

# Automatische aberratie correctie van een lage- energie elektronen microscop

Bachelor Thesis

T.P.M. van Steenis

6 Juni 2014

**De Haagse Hogeschool:** Technische Natuurkunde

**Studentnummer:** 10038418

**Coaches:** Dr. J.L. van Yperen

Dr. P. Menger

**Gecommitteerde:** Dhr. Ing. K. Hoogendijk

**LION, Universiteit Leiden**

**Begeleider:** Ing. M.B.S. Hesselberth



## Abstract

This report outlines the origin of aberrations in a Low Energy Electron Microscope (LEEM) and how to measure and automatically correct them. The aberrations in a LEEM are mainly caused by the cathode objective lens and are a limiting factor for the achievable resolution. To optimise the resolution of the microscope it is therefore necessary to cancel out the aberrations by using an electron mirror, which can introduce aberrations of the same magnitude but with opposite sign of the cathode objective lens. Aberrations can be measured by using Low Energy Electron Diffraction (LEED) patterns. In a LEED pattern spots are displaced from the centre spot as a function of the angle through the system. The magnitude of this displacement depends on defocus and spherical aberrations. By fitting a function of the form  $C_1\alpha + C_3\alpha^3$ , where  $C_1$  stands for defocus, and  $C_3$  stand for third order spherical aberrations, and  $\alpha$  for the angle, defocus en third order spherical aberrations can be extracted.

To automatically correct aberrations present in the microscope, LEED patterns are analysed with a program written with the programming language Python. Due to limiting time the present form of this program only contains a noise filter, which uses Wiener deconvolution in the frequency domain, and spot detection where a 3x3 pattern is shifted over the filtered pattern. To ensure that the spot detection works properly ten theoretical LEED patterns are generated with pre-set values for defocus en third order spherical aberration. These patterns are analysed by the written code to find the spot locations which are then exported to OriginPro<sup>tm</sup> to make a fit. From these fits the pre-set values for defocus en third order spherical aberrations are recovered. The results of the fits gave an average uncertainty of 0,30% and 1,31% for respectively defocus en third order spherical aberrations. This shows that the spot detection works properly. In the final part of the report there will be a description about how to implement order recognition using Delaunay-triangulation and Hasse-diagrams in the future.



# Inhoudsopgave

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| <b>ABSTRACT</b>                                                             | <b>I</b>  |
| <b>LIJST VAN AFKORTINGEN</b>                                                | <b>IV</b> |
| <b>1. INLEIDING</b>                                                         | <b>1</b>  |
| 1.1 KADER VAN DEZE SCRIPTIE                                                 | 1         |
| <b>2. DE LAGE-ENERGIE ELEKTRONEN MICROSCOOP</b>                             | <b>3</b>  |
| 2.1 AFBEELDINGSMODI                                                         | 4         |
| 2.2 ELEKTRONENOPTIEK                                                        | 5         |
| 2.2.1 Kathode objectieflens                                                 | 5         |
| 2.2.2 Lens aberraties                                                       | 6         |
| 2.2.3 Elektronenspiegel                                                     | 9         |
| <b>3. LAGE-ENERGIE ELEKTRONEN DIFFRACTIE</b>                                | <b>11</b> |
| 3.1 KRISTALSTRUCTUREN                                                       | 11        |
| 3.2 HET RECIPROKE ROOSTER                                                   | 12        |
| 3.3 INVALSHOEK VAN EEN DIFFRACTIEBUNDEL                                     | 13        |
| 3.4 DE EWALD CIRKEL CONSTRUCTIE                                             | 13        |
| 3.5 VOORBEELD: HET Si(111) OPPERVLAK                                        | 14        |
| <b>4. ABERRATIES METEN EN CORRIGEREN</b>                                    | <b>17</b> |
| 4.1 Tilt en astigmatisme                                                    | 17        |
| 4.2 Defocus en sferische aberraties                                         | 18        |
| 4.3 Chromatische aberraties                                                 | 18        |
| <b>5. AUTOMATISCHE ABERRATIECORRECTIE</b>                                   | <b>21</b> |
| 5.1 GEBRUIKTE ALGORITMEN                                                    | 21        |
| 5.1.1 Ruisonderdrukking                                                     | 21        |
| 5.1.2 Spotherkenning                                                        | 23        |
| 5.1.3 Orde toekenning                                                       | 24        |
| 5.1.4 Data fitten & parameters extraheren                                   | 26        |
| <b>6. CONCLUSIE</b>                                                         | <b>29</b> |
| <b>BIBLIOGRAFIE</b>                                                         | <b>30</b> |
| <b>BIJLAGE A DE PYTHON CODE</b>                                             | <b>31</b> |
| <b>BIJLAGE B SPOTDETECTIE OP VERSCHILLENDE LEED PATRONEN</b>                | <b>38</b> |
| <b>BIJLAGE C FIT VAN GEDETECTEERDE SPOTS BIJ GEGENEREERDE LEED PATRONEN</b> | <b>41</b> |
| <b>BIJLAGE D GRAFISCHE GEBRUIKERSOMGEVING</b>                               | <b>44</b> |
| <b>BIJLAGE E DE ORIGINELE OPDRACHTOMSCHRIJVING</b>                          | <b>46</b> |

## **Lijst van afkortingen**

|      |                                               |
|------|-----------------------------------------------|
| LEEM | Lage energie elektronen microscopie           |
| LEED | Lage energie elektronen diffractie            |
| MPA  | Magnetisch Prismastelsel/Magnetic Prism Array |

# 1. Inleiding

Lage-energie elektronen microscopie is een techniek die wordt gebruikt voor oppervlakte onderzoek en die is uitgevonden door Ernst Bauer in 1962. In 1985 kwam deze echter pas echt tot ontwikkeling. Bij een lage-energie elektronen microscoop (LEEM) worden elektronen met een energie van typische enkele tientallen elektronvolt op een preparaat geschoten. Door de lage energie van de elektronen is de indring diepte in het preparaat maar enkele Ångstroms waardoor het mogelijk is om alleen de bovenste atomaire lagen van het preparaat te bekijken. Hierdoor kunnen interacties op oppervlakken op een schaal van enkele atomen realtime worden waargenomen. Buiten afbeeldingen maken van het beeldvlak heeft een LEEM ook de mogelijkheid om in het diffractievlak te kijken. Hierdoor is het mogelijk om informatie te verkrijgen over de atomaire positie van de atomen in het kristal.

De theoretisch maximaal haalbare resolutie van een LEEM is ten gevolge van de diffractielimiet circa 0,5 nanometer [1]. In de praktijk is die door aberraties in de kathode objectieflens echter rond de vijf á acht nanometer afhankelijk van de energie van de inkomende elektronen. Door het toevoegen van een elektronenspiegel aan het systeem is het mogelijk om deze aberraties te corrigeren waardoor de resolutie van de LEEM richting de diffractielimiet gaat. De record resolutie van een LEEM staat op het moment op 1,4 nanometer en is gemeten met de LEEM opstelling bij de universiteit Leiden.

## 1.1 Kader van deze scriptie

Aberraties van de kathode objectieflens zijn een beperkende factor in de haalbare resolutie van een LEEM. Een elektronenspiegel is in staat om de eerste orde chromatische en derde orde sferische aberraties van de kathode objectieflens te corrigeren door gelijke aberraties maar met omgekeerd teken te introduceren. Om de elektronenspiegel correct te kunnen instellen is het nodig om de aanwezige aberraties te meten. Dit gebeurt door het bestuderen van lage-energie elektronen diffractie patronen (LEED). Het doel van deze scriptie is te beschrijven hoe de aberraties ontstaan (hoofdstuk 2) en hoe ze te corrigeren (hoofdstuk 4). Om de LEED patronen te kunnen interpreteren is er begrip nodig van het ontstaan van de LEED patronen (hoofdstuk 3). In het laatste hoofdstuk wordt beschreven hoe de aberraties automatisch gecorrigeerd kunnen worden door een programma geschreven in de programmeertaal Python.





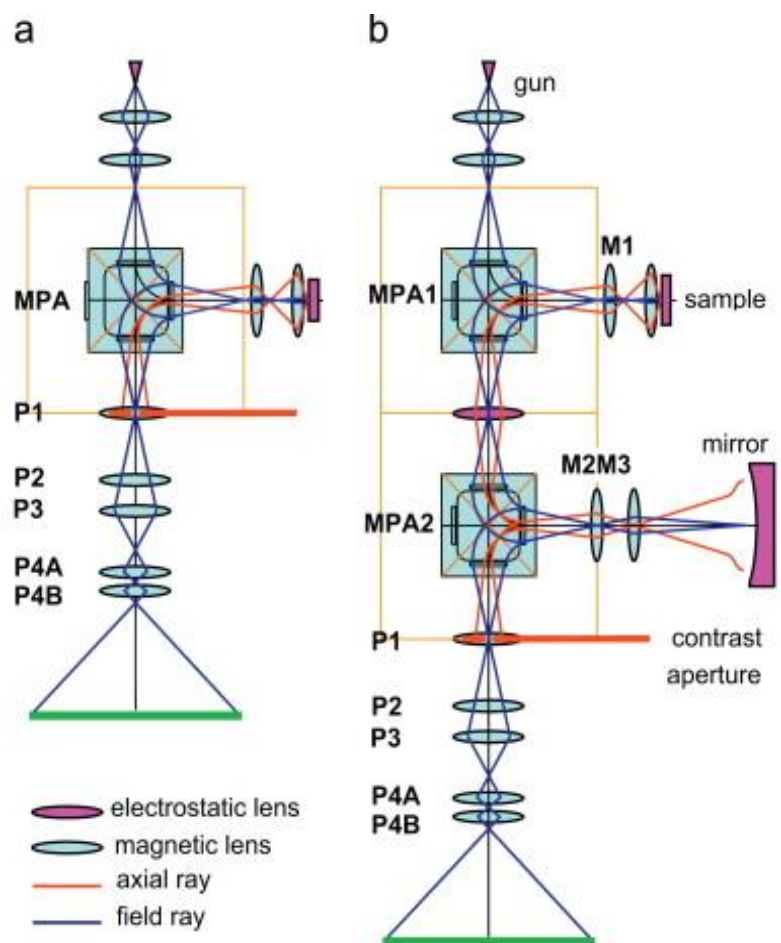
## 2. De lage-energie elektronen microscoop

Bij een LEEM wordt er gebruik gemaakt van een elektronenkanon dat elektronen met een energie van rond de 15 KeV wegschiet. De elektronen worden door 2 lenzen, waarmee de grootte van het belichte oppervlak wordt ingesteld, gefocusseerd in het eerste magnetische prisma stelsel (MPA) dat de bundel 90 graden afbuigt. Na het MPA wordt de bundel door het objectieflenssysteem, dat bestaat uit een overbrengingslens en de objectieflens, gecollimeerd op het preparaat. Het preparaat wordt op een negatieve potentiaal gehouden van ongeveer -15 kV waardoor de elektronen afgeremd worden en het preparaat raken met een energie van ongeveer 0 – 100 eV.

Op een kristallijn preparaat ondergaan de elektronen diffractie en worden vervolgens door de negatieve potentiaal terug versneld naar de objectieflens. Hierbij wordt er een diffractie patroon gevormd in het achterbrandvlak van de objectieflens en een afbeelding van het oppervlak op een grotere afstand. De overbrengingslens focuseert de elektronenbundel terug in het MPA waar deze weer 90 graden worden afgebogen: op deze manier worden de inkomende en uitgaande elektronenbundel gesplitst. In een ongecorrigeerde LEEM, zoals te zien in Figuur 2-1(a), gaan de elektronen hierna door het projectiesysteem en vormen een afbeelding op de detector. Door de lens P2 aan en uit te zetten kan er geschakeld worden tussen het diffractie vlak en het beeldvlak. [2, 3]

De LEEM in Leiden is een aberratie gecorrigeerde LEEM, zoals deze in Figuur 2-1(b) is te zien. Voor het corrigeren van de aberraties van de objectieflens wordt er na MPA1 een elektronenspiegel toegevoegd.

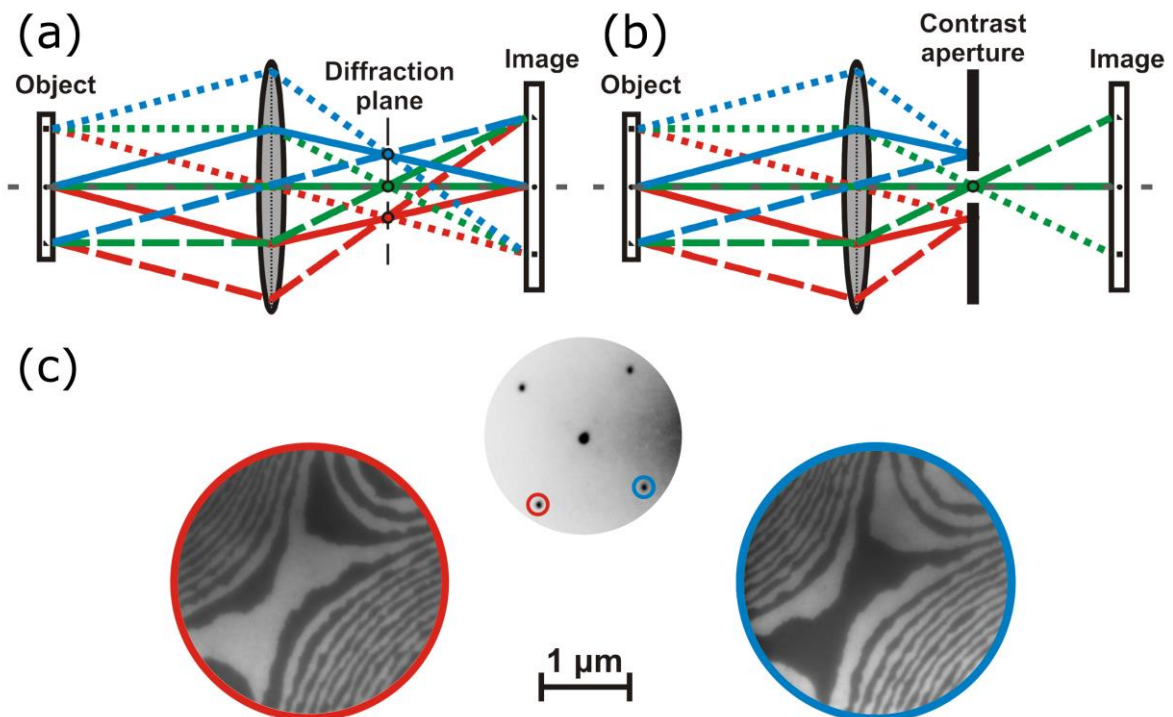
Voor het implementeren van de elektronenspiegel wordt een tweede MPA (MPA2) aan het systeem toegevoegd. MPA2 buigt de elektronenbundel wederom 90 graden af waarna de elektronen door twee magnetische lenzen naar de spiegel gaan. De spiegel corrigeert de aberraties van de objectieflens en reflecteert de bundel door de twee lenzen terug MPA2 in. MPA2 buigt de elektronen weer 90 graden af waarna de bundel het projectiesysteem ingaat en er afbeelding op de detector wordt gevormd. [2, 3]



Figuur 2-1: (a) Schematische weergave van een ongecorrigeerd LEEM systeem. (b) Schematische weergave van de aberratie gecorrigeerde LEEM zoals die bij de Universiteit van Leiden staat. De oranje vierkanten om de MPAs laten zien waar de diffractie vlakken liggen. De diagonale in de prisma's zijn beeldvlakken [2]

## 2.1 Afbeeldingsmodi

Met een LEEM is het mogelijk om verschillende afbeeldingsmethoden te gebruiken voor het vormen van het beeld op de detector. Door een diafragma te plaatsen in een diffractie vlak is het mogelijk om een diffractiespot te selecteren waaruit de afbeelding gevormd wordt. In Figuur 2-2(b) is dit schematisch weergegeven. Wanneer de speculaire diffractiespot (groene lijnen in Figuur 2-2(a) en (b)), ook nulde orde spot genoemd, geselecteerd wordt is er sprake van bright field imaging. Hier wordt het contrast in de afbeelding gevormd door de gelaagdheid van het preparaat en is afhankelijk van lokale amplitudes. Voor elke andere (hogere orde) spot is er sprake van dark field imaging. Hier verschijnen de gebieden die bijdragen aan de geselecteerde diffractie spot als heldere gebieden. In Figuur 2-2(c) is het verschil te zien tussen twee hogere orde spots. [3]



