelbesproken en uitdagend probleem omdat het in realiteit veel voorkomt en niet eenvoudig op te lossen is. Storingen in het signaal (vanwege complexe elektronica e.d.) zorgen ervoor dat het opgenomen beeld geen perfecte voorstelling is van het ideaalbeeld dat we eigenlijk zouden moeten waarnemen. In plaats daarvan wordt (in het geval van additieve ruis) de som gezien van een ideaalbeeld en een ongewenst ruissignaal. Ruisonderdrukking heeft als doel zo veel mogelijk van dit ruissignaal te verwijderen. De uitdaging zit hem dan in het zo goed mogelijk benaderen van het ideaalbeeld.

3.2.1.1 Laagdoorlaatfilter

Het laagdoorlaatfilter is een van de meest eenvoudige technieken om ruis te verwijderen. Aangezien deze binnen EM de meest bekende en gebruikte techniek is, beschrijven we de voor- en nadelen hiervan kort.

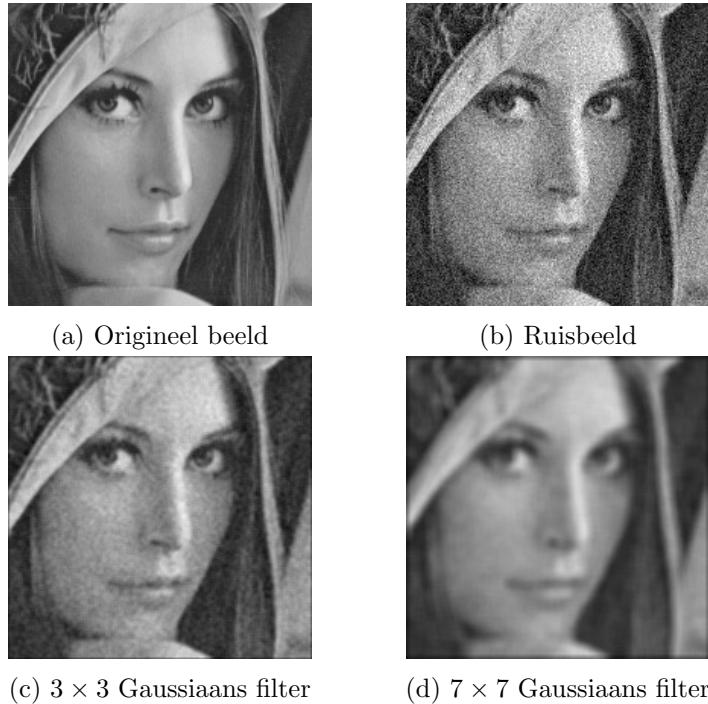
Het idee van een laagdoorlaatfilter is om lokaal pixels (al dan niet gewogen) uit te middelen. Aangezien ruis gemiddeld 0 is, zou door deze lokale uitmiddeling de ruis onderdrukt moeten worden.

Concreet start men van het volgende eenvoudig ruismodel:

$$\mathbf{y} = \mathbf{x} + \sigma \mathbf{n}$$

waarbij \mathbf{x} en \mathbf{y} respectievelijk het ideaalbeeld en waargenomen beeld voorstellen, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ stelt het ruissignaal voor en σ is een positieve constante die ervoor zorgt dat de ruis variantie σ^2 heeft. Een laagdoorlaatfilter \mathbf{H} zal dan de ruis proberen onderdrukken:

$$\begin{aligned} \mathbf{Hy} &= \mathbf{Hx} + \mathbf{H}\sigma\mathbf{n} \\ &\approx \mathbf{Hx} \end{aligned}$$



Figuur 3.1: Ruisonderdrukking met een laagdoorlaatfilter

Het is daarbij de moeilijke taak om de approximatie ($\mathbf{H}\sigma\mathbf{n} \approx \mathbf{0}$) zo correct mogelijk uit te voeren. Bovendien willen we \mathbf{x} benaderen en mag dus \mathbf{Hx} hier niet veel van verschillen. De keuze van het filter \mathbf{H} is dus allesbepalend.

Het grote voordeel van laagdoorlaatfilters is dat ze heel eenvoudig en snel werken. Het probleem is echter dat meestal een compromis gevonden moet worden tussen enerzijds ruis onderdrukken en anderzijds textuur en randen behouden, aangezien pixels uitmidden er voor zorgt dat randen en textuurregio's worden vervaagd. In Figuren 3.1c en 3.1d is te zien dat een 3×3 filter enerzijds niet genoeg ruis verwijdert, maar een 7×7 filter textuur en randen gaat vervagen. Dit zorgt er dus voor dat details in beelden zullen verdwijnen. Bijgevolg zijn deze filters niet aan te raden als op de gereconstrueerde beelden verdere analyse moet uitgevoerd worden.

3.2.1.2 Anisotrope diffusie

Laagdoorlaatfilteren is een effectieve manier om ruis te verwijderen. Ze zorgen er echter voor dat veel van de textuur en randen vervaagd worden in het gereconstrueerd beeld. Het is perfect toegelaten om sterk te filteren langs ‘vlakke’ regio’s, maar langs randen en textuur moet dit beperkt worden. Een slimmere techniek zou gebruik maken van de beeldstructuur om zo gericht te gaan filteren.

Anisotrope diffusie is een alternatieve reconstructietechniek die start van het idee om een laagdoorlaatfilter h iteratief toe te passen op het beeld $x^{(0)} = x$:

$$\begin{aligned} x^{(t)} &= h * x^{(t-1)} \\ &= h * h * \dots * h * x^{(0)} \end{aligned}$$

Merk op dat we terug overschakelen naar de convolutienotatie (Appendix A). Na elke toepassing van het laagdoorlaatfilter zal het beeld waziger en waziger worden, en zal dus meer en meer ruis verwijderd worden. Zoals reeds vermeld, zal dit echter er ook voor zorgen dat randen en textuur zullen vervagen. Het is dus de bedoeling om bij textuur- en randomgevingen niet te veel gebruik te maken van het laagdoorlaatfilter, maar in vlakkere regio's wel zodanig dat de ruis zo goed mogelijk verwijderd wordt. Het nadeel hierbij is echter dat ruis nabij randen en textuur eveneens niet verwijderd wordt. Anisotrope diffusie zal daarvoor in bepaalde richtingen laagdoorlaatfilteren, en in andere niet¹.

Wanneer de parameter t bij de gefilterde beelden als tijdsvariabele beschouwd wordt, kan hiermee de lineaire diffusievergelijking beschouwd worden:

$$\frac{\partial x^{(t)}}{\partial t} = \nabla^2 x^{(t)}$$

waarbij ∇^2 de Laplacian differentiaal operator voorstelt. Dit kan iteratief benaderd worden door:

$$x^{(t)} = x^{(t-1)} + \nabla^2 x^{(t-1)}$$

Anisotrope diffusie (Perona - Malik [23]) maakt echter gebruik van het volgende model:

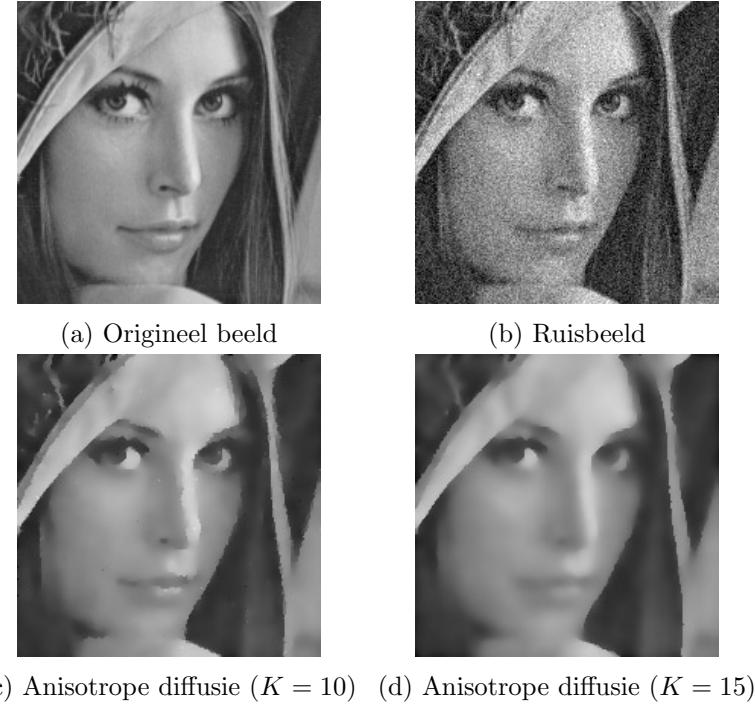
$$x^{(t)} = x^{(t-1)} + \nabla \cdot \left(\mathbf{C} \left(\nabla x^{(t-1)} \right) \nabla x^{(t-1)} \right) \quad (3.2)$$

Hierin zijn ∇ en $\nabla \cdot$ respectievelijk de gradiënt en divergentie operator:

$$\begin{aligned} \nabla x &= \begin{pmatrix} \frac{\partial x}{\partial i} \\ \frac{\partial x}{\partial j} \end{pmatrix} \\ \nabla \cdot \begin{pmatrix} f_i \\ f_j \end{pmatrix} &= \frac{\partial f_i}{\partial i} + \frac{\partial f_j}{\partial j} \end{aligned}$$

De matrix \mathbf{C} in Vergelijking 3.2 is in feite een functie van twee variabelen (de twee componenten van $\nabla x^{(t-1)}$) en zal a.h.w. ‘meten’ hoe sterk het beeld reliëf (randen) vertoont in de horizontale en verticale richting. Stellen we $\nabla x^{(t)} = (g_i, g_j)^T$, dan worden

¹Ruis nabij een verticale zwarte lijn, zal bv. verwijderd worden als we enkel in de verticale richting laagdoorlaatfilteren, zonder daarbij de zwarte lijn te vervagen.

(c) Anisotrope diffusie ($K = 10$) (d) Anisotrope diffusie ($K = 15$)

Figuur 3.2: Ruisonderdrukking met anisotrope diffusie

in [23] volgende definities voor \mathbf{C} voorgesteld:

$$\begin{aligned}\mathbf{C}(\nabla x^{(t)}) &= \begin{pmatrix} \exp\left(-\frac{g_i^2}{K^2}\right) & 0 \\ 0 & \exp\left(-\frac{g_j^2}{K^2}\right) \end{pmatrix} = \begin{pmatrix} c_i & 0 \\ 0 & c_j \end{pmatrix} \\ \mathbf{C}(\nabla x^{(t)}) &= \begin{pmatrix} \frac{1}{1+g_i^2/K^2} & 0 \\ 0 & \frac{1}{1+g_j^2/K^2} \end{pmatrix} = \begin{pmatrix} c_i & 0 \\ 0 & c_j \end{pmatrix}\end{aligned}$$

met K een vooraf bepaalde constante. Veronderstel dat we in het beeld op een sterke verticale rand zijn. Dan zal $g_i = \partial x / \partial i$ zeer groot zijn en $g_j = \partial x / \partial j$ zeer klein. Bijgevolg zal $c_i \approx 0$ en $c_j \approx 1$. Vergelijking 3.2 vereenvoudigt dan tot:

$$x^{(t)} = x^{(t-1)} + \frac{\partial^2 x^{(t-1)}}{\partial j^2}$$

We zien dus dat in dit geval enkel uitgemiddeld wordt in de verticale richting, wat er dus voor zorgt dat ruis langs deze randen eveneens zal verwijderd worden.

Merk op dat grote waarden voor K er voor zorgen dat details vervaagd zullen worden. Voor grote K waarden zal immers de matrix \mathbf{C} meer en meer op de eenheidsmatrix gelijken. Dit heeft als gevolg dat de richting van ruisverwijdering minder een rol zal spelen en de anisotrope diffusie zich als een laagdoorlaatfilter zal gedragen (Figuur 3.2d). Zoals vermeld in Sectie 3.2.1.1 zijn laagdoorlaatfilters ideaal om textuur te vervagen. Verder dienen we ook op te merken dat voor kleine K waarden het gereconstrueerd beeld

nog steeds ruis zal bevatten aangezien de matrix \mathbf{C} meer en meer op de nulmatrix begint te gelijken in dat geval. De iteratie in Vergelijking 3.2 zal bijgevolg weinig verandering brengen aan het ruisbeeld.

Anisotrope diffusie is een beeldreconstructietechniek die efficiënt ruis verwijdert (Figuur 3.2c). Toch heeft deze methode ook zijn minpunten. Door het feit dat enkel de horizontale en verticale richting in rekening gebracht worden, zal denoising langs schuine randen nog steeds een probleem vormen. Verder is deze methode zeer parameterafhankelijk en kan ze een significante hoeveelheid blur met zich meebrengen. Zo is in Figuur 3.2d een groot deel van de textuur verdwenen in het gereconstrueerd beeld. Randen die dan wel overgenomen wordt, komen soms eerder artificieel over in het gereconstrueerd beeld (bv. in het haar).

3.2.1.3 Non-local means

Laagdoorlaatfilters en anisotrope diffusie hebben beide als kenmerk dat ze lokaal pixels uitmidden. Op die manier onderdrukken ze enerzijds ruis, anderzijds zorgen ze ook voor beeldartefacten langs randen en textuur (de ene methode al wat meer dan de andere). Het is echter verstandiger om niet alleen lokaal pixels uit te middelen, maar over de hele afbeelding (of een grotere regio) gewogen te gaan uitmidden.

De non-local means (NMLS) techniek [24] is een van de betere hedendaagse ruisonderdrukkingsmethoden. In tegenstelling tot de meeste technieken die gebruik maken van lokale gelijkenissen in een beeld, zal NLMS eveneens niet-lokaal repetitieve structuren uitbuiten. Het idee achter NLMS is dat pixelwaarden gewogen worden uitgemiddeld. Hierbij hangen de gewichten echter af van de te vergelijken pixels. Hoe sterker de gelijkenis tussen twee pixels, hoe groter het corresponderend gewicht. Deze techniek blijkt sterke resultaten te geven en zal bovendien ook van fundamenteel belang zijn in ons voorgesteld algoritme.

Het NLMS algoritme start van een eenvoudig ruismodel:

$$\mathbf{y} = \mathbf{x} + \sigma \mathbf{n}$$

waarbij \mathbf{x} en \mathbf{y} respectievelijk het ideaalbeeld en waargenomen beeld voorstellen, \mathbf{n} is de ruiscomponent met verwachtingswaarde gelijk aan 0 en σ is de standaardafwijking van het ruissignaal.

De nieuwe pixelwaarde $\hat{\mathbf{x}}_i$ wordt bepaald als gewogen gemiddelde van de huidige pixels:

$$\hat{\mathbf{x}}_i = \frac{\sum_{j=0}^{MN-1} w_{ij} \mathbf{y}_j}{\sum_{j=0}^{MN-1} w_{ij}} \quad (3.3)$$

Hierbij zal de keuze van de gewichten van essentieel belang zijn om een optimale beeldreconstructie te garanderen. Deze zullen groot moeten zijn voor sterk gelijkaardige gebieden en klein (soms zelfs liefst 0) voor totaal verschillende gebieden. In [24] worden deze gedefinieerd a.d.h.v. de zogenaamde *Leclerc* functie:

$$w_{ij} = \exp\left(-\frac{\|\mathbf{y}_{N_i} - \mathbf{y}_{N_j}\|^2}{2h^2}\right) \quad (3.4)$$

waarbij \mathbf{y}_{N_i} een gebiedsvector rond de pixel op positie i is (bv. een 5×5 regio gecentreerd rond deze pixel) en h een parameter die bepaalt hoe sterk er uitgemiddeld wordt. Hoge waarden voor h zullen sterk gelijkende gebieden hogere gewichten geven tegenover lage h waarden. Hierdoor zal een hoge waarde voor h resulteren in een sterkere uitmiddeling (en dus meer ruisonderdrukking) dan lage waarden voor h .