Figuur 2-2: (a) Stralendiagram van een lens met aangegeven voorwerpsvlak, diffractievlak en beeldvlak, (b) Stralendiagram met ingevoegd contrast diafragma in het diffractievlak, (c) LEED patroon van het Si(001)-(2x1) oppervlak (gecentreerd) en de bijbehorende dark-field afbeeldingen voor de omringde spots [1]

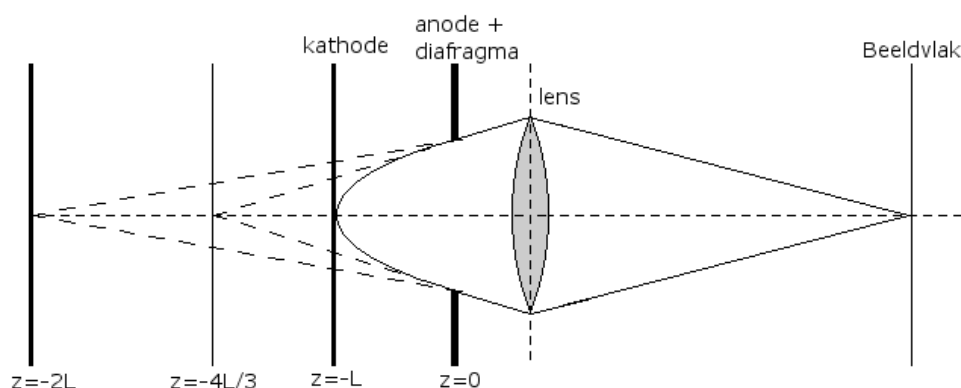
Een andere afbeeldingsmethode is foto elektron emissie microscopie (PEEM). Bij PEEM wordt er geen gebruik gemaakt van elektronen uit een elektronenkanon om een afbeelding te vormen maar van een UV ontladingslamp die middels het foto-elektrisch effect elektronen vrijmaakt uit het preparaat. Alleen elektronen die aan het oppervlak vrij worden gemaakt hebben een vrije weglengte groot genoeg om het preparaat te verlaten en dus is PEEM net als lage-energie elektronen microscopie een oppervlaktegevoelige techniek. [1]

## 2.2 Elektronenoptiek

Elektronen hebben net als fotonen een deeltjes- en een golfkarakter. Hierdoor heeft elektronenoptiek bepaalde overeenkomsten met lichtoptiek. Het is mogelijk om een elektronenbundel te beschrijven met behulp van stralen net als bij licht optiek. In licht optica worden de fotonen afgebogen door een transparant materiaal met een brekingsindex  $n$ . Bij elektronenoptica is het gebruik van materialen voor het afbuigen van elektronen niet mogelijk omdat deze snel worden geabsorbeerd door vaste materie. Hiervoor in de plaats wordt er gebruik gemaakt van elektrische en magnetische velden. Voor een uitgebreide beschrijving over de werking van elektronen lenzen zie [4].

### 2.2.1 Kathode objectieflens

De kathode objectieflens is één van de belangrijkste onderdelen van de LEEM aangezien de eigenschappen ervan de resolutie bepalen. In Figuur 2-3 is de kathode objectieflens schematisch weergegeven. Hierin is de kathode het preparaat dat op -15kV wordt gehouden. Het preparaat is daardoor onderdeel van de objectieflens. De anode is geaard en zit op anderhalve millimeter ( $L$ ) van de kathode. Het sterke elektrische veld dat hierdoor ontstaat tussen het preparaat en de anode is nodig om de aberraties zo klein mogelijk te houden. Figuur 2-3 laat zien dat de kathode objectieflens uit drie belangrijke beeldvormende elementen bestaat, te weten het elektrisch veld, het diafragma in de anode en de magnetische lens. [1]



Figuur 2-3: Schematische weergave van de kathode objectieflens. Het elektrisch veld alleen creëert een virtuele afbeelding op  $Z = -2L$  met een vergroting  $M = 1$ . Door de opening in de anode bevindt deze afbeelding zich in  $Z = -4/3L$  met een vergroting van  $M = 2/3$ . De lens maakt hier een afbeelding van in het beeldvlak.

Het elektrisch veld tussen preparaat en anode remt de inkomende elektronen af en versnelt ze na interactie met het preparaat weer terug richting de anode. De parabolische baan die de elektronen volgen bij het versnellen is weergegeven in Figuur 2-3. Door de versnelling verandert de hoek waarmee de elektronen door de anodeopening gaan. Hierdoor wordt er door het elektrisch veld een virtuele afbeelding geplaatst op een afstand  $Z = -2L$  van de anode met een vergroting van  $M = 1$ . In de anode bevindt zich een opening om de elektronen door te laten waardoor het elektrisch veld hier niet homogeen is. De afwijking in het elektrisch veld resulteert in een brandpuntsafstand. Welke gegeven wordt [5] door:

$$f = \frac{4V}{G_2 - G_1} \quad (2.1)$$

waarbij  $V$  de potentiaal is waarover de elektronen zijn versneld en  $G_2 - G_1$  het verschil in veldsterkte in is tussen anode en kathode. Voor de kathode objectieflens geeft dit een brandpuntsafstand van  $-4L$  ervan uitgaande dat de eindsnelheid van het elektron veel groter is dan de beginsnelheid. Het resultaat van de afwijking in het elektrische veld ten gevolge van het diafragma is dat het virtuele beeld van het elektrisch veld een virtueel beeld op een afstand  $Z = -4L/3$  van de anode oplevert met een vergroting  $M = 2/3$ . Van dit virtuele beeld wordt vervolgens door de elektromagnetische lens een reëel beeld gevormd. [1]

### 2.2.2 Lens aberraties

In een elektronenmicroscopie zijn de aberraties van lenzen een beperkende factor voor de resolutie van de afbeelding. De twee belangrijkste aberraties zullen hier behandeld worden. Dit zijn monochromatische en chromatische aberraties.

#### Monochromatische aberraties

Monochromatische aberraties zijn afwijkingen van de paraxiale (eerste orde) optica. Voorbeelden zijn astigmatisme, defocus, tilt en sferische aberraties. De afwijking in de afbeelding ten gevolge van deze aberraties wordt gegeven [6] door:

$$r_g = \frac{\partial \chi(\alpha)}{\partial \alpha} \quad (2.2)$$

Waarin  $\alpha$  de hoek in radialen is, die de elektronen maken met de optische as is en  $\chi(\alpha)$  de totale monochromatische aberratiecoëfficiënt in meters is. Welke tot de vijfde orde gegeven [6] wordt door:

$$\chi(\alpha) = C_1 \frac{(\alpha_x^2 + \alpha_y^2)}{2} + C_{1,2a} \frac{(\alpha_x^2 + \alpha_y^2)}{2} + C_{1,2b} \alpha_x \alpha_y + C_3 \frac{(\alpha_x^4 + 2\alpha_x^2 \alpha_y^2 + \alpha_y^4)}{4} + C_5 \frac{(\alpha_x^6 + 2\alpha_x^4 \alpha_y^2 + 2\alpha_x^2 \alpha_y^4 + \alpha_y^6)}{6} \quad (2.3)$$

Waarin:

|                      |                                                                              |       |
|----------------------|------------------------------------------------------------------------------|-------|
| $C_1$                | = Defocus coëfficiënt                                                        | [m]   |
| $C_3, C_5$           | = Respectievelijk derde en vijfde orde sferische aberratie coëfficiënt       | [m]   |
| $C_{1a}, C_{1b}$     | = De astigmatisme coëfficiënten                                              | [m]   |
| $\alpha_x, \alpha_y$ | = Hoek tussen elektronen en optische as voor respectievelijk de x en de y-as | [rad] |

Voor het gemak wordt in onderstaande de x- en y-component van de hoek die de elektronen maken met de optische as samengevoegd tot  $\alpha$ .

De defocus coëfficiënt is een weergave van de mate van uit focus zijn van de afbeelding. De afwijking als gevolg van defocus van de lens wordt gegeven door:

$$r_d = C_1 \alpha \quad (2.4)$$

Sferische aberratie is het effect dat elektronen die via een grotere hoek door de optische as van de lens gaan, sterker worden afgebogen dan elektronen die onder een kleinere hoek door de optische as van de lens gaan. In Figuur 2-4 is dit schematisch weergegeven. De verschuiving in de beeldafstand  $\Delta b$  is afhankelijk van de imperfecties van de lens en de hoek  $\alpha_1$  die de elektronen maken met de optische as. Via een machtreeks kan dit weergegeven [4] worden:

$$\Delta b = c_2 \alpha_1^2 + c_4 \alpha_1^4 + \dots \quad (2.5)$$

Waarin  $c_2$  en  $c_4$  coëfficiënten zijn die staan voor de imperfectie van de lens. Uit Figuur 2-4 is middels goniometrie een verband af te leiden tussen de beeldafstand en de hoek die de elektronen maken met de optische as volgens:

$$b_1 = \frac{\tan(\alpha_1) \cdot v}{\tan(\alpha_2)} = \frac{\alpha_1 \cdot v}{\alpha_2} \quad (2.6)$$

Waarin  $v$  de voorwerpsafstand is en verondersteld wordt dat de hoek die de elektronen maken met de optische as klein is. Wanneer de elektronen, die onder een grotere hoek door de optische as gaan, het beeldvlak bereiken zullen deze een afwijking  $r_s$  hebben ten opzichte van de elektronen met een kleinere hoek met de optische as volgens:

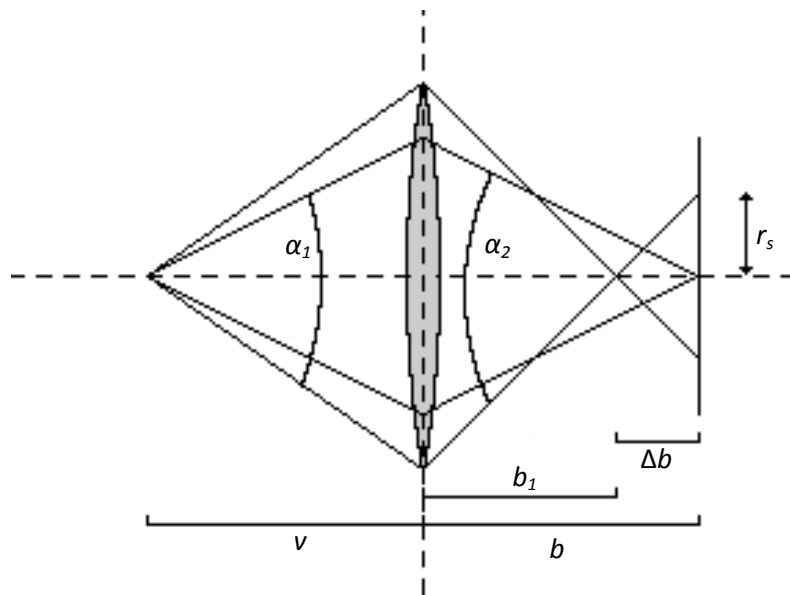
$$r_s = \Delta b \tan(\alpha_2) \approx \Delta b \alpha_2 \quad (2.7)$$

Wanneer verondersteld wordt dat  $\Delta b \ll b$  kan formule (2.6) omgeschreven worden tot:

$$\alpha_2 = \alpha_1 \frac{v}{b} \quad (2.8)$$

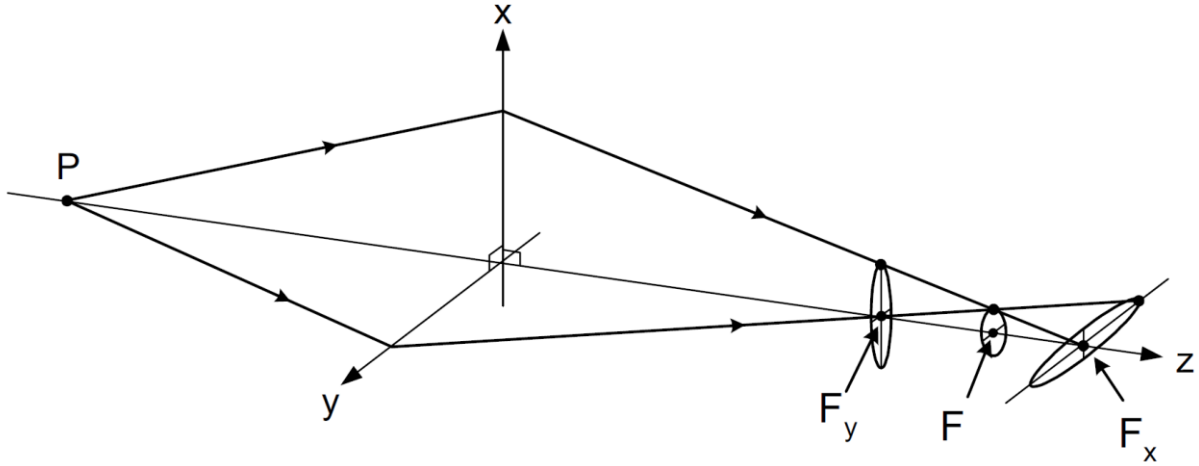
Door dit samen met formule (2.5) te substitueren in formule (2.7) volgt voor de afwijking  $r_s$  in de afbeelding:

$$r_s = (c_2 \alpha_1^2 + c_4 \alpha_1^4 + \dots) \cdot \alpha_1 \frac{v}{b} = C_3 \alpha_1^3 + C_5 \alpha_1^5 + \dots \quad (2.9)$$



Figuur 2-4: Schematische weergave van sferische aberraties bij een dunne lens.

Astigmatisme wordt veroorzaakt door een niet perfect symmetrische lens waardoor elektronen in het  $x$ - en  $y$ -vlak van de lens verschillend worden afgebogen. In Figuur 2-5 is te zien dat elektronen die door de  $x$ -as van de lens gaan minder sterk worden afgebogen dan elektronen die door de  $y$ -as van de lens gaan. De brandpuntsafstanden voor deze twee assen zijn dus verschillend. [4]



Figuur 2-5: Schematische weergave van astigmatisme. Elektronen verlaten een punt  $P$  en worden gefocust in een ellips in  $F_x$  en  $F_y$  of in een cirkel met straal  $R$  tussen de twee. [4]

Het ontstaan van tilt in de afbeelding vindt zijn oorsprong in elektronen die niet loodrecht op het preparaat invallen. Dit kan komen door bundel tilt of door preparaat tilt. Bij bundel tilt gaan de inkomende elektronen niet parallel aan de optische as door het systeem. Bij preparaat tilt staat het preparaat niet loodrecht op de optische as waardoor inkomende elektronen niet loodrecht op het preparaat invallen.

Aan de hand van de nulde orde spot kan tilt nader toegelicht worden. Wanneer er geen bundel tilt en geen preparaat tilt aanwezig is zal de nulde orde bundel op de optische as liggen. Wanneer er wel één van de twee aanwezig is zal de nulde orde bundel onder een hoek met de optische as door het systeem gaan. Om de tilt in formule (2.3) op te nemen moeten  $\alpha_x$  en  $\alpha_y$  vervangen worden met  $\alpha_x + \tau_x$  en  $\alpha_y + \tau_y$ . Hierin zijn  $\tau_x$  en  $\tau_y$  dan de tilt in respectievelijk het x- en het y-vlak. Deze correctie van de hoeken werkt door in de overige aberraties. [6]

### **Chromatische aberraties**

Chromatische aberratie is het verschijnsel dat elektronen met een grotere energie minder sterk worden afgebogen dan elektronen met een kleinere energie. Hierdoor zullen elektronen met een kleinere energie een afstand  $r_c$  verschuiven in het beeldvlak. Deze verschuiving is gegeven [4] door:

$$r_c = \Delta b \tan(\alpha_2) \approx \Delta b \cdot \alpha_2 \quad (2.10)$$

Hierin wordt er aangenomen dat de hoek die de elektronen maken met de optische as klein is. Voor de vergroting van de lens geldt:

$$M = \frac{\tan \alpha_1}{\tan \alpha_2} \approx \frac{\alpha_1}{\alpha_2} \quad (2.11)$$

Door formule (2.11) om te schrijven en te substitueren in formule (2.10) volgt voor de afwijking in het beeldvlak:

$$r_c = \Delta b \cdot \frac{\alpha_1}{M} \quad (2.12)$$

Wanneer verondersteld wordt dat de lens dun is, geldt de lenzenvergelijking  $1/b + 1/v = 1/f$ . De afgeleide hiervan bij constante voorwerpsafstand geeft:

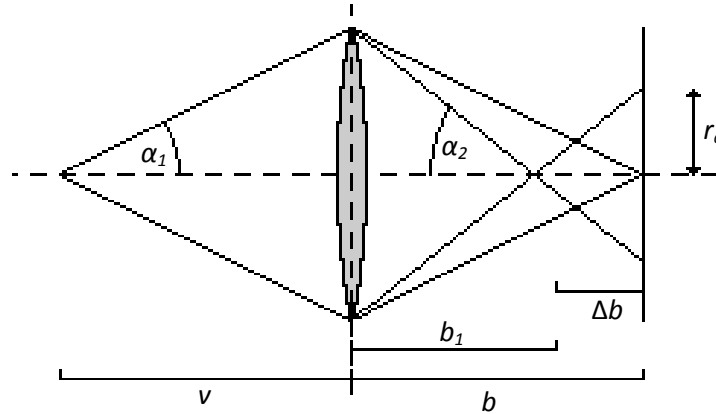
$$\Delta b = \frac{b^2}{f^2} \Delta f \quad (2.13)$$

Wanneer  $M \gg 1$  kan via de lenzenvergelijking gesteld worden dat  $v \approx f$  en dat  $b \approx Mf$  en gaat formule (2.13) over in  $\Delta b \approx M^2 \Delta f$ . Dit substitueren in formule (2.12) geeft:

$$r_c \approx M \alpha_1 \Delta f \quad (2.14)$$

Voor de brandpuntsafstand van een dunne lens geldt  $f = AE_0$  [4], waarin  $A$  een onafhankelijk constante is van de energie van het elektron in  $\text{m} \cdot \text{J}^{-1}$  zodat de afgeleide geeft  $\Delta f = A \Delta E_0 = (f/E_0) \Delta E_0$ . Wanneer dit gesubstitueerd wordt in formule (2.14) geeft dit voor het verlies in resolutie ten gevolge van chromatische aberratie:

$$r_c = M f \alpha_1 \left( \frac{\Delta E_0}{E_0} \right) = \alpha C_c \left( \frac{\Delta E_0}{E_0} \right) \quad (2.15)$$



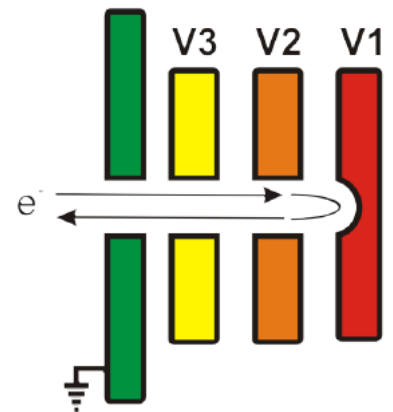
Figuur 2-6: Schematische weergave van chromatische aberraties bij een dunne lens.

### 2.2.3 Elektronenspiegel

De elektronenspiegel van de LEEM bestaat uit drie ringvormige elektroden en één ronde elektrode. De eerste ringvormige elektrode wordt op aardpotentiaal gehouden. De daaropvolgende elektroden worden op steeds groter wordende negatieve potentiaal gehouden  $V_3$ ,  $V_2$  en  $V_1$  gehouden. De ronde elektrode wordt op  $V_1$  gehouden, zoals te zien in Figuur 2-7.

Er wordt verondersteld dat de LEEM aberratie gecorrigeerd is wanneer het totaal van laagste orde chromatische en derde orde sferische aberraties gelijk zijn aan nul. Voor het correct instellen van de voltages van de spiegel moet er rekening worden gehouden met drie parameters van de spiegel. Deze parameters kunnen onafhankelijk van elkaar ingesteld worden en zijn  $C_{3, spiegel}$ ,  $C_{c, spiegel}$  en de brandpuntsafstand van de spiegel. De verbanden tussen de parameters en de potentialen op de spiegel zijn bij benadering:

$C_{c, spiegel} \propto V_1$ ,  $C_{3, spiegel} \propto V_3 - V_2$ , in het vervolg  $V_{32}$  genoemd, en de brandpuntsafstand wordt



Figuur 2-7: Schematische weergave van de elektronenspiegel die gebruikt wordt voor het corrigeren van chromatische en sferische aberraties. [1]

gecontroleerd met  $V_2$ . Voor een nauwkeurigere controle over de aberraties van de spiegel moeten  $C_{c, spiegel}$  en  $C_{3, spiegel}$  als functie van  $V_1$  en  $V_{32}$  genomen worden zoals weergegeven in Figuur 2-8(b). [1, 2, 3]

De aberratiecoëfficiënten van de spiegel schalen niet 1:1 met de aberratiecoëfficiënten van de objectieflens. De objectieflens vormt samen met alle optiek tot de elektronenspiegel een beeld dat als voorwerp dient voor de spiegel. Het verband tussen de aberraties van de objectieflens en de spiegel wordt dan gegeven [7] door:

$$\begin{aligned} C_{c, spiegel} \alpha_2 &= M \cdot C_{c, lens} \alpha_1 \\ C_{c, spiegel} &= M \cdot C_{c, lens} \left( \frac{\alpha_1}{\alpha_2} \right) \end{aligned} \quad (2.16)$$

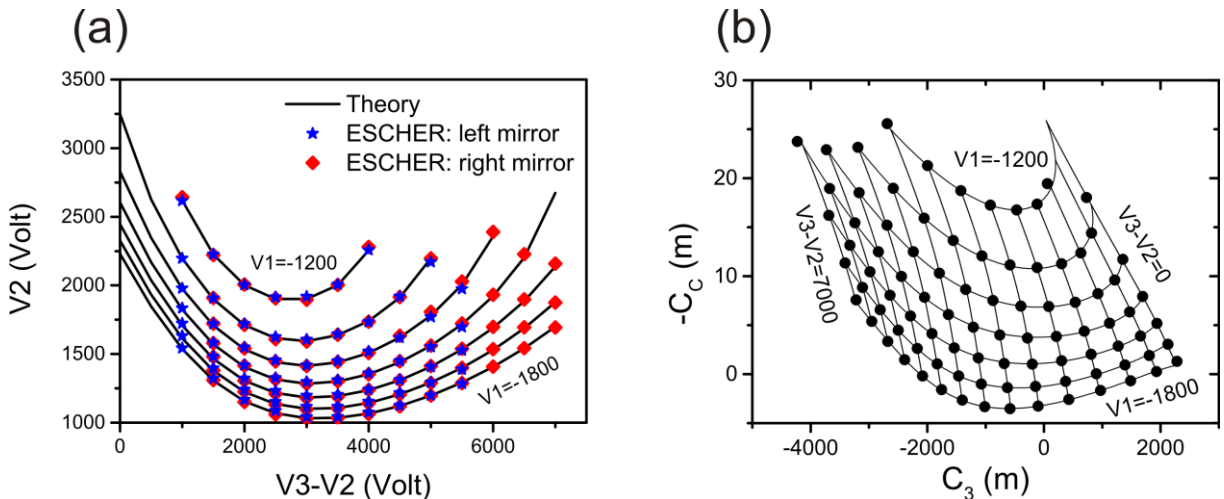
Door formule (2.11) om te schrijven naar functie van  $\alpha_2$  en  $M$  en dit te substitueren in formule (2.16) volgt voor het verband tussen  $C_{c, spiegel}$  en  $C_{c, lens}$ :

$$C_{c, spiegel} = M^2 \cdot C_{c, lens} \quad (2.17)$$

Tussen de chromatische aberraties van de spiegel en lens zit dus een factor  $M^2$ . Eén factor  $M$  voor de vergroting van de afbeelding en één factor  $M$  voor de afname van  $\alpha_2$  ten opzichte van  $\alpha_1$ . Op dezelfde manier kan ook het verband tussen  $C_{3, spiegel}$  en  $C_{3, lens}$  gegeven worden:

$$\begin{aligned} C_{3, spiegel} \alpha_2^3 &= M \cdot C_{3, lens} \alpha_1^3 \\ C_{3, spiegel} &= M \cdot C_{3, lens} \left( \frac{\alpha_1}{\alpha_2} \right)^3 \\ C_{3, spiegel} &= M^4 C_{3, lens} \end{aligned} \quad (2.18)$$

Tussen de sferische aberraties van de spiegel en de lens zit dus een factor  $M^4$ . Eén factor  $M$  voor de vergroting van de afbeelding en dus voor de vergroting van de aberraties en één factor  $M^3$  voor de afname van  $\alpha_2^3$  ten opzichte van  $\alpha_1^3$ .



Figuur 2-8: (a) Theoretische spanningscurves van  $V_{32}$ ,  $V_2$  en  $V_1$  op de spiegel voor in focus conditie. (b)  $C_{3, spiegel}$  en  $C_{c, spiegel}$  als functie van de voltages  $V_1$  en  $V_{32}$ . [1]



### 3. Lage-energie elektronen diffractie

Waar een LEEM in de reële ruimte een resolutie kan halen van 1,4 nanometer kan LEED informatie over de atomaire positie geven omdat een LEED patroon aan het reciproke rooster van een preparaat gekoppeld kan worden. Om diffractie te laten optreden aan een kristaloppervlak moet de golflengte van de elektronen maximaal in dezelfde orde van grootte zijn als die van de roosterconstante van dit oppervlak. Voor atomen en moleculen in een kristalstructuur is deze roosterconstante in de orde van grootte van enkele Ångströms. Voor niet-relativistische elektronen geldt:

$$\begin{aligned}\lambda &= \frac{h}{p} \\ p &= m_e \cdot v = \sqrt{2m_e E} \\ E &= \frac{h^2}{2 \cdot \lambda^2 \cdot m_e}\end{aligned}\tag{3.1}$$

Waarin:

|           |                                                    |       |
|-----------|----------------------------------------------------|-------|
| $\lambda$ | = Golflengte van het elektron                      | [m]   |
| $h$       | = Constante van Plank ( $6,626 \cdot 10^{-34}$ )   | [J·s] |
| $p$       | = Impuls                                           | [N·s] |
| $v$       | = Snelheid van het elektron                        | [m/s] |
| $m_e$     | = Massa van het elektron ( $9,11 \cdot 10^{-31}$ ) | [kg]  |
| $E$       | = Energie van het elektron                         | [J]   |

Voor het bestuderen van oppervlakken zijn elektronen met een energie van enkele tientallen elektronvolt zeer geschikt. Om een diffractiepatroon te kunnen beschrijven is het van belang om te weten wat de link is tussen het kristalrooster en het diffractiepatroon. Om deze link te kunnen maken zal eerst de kristalstructuur beschreven worden.

#### 3.1 Kristalstructuren

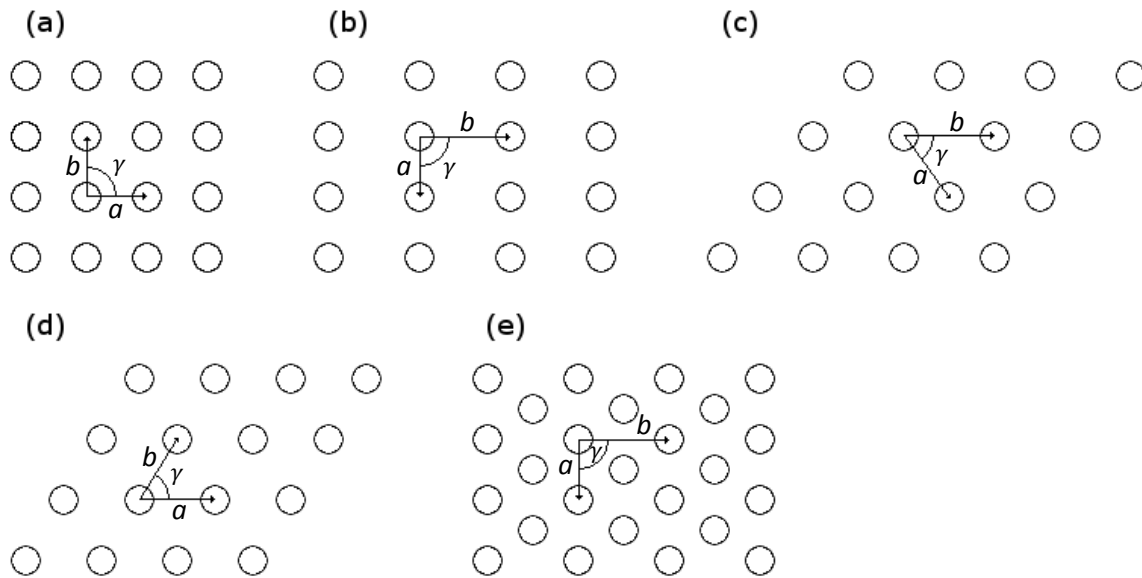
In een vaste stof willen de atomen in de toestand met de minste energie verkeren. Om in deze toestand te kunnen verkeren moeten alle atomen een indentieke omgeving hebben, met als resultaat een periodieke rangschikking. Deze rangschikking wordt de kristalstructuur genoemd. Een tweedimensionale kristalstructuur kan beschreven worden middels een Bravaisrooster met twee vectoren  $a$  en  $b$ . Elk punt in de structuur kan dan bereikt worden middels een roostervector in de vorm [8]:

$$r_n = n_1 a + n_2 b\tag{3.2}$$

Waarin  $n_1$  en  $n_2$  gehele getallen zijn. De regels voor het definiëren van de roostervectoren zijn dat  $b$  de langere van de twee is en dat  $b$  ten opzichte van  $a$  tegen de klok is ingedraaid. Afhankelijk van de lengte van de vectoren  $a$  en  $b$  en de hoek tussen de twee kunnen er verschillende eenheidscellen gevormd worden. Er zijn vier eenheidscellen die primitief worden genoemd. Deze eenheidscellen bevatten steeds maar één atoom. De primitieve eenheidscellen zijn de vierkante, rechthoekige, schuine en hexagonale. De vijfde en laatste eenheidscel bevat 2 atomen en wordt de gecentreerde rechthoekige eenheidscel genoemd. In Tabel 3-1 zijn de eigenschappen van de verschillende eenheidscellen weergegeven, in Figuur 3-1 zijn de verschillende eenheidscellen grafisch weergegeven. [8]

Tabel 3-1: De vijf verschillende basisvectoren voor kristalsymmetrieën in twee dimensies. [8]

| Basis vectoren | Hoeken                 | Eenheidscel            |
|----------------|------------------------|------------------------|
| $a = b$        | $\gamma = 90^\circ$    | Vierkant               |
| $a \neq b$     | $\gamma = 90^\circ$    | Rechthoek              |
| $a \neq b$     | $\gamma \neq 90^\circ$ | Schuine                |
| $a = b$        | $\gamma = 60^\circ$    | Hexagonaal             |
| $a \neq b$     | $\gamma = 90^\circ$    | Gecentreerde rechthoek |



Figuur 3-1: Tweedimensionale kristalstructuren. (a) Vierkante eenheidscel met  $a = b$  en  $\gamma = 90^\circ$ . (b) Rechthoekige eenheidscel met  $a \neq b$  en  $\gamma = 90^\circ$ . (c) Schuine eenheidscel met  $a \neq b$  en  $\gamma \neq 90^\circ$ . (d) Hexagonale eenheidscel met  $a = b$  en  $\gamma = 60^\circ$ . (e) Gecentreerde rechthoekige eenheidscel met  $a \neq b$  en  $\gamma = 90^\circ$ .