Opmerking 3.1. Merk op dat we voor de triviale keuze $\mathbf{y}_{N_i} = \mathbf{y}_i$ het zogenaamd *bilateraal filter* verkrijgen:

$$w_{ij} = \exp\left(-\frac{(\mathbf{y}_i - \mathbf{y}_j)^2}{2h^2}\right)$$

Dit filter is minder aan te raden aangezien deze enkel twee pixels met elkaar vergelijkt om het gewicht te bepalen. Bijgevolg zijn de gewichten zeer gevoelig voor ruis.

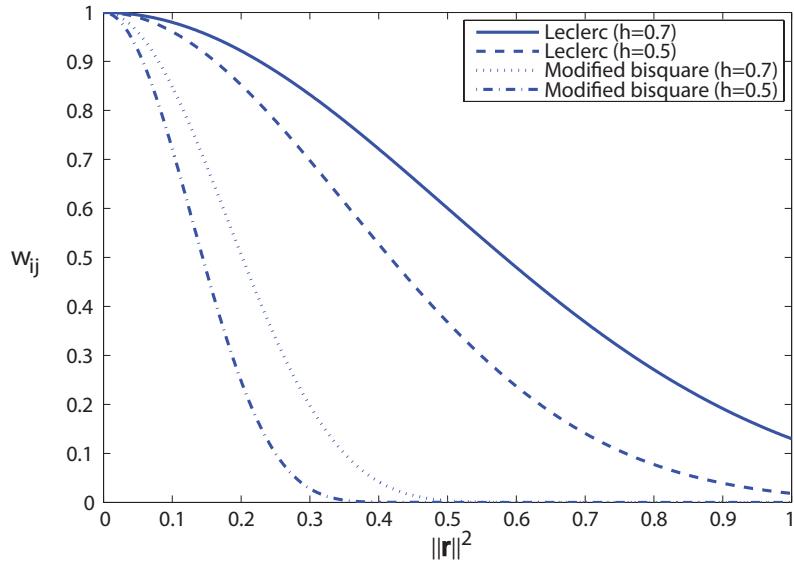
Opmerking 3.2. Als uitbreiding op Vergelijking 3.3 kan men bij het bepalen van $\hat{\mathbf{x}}_i$ gebruik maken van verschillende gewichten $w_{i+k,j+l}$ die corresponderen met de regio² ($k, l = -K, \dots, K$) rond pixel \mathbf{y}_i en \mathbf{y}_j . Deze gewichten kunnen dan geaggregeerd via een tweede verzameling gewichten $b_{k,l}$ worden tot één gewicht $w'_{ij} = \sum_{k=-K}^K \sum_{l=-K}^K b_{k,l} w_{i+k,j+l}$:

$$\begin{aligned} \hat{\mathbf{x}}_i &= \frac{\sum_{j=0}^{MN-1} w'_{ij} \mathbf{y}_j}{\sum_{j=0}^{MN-1} w'_{ij}} \\ &= \frac{\sum_{j=0}^{MN-1} \sum_{k=-K}^K \sum_{l=-K}^K b_{k,l} w_{i+k,j+l} \mathbf{y}_j}{\sum_{j=0}^{MN-1} \sum_{k=-K}^K \sum_{l=-K}^K b_{k,l} w_{i+k,j+l}} \end{aligned} \quad (3.5)$$

Het grote voordeel aan deze uitbreiding is dat lokale informatie extra in rekening gebracht wordt en bijgevolg de reconstructie optimaler kan verlopen. De extensie zorgt echter voor heel wat extra rekenwerk. Deze uitbreiding wordt ook wel *vectorgebaseerde NLMS* genoemd.

Hoewel de gewichten gedefinieerd als in Vergelijking 3.4 relatief goede resultaten geven, kunnen ze toch nog geoptimaliseerd worden. Ze geven namelijk nog steeds niet-nul gewichten aan sterk verschillende regio's. Een groot aantal niet-nul gewichten kan dus voor ongewenste effecten zorgen in de uitmiddeling. Het is daarom meer aan te raden

²Voor de eenvoud wordt een vierkante regio beschouwd. Dit kan echter eenvoudig veralgemeend worden naar een rechthoekig gebied.



Figuur 3.3: NLMS gewichtsfuncties

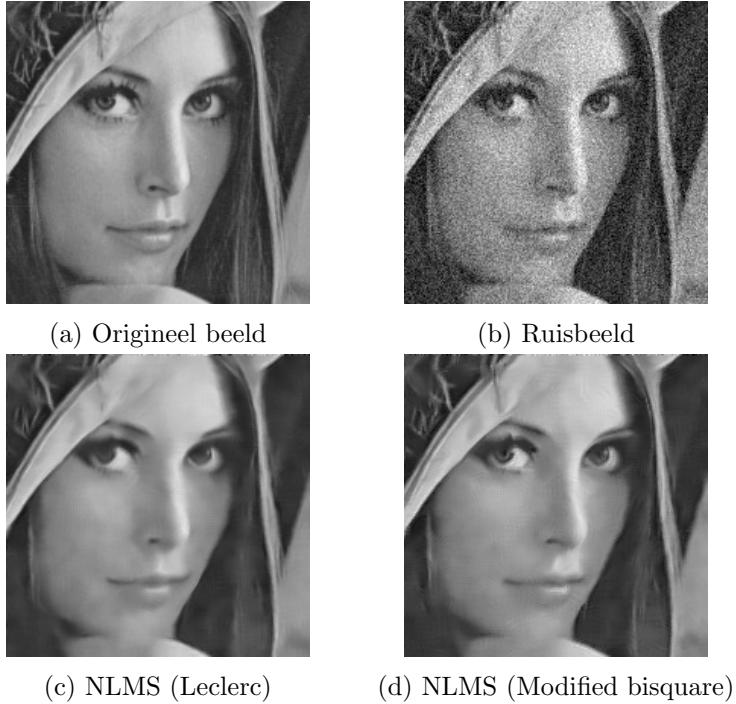
om gebruik te maken van een functie die nul is tot op een bepaalde grenswaarde. In [9] wordt de zogenaamde *modified bisquare* gewichtsfunctie voorgesteld:

$$w_{ij} = \begin{cases} \left(1 - \frac{\|\mathbf{y}_{N_i} - \mathbf{y}_{N_j}\|^2}{h^2}\right)^8 & (\|\mathbf{y}_{N_i} - \mathbf{y}_{N_j}\| \leq h \text{ en } i \neq j) \\ 0 & (\|\mathbf{y}_{N_i} - \mathbf{y}_{N_j}\| > h \text{ en } i \neq j) \\ 1 & (i = j) \end{cases} \quad (3.6)$$

Het verschil tussen de gewichten gedefinieerd in Vergelijkingen 3.4 en 3.6 wordt in Figuur 3.3 geïllustreerd. Hierin is $\mathbf{r} = \mathbf{y}_{N_i} - \mathbf{y}_{N_j}$. Merk op dat de modified bisquare functie een betere keuze is, aangezien deze nul gewichten geeft wanneer het patch \mathbf{y}_{N_j} voldoende verschilt van het beschouwde patch \mathbf{y}_{N_i} . Het punt vanaf wanneer deze nul gewichten worden gegeven, wordt bepaald door h .

Merk op dat bepaalde details (bv. de ogen) licht vervaagd worden bij de Leclerc gewichten (Figuur 3.4c), vergeleken met de modified bisquare resultaten (Figuur 3.4d).

Het NLMS algoritme is zoals geïllustreerd zeer effectief in het verwijderen van ruis, zonder daarbij het signaal significant te beïnvloeden. De parameter h dient echter goed afgestemd te worden. Te kleine h waarden zorgen voor meer nulgewichten en bijgevolg minder ruisonderdrukking. Grote h waarden echter zorgen voor meer vervaging aangezien veel verschillende pixels (met weliswaar lage gewichten) uitgemiddeld worden. Bovendien is de complexiteit van het NLMS algoritme $\mathcal{O}(M^2N^2)$ voor een M bij N afbeelding en dus relatief rekenintensief. De uitbreiding in Opmerking 3.2 zorgt zelfs voor een complexiteit van orde $\mathcal{O}((2K + 1)^2 M^2 N^2)$ (met K de gespecificeerde constante in



Figuur 3.4: Ruisonderdrukking met NLMS

Opmerking 3.2). Een efficiënte implementatie kan dit echter reduceren tot $\mathcal{O}(M^2N^2)$ [9]. Verder is het niet aan te raden om alle gewichten w_{ij} te bepalen voor de uitmiddeling. Dit zou zorgen voor een geheugenopslag van de orde $\mathcal{O}(M^2N^2)$ ³.

3.2.1.4 Bayesiaanse schatters

Naast de bestaande technieken die gebruik maken van het (niet-)lokaal uitmiddelen van pixels kan men ook a.d.h.v. kansrekening en de regel van Bayes een beeld schatten. Deze zogenaamde Bayesiaanse schatters maken gebruik van probabiliteitscalculus om het theoretisch meest waarschijnlijk ideaalbeeld te gaan bepalen, gebruik makend van de statistische eigenschappen van de ruis en ev. bijkomende kennis over het ideaalbeeld.

Hiervoor zal het ruismodel er als volgt uitzien:

$$\mathbf{y} = \mathbf{x} + \mathbf{n}$$

waarbij \mathbf{x} en \mathbf{y} respectievelijk het ideaalbeeld en waargenomen beeld voorstellen, \mathbf{n} is de ruiscomponent met verwachtingswaarde 0. Een Bayesiaanse schatter zal het theoretisch meest waarschijnlijk ideaalbeeld $\hat{\mathbf{x}}$ bepalen, gegeven dat \mathbf{y} waargenomen werd. Dit

³Voor een (relatief klein) beeld van 512 bij 512 pixels resulteert dit in de opslag van een 262.144×262.144 matrix! Iets wat in praktijk liever vermeden wordt.

gebeurt a.d.h.v. de regel van Bayes:

$$\begin{aligned}\hat{\mathbf{x}} &= \arg \max_{\mathbf{x}} p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}) \\ &= \arg \max_{\mathbf{x}} p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) p_{\mathbf{X}}(\mathbf{x}) \\ &= \arg \min_{\mathbf{x}} \left(-\log p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) - \log p_{\mathbf{X}}(\mathbf{x}) \right)\end{aligned}\quad (3.7)$$

Deze schatter wordt ook wel de maximum a posteriori (MAP) schatter genoemd. De eerste en tweede term in de minimalisatie worden daarbij respectievelijk de posterior en prior probability term genoemd. Een speciaal geval van de MAP schatter is de maximum likelihood (ML) schatter, waarbij elk beeld even waarschijnlijk wordt verondersteld ($p_{\mathbf{X}}(\mathbf{x}) = c$):

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} -\log p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})$$

We zullen nu een eenvoudig voorbeeld van een MAP schatter beschrijven. Daarbij veronderstellen we dat de ruis \mathbf{n} Gaussiaans, wit en signaalonafhankelijk is ($\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$). Aangezien $E[\mathbf{y}|\mathbf{x}] = \mathbf{x}$ en $\text{Var}(\mathbf{y}|\mathbf{x}) = \sigma^2 \mathbf{I}$ vinden we:

$$\begin{aligned}p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) &= \frac{1}{\sqrt{(2\pi)^{MN} |\sigma^2 \mathbf{I}|}} \exp \left(-\frac{1}{2} (\mathbf{y} - \mathbf{x})^T (\sigma^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{x}) \right) \\ -\log p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) &= -\log \left(\frac{1}{\sqrt{(2\pi)^{MN} |\sigma^2 \mathbf{I}|}} \right) + \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{x})^T (\mathbf{y} - \mathbf{x}) \\ &= c + \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{x}\|^2\end{aligned}$$

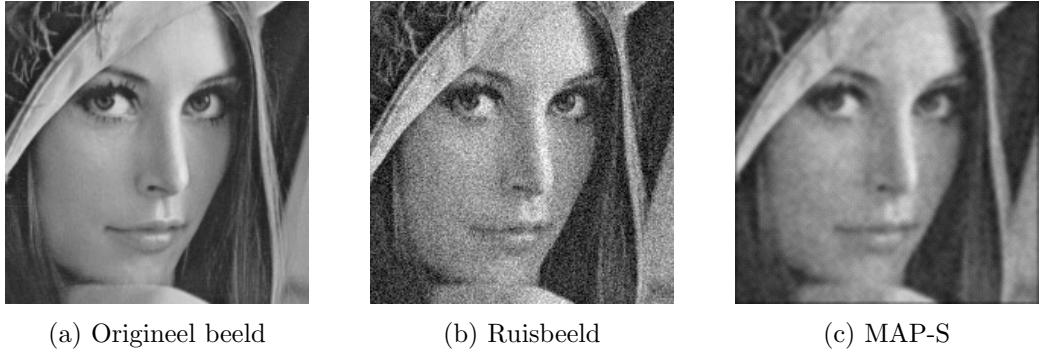
Merk op dat de constante $c = -\log \left(\frac{1}{\sqrt{(2\pi)^{MN} |\sigma^2 \mathbf{I}|}} \right)$ onafhankelijk is van \mathbf{x} . Voor de prior term kan voor de eenvoud ook een Gaussiaans proces verondersteld worden: $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \sigma_x^2 \mathbf{I})$. Hierin kan $\bar{\mathbf{x}}$ bv. een gefilterde versie van \mathbf{y} zijn en σ_x een parameter. Op analoge wijze krijgen we:

$$-\log p(\mathbf{y}|\mathbf{x}) = c' + \frac{1}{2\sigma_x^2} \|\mathbf{x} - \bar{\mathbf{x}}\|^2$$

en dus:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left(\|\mathbf{y} - \mathbf{x}\|^2 + \lambda \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \right)$$

waarbij $\lambda = \frac{\sigma_x^2}{\sigma_y^2}$ afhankelijk is van σ_x en dus een regularisatieparameter is. Merk op dat de constanten c en c' beide onafhankelijk zijn van \mathbf{x} en bijgevolg genegeerd mogen worden in de minimalisatie. Aangezien dit een kwadratisch probleem is, kunnen we dit



Figuur 3.5: Ruisonderdrukking met een MAP schatter

exact oplossen door de gradiënt van de te minimalizeren term gelijk te stellen aan 0.

$$\begin{aligned}
 \hat{\mathbf{x}} &= \arg \min_{\mathbf{x}} \left((\mathbf{y} - \mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \lambda (\mathbf{x} - \bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) \right) \\
 &= \arg \min_{\mathbf{x}} \left(\mathbf{y}^T \mathbf{y} + \mathbf{x}^T \mathbf{x} - 2\mathbf{y}^T \mathbf{x} + \lambda (\mathbf{x}^T \mathbf{x} + \bar{\mathbf{x}}^T \bar{\mathbf{x}} - 2\bar{\mathbf{x}}^T \mathbf{x}) \right) \\
 &\Downarrow \\
 2\hat{\mathbf{x}} + 2\lambda\hat{\mathbf{x}} - 2\mathbf{y} - 2\lambda\bar{\mathbf{x}} &= 0 \\
 \hat{\mathbf{x}} &= \frac{\mathbf{y} + \lambda\bar{\mathbf{x}}}{1 + \lambda}
 \end{aligned}$$

Er wordt m.a.w. een gewogen gemiddelde genomen van het waargenomen beeld en het benaderd ideaalbeeld en afhankelijk van de parameter λ zal het ideaalbeeld meer of minder bijdragen tot dit gemiddelde.