### 3.2 Het reciproke rooster

Een diffractiepatroon is geen directe weergave van de daadwerkelijke rangschikking van de atomen aan een oppervlakteroster. Uit het Bloch theorema volgt dat het diffractiepatroon van een periodiek kristal een Bravais rooster is dat uit het kristalrooster berekend kan worden middels de volgende [8] regels:

$$\vec{a} \cdot \vec{b}^* = 0 \quad (3.3)$$

$$\vec{b} \cdot \vec{a}^* = 0 \quad (3.4)$$

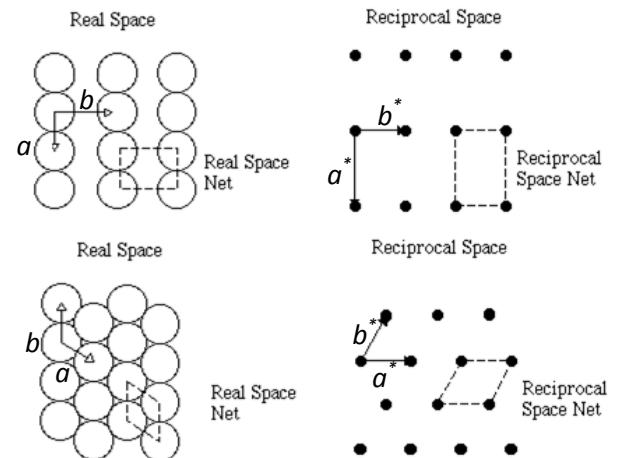
$$\vec{a} \cdot \vec{a}^* = 2\pi$$

$$\vec{b} \cdot \vec{b}^* = 2\pi$$

Waarin  $a^*$  en  $b^*$  de roostervectoren zijn van het reciproke rooster. Uit formule (3.3) volgt dat de roostervectoren  $a$  en  $b^*$  en  $b$  en  $a^*$  loodrecht op elkaar staan. En uit formule (3.4) volgt dat voor de lengte tussen  $a$  en  $a^*$  en  $b$  en  $b^*$  een inverse relatie bestaat. De reciproke roostervectoren worden dan gegeven door:

$$\vec{a}^* = 2\pi \frac{\vec{b} \times \vec{n}}{|\vec{a} \times \vec{b}|} \quad (3.5)$$

$$\vec{b}^* = 2\pi \frac{\vec{n} \times \vec{a}}{|\vec{a} \times \vec{b}|}$$



Figuur 3-2: Voorbeeld van positionering van de roostervectoren in de reële en reciproke ruimte. [11]

Waarin  $n$  de normaalvector is. In Figuur 3-2 is het verband tussen de reële ruimte en de reciproke ruimte weergegeven.

### 3.3 Invalshoek van een diffractiebundel

Voor een invallend elektron met een golfvector  $k_0 = 2\pi/\lambda_0$  en een verstrooid elektron met een golfvector  $k = 2\pi/\lambda$  wordt de diffractievoorwaarde gegeven door de Laue vergelijkingen [8]. Deze stellen dat:

$$\bar{k} - \bar{k}_0 = \bar{G}_{hkl} \quad (3.6)$$

Waarin de indices  $h, k$  en  $l$  gehele getallen zijn en  $G_{hkl}$  een reciproke roostervector is volgens:

$$\bar{G}_{hkl} = h\bar{a}^* + k\bar{b}^* + l\bar{c}^* \quad (3.7)$$

Aangezien de indringdiepte in het kristal maar enkele Ångströms bedraagt, is er geen sprake van een diffractie voorwaarde in de richting loodrecht op het preparaat. Daardoor vereenvoudigen formule (3.6) en (3.7) samen tot:

$$\bar{k}_{\parallel} - \bar{k}_{\parallel,0} = h\bar{a}^* + k\bar{b}^* \quad (3.8)$$

waarin  $k_{\parallel}$  en  $k_{\parallel,0}$  staan voor de component parallel aan het preparaat voor respectievelijk de gereflecteerde en de inkomende golfvector. In een LEED experiment is  $k_{\parallel,0} = 0$  door loodrechte inval op het preparaat. Doordat de elektronen elastisch botsen met het oppervlak van de kristalstructuur blijft de totale impuls van de elektronen behouden. Voor de loodrechte component van de golfvector geldt dan:

$$k_{\perp} = \sqrt{k_0^2 - k_{\parallel}^2} \quad (3.9)$$

De diffractiehoek aan het preparaat is dan:

$$\tan(\alpha) \approx \alpha = \frac{k_{\parallel}}{k_{\perp}} \quad (3.10)$$

Doordat de elektronen ten gevolge van het elektrische veld worden versneld richting de lens neemt de loodrechte component van de golfvector toe, zodat wanneer de elektronen bij de lens komen voor de hoek geldt:

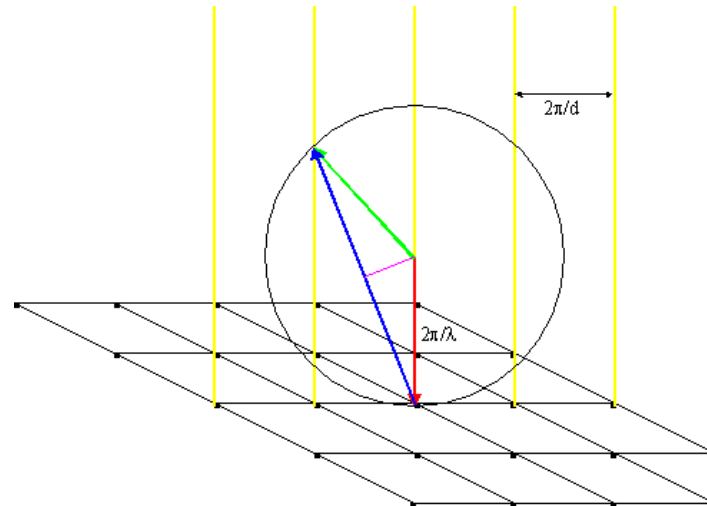
$$\tan(\alpha) \approx \alpha = \frac{k_{\parallel}}{k_{\perp} + k_{veld}} \quad (3.11)$$

Waarin  $k_{veld}$  de toename in de golfvector ten gevolge van het elektrische veld is.

### 3.4 De Ewald cirkel constructie

Om het punt waar het diffractiepatroon stopt te voorspellen wordt er gebruik gemaakt van de Ewald cirkel constructie, zoals in Figuur 3-3 weergegeven. Hierin wordt het diffractiepatroon voorgesteld als een verzameling van oneindig dunne en oneindig lange staven, de gele lijnen in Figuur 3-3. De inkomende golfvector richting het diffractie patroon is de rode pijl in Figuur 3-3. Om het beginpunt van de golfvector wordt een cirkel met straal  $2\pi/\lambda_0$  getekend, zijnde de lengte van de golfvector. Vanaf het beginpunt van de inkomende golfvector wordt naar een punt op de cirkel, waar de staven

van het diffractie patroon de cirkel kruisen, de gereflecteerde golfvector getekend, de groene pijl in Figuur 3-3. De reciproke kristalvector wordt dan getekend vanaf het eindpunt van beide golfvectoren, de blauwe pijl in Figuur 3-3. Aangezien de straal van de Ewald cirkel recht evenredig is met de energie van het inkomende elektron zijn er meer spots waar te nemen wanneer de inkomende elektronen een hoge energie hebben. [9]

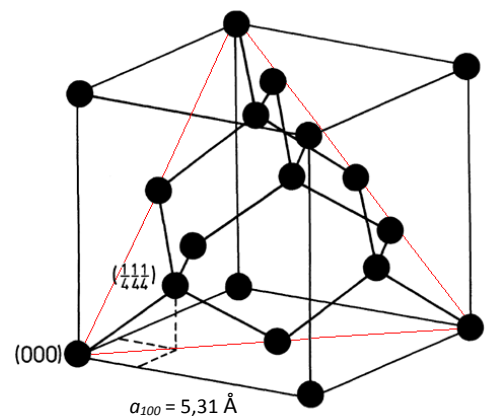


Figuur 3-3: De Ewald cirkel constructie. De rode pijl staat voor de inkomende golfvector, de groene pijl voor de gereflecteerde golfvector en de blauwe pijl voor de reciproke kristalvector. De gele lijnen representeren het diffractie patroon die bij de Ewald cirkel constructie als staven worden gezien [9]

### 3.5 Voorbeeld: Het Si(111) oppervlak

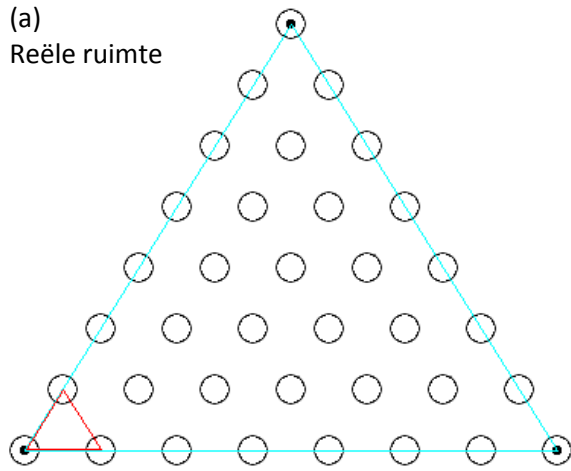
Het kristalrooster van Silicium heeft de zogenaamde diamantstructuur. De diamantstructuur bestaat uit twee face centered cubic (fcc) structuren welke in elkaar zitten. De tweede structuur is ten opzichte van de eerste in alle richtingen  $\frac{1}{4}$  verschoven. De oppervlaktestructuur van het silicium 111-vlak wordt weergegeven door het rode vlak in Figuur 3-4. Uit deze figuur is af te leiden dat silicium in het 111-vlak een hexagonale oppervlaktecel heeft met een roosterconstante van  $a_{111} = a_{100}/\sqrt{2}$ .

Doordat de periodieke structuur van het siliciumkristal aan het oppervlak wordt doorbroken hebben de atomen aan het oppervlak een andere periodieke rangschikking om in de laagste energie toestand te verkeren. Deze rangschikking wordt de 7x7 reconstructie van het siliciumoppervlak genoemd, omdat zich een superstructuur vormt waarbij de roosterconstante van het oppervlak verzevenvoudigd wordt, zoals in Figuur 3-5 weergegeven. Vertaald naar de reciproke ruimte komt dit neer op een reciproke roosterconstante van  $\frac{1}{7}a_{111}^*$ . In het diffractiepatroon ontstaan hierdoor spots op  $1/7$ ,  $2/7$  enzovoorts van de eerste orde spot. De hoeken die de diffractiebundels maken met de optische as van de lens zijn te bepalen met de formules beschreven in §3.3. Bij een versnelling van -15kV kan berekend worden dat de  $1/7$  orde spot een hoek van 4,3 mrad en de eerste orde spot een hoek van 30 mrad maakt met de optische as.

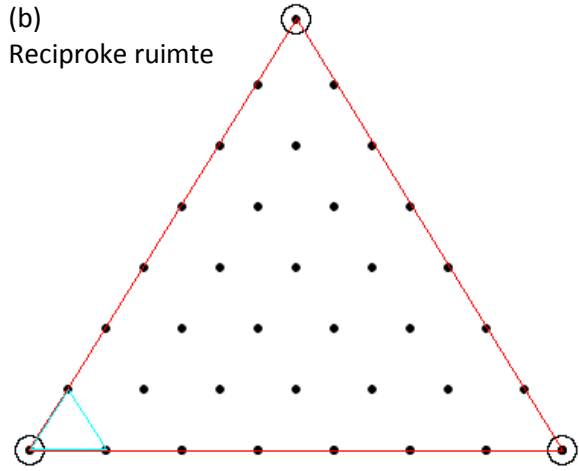


Figuur 3-4: Kristalstructuur van silicium in drie dimensies, ook bekend als de diamant structuur. De structuur bestaat uit twee fcc structuren welke ten opzichte van elkaar  $\frac{1}{4}$  in elke richting zijn verschoven. De rode lijnen geven aan waar het 111-vlak zich bevindt. [8]

(a)  
Reële ruimte



(b)  
Reciproke ruimte



Figuur 3-5: 7x7 reconstructie van het Si(111) oppervlak in de reële ruimte (a) en in de reciproke ruimte (b). De rode driehoeken representeren de eenheidscel met roosterconstante  $a_{111}$  en de licht blauwe driehoeken representeren de eenheidscel van het gereconstrueerde oppervlak met roosterconstante  $7a_{111}$ .



## 4. Aberraties meten en corrigeren

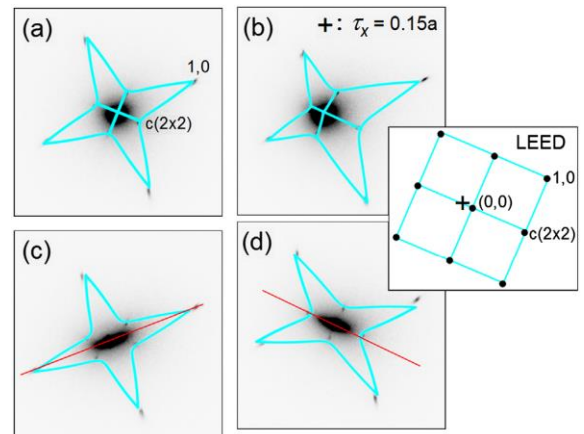
Sferische en chromatische aberraties bij een LEEM worden gecorrigeerd met de elektronenspiegel en worden voor het grootste deel veroorzaakt door de objectieflens. De andere optische elementen hebben een beduidend geringere bijdragen in deze aberraties. De aberraties zijn gecorrigeerd wanneer de spiegel aberraties van dezelfde grootte maar met omgekeerd teken introduceert. Tilt en astigmatisme worden daarentegen niet gecorrigeerd met de elektronen spiegel. Tilt wordt afhankelijk van de soort tilt, bundel of preparaat tilt, gecorrigeerd door een XY-stuurspoel of door de preparaathouder, en dus het preparaat zelf, onder een hoek te kantelen. Astigmatisme wordt gecorrigeerd door gebruik te maken van een magnetische quadrupool.

### 4.1 Tilt en astigmatisme

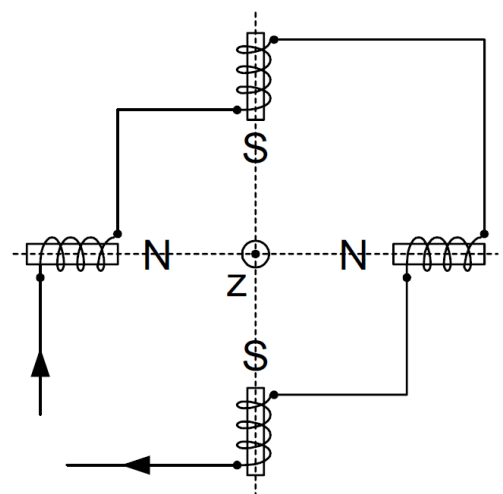
Tilt kan gemeten worden door te kijken naar de symmetrie in een LEED patroon. In Figuur 4-1(b) is het effect te zien van positieve tilt in de x-richting in combinatie met hogere orde sferische aberraties. Door het middelpunt te nemen van alle spots en dit te vergelijken met de positie van de nulde orde spot kan bepaald worden of er tilt aanwezig is. Wanneer het middelpunt van alle spots niet op dezelfde plaats uit komt als de nulde orde spot is er sprake van tilt. Aangezien de totale tilt een combinatie is van bundel- en preparaat tilt is het verwijderen hiervan een iteratief proces.

Astigmatisme kan net als tilt gemeten worden aan de hand van de symmetrie van LEED patronen. In Figuur 4-1(c) en (d) zijn voorbeelden te zien van astigmatisme in een LEED patroon. Deze patronen zijn vergeleken met het patroon in Figuur 4-1(a) langs de rode lijn uitgerekt. Voor het bepalen van astigmatisme wordt er gekeken naar spots waarvan de hoek door de lens dezelfde grote heeft, maar waarvan de x- en y-component verschillende tekens hebben. Wanneer de afstanden tussen spots die tegenover elkaar liggen met elkaar worden vergeleken en deze niet gelijk zijn aan elkaar is er sprake van astigmatisme. [6]

Zoals beschreven in §2.2.2 is de oorzaak van astigmatisme dat elektronen, die door de x- en y-as van een lens gaan, met een verschillende sterkte worden afgebogen. Met behulp van een magnetische quadrupool is hiervoor te compenseren. Een quadrupool bestaat uit vier korte spoelen die ten opzichte van elkaar 90° zijn gedraaid. De stroom door de spoelen heeft een zodanige richting dat de noord- en zuidpolen van de spoelen tegenover elkaar staan, zoals in Figuur 4-2. Door het magneetveld dat de quadrupool opwekt is het mogelijk om gecontroleerd astigmatisme op te laten treden. Door dit in tegenovergestelde richting te doen van



Figuur 4-1: Diffractie patronen van een Cu(001)-C(2x2) preparaat met hogere orde sferische aberraties en verschillende vormen van tilt en astigmatisme. (a) Geen tilt en geen astigmatisme. (b) Tilt in de x richting, geen astigmatisme. (c) Geen tilt, astigmatisme langs de rode lijn (45°). (d) Geen tilt, astigmatisme langs de rode lijn (0°). Rechts schematisch ideaal LEED patroon. De blauwe lijnen in het LEED patroon komen overeen met die in (a)-(d). Het + teken geeft de verplaatsing van de nulde orde spot ten gevolge van de tilt in (b) aan. [6]



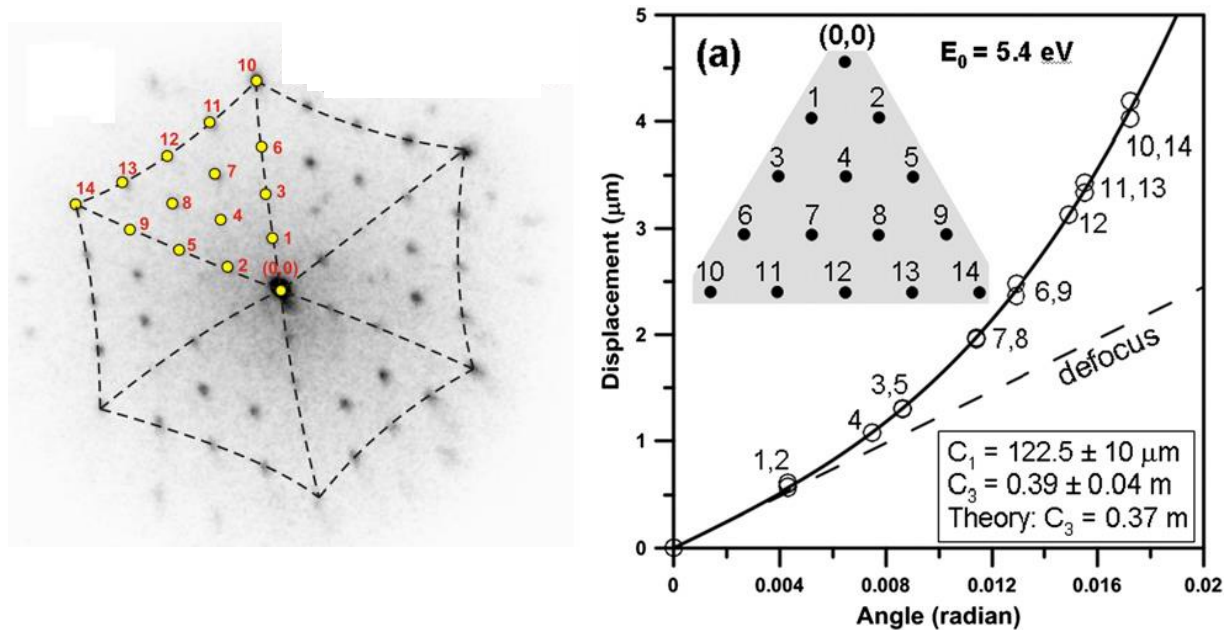
Figuur 4-2: Een magnetische quadrupool die wordt gebruikt voor het verwijderen van astigmatisme. [4]

de andere optiek is het mogelijk om een afbeelding te krijgen die vrij is van astigmatisme. Omdat in de praktijk niet van te voren is te voorspellen in welke richting er astigmatisme optreedt wordt er nog een tweede quadrupool toegevoegd welke 45° is gedraaid ten opzichte van de eerste quadrupool. Door deze tweede quadrupool is het mogelijk om astigmatisme in elke richting op te wekken en dus te compenseren. [4]

#### 4.2 Defocus en sferische aberraties

Voor het meten van de aanwezige defocus en sferische aberraties wordt er vanuit gegaan dat tilt en astigmatisme al zijn verwijderd en wordt er gebruik gemaakt van een preparaat met een rijk diffractie patroon. Zo'n preparaat is Si(111)-(7x7), een voorbeeld van dit patroon is te zien in Figuur 4-3(a).

De hoeken die de diffractiebundels met de optische as maken zijn bekend uit de literatuur [8]. Door de afwijking ten opzichte van de nulde orde spot te plotten als functie van de diffractiehoek, oftewel door een fit door de data te maken gelijk aan de som van de formules (2.4) en (2.9), kunnen de defocus en hogere orde sferische aberraties hieruit geëxtraheerd worden. Voor het corrigeren van de vijfde en hogere orde sferische aberraties is een voltage op de spiegel nodig met een stabiliteit op ppb niveau. Met de huidige technologie is deze stabiliteit niet te halen waardoor in de praktijk de vijfde en hogere orde sferische aberraties niet gecorrigeerd kunnen worden. De fit die dan door de data getrokken wordt is gelijk aan de functie  $r_s = C_1\alpha + C_3\alpha^3$ . In Figuur 4-3(b) is zo'n fit te zien. In Figuur 2-8(b) zijn vervolgens de voltages op te zoeken die naar de spiegel gestuurd moeten worden. [1, 3]



Figuur 4-3: (a) Diffractiepatroon van een Si(111)-(7x7) preparaat. [1] (b) Fit door de afwijking van de nulde orde spot als functie van de diffractiehoek. [3]

#### 4.3 Chromatische aberraties

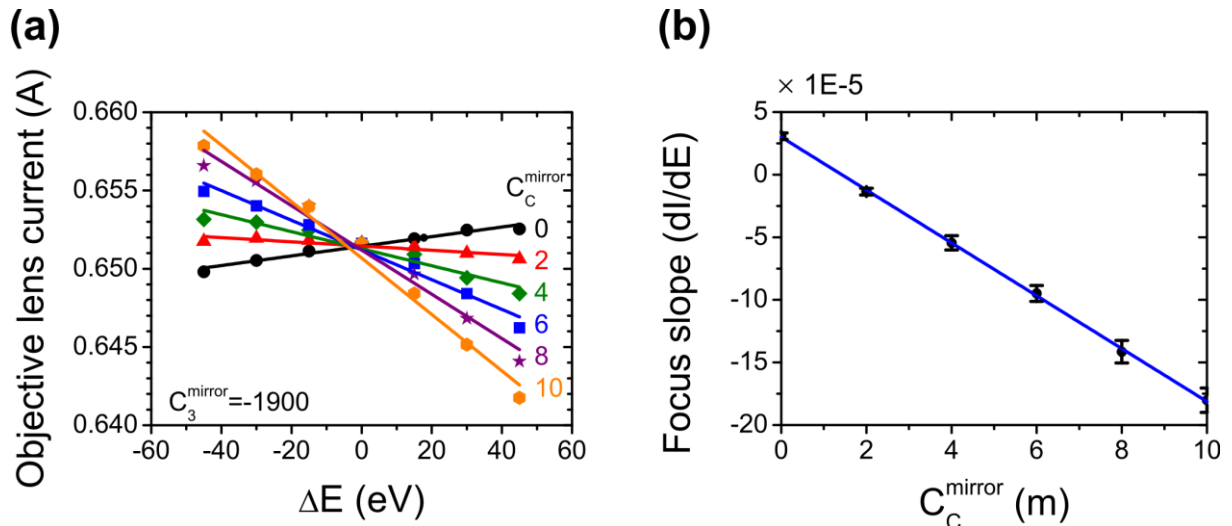
Voor het meten van de chromatische aberraties van de objectieflens is het belangrijk om te weten dat deze bestaat uit twee delen, te weten een startenergie afhankelijke deel ten gevolge van het elektrostatische veld tussen objectieflens en preparaat ( $C_{ce}$ ), en een startenergie onafhankelijk deel



ten gevolge van het magnetische veld in de objectieflens ( $C_{cm}$ ). De totale chromatische aberratie wordt dan gegeven door:

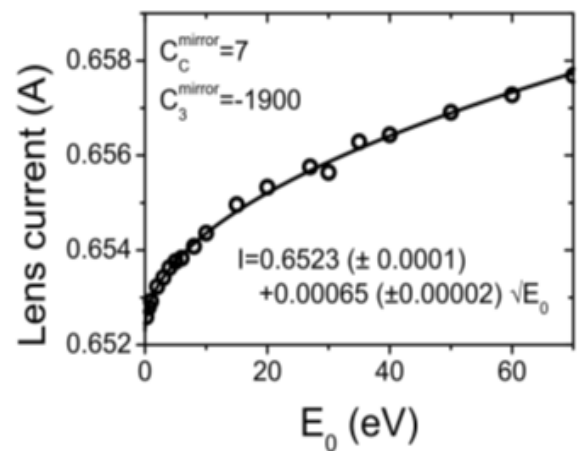
$$C_{C,totaal} = C_{Ce} + C_{Cm} + \frac{C_{C,spiegel}}{M^2} \quad (4.1)$$

Voor het bepalen van  $C_{C,totaal}$  is het nodig om  $C_{Ce}$  en  $C_{Cm} + C_{C,spiegel} / M^2$  apart van elkaar te bepalen. Doormiddel van een PEEM experiment kan  $C_{Cm} + C_{C,spiegel} / M^2$  worden bepaald. Hierbij wordt het voltage, waarop het preparaat wordt gehouden, gevarieerd ( $E$ ) en de energie waarmee de elektronen het preparaat verlaten ( $E_0$ ) gelijk gehouden. De fotonen maken immers middels het foto-elektrisch effect elektronen met een bepaalde energie vrij. In Figuur 4-5 zijn metingen te zien van de stroomsterkte door de objectieflens om de afbeelding in focus te houden wanneer  $E$  gevarieerd wordt. De metingen zijn herhaald voor verschillende waarden van  $C_{C,spiegel}$  van 0 tot 10 m in stappen van 2 m. De helling van de curves in Figuur 4-5(a) zijn in Figuur 4-5(b) geplot als functie van  $C_{C,spiegel}$  met een fit er door heen in de vorm van  $S = S_0 - aC_{C,spiegel}$ . Deze functie is recht evenredig met  $C_{Cm} + C_{C,spiegel} / M^2$ . [1]



Figuur 4-5: (a) Stroomsterkte door de objectieflens voor infocus condities als functie van de preparaatspanning voor verschillende waarden van  $C_{C,spiegel}$ . Bij  $\Delta E = 0$  is de preparaatspanning gelijk aan 15kV. (b) Helling van curves in (a) als functie van  $C_{C,spiegel}$ . [1]

Doormiddel van een LEEM experiment kan  $C_{Ce}$  worden bepaald. Hierbij wordt de kanonpotentiaal ( $G$ ) en dus  $G = E + E_0$  constant gehouden en wordt het voltage op het preparaat gevarieerd waardoor de energie waarmee de elektronen het preparaat verlaten ook varieert. In Figuur 4-4 zijn metingen te zien van de stroomsterkte door de objectieflens om de afbeelding in focus te houden wanneer  $E_0$  gevarieerd wordt. De lijn in Figuur 4-4 is een fit in de vorm  $I = I_0 + c\sqrt{E_0}$ . De helling van deze curve,  $dI/dE_0 = c / 2\sqrt{E_0}$ , is recht evenredig met  $C_{Ce}$ . De totale chromatische aberraties zijn gecorrigeerd wanneer de uitkomst van formule (4.1) gelijk is aan nul en dus ook wanneer:



Figuur 4-4: Stroomsterkte door de objectieflens voor in focus condities als functie van elektron energie bij verlaten van het preparaat. [1]

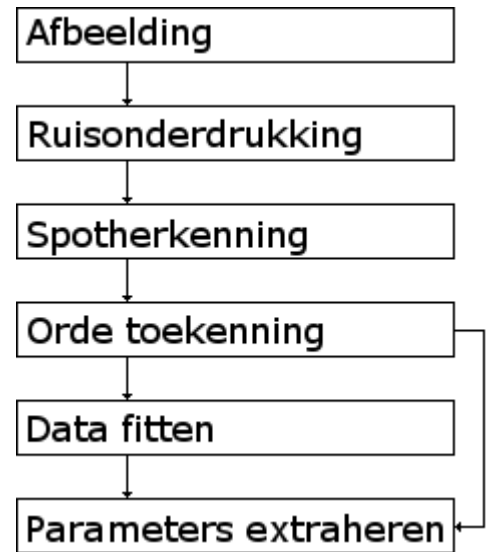
$$\frac{c}{2\sqrt{E_0}} + S_0 - aC_{c,spiegel} = 0 \quad (4.2)$$

Door  $C_{c,spiegel}$  te isoleren uit formule (4.2) wordt de waarde gevonden waarvoor de spiegel de chromatische aberraties van de objectieflens compenseert. Via formule (2.18) is vervolgens  $C_{c,lens}$  te bepalen en kunnen in Figuur 2-8(b) de voltages, die naar de spiegel gestuurd moeten worden, opgezocht worden. [1]

## 5. Automatische aberratiecorrectie

Voor het implementeren van de automatische aberratiecorrectie is gebruik gemaakt van de programmeertaal Python. De keuze voor Python is gemaakt omdat de al aanwezige software voor de LEEM in deze taal is geschreven en integratie van het aberratie correctieprogramma hierdoor makkelijker te realiseren is. Python is een programmeertaal die begin jaren 90 is ontworpen door Guido van Rossum. De taal heeft tegenwoordig een grote community van ontwikkelaars met als gevolg dat veel bibliotheken die geschreven worden gratis ter beschikking worden gesteld aan anderen.

Het programma voor de aberratiecorrectie kan opgedeeld worden in vijf onderdelen. Deze onderdelen zijn weergegeven in Figuur 5-1: Ruisonderdrukking, Spotherkenning, Orde toekenning, Data fitten en Parameters extraheren. Het onderdeel ruisonderdrukking filtert de ruis in de afbeelding eruit waardoor de spotherkenning van het continue probleem, een discreet probleem kan maken door alle spots een coördinaat in de afbeelding te geven. Bij de ordetoekenning wordt aan elke gedetecteerde spot een hoek toegekend. Vervolgens wordt in het onderdeel Data fitten een derde orde polynoom gefit zoals beschreven in §4.2. Het laatste onderdeel, Parameters extraheren, haalt uit de herkende spots het astigmatisme en de tilt uit het diffractie patroon. Uit de fit die is verkregen uit het onderdeel data fitten worden de aanwezige defocus en sferische aberraties verkregen.



Figuur 5-1: Programmastructuur diagram van het aberratie correctie programma

### 5.1 Gebruikte algoritmen

In elk onderdeel van het programma worden er verschillende algoritmen gebruikt voor het verkrijgen van het resultaat. Hier zullen de stappen die zijn gezet voor het verkrijgen van het gewenste resultaat één voor één beschreven worden. De geprogrammeerde code is te vinden in Bijlage A.

#### 5.1.1 Ruisonderdrukking

In een verkregen signaal zal altijd ruis aanwezig zijn. Wanneer er een schatting gemaakt kan worden van de aard van de ruis is het mogelijk om deze via een Wiener deconvolutie eruit te filteren.