We zullen dit voorbeeld van een MAP schatter in het verder verloop als MAP-S (MAP-Simple) benoemen.

De MAP schatter heeft als voordeel dat hij een theoretisch ideaalbeeld bepaalt, gegeven het waargenomen beeld \mathbf{y} . We dienen echter wel op te merken dat dit ideaalbeeld niet altijd overeenstemt met de beste oplossing. De MAP schatter hangt af van de veronderstelde ruiseigenschappen en het priormodel. Verder dienen we ook te benadrukken dat het voorafgaande voorbeeld (Figuur 3.5c) zeer eenvoudig is. In praktijk komt men niet altijd een kwadratisch probleem en kan het minimum in Vergelijking 3.7 niet altijd (of toch niet altijd even efficiënt) bepaald worden.

3.2.1.5 Alternatieve ruisonderdrukkingstechnieken

De vier technieken die hierboven beschreven worden behoren tot de bekendste ruisonderdrukkingstechnieken. Dit is echter maar een fractie van het totaal aanbod. We vermelden nog enkele bestaande technieken (minder in detail vanwege ons tijdsbestek):

- Wavelet gebaseerde ruisonderdrukking [25, 26]: om zowel spatiale als frequentie-informatie in rekening te houden bij beeldreconstructie kan men er voor kiezen om gebruik te maken van de zogenaamde wavelets. Dit is een transformatie van het beeld die een opsplitsing maakt volgens het frequentieniveau. Op die manier krijgen we enkele hoogdoorlaatgefilterde (H -)beelden en een laagdoorlaatgefilterd (L -)beeld. Wanneer ruis op de beelden te zien is, uit zich dit tot ruis in de H -beelden, aangezien ruis zich typisch binnen de hoge frequentiebanden bevindt. Een eenvoudige ruisonderdrukkingstechniek stelt bv. alle coëfficiënten kleiner dan een bepaalde drempelwaarde gelijk aan nul (hard thresholding), een beter idee is om deze coëfficiënten in te krimpen naar nul met die drempelwaarde (soft thresholding). Andere technieken maken gebruik van de beeld- en ruisstructuur om de drempelwaarde adaptief te kiezen [27]. Het gereconstrueerd beeld kan dan bekomen worden via de inverse wavelettransformatie.
- Block matching & 3D filtering (BM3D) [15, 18]: een ruizig beeld kan onderverdeeld worden in verschillende overlappende referentieblokken. Voor elk referentieblok bepaalt men vervolgens een verzameling van blokken die sterk gelijken op dit referentieblok⁴. Deze gelijkaardige blokken worden gegroepeerd tot een 3D blok. Vervolgens zorgt een filterstap a.d.h.v. een transformatie naar een ander domein voor het verwijderen van de ruis in dit 3D blok. Een aggregatiestap zorgt vervolgens voor de samenvoeging van de 3D blokken tot een gereconstrueerd beeld.
- Total variation (TV) [28, 29]: een alternatieve methode minimaliseert de totale variatie in een beeld. Dagdagelijkse afbeeldingen bestaan voornamelijk uit vlakke gebieden die van elkaar afgescheiden zijn door randen en textuur. De toevoeging van ruis zorgt bijgevolg voor extra ongewenste ‘randen en textuur’. De totale variatie kan in deze zin gedefinieerd worden als de totale grootte van overgangen tussen lage en hoge pixelintensiteiten. Bij een constant beeld zou dit eenvoudigweg gelijk zijn aan nul. Afbeeldingen die veel ruis bevatten hebben typisch een hoge totale variatie. Het minimalizeren zal bijgevolg de ruis in het beeld onderdrukken. TV wordt dikwijls ook gebruikt als prior bij MAP schatters.

3.2.2 Deconvolutie

Het probleem waarbij vervaging uit een beeld moet verwijderd worden, noemen we *deconvolutie*. Het beeld is a.h.w. geconvoluteerd met een laagdoorlaatfilter h en het is aan ons deze convolutie ongedaan te maken. Hierbij worden twee types onderscheiden: enerzijds kan het zijn dat de blurkernel h gekend is, in dat geval spreken we van *gewone*

⁴Merk de gelijkenis op met NLMS.

deconvolutie. Anderzijds is het mogelijk dat deze ongekend is, in dit geval spreken we van *blinde deconvolutie*.

De studie in Sectie 2.4 heeft ons nog geen concrete blurkernel gegeven, alhoewel deze wel nog kan gevonden worden. We zullen dus voor de eenvoud in wat volgt geen blinde deconvolutie beschouwen, maar er van uitgaan dat de blurkernel gekend is.

3.2.2.1 Invers filter

De meest eenvoudige deconvolutiemethode is het invers filter. Deze start van volgend ruismodel:

$$y_{i,j} = (h * x)_{i,j} \quad (3.8)$$

waarbij x en y respectievelijk het ideaalbeeld en waargenomen beeld in convolutienotatie (Appendix A) zijn en h de gekende blurkernel is. In het fourierdomein kan Vergelijking 3.8 herschreven worden als (Eigenschap C.2):

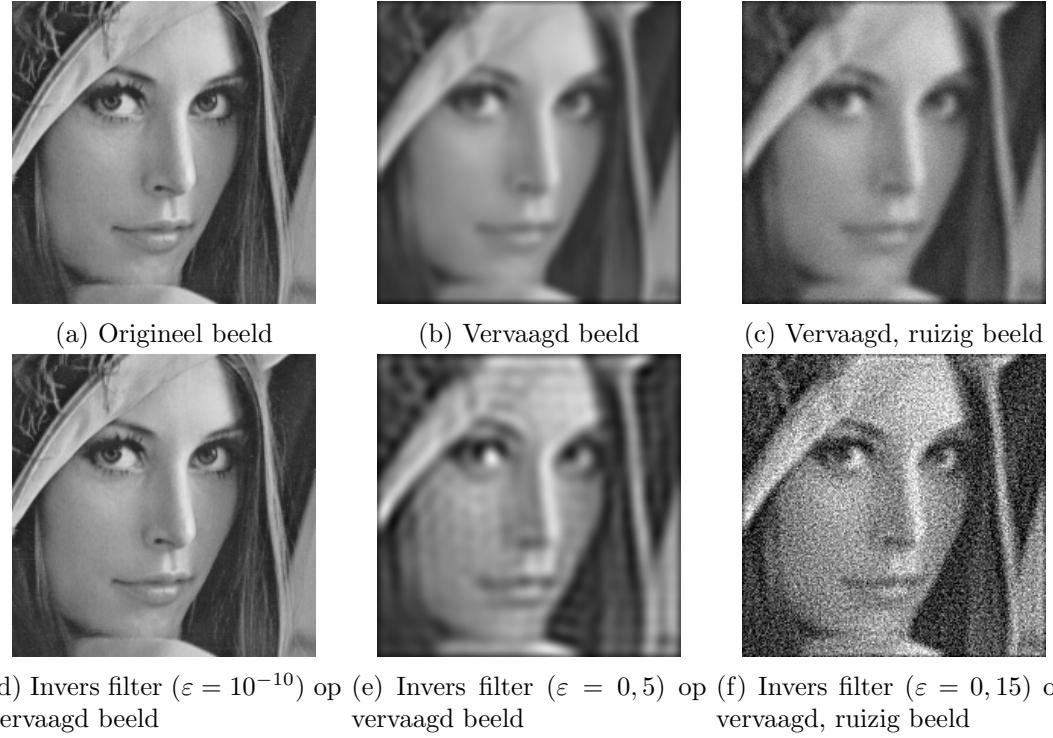
$$Y_{\omega_i, \omega_j} = H_{\omega_i, \omega_j} X_{\omega_i, \omega_j}$$

Bijgevolg kan het ideaalbeeld in het fourierdomein geschat worden als:

$$\hat{X}_{\omega_i, \omega_j} = \frac{Y_{\omega_i, \omega_j}}{\max(\varepsilon, H_{\omega_i, \omega_j})} \quad (3.9)$$

Merk op dat Vergelijking 3.9 in praktijk moeilijk uit te rekenen is, door de slechte conditionering. De parameter ε zal numerieke instabiliteit vermijden.

De eenvoud in het principe van het invers filter doet al vermoeden dat hier enkele nadelen aan verbonden zijn. Zo zal de deling in Vergelijking 3.9 slecht geconditioneerd zijn en bijgevolg de keuze van ε essentieel zijn. Zeer kleine waarden voor ε zullen leiden tot numeriek instabiele oplossingen, terwijl grote waarden zullen zorgen voor een significante verandering in het frequentiedomein. Bijgevolg ontstaan allerlei ringeffecten zoals geïllustreerd in Figuur 3.6e. Een ander belangrijk nadeel is dat in praktijk meestal het waargenomen beeld niet voldoet aan het model gegeven in Vergelijking 3.8. Meestal is het beeld onderhevig aan ruis en dient hier dus nog een extra ruisterm aan toegevoegd te worden. Toepassing van een invers filter zal er voor zorgen dat deze ruis versterkt wordt en het beeld dus kwalitatief slechter zal zijn (Figuur 3.6f).



Figuur 3.6: Deconvolutie met het invers filter

3.2.2.2 Bayesiaans schatter als deconvolutiemethode

De Bayesiaanse schatter kan eveneens gebruikt worden om een deconvolutieresultaat te verkrijgen. In dit geval wordt uitgegaan van het volgend ruismodel:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

waarbij \mathbf{x} en \mathbf{y} respectievelijk het ideaalbeeld en waargenomen beeld voorstellen, \mathbf{n} en \mathbf{H} zijn respectievelijk de ruiscomponent en gekende blurkernel.

Zoals in Sectie 3.2.1.4 verklaard wordt, is het de bedoeling om het volgende minimalisatieprobleem op te lossen:

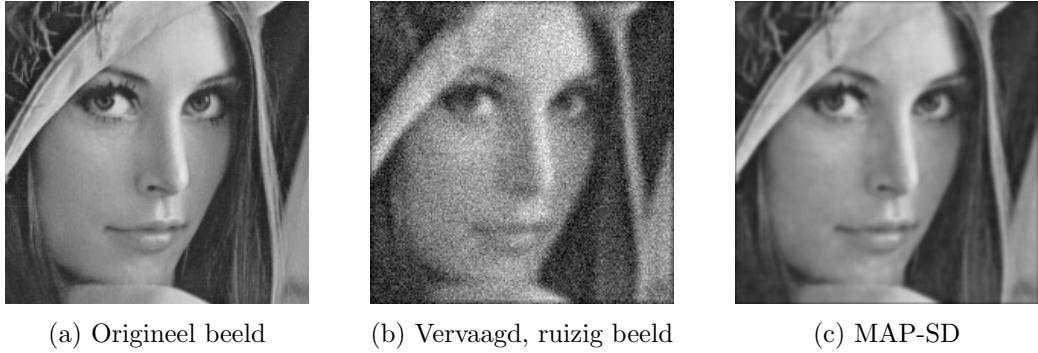
$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left(-\log p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) - \log p_{\mathbf{X}}(\mathbf{x}) \right) \quad (3.10)$$

Als we voor de eenvoud veronderstellen dat we te maken hebben met witte, Gaussiaanse, signaalafhankelijke ruis met variantie σ^2 en \mathbf{x} een normale verdeling volgt met gemiddelde $\bar{\mathbf{x}}$ ⁵ en variantie $\sigma_{\mathbf{x}}^2$, dan herleidt⁶ Vergelijking 3.10 zich tot:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left(\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \right) \quad (3.11)$$

⁵Voor $\bar{\mathbf{x}}$ kan een snelle ruisonderdrukkingstechniek (bv. laagdoorlaatfilter) gebruikt worden om de hoeveelheid ruis in het gereconstrueerd beeld te doen afnemen.

⁶De afleiding is volledig analoog aan die in Sectie 3.2.1.4



Figuur 3.7: Deconvolutie met een MAP schatter

met $\lambda = \frac{\sigma^2}{\sigma_x^2}$ een regularisatieconstante. Merk op dat de eerste term zich bezighoudt met deconvolutie, terwijl de tweede term zorgt voor ruisonderdrukking. Een goede ruisonderdrukkingstechniek is dus aan te raden bij het bepalen van \bar{x} . De gradiënt van Vergelijking 3.11 gelijkstellen aan nul, herleidt zich tot:

$$[\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I}] \hat{\mathbf{x}} = \mathbf{H}^T \mathbf{y} + \bar{\mathbf{x}} \quad (3.12)$$

Dit is een stelsel van MN vergelijkingen in MN onbekenden ($\mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$). De oplossing hiervan kan benaderd worden a.d.h.v. het conjugate gradient algoritme (Appendix D). We zullen dit voorbeeld van een MAP schatter in het verder verloop als MAP-SD (MAP-Simple Deconvolution) benoemen.

Opmerking 3.3. Merk op dat in de afwezigheid van ruis het voldoende is om de ML schatter te beschouwen, aangezien geen schatting meer nodig is van het beeld zonder ruis ($\bar{\mathbf{x}}$). Bovenstaande vergelijkingen herleiden zich dan tot:

$$\begin{aligned} \hat{\mathbf{x}} &= \arg \min_{\mathbf{x}} -\log p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) \\ &= \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{Hx}\|^2 \\ \mathbf{H}^T \mathbf{H} \hat{\mathbf{x}} &= \mathbf{H}^T \mathbf{y} \end{aligned} \quad (3.13)$$

Vergelijking 3.13 kan eveneens via het conjugate gradient algoritme (Appendix D) benaderd opgelost worden.

Het voordeel aan de MAP schatter bij deconvolutie is dat deze eveneens rekening houdt met de mogelijke aanwezigheid van ruis. Zoals te zien is in Figuur 3.7c is het gereconstrueerd beeld zowel verscherpt als minder onderhevig aan ruis. Een goede ruiskennis (als die kan verkregen worden) is echter wel een vereiste in de reconstructie.

3.3 NLMS als MAP schatter

Om tot het voorgesteld optimalisatie-algoritme te komen, zullen we gebruik maken van een belangrijke eigenschap: De NLMS reconstructietechniek is een speciaal geval van de MAP schatter. We zullen deze eigenschap allereerst aantonen.

Eigenschap 3.4. *Gegeven het ruismodel waar witte, Gaussiaanse, signaal en afhankelijke ruis verondersteld wordt:*

$$\mathbf{y} = \mathbf{x} + \sigma \mathbf{n}$$

met \mathbf{x} en \mathbf{y} respectievelijk het ideaalbeeld en waargenomen beeld, \mathbf{n} een ruisvector die een standaardnormale verdeling volgt en σ de standaardafwijking van de ruisterm. De MAP schatter kan dan als volgt geschreven worden:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left(-\log p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) - \log p_{\mathbf{X}}(\mathbf{x}) \right) \quad (3.14)$$

Er bestaat een prior kansverdelingsfunctie $p_{\mathbf{X}}(\mathbf{x})$ zodanig dat het minimum in Vergelijking 3.14 bereikt wordt bij de NLMS oplossing:

$$\hat{\mathbf{x}}_i = \frac{\sum_{j=0}^{MN} w_{ij} \mathbf{y}_j}{\sum_{j=0}^{MN} w_{ij}}$$

waarbij w_{ij} de NLMS gewichten zijn, gedefinieerd in Vergelijking 3.6.