Veronderstel een systeem gegeven door:

$$y(t) = h(t) * x(t) + n(t) \quad (5.1)$$

Waarin \* convolutie aangeeft en:

- $y(t)$  = Verkregen signaal
- $h(t)$  = Overdrachtsfunctie van het systeem
- $x(t)$  = Ingangssignaal
- $n(t)$  = Ruis signaal onafhankelijk van ingangssignaal

Gewenst is een functie  $g(t)$  te vinden die een schatting kan maken van  $x(t)$  volgens:

$$\hat{x}(t) = g(t) * y(t) \quad (5.2)$$

Waarin  $\hat{x}(t)$  een schatting is van  $x(t)$ . De Wiener deconvolutie is het makkelijkst uit te voeren in het frequentiedomein zodat voor (5.2) geldt:

$$\hat{X}(f) = G(f)Y(f) \quad (5.3)$$

Waarin:

$\hat{X}(f)$  = De Fourier getransformeerde van  $\hat{x}(t)$

$G(f)$  = De Fourier getransformeerde van  $g(t)$

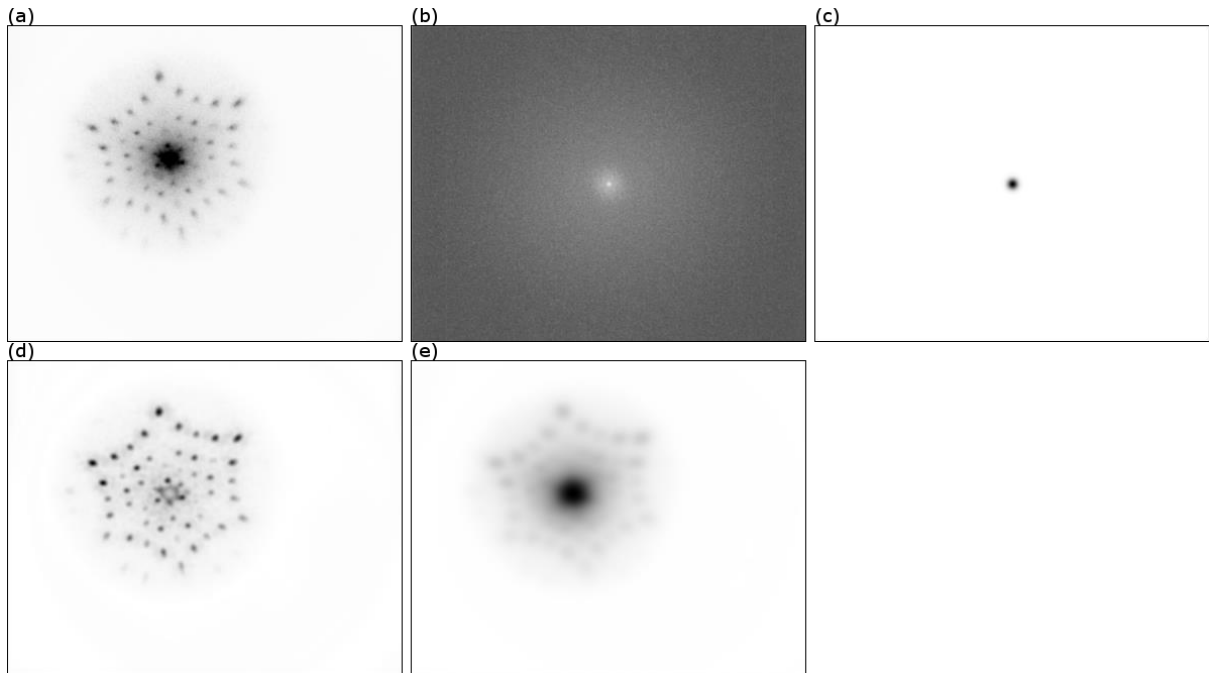
$Y(f)$  = De Fourier getransformeerde van  $y(t)$

Wanneer ongeveer bekend is in welk frequentiegebied het ruissignaal aanwezig is moet het filter  $G(f)$  zo gemaakt worden dat deze frequenties geblokkeerd worden. Figuur 5-2(a) is een voorbeeld van een LEED patroon waar de ruis uitgefilterd dient te worden. Te zien is dat de ruis ten opzichte van het signaal een lage frequentie heeft; de ruis neemt geleidelijke toe richting het midden. De lage frequenties worden middels een gaussisch filter onderdrukt. Dit filter wordt gegeven door:

$$G(f) = \exp\left(-\frac{x^2+y^2}{\sigma^2}\right) \quad (5.4)$$

waarin  $x$  en  $y$  de afstanden zijn tot het middelpunt van de afbeelding voor respectievelijk de  $x$ - en  $y$ -richting en  $\sigma$  gegeven wordt door:

$$\sigma = \frac{FWTM}{2\sqrt{2} \ln 10} \quad (5.5)$$

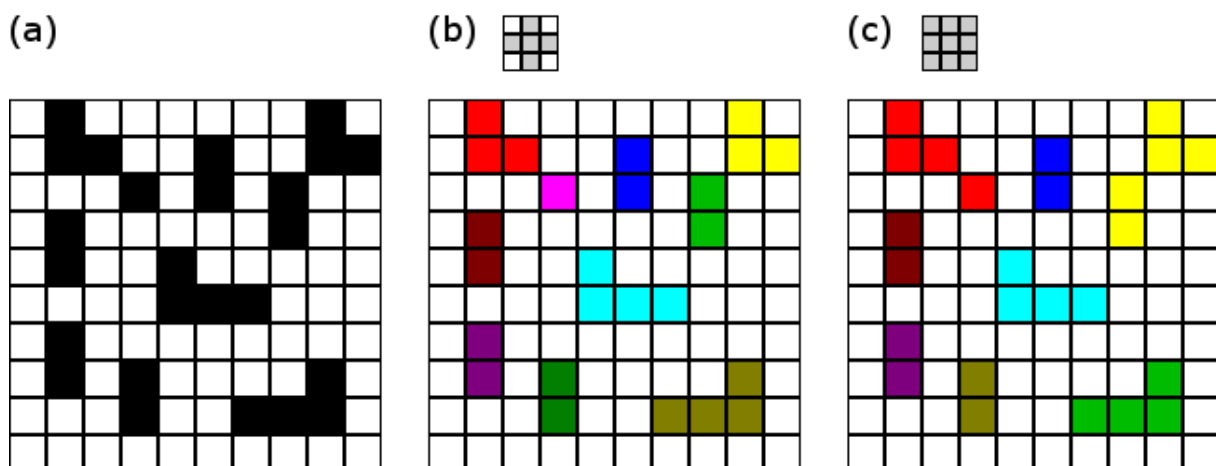


Figuur 5-2: Ruisonderdrukking aan de hand van Wiener deconvolutie. (a) Oorspronkelijke afbeelding. (b) Fourier getransformeerde van de oorspronkelijke afbeelding. (c) Toegepast filter op de Fourier getransformeerde. (d) Terug getransformeerde Fourier afbeelding bij onderdrukking van de lage frequenties. (e) Terug getransformeerde Fourier afbeelding bij onderdrukking van de hoge frequenties met het geïnverteerde filter van (c).

Hierin is *FWTM* de volledige breedte op een tiende van het maximum. Na experimenteel de *FWTM* te variëren en het resultaat te beoordelen is gebleken dat een waarde van 70 het beste resultaat gaf. In Figuur 5-2(c) is het zojuist beschreven filter weergegeven. Figuur 5-2(d) laat het resultaat zien van de teruggetransformeerde Fourier afbeelding bij van het onderdrukken van de lage frequenties met het beschreven filter. Figuur 5-2(e) laat het resultaat zien van de teruggetransformeerde Fourier afbeelding bij het onderdrukken van de hoge frequenties met het geïnverteerde filter.

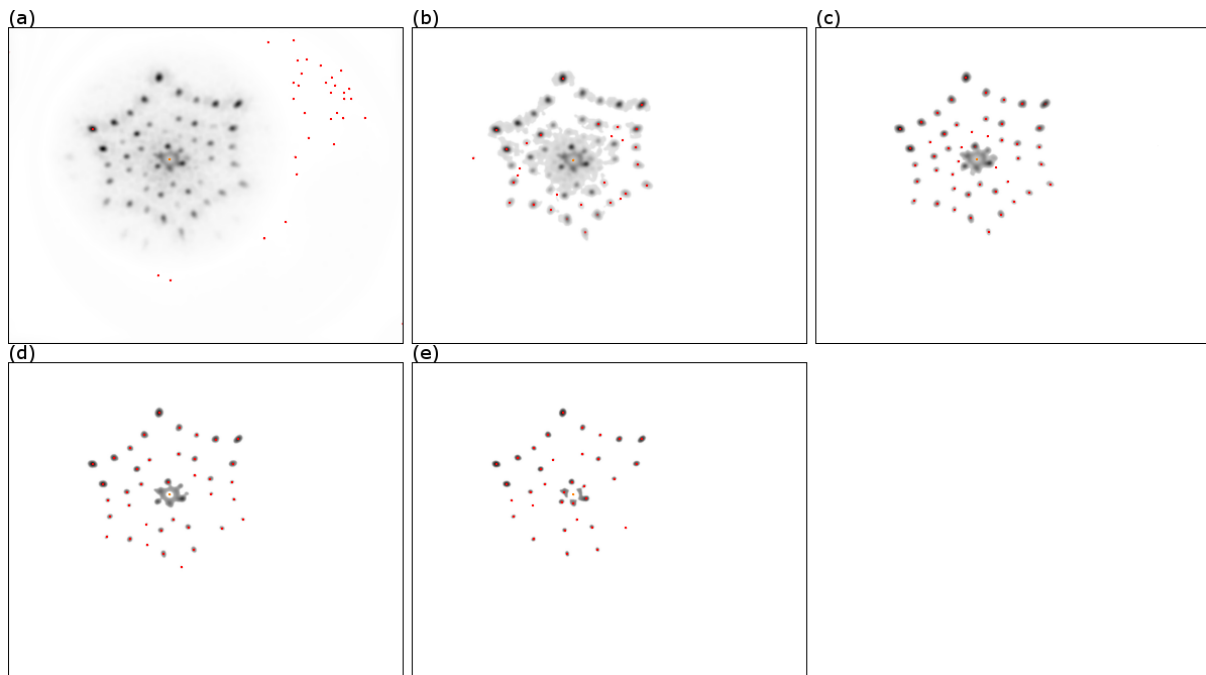
### 5.1.2 Spotherkenning

Voor het detecteren van de spots worden aaneengesloten gebieden met waarden groter dan 0 gecreëerd. Dit wordt gedaan door een 3x3 patroon over de afbeelding te schuiven. Aan het 3x3 patroon kunnen verschillende indelingen gegeven worden. In Figuur 5-3(b) en (c) zijn twee verschillende indelingen weergegeven. In (b) is de indeling van het 3x3 patroon een kruis en bij (c) is het een vierkant. Het 3x3 patroon wordt gecentreerd over elke pixel geschoven. Wanneer een waarde groter dan 0 in de indeling van het 3x3 patroon valt wordt deze pixel tot het zelfde gebied gerekend als waar het 3x3 patroon over gecentreerd is. Zoals te zien in Figuur 5-3(b) en (c) is het mogelijk om aan de hand van de indeling van het 3x3 patroon van dezelfde afbeelding verschillende gebieden te onderscheiden. Voor de spotdetectie is het 3x3 patroon met kruisindeling de beste optie, omdat gebieden dan sneller van elkaar worden onderscheiden.



Figuur 5-3: Gebiedsherkenning door het schuiven van een 3x3 patroon over de afbeelding. (a) Originele afbeelding. (b) Afbeelding met herkende gebieden door een 3x3 patroon met kruis indeling over de afbeelding te schuiven. (c) Afbeelding met herkende gebieden door een 3x3 patroon met vierkante indeling over de afbeelding te schuiven. De verschillende kleuren in (b) en (c) representeren verschillende herkende gebieden.

Wanneer de afbeelding is opgedeeld in verschillende gebieden, door het 3x3 patroon over de afbeelding te schuiven, is in elk gebied de pixel met de hoogste intensiteit als locatie van de spot aangewezen. Wanneer direct na het filteren van de ruis een 3x3 patroon over de afbeelding geschoven wordt en de maximale intensiteit per herkend gebied als spot wordt gemarkeerd, worden de herkende spots weergegeven zoals in Figuur 5-4(a). Hierin is te zien dat de spots niet goed herkend zijn. Dit komt doordat het hele LEED patroon een grotere pixel waarde heeft dan 0 met als gevolg dat dit als 1 gebied wordt gezien. Verder zijn er buiten het LEED patroon nog overblijfselen van de ruis welke ook als een gebied met een spot worden gezien. Om deze methode te laten werken is het nodig om een drempelwaarde in te stellen. Alle pixels die onder de drempelwaarde zitten worden dan een nul. In Figuur 5-4(b)-(e) wordt de drempelwaarde in stappen van 25 vergroot. Aan het resultaat is te zien dat de spots voor dit LEED patroon bij een drempelwaarde van 50 het beste worden herkend.



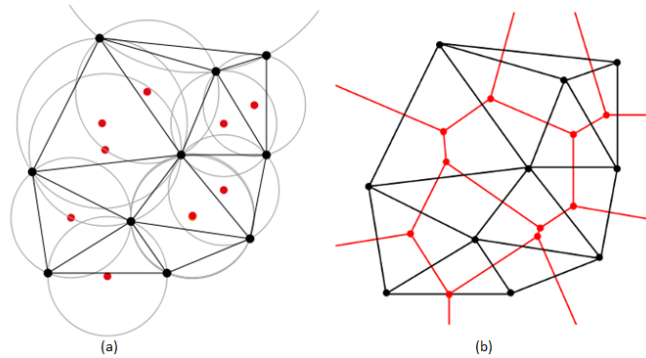
Figuur 5-4: Werking van het spotdetectie algoritme. (a) Afbeelding zoals die uit de ruisonderdrukking komt. (b) Afbeelding met een ingestelde drempelwaarde van 25. (c) Afbeelding met een ingestelde drempelwaarde van 50. (d) Afbeelding met een ingestelde drempelwaarde van 75. (e) Afbeelding met een ingestelde drempelwaarde van 100.

Voor de detectie van de nulde orde spot wordt er gebruik gemaakt van de teruggetransformeerde Fouries afbeelding waaruit de hoge frequenties zijn weg gefilterd, Figuur 5-2(e). Hierbij is het, net als bij de detectie van de hogere orde spots, nodig om een drempelwaarde in te stellen om de overgebleven ruis weg te filteren. In dit geval geeft een constante drempelwaarde van 20 een betrouwbaar resultaat.

Voor het controleren van de werking van de spotherkenning is er op nog 14 andere beschikbare LEED patronen een test gedaan. Hieruit is gebleken dat de drempelwaarde voor de centrale spot bij een waarde van 20 altijd een goed resultaat geeft. Voor het detecteren van de hogere orde spots is de drempelwaarde afhankelijk van het LEED patroon. De resultaten voor de overige 14 patronen zijn te vinden in Bijlage B.

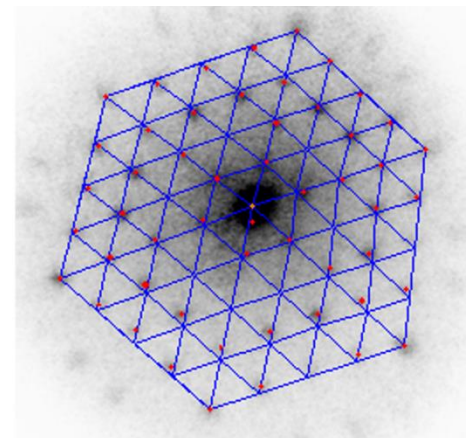
### 5.1.3 Orde toekenning

Voor het toekennen van een orde aan de spots is er gebruik gemaakt van Delaunay-triangulaties en Voronoi-diagrammen. Voor het creëren van de Delaunay-triangulatie van een gegeven reeks punten moet voor elke driehoek de omschreven cirkel geen enkel ander punt uit de reeks bevatten dan de 3 hoekpunten van de driehoek. De middelpunten van de omschreven cirkels zijn dan de punten waar het Voronoi-diagram overheen loopt. Een andere manier voor het vinden van het Voronoi-diagram is om elk punt in een afbeelding aan het dichtstbijzijnde punt van de reeks toe te wijzen. De randen tussen twee cellen in het Voronoi-diagram geven de plek aan waar de punten het dichtst bij twee punten van de reeks zitten. In Figuur 5-5 is een voorbeeld gegeven van een Delaunay-triangulatie en de bijbehorend Voronoi-diagram. [10]



Figuur 5-5: (a) Delaunay-triangulaties met omschreven cirkels. (b) Voronoi-diagram (rode lijnen) en Delaunay-triangulaties (zwarte lijnen) van (a). [10]

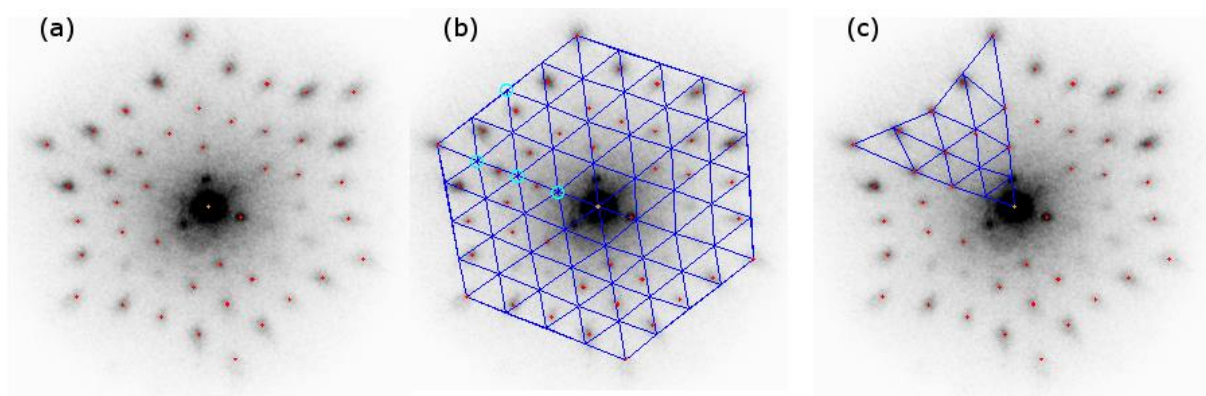
Om Delaunay-triangulatie toe te kunnen passen op het LEED patroon van Si(111) is herkenning van de uiterste hoekpunten in het patroon van cruciaal belang. Dit omdat de hoekpunten de basis zijn voor het voorspellen van de rest van het patroon. In Figuur 5-6(a) is een voorspelling te zien van de locaties aan de hand van de nulde orde spot en de zes uiterste hoekpunten van het patroon. Te zien is dat deze voorspelling in de buurt komt van de daadwerkelijke plek van de spots. Door het gebruik van naaste buurrelaties op de voorspelde spots en de gemeten spots toe te passen moet het mogelijk zijn om de voorspelde spots te vervangen door de gemeten en zo de Delaunay-triangulatie te corrigeren en door de gemeten spots te laten gaan. Als de Delaunay-triangulatie is gecorrigeerd voor de gemeten spots is het mogelijk om voor elke spot via een Hasse-diagram van een partiële ordening de afstand naar een hoekpunt van de ring waartoe de spot behoort te bepalen. Als de ring waartoe de spot behoort en de afstand tot het hoekpunt van die ring bekend is kunnen de waardes voor  $h$  en  $k$  in formule (3.8) ingevuld worden en zo de hoek en dus de orde berekend worden.



Figuur 5-6: Delaunay-triangulaties op het LEED patroon van Si(111).

In Figuur 5-6(a) was een voorspelling te zien van de locatie van de spots voor een bijna volledige gecorrigeerd patroon. In Figuur 5-7(a) is een ongecorrigeerd patroon weergegeven. Figuur 5-7(b) laat het resultaat zien wanneer op dezelfde manier als hierboven de locatie van de spots voorspeld worden. In dit geval is te zien dat de locaties als gevolg van de aberraties niet goed voorspeld zijn. Om toch gebruik te kunnen maken van Delaunay-triangulaties is het nodig om handmatig de omcirkelde punten in Figuur 5-7(b) naar de gewenste spot te verplaatsen. Door de voorspelde spots met zelfde orde aan elkaar te koppelen is het mogelijk om door het verplaatsen van één spot de andere spots van die orde over een zelfde afstand te verschuiven. In Figuur 5-7(c) is een voorbeeld te zien van een deel van de Delaunay-triangulatie wanneer deze handmatig is getekend. Hieruit blijkt dat door alleen de vier omcirkelde punten in Figuur 5-7(b) te verplaatsen, en gelijke ordes mee te verplaatsen, de voorspelling van de locatie van de spots weer een goede benadering is. Door het toepassen van dezelfde naaste buurrelaties en een Hasse-diagram zoals eerder beschreven moet het nu ook mogelijk zijn om voor ongecorrigeerde patronen de hoek en orde te bepalen.

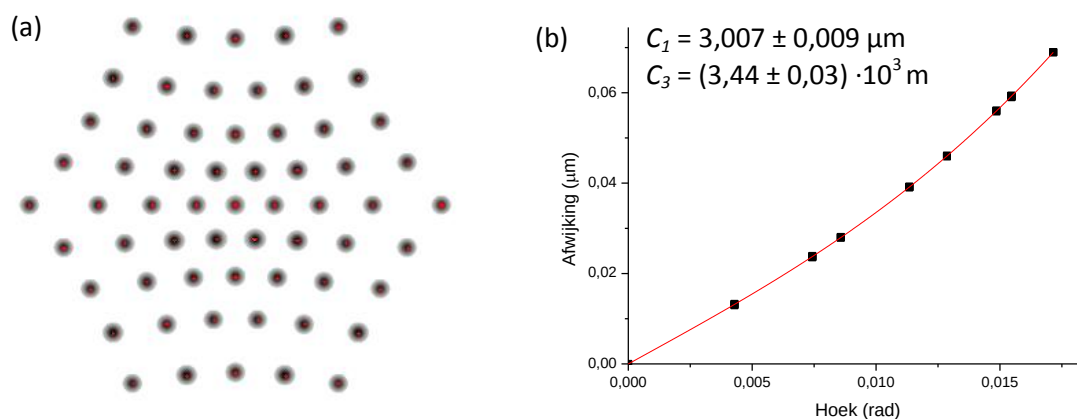
Het implementeren van de ordetoekenning is nog niet gerealiseerd. De hier boven beschreven methode is daarom dus hypothetisch en de werking hiervan kan niet gegarandeerd worden.



Figuur 5-7: Delaunay-triangulaties op het LEED patroon van Si(111). (a) LEED patroon met gemeten spots. (b) LEED patroon met voorspelde Delaunay-triangulaties. (c) LEED patroon met handmatig getekende Delaunay-triangulaties.

#### 5.1.4 Data fitten & parameters extraheren

Doordat er geen automatische orde detectie is gerealiseerd is het niet mogelijk om in Python een fit te maken met de gedetecteerde spots. Om toch aan te kunnen tonen dat deze methode werkt is er door M. Hesselberth een programma geschreven dat een LEED patroon kan genereren met ingestelde waarden voor de defocus en derde orde sferische aberratie. De gegenereerde LEED patronen worden ingeladen in de spotdetectie waarna de gedetecteerde spots geëxporteerd worden naar OriginPro<sup>tm</sup>. In OriginPro<sup>tm</sup> wordt door de gedetecteerde spots een fit gemaakt, zoals beschreven in §4.2, en worden de ingestelde waarden voor de defocus en derde orde sferische aberratie hieruit geëxtraheerd. In Figuur 5-8(a) is een voorbeeld te zien van een gegenereerd LEED patroon met in (b) een fit door de afwijking ten opzichte van de nulde orde spot. In Tabel 5-1 zijn de resultaten te zien voor andere gegenereerde LEED patronen deze patronen zijn te vinden Bijlage C.



Figuur 5-8: (a) Gegenereerd patroon met ingestelde defocus van 3,00  $\mu\text{m}$  en derde orde sferische aberraties van 3400m met toegepaste spotdetectie. (b) Fit van de vorm  $r_s = C_1\alpha + C_3\alpha^3$  door de afwijking van de spots als functie van de hoek met de optische as.

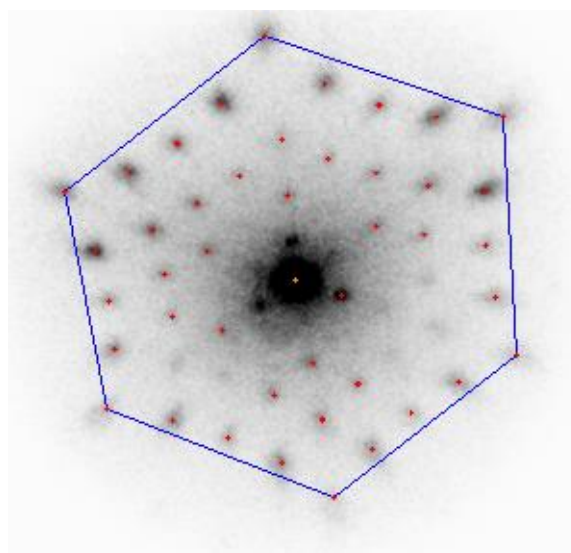


Tabel 5-1: Gemeten en ingestelde waarden voor  $C_1$  en  $C_3$  aan de hand van de genereerde LEED patronen in Bijlage C.

|            | Ingestelde $C_1$ [ $\mu\text{m}$ ] | Gemeten $C_1$ [ $\mu\text{m}$ ] | Ingestelde $C_3$ [m] | Gemeten $C_3$ [m]            |
|------------|------------------------------------|---------------------------------|----------------------|------------------------------|
| Patroon 1  | 3,00                               | $3,007 \pm 0,009$               | $3,400 \cdot 10^3$   | $(3,44 \pm 0,03) \cdot 10^3$ |
| Patroon 2  | 2,75                               | $2,767 \pm 0,007$               | $3,250 \cdot 10^3$   | $(3,28 \pm 0,03) \cdot 10^3$ |
| Patroon 3  | 1,50                               | $1,508 \pm 0,008$               | $5,000 \cdot 10^3$   | $(5,03 \pm 0,04) \cdot 10^3$ |
| Patroon 4  | 4,50                               | $4,513 \pm 0,009$               | $4,150 \cdot 10^3$   | $(4,18 \pm 0,04) \cdot 10^3$ |
| Patroon 5  | 3,00                               | $3,012 \pm 0,007$               | $0,250 \cdot 10^3$   | $(0,23 \pm 0,03) \cdot 10^3$ |
| Patroon 6  | 1,85                               | $1,854 \pm 0,006$               | $1,400 \cdot 10^3$   | $(1,42 \pm 0,03) \cdot 10^3$ |
| Patroon 7  | 3,16                               | $3,167 \pm 0,006$               | $2,130 \cdot 10^3$   | $(2,17 \pm 0,03) \cdot 10^3$ |
| Patroon 8  | 1,26                               | $1,27 \pm 0,01$                 | $1,910 \cdot 10^3$   | $(1,90 \pm 0,05) \cdot 10^3$ |
| Patroon 9  | 2,14                               | $2,143 \pm 0,006$               | $2,570 \cdot 10^3$   | $(2,60 \pm 0,03) \cdot 10^3$ |
| Patroon 10 | 1,56                               | $1,568 \pm 0,006$               | $1,620 \cdot 10^3$   | $(1,62 \pm 0,03) \cdot 10^3$ |

De absolute afwijking in de defocus en derde orde sferische aberraties zijn gemiddeld respectievelijk 8 nanometer en 30 meter. Procentueel is de afwijking gemiddeld 0,30% en 1,31% voor respectievelijk de defocus en derde orde sferische aberraties. Uit deze data is te concluderen dat de spots op correcte wijze worden gedetecteerd en dat de methode om defocus en derde orde sferische aberraties te meten werkt.

Voor het extraheren van de tilt en astigmatisme wordt er gebruik gemaakt van het convex omhulsel. Het convex omhulsel bestaat uit de zes uiterste hoekpunten van het LEED patroon en omvat alle andere spots, zoals weergegeven in Figuur 5-9. Door het middelpunt van het convex omhulsel te berekenen en dit te vergelijken met de afstand tot de nulde orde spot is het mogelijk om te detecteren of er tilt aanwezig is. Voor het bepalen van negatieve of positieve tilt wordt er gekeken of de locatie van het middelpunt van het convex omhulsel links of rechts en boven of onder de nulde orde spot ligt. Er is sprake van positieve tilt in de x- en/of y-richting wanneer het middelpunt respectievelijk rechts of boven de nulde orde spot ligt. Wanneer de spot links of onder de nulde orde spot ligt is er sprake van negatieve tilt in respectievelijk de x- en y-richting.



Figuur 5-9: Convex omhulsel van een LEED patroon

Wanneer patronen overgecorrigeerd zijn bestaat het convex omhulsel uit meer spots van de buitenste ring. Om toch de gewenste bewerking te kunnen doen wordt er een voorwaarde gesteld aan de zijden van het convex omhulsel. Het verschil in de hoek die twee aangrenzende zijden maken met de x-as mag maximaal tien graden bedragen. Wanneer het verschil in deze hoek kleiner is dan tien graden wordt het gedeelte punt verwijderd uit het convex omhulsel.

Voor het extraheren van het astigmatisme worden de afstanden van de tegenover elkaar staande hoekpunten met elkaar vergeleken. Wanneer deze afstanden niet gelijk zijn is er sprake van astigmatisme. De lijn waarover het astigmatisme optreedt is te bepalen aan de hand van de verhoudingen tussen de afstanden.



## 6. Conclusie

Om de aberraties automatisch te corrigeren zijn er LEED patronen automatisch geanalyseerd met de programmeertaal Python. Het programma in de huidige vorm bestaat uit een ruisfilter en een spotdetectie. Het ruisfilter maakt gebruik van Wiener deconvolutie toepast in het frequentie domein om de ruis op het signaal te verwijderen. Voor het detecteren van de spots wordt er over het gefilterde patroon een 3x3 patroon geschoven, om gebieden waar spots zitten aan te wijzen. In deze gebieden wordt de maximale intensiteit als spot aangewezen. Voor het toekennen van een orde aan de spots kan er gebruik gemaakt worden van Delaunay-triangulaties en Hasse-diagrammen, waarbij de Delaunay-triangulaties een voorspelling doen van de locatie van de spots en met het Hasse-diagram de orde wordt toegewezen door te kijken wat de afstand is tot de hoekpunten. Het implementeren van dit deel van het programma is nog niet gerealiseerd waardoor dit een hypothetische methode is en geen garantie over de werking hiervan gegeven kan worden.

Doordat in het geschreven programma nog geen orde toekenning aanwezig is is het niet mogelijk om in Python een fit te maken door de afwijking, ten opzichte van de nulde orde spot, als functie van de diffractiehoek. Om de werking van het tot nu toe geschreven programma te controleren zijn er tien LEED patronen gegenereerd, met ingestelde waardes voor defocus ( $C_1$ ) en derde orde sferische aberraties ( $C_3$ ). Deze LEED patronen zijn geanalyseerd met de geprogrammeerde code waarna de posities van de gedetecteerde spots zijn geëxporteerd naar OriginPro<sup>tm</sup>. In OriginPro<sup>tm</sup> is er een fit door de afwijking als functie van de diffractiehoek gemaakt, waaruit  $C_1$  en  $C_3$  geëxtraheerd kunnen worden. De absolute afwijking in  $C_1$  en  $C_3$  zijn gemiddeld respectievelijk 8 nanometer en 30 meter, wat een gemiddelde procentueel afwijking geeft van 0,30% en 1,31%. Hieruit kan geconcludeerd worden dat de spots op correcte wijze worden gedetecteerd.

## Bibliografie

- [1] S. Schramm, „Imaging with Aberration-Corrected Low Energy Electron Microscopy,” BOXPress, Leiden, 2013.
- [2] R. Tromp, J. Hannon, W. Wan, A. Berghaus en O. Schaff, „A new aberration-corrected, energy-filtered LEEM/PEEM instrument. I. Principles and design,” *Ultramicroscopy*, vol. 110, pp. 852-861, 2010.
- [3] R. Tromp, J. Hannon, W. Wan, A. Berghaus en O. Schaff, „A new aberration-corrected, energy-filtered LEEM/PEEM instrument II. Operation and results,” *Ultramicroscopy*, vol. 127, pp. 25-39, 2013.
- [4] R. F. Egerton, „Physical Principles of Electron Microscopy,” New York, Springer, 2005, pp. 27-55.
- [5] R. Tromp, W. Wan en S. Schramm, „Aberrations of the cathode objective lens up to fifth order,” *Ultramicroscopy*, vol. 119, pp. 33-39, 2012.
- [6] R. Tromp, „Characterization of the cathode objective lens by Real-Space,” *Ultramicroscopy*, vol. 130, pp. 2-6, 2013.
- [7] R. Tromp, „Measuring and correcting aberrations of a cathode objective lens,” *Ultramicroscopy*, vol. 111, pp. 273-281, 2011.
- [8] H. Ibach en H. Lüth, Solid-State Physics, Aachen: Springer, 2009.
- [9] V. Zamlynny, „The Ewald Construction for the surface,” University of Guelph, 18 April 1997. [Online]. Available: <http://www.chembio.uoguelph.ca/educmat/chm729/recipe/9surew.htm>. [Geopend 31 Mei 2014].
- [10] „Delaunay-triangulatie,” Wikimedia Foundation, 28 December 2013. [Online]. Available: <http://nl.wikipedia.org/wiki/Delaunay-triangulatie>. [Geopend 1 Juni 2014].
- [11] S. J. Garrett, „Special Topics in Analytical Chemistry (CEM 924): Lecture Notes,” 2001. [Online]. Available: <http://www.cem.msu.edu/~cem924sg/LectureNotes.html>. [Geopend 7 mei 2014].