Bewijs. We stellen de volgende prior verdelingsfunctie voor:

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Z} \prod_{i,j=0}^{MN-1} \exp \left(- \left(\frac{\mathbf{x}_i - \mathbf{x}_j}{\sigma_{ij}} \right)^2 \right) \quad (3.15)$$

waarbij Z een normaliserende constante is en σ_{ij} parameters zijn die in het verloop van het bewijs vastgelegd zullen worden en bovendien ook symmetrisch in i en j verondersteld worden. Merk op dat de prior verdelingsfunctie uitdrukt dat we verwachten van het ideaalbeeld \mathbf{x} dat het bestaat uit veel vlakke regio's⁷, een aanname die bij de meeste beelden in realiteit kan vastgesteld worden. Definiëren we de matrices \mathbf{T}_i als volgt:

$$(\mathbf{T}_i)_{k,l} = \delta_{k,1} \delta_{l,i}$$

met $\delta_{m,n}$ de kronecker delta, dan kunnen we Vergelijking 3.15 herschrijven tot:

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Z} \prod_{i,j=0}^{MN-1} \exp \left(- \frac{\|(\mathbf{T}_i - \mathbf{T}_j)\mathbf{x}\|^2}{\sigma_{ij}^2} \right)$$

⁷Het extreem geval waarbij $\mathbf{x} = \mathbf{c}$ met c een constante bevestigt dit. In dit geval is namelijk $p(\mathbf{x}) = 1$

en bijgevolg vinden we, op een constante (onafhankelijk van \mathbf{x}) na, de prior die in [31] wordt voorgesteld:

$$-\log p_{\mathbf{X}}(\mathbf{x}) = c + \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \|(\mathbf{T}_i - \mathbf{T}_j)\mathbf{x}\|^2$$

De MAP schatter in Vergelijking 3.14 reduceert zich bijgevolg tot

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left(\|\mathbf{y} - \mathbf{x}\|^2 + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \|(\mathbf{T}_i - \mathbf{T}_j)\mathbf{x}\|^2 \right)$$

met $\lambda > 0$ een regularisatieparameter. Dit kan verder herschreven worden tot:

$$\begin{aligned} \hat{\mathbf{x}} &= \arg \min_{\mathbf{x}} \left((\mathbf{y} - \mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{x}^T (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \mathbf{x} \right) \\ &= \arg \min_{\mathbf{x}} \left(\mathbf{y}^T \mathbf{y} + \mathbf{x}^T \mathbf{x} - 2\mathbf{y}^T \mathbf{x} + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{x}^T (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \mathbf{x} \right) \\ &= \arg \min_{\mathbf{x}} \left(\mathbf{x}^T \left[\mathbf{I} + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \right] \mathbf{x} - 2\mathbf{y}^T \mathbf{x} \right) \end{aligned} \quad (3.16)$$

Het minimum van deze functie correspondeert met het nulpunt van de gradiënt:

$$\left[\mathbf{I} + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \right] \hat{\mathbf{x}} = \mathbf{y}$$

Dit kan herschreven worden als:

$$\begin{aligned} \hat{\mathbf{x}} + \lambda \left(\sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{T}_i^T \mathbf{T}_i \hat{\mathbf{x}} + \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{T}_j^T \mathbf{T}_j \hat{\mathbf{x}} \right. \\ \left. - \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{T}_i^T \mathbf{T}_j \hat{\mathbf{x}} - \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{T}_j^T \mathbf{T}_i \hat{\mathbf{x}} \right) = \mathbf{y} \end{aligned}$$

Aangezien de parameters σ_{ij} symmetrisch verondersteld zijn, reduceert dit tot:

$$\hat{\mathbf{x}} + 2\lambda \left(\sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{T}_i^T \mathbf{T}_i \hat{\mathbf{x}} - \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{T}_i^T \mathbf{T}_j \hat{\mathbf{x}} \right) = \mathbf{y} \quad (3.17)$$

We zullen Vergelijking 3.17 nu componentsgewijs beschouwen. Daarbij maken we gebruik van het feit dat $(\mathbf{T}_i^T \mathbf{T}_i \hat{\mathbf{x}})_k = \delta_{ik} \hat{\mathbf{x}}_i$ en $(\mathbf{T}_i^T \mathbf{T}_j \hat{\mathbf{x}})_k = \delta_{ik} \hat{\mathbf{x}}_j$ ⁸:

$$\begin{aligned} \hat{\mathbf{x}}_k + 2\lambda \left(\sum_{j=0}^{MN-1} \frac{1}{\sigma_{kj}^2} \hat{\mathbf{x}}_k - \sum_{j=0}^{MN-1} \frac{1}{\sigma_{kj}^2} \hat{\mathbf{x}}_j \right) &= \mathbf{y}_k \\ \Updownarrow \\ \left(1 + 2\lambda \sum_{j=0}^{MN-1} \frac{1}{\sigma_{kj}^2} \right) \hat{\mathbf{x}}_k &= \mathbf{y}_k + 2\lambda \sum_{j=0}^{MN-1} \frac{1}{\sigma_{kj}^2} \hat{\mathbf{x}}_j \end{aligned}$$

Maken we nu de approximatie $\hat{\mathbf{x}}_i = \mathbf{y}_i$ en stellen we $\sigma_{ij}^2 = \frac{2\lambda}{w_{ij}}$ met w_{ij} gedefinieerd zoals in Vergelijking 3.6, dan vereenvoudigt bovenstaande gelijkheid tot:

$$\begin{aligned} \hat{\mathbf{x}}_k + \left(\sum_{j=0}^{MN-1} w_{kj} \right) \hat{\mathbf{x}}_k &= \mathbf{y}_k + \sum_{j=0}^{MN-1} w_{kj} \hat{\mathbf{x}}_j \\ \Updownarrow \\ \left(\sum_{j=0}^{MN-1} w_{kj} \right) \hat{\mathbf{x}}_k &= \sum_{j=0}^{MN-1} w_{kj} \mathbf{y}_j \\ \Updownarrow \\ \hat{\mathbf{x}}_k &= \frac{\sum_{j=0}^{MN-1} w_{kj} \mathbf{y}_j}{\sum_{j=0}^{MN-1} w_{kj}} \end{aligned}$$

Dit is niets anders dan de NLMS oplossing. □

3.4 NLMS-SCB

In de vorige sectie hebben we aangetoond dat NLMS als een MAP schatter kan beschouwd worden. Hiervoor wordt van het ruismodel echter verwacht dat de ruis wit, Gaussiaans en signaalafhankelijk is. Iets wat in ons ruismodel zeker en vast niet van toepassing is. We kunnen een analoge redenering uitvoeren vertrekende van ons ruismodel (Vergelijking 2.2). Het resultaat dat overeenstemt met Vergelijking 3.16 in

⁸Dit volgt rechtstreeks uit de definitie van \mathbf{T}_i .

Eigenschap 3.4 zou er dan als volgt uit zien:

$$\begin{aligned}
\hat{\mathbf{x}} &= \arg \min_{\mathbf{x}} \left((\mathbf{y} - \mathbf{Gx})^T (\mathbf{H}^T \Sigma(\mathbf{x})^2 \mathbf{H})^{-1} (\mathbf{y} - \mathbf{Gx}) \right. \\
&\quad \left. + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{x}^T (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \mathbf{x} \right) \\
&= \arg \min_{\mathbf{x}} \left(\mathbf{y}^T (\mathbf{H}^T \Sigma(\mathbf{x})^2 \mathbf{H})^{-1} \mathbf{y} + \mathbf{x}^T \mathbf{G}^T (\mathbf{H}^T \Sigma(\mathbf{x})^2 \mathbf{H})^{-1} \mathbf{Gx} \right. \\
&\quad \left. - 2\mathbf{y}^T (\mathbf{H}^T \Sigma(\mathbf{x})^2 \mathbf{H})^{-1} \mathbf{Gx} + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{x}^T (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \mathbf{x} \right) \\
&= \arg \min_{\mathbf{x}} \left(\mathbf{x}^T \left[\mathbf{G}^T (\mathbf{H}^T \Sigma(\mathbf{x})^2 \mathbf{H})^{-1} \mathbf{G} + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \right] \mathbf{x} \right. \\
&\quad \left. - 2\mathbf{y}^T (\mathbf{H}^T \Sigma(\mathbf{x})^2 \mathbf{H})^{-1} \mathbf{Gx} \right)
\end{aligned}$$

Hierbij hebben we gebruik gemaakt van $\mathbb{E}[\mathbf{y}|\mathbf{x}] = \mathbf{Gx}$, $\text{Var}[\mathbf{y}|\mathbf{x}] = \mathbf{H}^T \Sigma(\mathbf{x})^2 \mathbf{H}$ en $\Sigma(\mathbf{x}) \approx \Sigma(\mathbf{y})$ ⁹. De gradient hiervan gelijkstellen aan nul in $\hat{\mathbf{x}}$, levert:

$$\left[\mathbf{G}^T (\mathbf{H}^T \Sigma(\hat{\mathbf{x}})^2 \mathbf{H})^{-1} \mathbf{G} + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \right] \hat{\mathbf{x}} = \mathbf{G}^T (\mathbf{H}^T \Sigma(\hat{\mathbf{x}})^2 \mathbf{H})^{-1} \mathbf{y} \tag{3.18}$$

We zullen nu de vereenvoudiging maken dat de ruis ongecorreleerd is ($\mathbf{H} = \mathbf{I}$) en dat er geen blur aanwezig is ($\mathbf{G} = \mathbf{I}$). De laatste vergelijking herleidt zich dan tot:

$$\left[\mathbf{I} + \lambda \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \Sigma(\hat{\mathbf{x}})^2 (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \right] \hat{\mathbf{x}} = \mathbf{y}$$

De aandachtige lezer herkent hier een vergelijking van de vorm $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$ en besluit bijgevolg dat dit iteratief kan opgelost worden (d.m.v. bv. het conjugate gradient algoritme). Merk echter op dat de matrix \mathbf{A} afhankelijk is van $\hat{\mathbf{x}}$ en dus een iteratieve zoekmethode niet van toepassing kan zijn.

Een componentsgewijze beschouwing levert, op gelijkaardige manier als in het bewijs van Eigenschap 3.4:

$$\left(1 + 2\lambda(\alpha\hat{\mathbf{x}}_k + \sigma_0) \sum_{j=0}^{MN-1} \frac{1}{\sigma_{kj}^2} \right) \hat{\mathbf{x}}_k = \mathbf{y}_k + 2\lambda(\alpha\hat{\mathbf{x}}_k + \sigma_0) \sum_{j=0}^{MN-1} \frac{1}{\sigma_{kj}^2} \hat{\mathbf{x}}_j$$

⁹Hierdoor mag de term $\mathbf{y}^T (\mathbf{H}\Sigma(\mathbf{x})^2 \mathbf{H}^T)^{-1} \mathbf{y}$ genegeerd worden in de minimalisatie naar \mathbf{x} .

Net als in het bovenstaand bewijs, zou de veronderstelling $\hat{\mathbf{x}}_i = \mathbf{y}_i$ samen met $\sigma_{ij}^2 = \frac{2\lambda(\alpha\hat{\mathbf{x}}_i + \sigma_0)}{w_{ij}}$ de NLMS oplossing geven. Deze veronderstelling is echter in strijd met de symmetrie die we verwachten van σ_{ij} . De MAP schatter kan dus niet gebruikt worden als extensie van het NLMS algoritme in het speciaal geval $\mathbf{H} = \mathbf{I}$ en $\mathbf{G} = \mathbf{I}$. Bijgevolg is dit eveneens zo voor ons breder ruismodel (Vergelijking 2.2).

Om Eigenschap 3.4 dus te gebruiken binnen ons model, zullen we de ruis in een pre-processing stap a.h.w. ongecorreleerd en signaalonafhankelijk moeten maken. We zullen dit probleem oplossen door de NLMS gewichten gedefinieerd in Vergelijking 3.6 te bepalen a.d.h.v. een signaalonafhankelijke, ongecorreleerde versie van het waargenomen beeld. De gewichten zijn dus als volgt gedefinieerd:

$$w_{ij} = \begin{cases} \left(1 - \frac{\|\mathbf{y}_{N_i}'' - \mathbf{y}_{N_j}''\|^2}{h^2}\right)^8 & (\|\mathbf{y}_{N_i}'' - \mathbf{y}_{N_j}''\| \leq h \text{ en } i \neq j) \\ 0 & (\|\mathbf{y}_{N_i}'' - \mathbf{y}_{N_j}''\| > h \text{ en } i \neq j) \\ 1 & (i = j) \end{cases}$$

met $\mathbf{y}_{N_i}'' = (\mathbf{H}\Sigma(\mathbf{x}_i))^{-1}\mathbf{y}_{N_i}$. Natuurlijk is een rechtstreeks implementatie van deze gewichten niet echt efficiënt. Daarom zullen we de ruis in het beeld op een gelijkaardige manier als in [9] eerst ongecorreleerd maken:

$$\tilde{\mathbf{Y}}' = \frac{\tilde{\mathbf{Y}}}{\max(\varepsilon, \tilde{\mathbf{H}})} \quad (3.19)$$

met $\tilde{\mathbf{Y}}$, $\tilde{\mathbf{H}}$ respectievelijk de fouriergetransformeerden van \mathbf{y} en \mathbf{H} en ε een kleine positieve constante die zorgt voor numerieke stabiliteit. Het ‘witgemaakt’ signaal \mathbf{y}' is bijgevolg de invers fouriergetransformeerde van $\tilde{\mathbf{Y}}'$. Vervolgens zorgen we ervoor dat de ruis eveneens signaalonafhankelijk wordt gemaakt:

$$\mathbf{y}_i'' = \frac{\mathbf{y}_i'}{\alpha\mathbf{y}_i' + \sigma_0} \quad (3.20)$$

met α en σ_0 zoals geschat in Vergelijking 2.14. Vervolgens kunnen de NLMS gewichten bepaald worden a.d.h.v. deze ongecorreleerde, signaalonafhankelijke versie.

Het gereconstrueerd beeld wordt gevormd op basis van deze gewichten en het waargenomen beeld:

$$\hat{\mathbf{x}}_i = \frac{\sum_{j=0}^{MN-1} w_{ij} \mathbf{y}_j}{\sum_{j=0}^{MN-1} w_{ij}}$$

Merk op dat Vergelijkingen 3.19 en 3.20 niet altijd nodig zijn (bv. bij signaalonafhankelijke of ongecorreleerde ruis). We zullen de volgende NLMS varianten beschouwen in onze resultaten:

- NLMS-S: we gaan uit van ongecorreleerde, signaalafhankelijke ruis en stellen bijgevolg $\tilde{\mathbf{Y}}' = \tilde{\mathbf{Y}}$ (en dus $\mathbf{y}' = \mathbf{y}$) i.p.v. Vergelijking 3.19.
- NLMS-C: we gaan uit van gecorreleerde, signaalafhankelijke ruis en stellen bijgevolg $\mathbf{y}'' = \mathbf{y}'$ i.p.v. Vergelijking 3.20.
- NLMS-SC: we gaan uit van gecorreleerde, signaalafhankelijke ruis en leggen $\tilde{\mathbf{Y}}'$ (en dus ook \mathbf{y}') en \mathbf{y}'' vast a.d.h.v. Vergelijkingen 3.19 en respectievelijk 3.20.

Opmerking 3.5. Bovenstaande algoritmen zijn een uitbreiding van het NLMS algoritme. We kunnen echter een gelijkaardige redenering toepassen als in Opmerking 3.2 ter extensie van deze algoritmen. Dit zal hoogstwaarschijnlijk resulteren in kwalitatievere resultaten.

Er rest ons nu nog de vervaging van het beeld op te nemen. Dit is een extra stap die het probleem veel gecompliceerder maakt, aangezien zowel blur als ruis verwijderen twee acties zijn die elkaar tegenwerken¹⁰. Bovendien is er in de literatuur nog vrijwel geen werk gemaakt van de gelijktijdige verwijdering van signaalafhankelijke, gecorreleerde ruis en blur.

We merken eerst op dat Vergelijking 3.18 nog steeds opgaat wanneer er sprake is van blur en witte, Gaussiaanse, signaalafhankelijke ruis ($\mathbf{H} = \mathbf{I}$ en $\Sigma(\mathbf{x}) = \sigma\mathbf{I}$):

$$\left[\mathbf{G}^T \mathbf{G} + \lambda' \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} (\mathbf{T}_i - \mathbf{T}_j)^T (\mathbf{T}_i - \mathbf{T}_j) \right] \hat{\mathbf{x}} = \mathbf{G}^T \mathbf{y} \quad (3.21)$$

waarbij $\lambda' = \frac{\lambda}{\sigma^2}$ een regularisatieparameter is. Bovenstaande vergelijking kan (analoog als in het bewijs van Eigenschap 3.4) verder geschreven worden als:

$$\mathbf{G}^T \mathbf{G} \hat{\mathbf{x}} + 2\lambda' \left(\sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{T}_i^T \mathbf{T}_i \hat{\mathbf{x}} - \sum_{i,j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \mathbf{T}_i^T \mathbf{T}_j \hat{\mathbf{x}} \right) = \mathbf{G}^T \mathbf{y}$$

hetgeen componentsgewijs niets anders is dan:

$$\begin{aligned} (\mathbf{G}^T \mathbf{G} \hat{\mathbf{x}})_i + 2\lambda' \left(\sum_{j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \hat{\mathbf{x}}_i - \sum_{j=0}^{MN-1} \frac{1}{\sigma_{ij}^2} \hat{\mathbf{x}}_j \right) &= (\mathbf{G}^T \mathbf{y})_i \\ \Updownarrow \\ (\mathbf{G}^T \mathbf{G} \hat{\mathbf{x}})_i + \hat{\mathbf{x}}_i \left(\sum_{j=0}^{MN-1} w_{ij} \right) - \sum_{j=0}^{MN-1} w_{ij} \hat{\mathbf{x}}_j &= (\mathbf{G}^T \mathbf{y})_i \end{aligned} \quad (3.22)$$

¹⁰Meestal zorgt ruisonderdrukking voor meer blur en deconvolutie voor extra ruis.

Stellen we nu:

$$W_i = \sum_{j=0}^{MN-1} w_{ij}$$

en

$$\begin{aligned}\hat{\mathbf{x}}_i &= \text{NLMS}(\hat{\mathbf{x}})_i \\ &= \frac{\sum_{j=0}^{MN-1} w_{ij} \hat{\mathbf{x}}_j}{\sum_{j=0}^{MN-1} w_{ij}}\end{aligned}$$

m.a.w. W_i is de som van alle gewichten die corresponderen met pixel i en $\hat{\mathbf{x}} = \text{NLMS}(\hat{\mathbf{x}})$ is het gereconstrueerd beeld uit $\hat{\mathbf{x}}$ via het NLMS algoritme. De derde term in Vergelijking 3.22 $\sum_{j=0}^{MN-1} w_{ij} \hat{\mathbf{x}}_j$ kan bepaald worden door het NLMS algoritme toe te passen op $\hat{\mathbf{x}}$ en de i -de component te vermenigvuldigen met W_i^{11} :

$$\begin{aligned}\sum_{j=0}^{MN-1} w_{ij} \hat{\mathbf{x}}_j &= \left(\sum_{j=0}^{MN-1} w_{ij} \right) \frac{\sum_{j=0}^{MN-1} w_{ij} \hat{\mathbf{x}}_j}{\sum_{j=0}^{MN-1} w_{ij}} \\ &= W_i \text{NLMS}(\hat{\mathbf{x}})_i\end{aligned}$$

Aangezien W_i door het NLMS algoritme bepaald worden, kan ook de tweede term in Vergelijking 3.22 $\hat{\mathbf{x}}_i \left(\sum_{j=0}^{MN-1} w_{ij} \right)$ berekend worden door W_i te vermenigvuldigen met $\hat{\mathbf{x}}_i$:

$$\hat{\mathbf{x}}_i \left(\sum_{j=0}^{MN-1} w_{ij} \right) = W_i \hat{\mathbf{x}}_i$$

We stellen dus vast dat het volledig linkerlid van Vergelijking 3.22 bepaald kan worden. We kunnen nu Vergelijking 3.21 beschouwen als een stelsel $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$ en een benaderende oplossing bepalen a.d.h.v. het conjugate gradient algoritme (Appendix D). Dit algoritme vereist echter de uitvoering van een vermenigvuldiging $\mathbf{A}\hat{\mathbf{x}}$. Door de voorgaande analyse stellen we vast dat dit mogelijk is dankzij het NLMS algoritme. Een betere benadering is zelfs om i.p.v. het NLMS algoritme, het NLMS-SC algoritme te gebruiken, aangezien deze eveneens ruiscorrelatie en signaalafhankelijkheid in rekening brengt.

Het algoritme zoals hierboven beschreven zullen we verder vernoemen als NLMS-SCB.

¹¹In praktijk is deze vermenigvuldiging niet nodig, aangezien een efficiënte implementatie van het NLMS algoritme eerst $\sum_{j=0}^{MN-1} w_{ij} \hat{\mathbf{x}}_j$ bepaalt en vervolgens deze component deelt door W_i . Die laatste deling kan dus nu overgeslagen worden.

Hoofdstuk 4

Experimentele resultaten

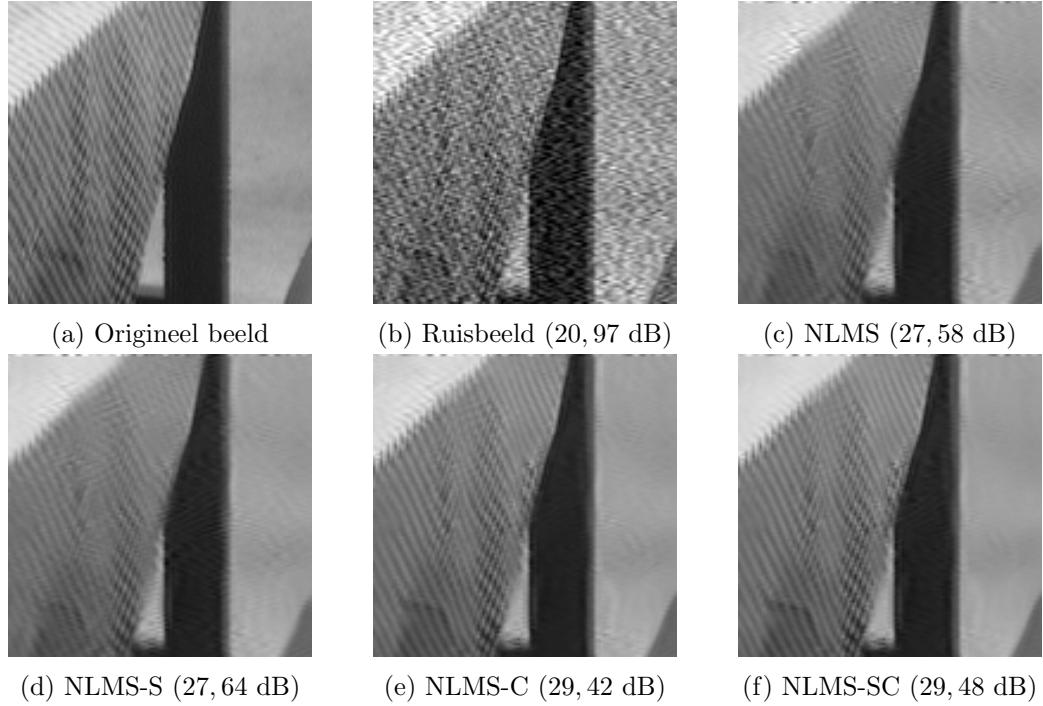
In Hoofdstuk 3 werden vier voorgestelde NLMS varianten besproken. In Hoofdstuk 2 werd een uitgebreide studie van de ruis en blur uitgevoerd. Een extra voordeel hiervan is dat we over de nodige kennis beschikken om de ruis en blur te simuleren op referentiebeelden. Bijgevolg kunnen we de vier varianten evalueren a.d.h.v. numerieke waarden (PSNR, MSE, ...) door een vergelijking van het gereconstrueerd beeld en het referentiebeeld.

4.1 Beeldoptimalisatie bij artificiële ruisbeelden

De varianten van het NLMS algoritme hebben elk op zich als doel al dan niet gecorreleerde en/of signaalafhankelijke ruis te verwijderen. In deze evaluatie zullen we nagaan of bv. het NLMS-S algoritme betere resultaten geeft bij het verwijderen van signaalafhankelijke ruis dan het NLMS algoritme. Dit zou betekenen dat de extra signaalafhankelijkheidsinformatie inderdaad van nut is.

Het probleem bij EM-beelden is dat er geen *ground truth*¹ beschikbaar is en we dus geen vergelijking kunnen maken met het ‘ideaalbeeld’. Om dit probleem op te lossen, hebben we gebruik gemaakt van de kenmerkende eigenschappen die EM ruis vertoont (Sectie 2.3). Deze eigenschappen laten ons toe de EM-ruis (signaalafhankelijk en gecorreleerd) artificieel toe te voegen aan referentiebeelden. Dit werd uitgevoerd op een gedeelte van het *Barbara* beeld (Figuur 4.1b). We kunnen bijgevolg een vergelijkende studie maken tussen enerzijds het gereconstrueerde beeld en anderzijds het origineel referentiebeeld. Deze vergelijkende studie gebeurt bovendien a.d.h.v. een numeriek resultaat, de Peak

¹Theoretisch gezien zouden we deze beelden kunnen benaderen door eenzelfde beeld verschillende keren uit te middelen. Aangezien de additieve ruis gemiddeld nul is, zou dit er voor zorgen dat alle ruis verwijderd wordt. Praktisch gezien is het echter onmogelijk om met een 3D EM eenzelfde beeld meerdere keren op te nemen. Het staal is immers vernietigd na opname.



Figuur 4.1: Beeldreconstructie met pixelgebaseerde NLMS varianten bij gecorreleerde, signaalafhankelijke ruis.

De toevoeging van extra kenmerkende eigenschappen van de ruis zorgt voor een betere beeldreconstructie. De corresponderende PSNR-waarden worden tussen haakjes vermeld.

Signal to Noise Ratio (PSNR):

$$\text{PSNR}(\mathbf{x}, \mathbf{y}) = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i=0}^{MN-1} (\mathbf{x}_i - \mathbf{y}_i)^2} \quad (4.1)$$

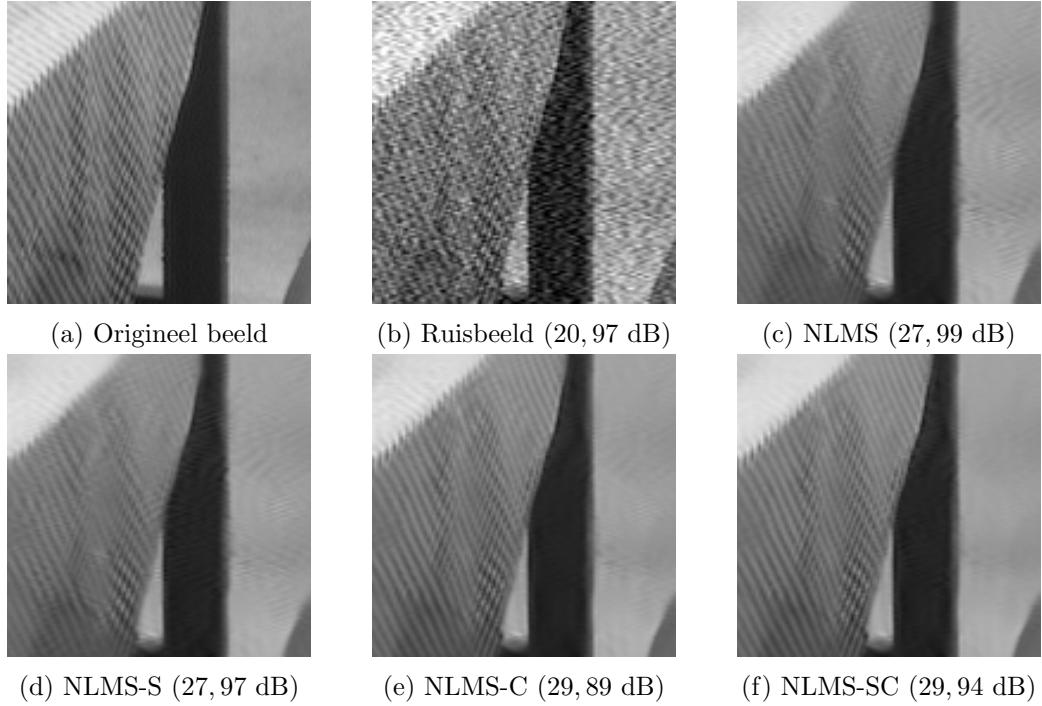
Als \mathbf{x} en \mathbf{y} dus respectievelijk het origineel en gereconstrueerd beeld is, zal een grote PSNR waarde duiden op een goede reconstructie aangezien de noemer in Vergelijking 4.1 (dit is de MSE van \mathbf{x} en \mathbf{y}) klein is.

Opmerking 4.1. We dienen op te merken dat de gebruikte algoritmen gebruik maken van verschillende parameters. De afstelling van deze parameters is geoptimaliseerd naar een maximale PSNR waarde.

We merken bij Figuur 4.1c op dat NLMS een groot deel van de ruis verwijdert zonder daarbij het beeld kwalitatief te verslechtern. Er zijn echter veel correlatie-artefacten te bespeuren. Ook zien we dat in regio's van hoge intensiteit te weinig ruis wordt verwijderd.

De NLMS-S methode (Figuur 4.1d) neemt enkel de signaalafhankelijkheid van de ruis in acht en zorgt voor een lichte verbetering in reconstructie vergeleken met NLMS. Er blijven echter nog correlatie-artefacten aanwezig.

NLMS-C (Figuur 4.1e) neemt enkel de correlatie van de ruis in acht en zorgt er voor



Figuur 4.2: Beeldreconstructie met vectorgebaseerde NLMS varianten bij ge-correleerde, signaalafhankelijke ruis.

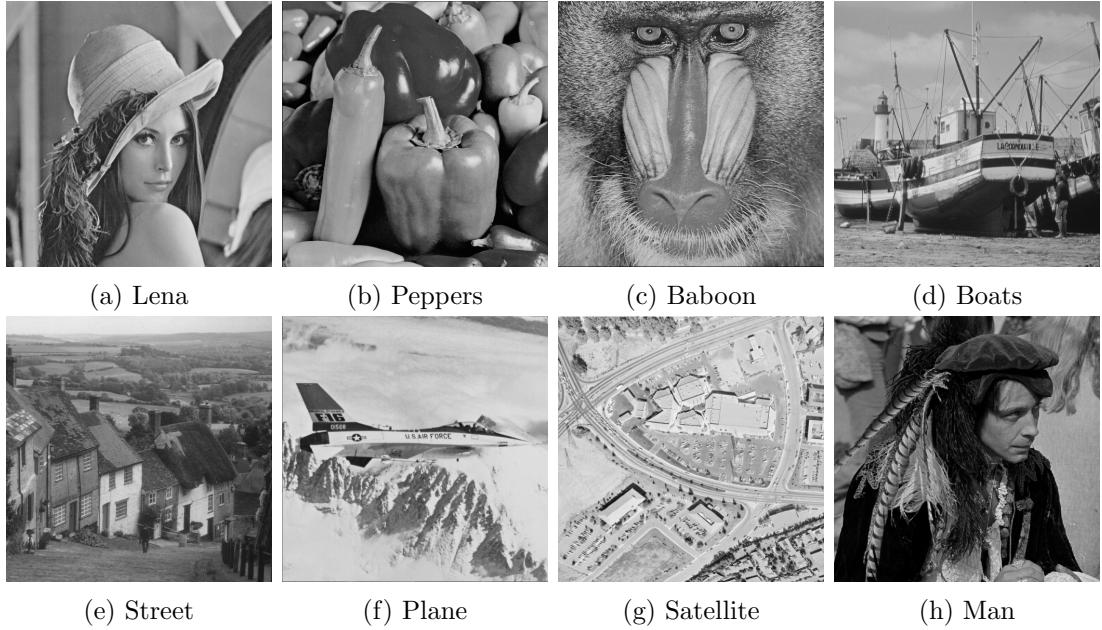
De corresponderende PSNR-waarden worden tussen haakjes vermeld.

dat de correlatie-artefacten bij NLMS-S nagenoeg verdwenen zijn. Echter, in dit geval wordt in de donkere regio's te veel ruis verwijderd en in de lichte regio's te weinig. De oorzaak hiervan ligt in het feit dat de ruis signaalafhankelijk wordt verondersteld door het algoritme.

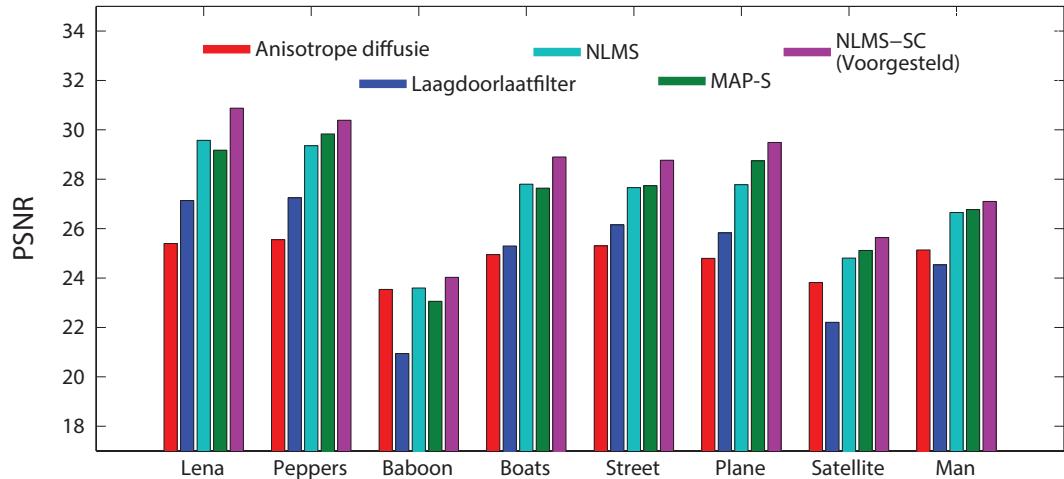
Het NLMS-SC algoritme (Figuur 4.1f) neemt zowel de signaalafhankelijkheid als de correlatie in de ruis in acht en zorgt op die manier voor de beste reconstructie. Correlatie-artefacten zijn niet meer te bespeuren en in regio's met hoge intensiteit wordt meer ruis verwijderd dan in regio's met lage intensiteit. Het verwijderen van de ruis gaat bovendien niet gepaard met significante artefacten zoals het vervagen van randen en textuur. Over het algemeen stellen we dus vast dat de kennis van correlatie en/of signaalafhankelijkheid in de ruis van belang is in het reconstructieproces. Onze voorgestelde NLMS varianten geven zowel op numeriek als op visueel vlak een beter resultaat dan het oorspronkelijk NLMS algoritme.

Opmerking 4.2. Merk op dat de vectorgebaseerde extensie uit Opmerking 3.5 inderdaad voor betere resultaten zorgt (Figuur 4.2). Gemiddeld ligt de PSNR waarde hier 0,5 dB hoger.

Uiteraard stellen we ons ook de vraag of het NLMS-SC algoritme voor een betere reconstructie zorgt, vergeleken met andere bestaande technieken. Deze vergelijking werd op verschillende referentiebeelden (Figuur 4.3) toegepast. Figuur 4.4 geeft de corresponde-



Figuur 4.3: Referentiebeelden



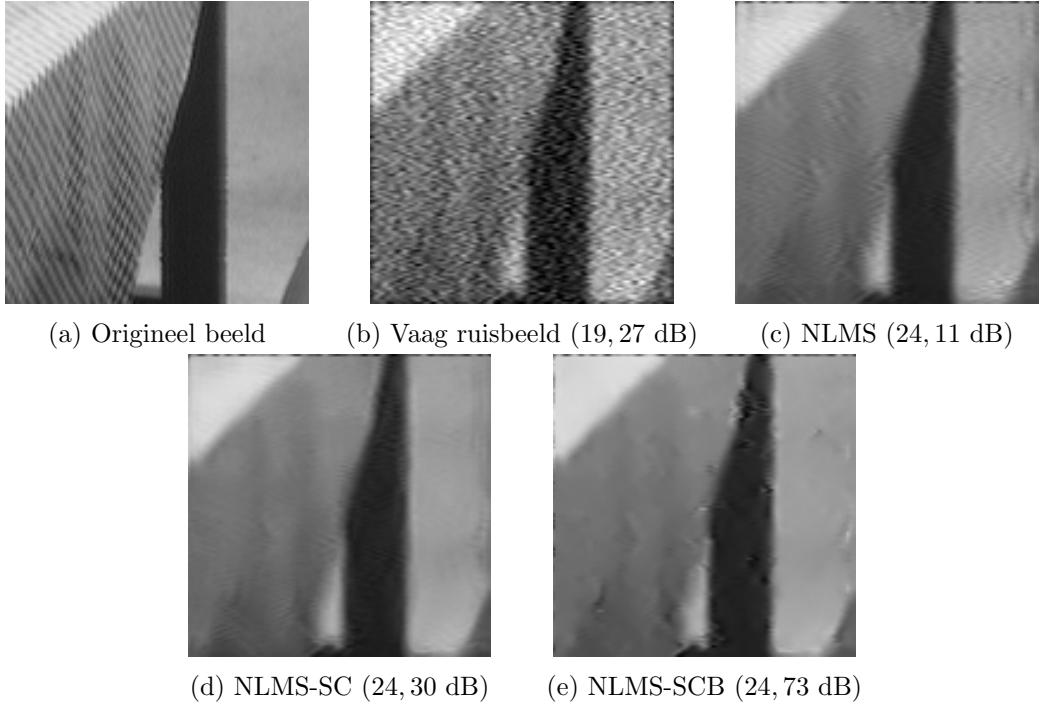
Figuur 4.4: NLMS-SC vergelijking: PSNR i.f.v. referentiebeeld.

Het NLMS-SC algoritme zorgt in alle gevallen voor een reconstructie die beter is (volgens de PSNR waarde) vergeleken met andere reconstructietechnieken.

rende PSNR waarden in functie van het gebruikte algoritme en referentiebeeld.

We merken op dat NLMS-SC over het algemeen de beste prestaties biedt.

We zullen nu het NLMS-SCB algoritme evalueren. Daarvoor gaan we eerst het beeld vervagen en vervolgens een EM ruissignaal hieraan toevoegen. Beeldreconstructie in dit geval is een veel moeilijker probleem aangezien ruisonderdrukkingstechnieken meestal voor extra blur zorgen. Bijgevolg is het gereconstrueerd beeld in veel gevallen een heel wazigere versie van het ideaalbeeld.



Figuur 4.5: Beeldreconstructie met NLMS varianten bij gecorreleerde, signaal-afhankelijke ruis en beeldvervaging.

De corresponderende PSNR-waarden worden tussen haakjes vermeld.

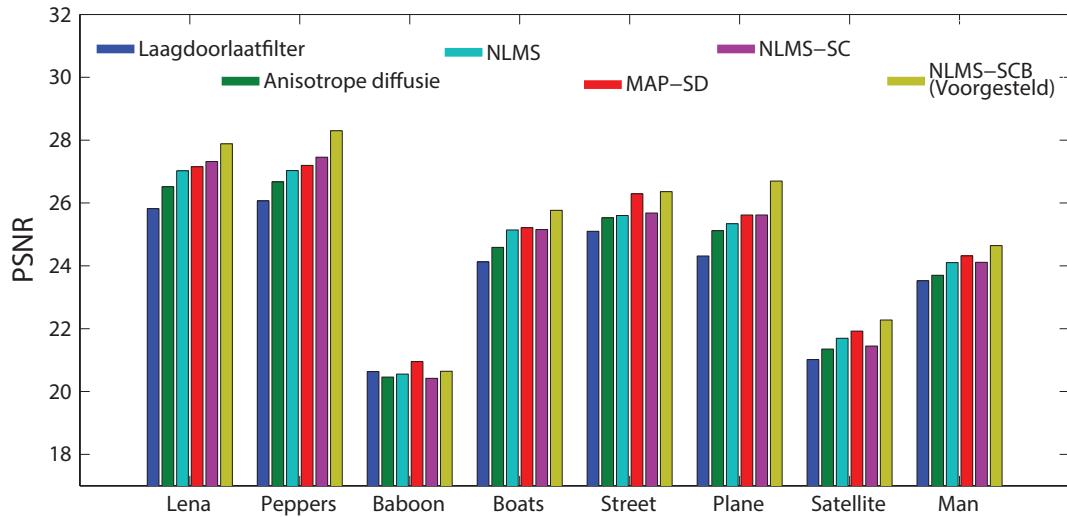
Opmerking 4.3. Merk op dat we bij het reconstrueren van EM-beelden (nog) geen kennis hebben van de blurkernel. We zullen voor de eenvoudig een relatief kleine 3×3 Gaussiaanse blurkernel gebruiken met $\sigma = 1$ hiervoor gebruiken. We dienen dus in het achterhoofd te houden dat deze kernel nog geoptimaliseerd dient te worden.

Figuur 4.5 toont het verschil in reconstructie wanneer we een geblurred ruizig beeld proberen te restoreren a.d.h.v. de NLMS algoritmen. Zoals verwacht is een groot deel van de textuur verdwenen door de vervaging. We zien echter dat de NLMS-SCB reconstructie een kwalitatievere reconstructie biedt tegenover NLMS(-SC). De gereconstrueerde beelden zijn licht verscherpt. Daarbij merken we wel op dat bij het NLMS-SCB algoritme NLMS-SC in elke iteratie opgeroepen wordt en bijgevolg dus een stuk trager is.

Vervolgens maken we ook nog een vergelijkende studie met hedendaagse technieken (Figuur 4.6). Het NLMS-SCB algoritme zorgt over het algemeen voor de beste reconstructieresultaten.

4.2 Beeldoptimalisatie bij EM-beelden

We komen nu tot de uiteindelijke toepassing van onze ontwikkelde algoritmen op de EM-beelden. Een bijkomende moeilijkheid hierbij is het bepalen van de parameters



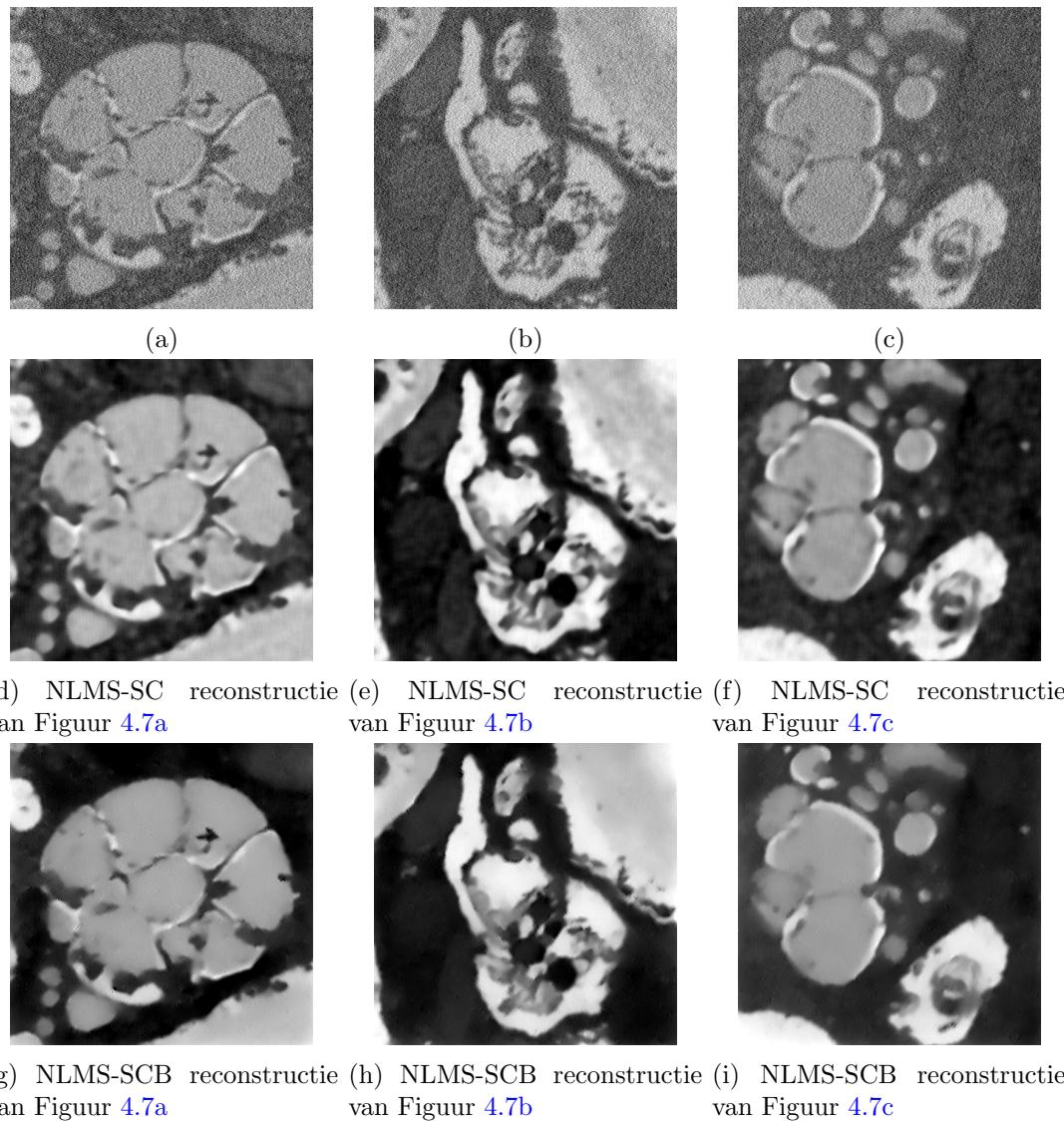
Figuur 4.6: NLMS-SCB vergelijking: PSNR i.f.v. referentiebeeld.
Het NLMS-SCB algoritme zorgt in nagenoeg alle gevallen voor een reconstructie die beter is (volgens de PSNR waarde) vergeleken met andere reconstructietechnieken.

aangezien deze nu niet meer kunnen geoptimaliseerd worden naar een maximale PSNR. Bijgevolg hebben we deze op zo een manier bepaald dat een maximale hoeveelheid ruis wordt verwijderd zonder daarbij inhoudelijke structuur, randen en textuur te verliezen (of toch zo veel mogelijk te behouden). De resultaten van het NLMS-SC en NLMS-SCB algoritme zijn te zien in Figuur 4.7.

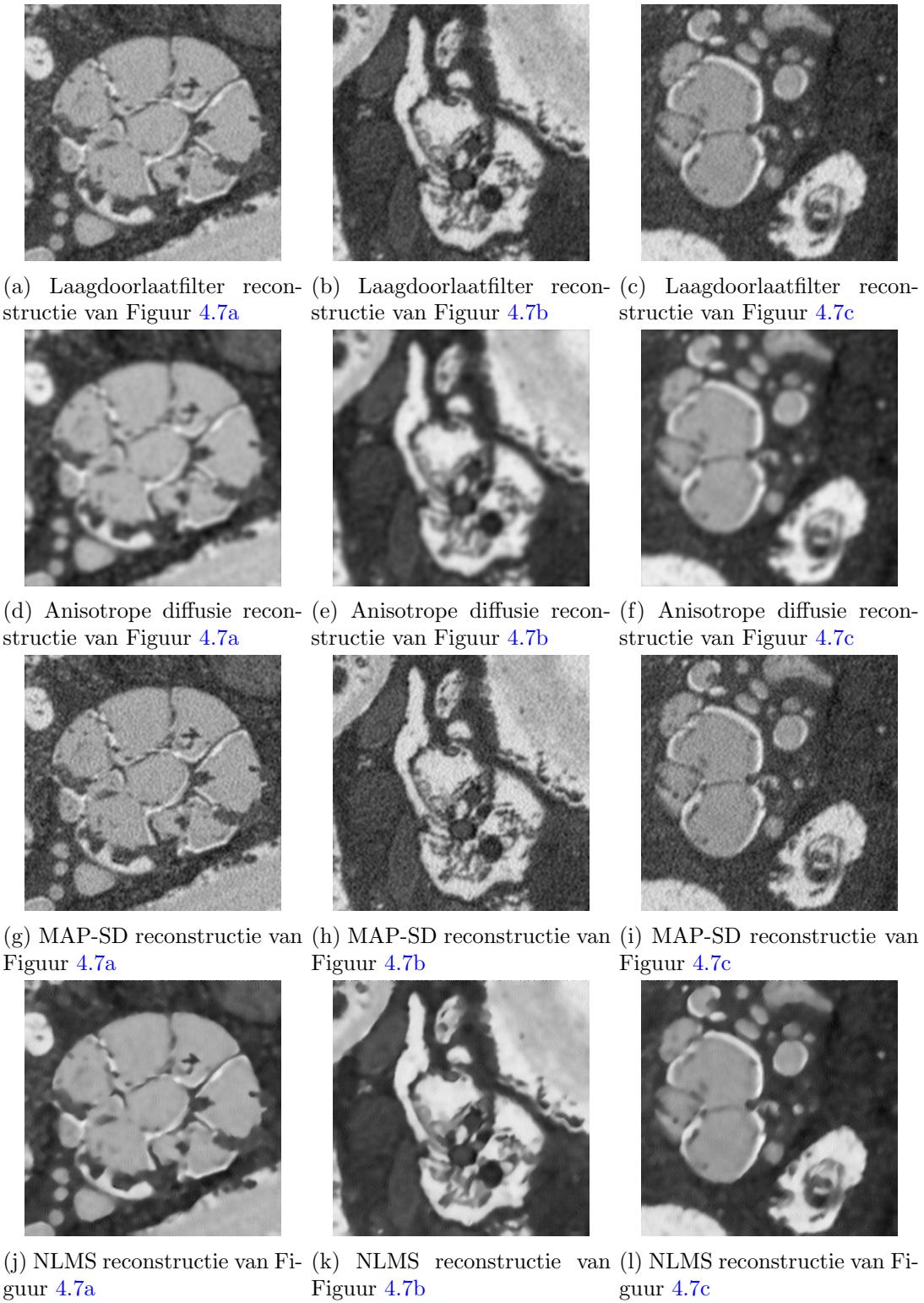
We merken op dat beide algoritmen een grote hoeveelheid ruis verwijderen. Dit gaat echter gepaard met een lichte vervaging in kleinere beeldstructuren. Deze vervaging kan vermeden worden door de parameters zodanig af te stellen dat minder ruis wordt verwijderd. In de uiteindelijke reconstructie dient dus een afweging gemaakt te worden tussen het belang van beeldstructuren en ruisonderdrukking.

Tenslotte hebben we dezelfde EM-beelden gereconstrueerd a.d.h.v. eerder vermelde reconstructietechnieken: laagdoorlaatfilter, anisotrope diffusie, MAP-SD en NLMS. De gereconstrueerde beelden worden weergegeven in Figuur 4.8. We kunnen hier duidelijk uit concluderen dat het laagdoorlaatfilter² (Figuur 4.8a, 4.8b en 4.8c) en MAP-SD (Figuur 4.8g, 4.8h en 4.8i) een grote hoeveelheid ruis achterlaten in de reconstructies. Verder zorgt anisotrope diffusie (Figuur 4.8d, 4.8e en 4.8f) voor een betere ruisonderdrukking, dit gaat echter gepaard met wazigere randen. Het NLMS algoritme zorgt voor een kwalitatievere ruisonderdrukking, hoewel er nog steeds artefacten zichtbaar zijn vanwege o.a. de ruiscorrelatie.

²Merk op dat dit de huidig gebruikte reconstructietechniek binnen EM is.



Figuur 4.7: Beeldreconstructie van EM beelden met het NLMS-SC(B) algoritme



Figuur 4.8: Beeldreconstructie van EM beelden met alternatieve algoritmen

Hoofdstuk 5

Conclusie en toekomstig werk

Na het bekijken van de opgenomen SEM-beelden, werd een sterke aanwezigheid van ruis vastgesteld. Dit wordt o.a. veroorzaakt door complexe elektronische systemen die in de EM geïntegreerd zijn. Verder zijn de beelden sterk vervaagd door o.a. de technische limietgevingen van (magnetische) lenzen.

Een gedetailleerde analyse van de ruis gaf een speciaal type bij EM vrij. In de eerste plaats is de aanwezige ruis Gaussiaans verdeeld en stationair (d.w.z. de ruisstatistieken zijn onafhankelijk van de positie in het beeld). Verder vonden we ook dat EM beelden signaalafhankelijke ruis vertoonden, iets wat gemodelleerd wordt via een Poissonmodel. Bijgevolg is de EM-ruis een combinatie van Gaussiaanse ruis en Poissonruis. Vervolgens werd ook vastgesteld dat de ruis sterk gecorreleerd is in de horizontale richting.

We hebben een methode uitgewerkt die ons zou moeten toelaten de blurkernel van de EM ruis te bepalen. Jammer genoeg hebben we deze niet kunnen vastleggen door een technisch defect. Dit vormt echter geen hindernis voor de ontwikkeling van onze techniek aangezien de blurkernel eveneens als parameter gemodelleerd kan worden.

De meeste beeldreconstructietechnieken gaan in hun model uit van een specifiek type ruis en/of blur. EM-beeldreconstructie vormt hier echter een uitzondering op aangezien vrijwel alle uitdagende aspecten op vlak van ruisonderdrukking en deconvolutie tegelijk aan bod komen.

Om de signaalafhankelijke, gecorreleerde ruis en blur te verwijderen uit de EM-beelden zijn we vertrokken van de recente NLMS techniek. Deze techniek is gebaseerd op het uitmiddelen van pixels waarvan de corresponderende regio's rond die pixels gelijkaardig zijn. Op basis van deze techniek hebben we twee NLMS varianten beschreven die signaalafhankelijke en gecorreleerde ruis verwijderen (NLMS-S en respectievelijk NLMS-C). Een derde variant (NLMS-SC) werd vervolgens afgeleid die zowel signaalafhankelijkheid als ruiscorrelatie in het spel brengt. Zowel NLMS-S als NLMS-C zorgden voor een verbetering in reconstructie. De combinatie van de twee, NLMS-SC, gaf echter de grootste

verbetering op vlak van beeldreconstructie. Bovendien hebben we ook aangetoond dat het NLMS-SC algoritme kwalitatievere reconstructies geeft dan alternatieve ruisonderdrukkingsalgoritmen. Het is dus van essentieel belang om de eventuele aanwezigheid van signaalafhankelijkheid of correlatie in de ruis, uit te buiten om een zo optimaal resultaat te garanderen.

Vervolgens werd een laatste NLMS variant (NLMS-SCB) ontworpen die d.m.v. een Bayesiaanse schatter de blurkernel mee in rekening bracht. Deze Bayesiaanse schatting was mogelijk aangezien NLMS van een wit, Gaussiaans ruismodel vertaald kan worden in een MAP schatter met aangepaste prior. Het NLMS-SCB algoritme geeft betere reconstructies vergeleken met het NLMS-SC algoritme en andere hedendaags gebruikte technieken.

Tot slot werden de NLMS-SC en NLMS-SCB algoritmen toegepast op EM-beelden. Experts hebben daarbij vastgesteld dat de beelden kwalitatief enorm verbeterd werden. Beide technieken verwijderen nagenoeg alle ruis, zonder daarbij de randen en texturen significant te beïnvloeden. Het NLMS-SCB algoritme slaagt erin om randen te verscherpen. Kleine structuren worden echter iets sterker vervaagd dan bij het NLMS-SC algoritme.

We hebben dus een state-of-the-art beeldoptimalisatie-algoritme opgesteld dat beelden met signaalafhankelijke, gecorreleerde ruis en blur reconstrueert. We hebben aangetoond dat dit algoritme een grote hoeveelheid ruis en blur kan onderdrukken zonder daarbij de beeldinhoud significant te beïnvloeden. Dit hebben we niet alleen vastgesteld op artificieel ruizige en geblurde beelden, maar eveneens op EM-beelden. Voor die eerste hebben we een numerieke evaluatie uitgevoerd en op die manier vastgesteld dat ons algoritme de huidige reconstructietechnieken overtreft. Verder hebben experts eveneens vastgesteld dat de gereconstrueerde EM-beelden van hogere kwaliteit waren dan de huidige reconstructies. Bijgevolg is een kwalitatievere beeldanalyse mogelijk gemaakt op EM-beelden.

De ontwikkeling van het NLMS-SC en NLMS-SCB algoritme zijn zeker en vast niet het slot betreffende beeldoptimalisatie bij EM-beelden.

Eerst en vooral werd in onze analyse nog geen EM-blurkernel vastgelegd. Kennis van deze blurkernel zal zeker en vast in het voordeel spelen bij de uitvoering van het NLMS-SCB algoritme.

Vervolgens dienen we op te merken dat we een vectorgebaseerde uitbreiding (Opmerking 3.2) voorzien hebben in het NLMS-SC algoritme. Dit kan echter ook op bij NLMS-SCB toegepast worden om kwalitatievere resultaten te bekomen.

Verder hebben we bij de ruisanalyse vastgesteld dat de ruisstatistieken afhankelijk zijn van de dwell time. Een grotere dwell time resultereert namelijk in minder ruizige beelden.

Een bijkomend nadeel van deze grotere dwell time is de langere opnametijd. Een analyse van de dwell time afhankelijkheid kan alleen maar in het voordeel spelen van het degradatiemodel.

De gereconstrueerde beelden zijn enorm parameterafhankelijk. In het eenvoudig geval dat we de originele beelden ter beschikking hadden, konden deze geoptimaliseerd worden naar een maximale numerieke evaluatiewaarde (bv. PSNR). In praktische toepassingen is een origineel beeld echter niet beschikbaar en zullen de parameters idealiter geoptimaliseerd moeten worden a.d.h.v. een ander criterium.

Tot slot dienen we op te merken dat de NLMS-varianten enerzijds goede resultaten geven op vlak van beeldreconstructie. Anderzijds zijn de algoritmen computationally minder efficiënt. Vooral het NLMS-SCB algoritme, dat iteratief NLMS-SC oproept, heeft nood aan optimalisatie om in praktijk meer van toepassing te kunnen zijn.

Bijlage A

Notaties

Met het oog op duidelijkheid en eenvoud is het in sommige omstandigheden aan te raden een beeld ofwel voor te stellen als matrix ofwel als vector. We dienen echter consistent te zijn bij dit notatieverschil zodat het eindresultaat na een beeldtransformatie (het toepassen van een filter) equivalent is in beide notaties. De volgende secties verduidelijken de twee gekozen notaties die in deze scriptie gebruikt worden.

A.1 Convolutienotatie

In de convolutienotatie zal een $M \times N$ beeld rechtstreeks voorgesteld worden als een matrix $x \in \mathbb{R}^{M \times N}$:

$$x = \begin{pmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,N-1} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M-1,0} & x_{M-1,1} & \cdots & x_{M-1,N-1} \end{pmatrix}$$

Een $M \times N$ filter zal eveneens rechtstreeks voorgesteld worden als een matrix $h \in \mathbb{R}^{M \times N}$:

$$h = \begin{pmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,N-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M-1,0} & h_{M-1,1} & \cdots & h_{M-1,N-1} \end{pmatrix}$$

Het toepassen van een filter h op een beeld x komt in dit geval neer op een convolutie $h * x$ zoals gedefinieerd in Appendix B:

$$(h * x)_{i,j} = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h_{k,l} x_{i+k,j+l} \quad (\text{A.1})$$

waarbij de indices $i - k$ en $j - l$ respectievelijk modulo M en N worden beschouwd (of equivalent, het beeld wordt periodiek uitgebreid).

A.2 Matrixproductnotatie

In de matrixproductnotatie zal een $M \times N$ beeld voorgesteld worden als een vector $\mathbf{x} \in \mathbb{R}^{MN}$ door de pixels rij per rij te overlopen:

$$\mathbf{x} = \begin{pmatrix} x_{0,0} \\ x_{0,1} \\ \vdots \\ x_{0,N-1} \\ x_{1,0} \\ x_{1,1} \\ \vdots \\ x_{M-1,N-1} \end{pmatrix}$$

We willen een filter $h \in \mathbb{R}^{M \times N}$ (in rechtstreekse matrixrepresentatie) voorstellen als een matrix $\mathbf{H} \in \mathbb{R}^{MN \times MN}$ zodat de convolutie-operatie gegeven door Vergelijking A.1 equivalent kan geschreven worden als een matrixproduct $\mathbf{H}\mathbf{x}$. Hiervoor zullen we op de eerste rij van \mathbf{H} de elementen van h terug rij per rij moeten invoegen om $(\mathbf{H}\mathbf{x})_0 = (h * x)_{0,0}$ te berekenen. De berekening van $(\mathbf{H}\mathbf{x})_1 = (h * x)_{0,1}$ zal echter verwachten dat de tweede rij van \mathbf{H} identiek is aan de eerste rij op een translatie van één positie na. Een filter h zal dus in matrixproductnotatie geschreven worden als:

$$\mathbf{H} = \begin{pmatrix} h_{0,0} & h_{0,1} & h_{0,2} & \cdots & h_{M-1,N-1} \\ h_{M-1,N-1} & h_{0,0} & h_{0,1} & \cdots & h_{M-1,N-2} \\ h(M, N - 1) & h_{M-1,N-1} & h_{0,0} & \cdots & h_{M-1,N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{0,1} & h_{0,2} & h_{0,3} & \cdots & h_{0,0} \end{pmatrix}$$

Bijgevolg is \mathbf{H} een circulante matrix. Het toepassen van een filter h wordt voorgesteld door een matrixproduct:

$$\mathbf{H}\mathbf{x} = \begin{pmatrix} h_{0,0} & h_{0,1} & h_{0,2} & \cdots & h_{M-1,N-1} \\ h_{M-1,N-1} & h_{0,0} & h_{0,1} & \cdots & h_{M-1,N-2} \\ h(M, N - 1) & h_{M-1,N-1} & h_{0,0} & \cdots & h_{M-1,N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{0,1} & h_{0,2} & h_{0,3} & \cdots & h_{0,0} \end{pmatrix} \begin{pmatrix} x_{0,0} \\ x_{0,1} \\ x_{0,2} \\ \vdots \\ x_{M-1,N-1} \end{pmatrix}$$

Merk op dat dit bij constructie equivalent is aan Vergelijking A.1. Bovendien zal vanwege de associativiteit van de convolutie-operatie (Eigenschap B.2) het toepassen van twee filters (in matrixproductnotatie) \mathbf{F} en \mathbf{G} hetzelfde resultaat geven als het toepassen van één filter \mathbf{FG} , de matrixvermenigvuldiging van \mathbf{F} en \mathbf{G} . In convolutienotatie zou dit neerkomen op een convolutie $f * g$.

Bijlage B

Eigenschappen van convoluties

De convolutie-operator $*$ is onmisbaar binnen de beeldverwerking. Hier is enerzijds een continue en anderzijds een discrete variant van. Aangezien digitale beelden opgebouwd zijn uit een discreet aantal pixels, is het voldoende voor onze doeleinden om enkel de discrete convolutie te beschouwen.

Definitie B.1. Zij $f, g \in \mathbb{R}^{M \times N}$, dan is de discrete convolutie van f en g gedefinieerd als

$$(f * g)(i, j) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f(k, l)g(i+k, j+l) \quad (\text{B.1})$$

waarbij we aannemen dat f en g periodiek uitgebreid worden, d.w.z. $f(i, j) = f(i + \alpha M, j + \beta N)$ en $g(i, j) = g(i + \alpha M, j + \beta N)$ voor alle $\alpha, \beta \in \mathbb{Z}$.

Zoals vermeld in Appendix A wordt dit in matrixproductnotatie equivalent voorgesteld door een matrixvermenigvuldiging $\mathbf{F}\mathbf{G}$ of $\mathbf{F}\mathbf{g}$ (afhankelijk of het tweede element een beeld of convolutiekernel voorstelt).

We leiden nu enkele eigenschappen van convoluties af.

Eigenschap B.2. Zij $f, g, h \in \mathbb{R}^{M \times N}$, dan geldt de gelijkheid:

$$(f * g) * h = f * (g * h) \quad (\text{B.2})$$

Bewijs. We tonen dit rechtstreeks aan via de definitie:

$$\begin{aligned}
 ((f * g) * h)_{i,j} &= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} (f * g)_{k,l} h_{i+k,j+l} \\
 &= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n} g_{k+m,l+n} h_{i+k,j+l} \\
 &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} g_{k+m,l+n} h_{i+k,j+l} \\
 &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n} \sum_{k'=0}^{M-1} \sum_{l'=0}^{N-1} g_{k',l'} h_{i+m+k',j+n+l'} \\
 &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n} (g * h)_{i+m,j+n}
 \end{aligned}$$

waarbij we gebruik hebben gemaakt van de periodieke uitbreiding van g en h . \square

Eigenschap B.3. *Zij $x \in \mathbb{R}^{M \times N}$ een constant beeld ($x(i,j) = c \in \mathbb{R}$) en $h \in \mathbb{R}^{M \times N}$ een genormaliseerd filter, dan geldt:*

$$h * x = x$$

of in matrix-vector representatie:

$$\mathbf{H}\mathbf{x} = \mathbf{x}$$

Bewijs. Aangezien beide representaties equivalent zijn, is het voldoende om dit aan te tonen voor één van beide. We zullen de eerste gelijkheid aantonen:

$$\begin{aligned}
 (h * x)_{i,j} &= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h_{k,l} x_{i+k,j+l} \\
 &= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h_{k,l} c \\
 &= c \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h_{k,l} \\
 &= c \\
 &= x_{i,j}
 \end{aligned}$$

Bijgevolg is $h * x = x$ en, equivalent, $\mathbf{H}\mathbf{x} = \mathbf{x}$. \square

Bijlage C

De fouriertransformatie

De fouriertransformatie is een essentiële transformatie die een beeld in het spatiale domein kan transformeren naar een beeld in het frequentiedomein en omgekeerd. Hier is enerzijds een continue en anderzijds een discrete variant van. Aangezien digitale beelden opgebouwd zijn uit een discreet aantal pixels, is het voldoende voor onze doeleinden om enkel de discrete fouriertransformatie te beschouwen.

Definitie C.1. Zij $f \in \mathbb{R}^{M \times N}$, dan is de discrete fouriertransformatie (DFT) van f gedefinieerd als

$$F = (\mathcal{F}(f))_{\omega_x, \omega_y} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n} \exp \left(-j2\pi \left(\frac{\omega_x}{M} m + \frac{\omega_y}{N} n \right) \right)$$

met j de complexe eenheid.

Het is eenvoudig na te gaan dat de inverse discrete fouriertransformatie (IDFT) van F bijgevolg gegeven is door:

$$f = (\mathcal{F}^{-1}(F))_{k,l} = \sum_{\omega_x=0}^{M-1} \sum_{\omega_y=0}^{N-1} F_{\omega_x, \omega_y} \exp \left(j2\pi \left(\frac{\omega_x}{M} k + \frac{\omega_y}{N} l \right) \right)$$

We geven nog een belangrijke eigenschap van de discrete fouriertransformatie uit de beeldverwerking zonder bewijs die we in deze scriptie zullen gebruiken.

Eigenschap C.2. Zij $f, g \in \mathbb{R}^{M \times N}$, dan geldt de gelijkheid:

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g)$$

waarbij de operator \cdot wijst op een puntsgewijs product.

Deze eigenschap heeft een groot voordeel op implementatieniveau aangezien de fourier-transformatie in de meeste gevallen sneller kan uitgevoerd worden dan een convolutie.

Bijlage D

Het conjugate gradient algoritme

Het conjugate gradient algoritme [30] is een iteratieve methode om stelsels van de vorm

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (\text{D.1})$$

op te lossen. Hierbij wordt verondersteld dat $\mathbf{A} \in \mathbb{R}^{M \times M}$ symmetrisch en positief definit is en $\mathbf{x}, \mathbf{b} \in \mathbb{R}^M$. De bedoeling is om de vector \mathbf{x} zo goed mogelijk benaderen met een vector $\hat{\mathbf{x}}$.

Definitie D.1. Twee vectoren $\mathbf{u}, \mathbf{v} \in \mathbb{R}^M$ worden \mathbf{A} -geconjugeeerd genoemd als:

$$\mathbf{u}^T \mathbf{A} \mathbf{v} = 0$$

Aangezien $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{A}} = \mathbf{u}^T \mathbf{A} \mathbf{v}$ een inproduct definieert, zullen we zeggen dat twee vectoren orthogonaal zijn als ze \mathbf{A} -geconjugeeerd zijn.

Veronderstel dat we een basis van M onderling orthogonale vectoren \mathbf{p}_i hebben. Dan kunnen we \mathbf{x} ontbinden in deze basis:

$$\mathbf{x} = \sum_{i=0}^M \alpha_i \mathbf{p}_i \quad (\text{D.2})$$

met $\alpha_i \in \mathbb{R}$. Bijgevolg geldt dan vanwege de onderlinge orthogonaliteit:

$$\begin{aligned} \mathbf{p}_k^T \mathbf{b} &= \mathbf{p}_k^T \mathbf{A} \mathbf{x} \\ &= \sum_{i=0}^M \alpha_i \mathbf{p}_k^T \mathbf{A} \mathbf{p}_i \\ &= \alpha_k \mathbf{p}_k^T \mathbf{A} \mathbf{p}_k \end{aligned}$$

en we vinden dus:

$$\alpha_k = \frac{\mathbf{p}_k^T \mathbf{b}}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k} \quad (\text{D.3})$$

Om \mathbf{x} te vinden, dienen we dus een basis van onderling \mathbf{A} -geconjugeerde vectoren \mathbf{p}_i te vinden en vervolgens de coëfficiënten α_i te bepalen zoals in Vergelijking D.3. De oplossing wordt dan gegeven door Vergelijking D.2. In praktijk zal men zich echter niet bezig houden met het berekenen van deze volledige basis.

We zijn op zoek naar een minimum van de volgende kwadratische functie:

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}$$

Een huidige oplossing \mathbf{x}_{k-1} zal dus in een volgende iteratie de waarde van f in \mathbf{x}_i zo klein mogelijk trachten te houden. Een manier hoe dit kan gebeuren is door in de negatieve gradiëntrichting te lopen. Aangezien de gradiënt van f gelijk is aan $\mathbf{A}\mathbf{x} - \mathbf{b}$, wordt de residuvector dus gegeven door:

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$$

Het gradient descent algoritme zou zeggen dat de volgende iteratie $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{r}_k$. We zullen nu echter als bijkomende voorwaarde van de zoekrichtingen \mathbf{p}_k (\mathbf{r}_k bij gradient descent) veronderstellen dat ze onderling geconjugeerd zijn. Bovendien wordt de volgende zoekrichting geconstrueerd uit de huidige residuvector en alle eerder afgeleide zoekrichtingen (op een gelijkaardige manier als bij Gram-Schmidt orthogonalisatie):

$$\mathbf{p}_k = \mathbf{r}_k - \sum_{i < k} \frac{\mathbf{p}_i^T \mathbf{A} \mathbf{r}_k}{\mathbf{p}_i^T \mathbf{A} \mathbf{p}_i} \mathbf{p}_i$$

Bijgevolg wordt de volgende benaderende oplossing gegeven door:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

met

$$\begin{aligned} \alpha_k &= \frac{\mathbf{p}_k^T \mathbf{b}}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k} \\ &= \frac{\mathbf{p}_k^T (\mathbf{r}_{k-1} + \mathbf{A}\mathbf{x}_{k-1})}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k} \\ &= \frac{\mathbf{p}_k^T \mathbf{r}_{k-1}}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k} \end{aligned}$$

aangezien \mathbf{p}_k en \mathbf{x}_{k-1} orthogonaal zijn.

De bovenstaande procedure wordt iteratief verder gezet, startend van een initiële oplossing \mathbf{x}_0 , tot een bepaald stopcriterium voldaan is.

Bibliografie

- [1] Hooke R. *Micrographia*. Courier Dover Publications, 1665.
- [2] M. Malpighi. *De Pulmonibus Observationes Anatomicae*. 1663.
- [3] Optical microscope, . URL http://en.wikipedia.org/wiki/Optical_microscope.
- [4] Enkelvoudige microscoop, . URL http://upload.wikimedia.org/wikipedia/commons/4/4f/Microscope_simple_diagram.png.
- [5] Samengestelde microscoop, . URL http://upload.wikimedia.org/wikipedia/commons/f/fe/Microscope_compound_diagram.png.
- [6] Electron microscope, . URL http://en.wikipedia.org/wiki/Electron_microscope.
- [7] Scherptediepte in beelden, . URL <http://img407.imageshack.us/img407/6410/xwingcomparisonwf6.jpg>.
- [8] H. Luong S. Lippens Y. Saeys B. Goossens J. Aelterman, J. De Vylder and W. Philips. The image processing revolution: The next step in electron microscopy. *Proc. 3View Users Group Meeting and Symposium*, 2014.
- [9] A. Pižurica B. Goossens, H.Q. Luong and W. Philips. An improved non-local means algorithm for image denoising. *International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.
- [10] A. Pižurica J. Aelterman, B. Goossens and W. Philips. Removal of correlated rician noise in magnetic resonance imaging. *16th European Signal Processing Conference*, 2008.
- [11] A. Pižurica B. Goossens and W. Philips. Removal of correlated noise by modeling the signal of interest in the wavelet domain. *IEEE Transactions on Image Processing*, 2009.

- [12] D. Giacalone S. Battiato A. Bosco, R. A. Bruna and R. Rizzo. Signal dependent raw image denoising using sensor noise characterization via multiple acquisitions. *Digital Photography VI*, 2010.
- [13] A. Foi. Clipped noisy images: Heteroskedastic modeling and practical denoising. *Signal Processing*, 2009.
- [14] P. Weiss J. Fehrenbach and C. Lorenzo. Variational algorithms to remove stationary noise: Application to spim imaging. *IEEE Transactions on Image Processing*, 2011.
- [15] G. Poggi M. Matrecano and L. Verdoliva. Improved bm3d for correlated noise removal. *VISAPP (1)*, 2012.
- [16] V. V. Lukin N. N. Ponomarenko R. Öktem, K. Egiazarian and O. V. Tsymbal. Locally adaptive dct filtering for signal-dependent noise removal. *EURASIP Journal on Advances in Signal Processing*, 2007.
- [17] M. Tanaka X. Liu and M. Okutomi. Estimation of signal dependent noise parameters from a single image. *IEEE International Conference on Image Processing*, 2013.
- [18] V. Katkovnik K. Dabov, A. Foi and K. Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 2007.
- [19] P.T. Vanathi V.R. Vijaykumar and P. Kanagasabapathy. Fast and efficient algorithm to remove gaussian noise in digital images. *IAENG International Journal of Computer Science*, 2010.
- [20] M. Chandra S. Lal and G. K. Upadhyay. Noise removal algorithm for images corrupted by additive gaussian noise. *International Journal of Recent Trends in Engineering*, 2009.
- [21] Chi-square goodness-of-fit, . URL <http://www.stat.yale.edu/Courses/1997-98/101/chigf.htm>.
- [22] V. Katkovnik A. Foi, M. Trimeche and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 2007.
- [23] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.
- [24] B. Coll A. Buades and J.-M. Morel. A non-local algorithm for image denoising. *Image Processing On Line*, 2011.

- [25] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989.
- [26] R. R. Coiffman and D. L. Donoho. Translation-invariant de-noising. *Lecture Notes in Statistics: Wavelets and Statistics*, 1995.
- [27] B. Yu S. G. Chang and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 2000.
- [28] S. Osher L. I. Rudin and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 1992.
- [29] F. Park T. Chan, S. Esedoglu and A. Yip. Recent developments in total variation image restoration. *In Mathematical Models of Computer Vision*, 2005.
- [30] Conjugate gradient method, . URL http://en.wikipedia.org/wiki/Conjugate_gradient_method.
- [31] H. Luong J. De Vylder A. Pizurica J. Aelterman, B. Goossens and W. Philips. Combined non-local and multi-resolution sparsity prior in image restoration. *IEEE International Conference on Image Processing*, 2012.