## Bijlage A De Python code

Deze code is getest en werkend met de volgende bibliotheekversies: NumPy 1.8.0, SciPy 0.13.3, Matplotlib 1.3.1 en PIL 1.1.7.

```
# -*- coding: utf-8 -*-
"""
Program for automatic aberration correction
Created by Timothy van Steenis as graduation internship at University Leiden
"""
# -*- coding: utf-8 -*-

import Tkinter as Tk
#import time
import image_filter #exern script
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image, ImageTk
from scipy import ndimage
from scipy.spatial import (Delaunay, ConvexHull, Voronoi, voronoi_plot_2d,
delaunay_plot_2d)
import math
import PIL.ImageOps

class Application(Tk.Frame):
    def __init__(self, parent):
        Tk.Frame.__init__(self)
        self.root = parent
        self.pos = []
        self.var_spot_check = Tk.IntVar()
        self.var_spot_check.set(0)
        self.var_image_check = Tk.IntVar()
        self.var_image_check.set(0)
        self.grid()
        self.Canvas_Img()
        self.System_Buttons()
        self.Scrollbar_Threshold()
        self.Spot_Button()
        self.Checkbutton()
        self.Order_Detection()

    def Canvas_Img(self):#draw image on canvas
        self.canvas_image = Tk.Canvas(self, width = 640, height = 512)
        self.original = Image.open("Path_here")
        self.original_resized = self.original.resize((640, 512), Image.ANTIALIAS)
        self.image = ImageTk.PhotoImage(self.original_resized)
        self.canvas_image.create_image(320, 256, image = self.image, tag = "image")
        self.canvas_image.grid(column = 0, row = 0, rowspan = 10)
        array = np.array(self.original)
        self.scale = array.shape[0] / 512
        print "self.scale =", self.scale
        spot_image, center_image = image_filter.FFT(array)
        self.filter_image = Image.fromarray(spot_image)
        labeled_array, num_features = ndimage.measurements.label(center_image,
                                                                    structure = [[0,1,0],[1,1,1],[0,1,0]])
        self.pos_center = ndimage.measurements.maximum_position(center_image,
                                                                    labels = labeled_array, index = np.arange(1,
                                                                    num_features + 1))

    def Scrollbar_Threshold(self):
        def Scrollbar(x):
            self.canvas_image.delete("all")
            array = np.array(self.filter_image)
            bool_ = array > int(x)
            self.ndarray = self.filter_image * bool_
```

```

self.ndarray = self.filter_image * bool_
img_threshold = PIL.ImageOps.invert(Image.fromarray(self.ndarray)
    ).resize((640, 512), Image.ANTIALIAS)
self.image = ImageTk.PhotoImage(img_threshold)
self.canvas_image.create_image(320, 256, image = self.image,
    tag = "image")

y, x = self.pos_center[0]
self.canvas_image.create_oval((x/self.scale)+ 1,(y/self.scale)+ 1,
    (x/self.scale)- 1,(y/self.scale)- 1,
    fill = "DarkOrange1", outline =
    "DarkOrange1", tag = "center")
self.scrollbar = Tk.Scale(self, orient = Tk.HORIZONTAL, length = 200,
    command = Scrollbar, to = 255,
    label = "Threshold", tickinterval = 50)
self.scrollbar.grid(column = 2, row = 1, columnspan = 3)

def Confirm():#spot detection
    self.canvas_image.delete("spot")
    labeled_array, num_features = ndimage.measurements.label(self.ndarray,
        structure = [[0,1,0],[1,1,1],[0,1,0]])
    self.pos = ndimage.measurements.maximum_position(self.ndarray,
        labels = labeled_array, index = np.arange(1,
        num_features + 1))
    print self.pos_center
    y_center, x_center = self.pos_center[0]
    for n in xrange(len(self.pos)):
        y, x = self.pos[n]
        print "x=", x, "y=", y
        self.canvas_image.create_oval((x/self.scale)+ 1,(y/self.scale)+ 1,
            (x/self.scale)- 1,(y/self.scale)- 1,
            fill = "red", outline = "red",
            tag = "spot")

        self.button_remove_spot.configure(state = Tk.ACTIVE)
        self.button_add_spot.configure(state = Tk.ACTIVE)
        self.spot.configure(state = Tk.ACTIVE)
        self.check_image.select()
        self.spot.select()

    print "self.pos_center =", self.pos_center
    print "self.pos =", self.pos
    self.button_confirm_threshold = Tk.Button(self, text = "Confirm",
        fg = "red", command = Confirm)
    self.button_confirm_threshold.grid(column = 1, row = 1)

def Checkbutton(self):
    #checkboxbutton show spots on/off
    def Spot_Switch():
        if self.var_spot_check.get() == 1:
            print "Show spots on"
            try:
                for n in xrange(len(self.pos)):
                    y, x = self.pos[n] #y, x, r, vec_x, vec_y
                    self.canvas_image.create_oval((x/self.scale)+ 1,
                        (y/self.scale)+ 1,(x/self.scale)- 1,(y/self.scale)- 1,
                        fill = "red", outline = "red", tag = "spot")
            except TypeError:
                print "No spots detected please confirm threshold"
                pass
            try:
                y, x = self.pos_center[0]

```

```

        self.canvas_image.create_oval((x/self.scale)+ 1,
        (y/self.scale)+ 1,(x/self.scale)- 1,(y/self.scale)-1,
        fill = "DarkOrange1", outline = "DarkOrange1", tag = "center")
    except IndexError:
        print "No center spot found"
        pass

    else:
        print "Show spots off"
        self.canvas_image.delete("spot")
        self.canvas_image.delete("center")

    self.spot = Tk.Checkbutton(self, command = Spot_Switch, state = Tk.DISABLED,
        variable = self.var_spot_check,
        text = "Show spots on/off")
    self.spot.grid(row = 3, column = 3, sticky = Tk.S)

    #checkboxbutton original/filtered image
    def Change_image():
        if self.var_image_check.get() == 1:
            print "fliter"
            self.canvas_image.delete("all")
            self.canvas_image.create_image(320, 256, image = self.image,
                tag = "image")
            Spot_Switch()
        else:
            print "original"
            self.canvas_image.delete("all")
            self.original_image = ImageTk.PhotoImage(PIL.ImageOps.invert(
                self.original_resized))
            self.canvas_image.create_image(320, 256,
            image = self.original_image, tag = "image")
            Spot_Switch()

    self.check_image = Tk.Checkbutton(self, command = Change_image,
        variable = self.var_image_check,
        text = "Show original/filtered image")
    self.check_image.select()
    self.check_image.grid(row = 3, column = 1, columnspan = 2, sticky = Tk.S)

def Spot_Button(self):
    #Remove spot button
    def Activate_Remove():
        def Remove_Spot(event):
            id_spot = self.canvas_image.find_overlapping(event.x - 5,
                event.y - 5, event.x + 5, event.y + 5)

            try:
                x1, y1, x2, y2 = self.canvas_image.coords(id_spot[1])
                x1 = (x1 * 2) + 2; y1 = (y1 * 2) + 2
                x1 = np rint(x1); y1 = np rint(y1)
                k = [x for x in self.pos].index((y1, x1))
                del self.pos[k]
                self.canvas_image.delete(id_spot[1])
            except IndexError:
                print "None spot found."

            self.canvas_image.unbind("<Button-1>")
            self.canvas_image.bind("<Button-1>", Remove_Spot)
    self.button_remove_spot = Tk.Button(self, text = "Remove Spot", fg = "red",

```

```

        command = Activate_Remove, state = Tk.DISABLED)
self.button_remove_spot.grid(column = 1, row = 2)

#Add spot button
def Activate_Add():
    def Add_Spot(event):
        self.pos += ((event.y * 2, event.x * 2),)
        self.canvas_image.create_oval(event.x + 1, event.y + 1, event.x - 1,
                                       event.y - 1, fill = "red",
                                       outline = "red", tag = "spot")
        self.canvas_image.unbind("<Button-1>")
        self.canvas_image.bind("<Button-1>", Add_Spot)
    self.button_add_spot = Tk.Button(self, text = "Add Spot", fg = "red",
                                     command = Activate_Add,
                                     state = Tk.DISABLED)
    self.button_add_spot.grid(column = 2, row = 2)

def Activate_Info():
    def Print_Info(event):
        id_spot = self.canvas_image.find_overlapping(event.x - 5,
                                                       event.y - 5, event.x + 5, event.y + 5)
        x1, y1, x2, y2 = self.canvas_image.coords(id_spot[1])
        x1 = (x1 * 2) + 2; y1 = (y1 * 2) + 2
        x1 = np rint(x1); y1 = np rint(y1)
        y_center, x_center = self.pos_center[0]
        r = np.sqrt((x1 - x_center)**2 + (y1 - y_center)**2)
        vec_x = x1 - x_center
        vec_y = y1 - y_center
        print "\nx =", x1, "& y =", y1
        print "r =", r
        print "vec_x =", vec_x, "& vec_y =", vec_y
        print "id spot", id_spot
        self.canvas_image.unbind("<Button-1>")

        self.canvas_image.bind("<Button-1>", Print_Info)
    self.button_info_spot = Tk.Button(self, text = "spot info", fg = "red",
                                       command = Activate_Info)
    self.button_info_spot.grid(column = 3, row = 2)

def System_Buttons(self):
    #quit button
    self.button_quit = Tk.Button(self, text = "Quit", fg = "red",
                                  command = self.quit)
    self.button_quit.grid(column = 1, row = 0, sticky = Tk.S)

    #save button
    def Save():
        print "kaas"
        #mahotas.imshow('Plaatjes_verslag/test.bmp', self.ndarray)
    self.button_save = Tk.Button(self, text = "Save", fg = "red",
                                  command = Save)
    self.button_save.grid(column = 3, row = 0, sticky = Tk.S)

    #reset button
    def Reset():
        print "Coords have been reset"
        self.canvas_image.delete("line")
        self.canvas_image.delete("spot")
        self.pos = []
        self.pos_center = []

```



```

        print "Canvas has been cleared"
    self.button_reset = Tk.Button(self, text = "Reset", fg = "red",
                                  command = Reset)
    self.button_reset.grid(column = 2, row = 0, sticky = Tk.S)

def Order_Detection(self):
    def Order():
        spots = np.zeros((len(self.pos)+1, 2))
        for i in xrange(len(self.pos)):
            y, x = self.pos[i]
            spots[i,] = x, y
        y_center, x_center = self.pos_center[0]
        spots[i+1,] = x_center, y_center

        hull = ConvexHull(spots)
        hull_point = hull.vertices
        hull_state = True
        while(hull_state):
            for i in xrange(len(hull_point)):
                x1, y1 = spots[hull_point[-2+i],]
                x2, y2 = spots[hull_point[-1+i],]
                x3, y3 = spots[hull_point[0+i],]
                print math.atan((y1-y2)/(x1-x2)) - math.atan((y2-y3)/(x2-x3))
                if (-0.18 < (math.atan((y1-y2)/(x1-x2)) -
                             math.atan((y2-y3)/(x2-x3))) < 0.18):
                    hull_point = np.delete(hull_point, -1+i)
                    break
            if i == len(hull_point)-1:
                hull_state = False

        print "hull_point =", hull_point

        print "hull.simplices\n", hull.simplices

        print "hull.neighbors\n", hull.neighbors

        virtual_spot = np.zeros((len(hull_point)+1, 2))

        for i in xrange(len(hull_point)):
            virtual_spot[i,] = spots[hull_point[i]]

        y_center, x_center = self.pos_center[0]
        virtual_spot[i+1,] = x_center, y_center

        tri = Delaunay(virtual_spot)

        plt.subplot(1,3,1),plt.triplot(virtual_spot[:,0], virtual_spot[:,1],
                                       tri.simplices.copy())
        plt.plot(virtual_spot[:,0], virtual_spot[:,1], 'o')

    '''ABERATION FUNC'''

    def tri_line(a, b, vir_spot):#add new virtual spots
        a_1 = vir_spot[a][0]; a_2 = vir_spot[a][1]
        b_1 = vir_spot[b][0]; b_2 = vir_spot[b][1]
        c = np.array([np rint((a_1 + b_1) / 2), np rint((a_2 + b_2) / 2)])
        vir_spot = np.vstack((vir_spot, c))
        return vir_spot

    for j in xrange(2):

```

```

self.canvas_image.delete("Delaunay")
tri_index = []
for i in xrange(len(tri.simplices)):
    if (((tri.simplices[i][0], tri.simplices[i][1]) not in
        tri_index) and ((tri.simplices[i][1],
            tri.simplices[i][0]) not in tri_index)):
        tri_index.append((tri.simplices[i][0],
            tri.simplices[i][1]))
        virtual_spot = tri_line(tri.simplices[i][0],
            tri.simplices[i][1], virtual_spot)

    if (((tri.simplices[i][1], tri.simplices[i][2]) not in
        tri_index) and ((tri.simplices[i][2],
            tri.simplices[i][1]) not in tri_index)):
        tri_index.append((tri.simplices[i][1],
            tri.simplices[i][2]))
        virtual_spot = tri_line(tri.simplices[i][1],
            tri.simplices[i][2], virtual_spot)

    if (((tri.simplices[i][0], tri.simplices[i][2]) not in
        tri_index) and ((tri.simplices[i][2],
            tri.simplices[i][0]) not in tri_index)):
        tri_index.append((tri.simplices[i][0], tri.simplices[i][2]))
        virtual_spot = tri_line(tri.simplices[i][0],
            tri.simplices[i][2], virtual_spot)

tri = Delaunay(virtual_spot)
plt.subplot(1,3,2+j),plt.triplot(virtual_spot[:,0],
                                virtual_spot[:,1], tri.simplices.copy())
plt.plot(virtual_spot[:,0], virtual_spot[:,1], 'o')

def draw_tri(a, b, vir_spot):#draw delaunay triangulation on canvas
    a_1 = vir_spot[a][0]; a_2 = vir_spot[a][1]
    b_1 = vir_spot[b][0]; b_2 = vir_spot[b][1]
    self.canvas_image.create_line(a_1/self.scale, a_2/self.scale,
                                b_1/self.scale, b_2/self.scale,
                                fill = "Blue", tags = "Delaunay")

for i in xrange(len(tri.simplices)):#loop through delaunay tri for drawi
    if (((tri.simplices[i][0], tri.simplices[i][1]) not in tri_index)
        and ((tri.simplices[i][1], tri.simplices[i][0]) not in
            tri_index)):
        tri_index.append((tri.simplices[i][0], tri.simplices[i][1]))
        draw_tri(tri.simplices[i][0], tri.simplices[i][1],
            virtual_spot)

    if (((tri.simplices[i][1], tri.simplices[i][2]) not in tri_index)
        and ((tri.simplices[i][2], tri.simplices[i][1]) not in
            tri_index)):
        tri_index.append((tri.simplices[i][1], tri.simplices[i][2]))
        draw_tri(tri.simplices[i][1], tri.simplices[i][2],
            virtual_spot)

    if (((tri.simplices[i][0], tri.simplices[i][2]) not in tri_index)
        and ((tri.simplices[i][2], tri.simplices[i][0]) not in
            tri_index)):
        tri_index.append((tri.simplices[i][0], tri.simplices[i][2]))
        draw_tri(tri.simplices[i][0], tri.simplices[i][2],
            virtual_spot)
vor = Voronoi(virtual_spot)
voronoi_plot_2d(vor)
delaunay_plot_2d(tri, ax=None)
plt.show()

self.start_order_detection = Tk.Button(self, text = "Order Detection",
                                       fg = "red", command = Order)
self.start_order_detection.grid(column = 1, row = 4, sticky = Tk.S)

root = Tk.Tk()
app = Application(root)
app.master.title("Spot detection")
root.mainloop()
root.destroy()

```

Extern ingeladen script voor de Fourier transformatie.

```
import numpy as np
import math
from scipy import ndimage

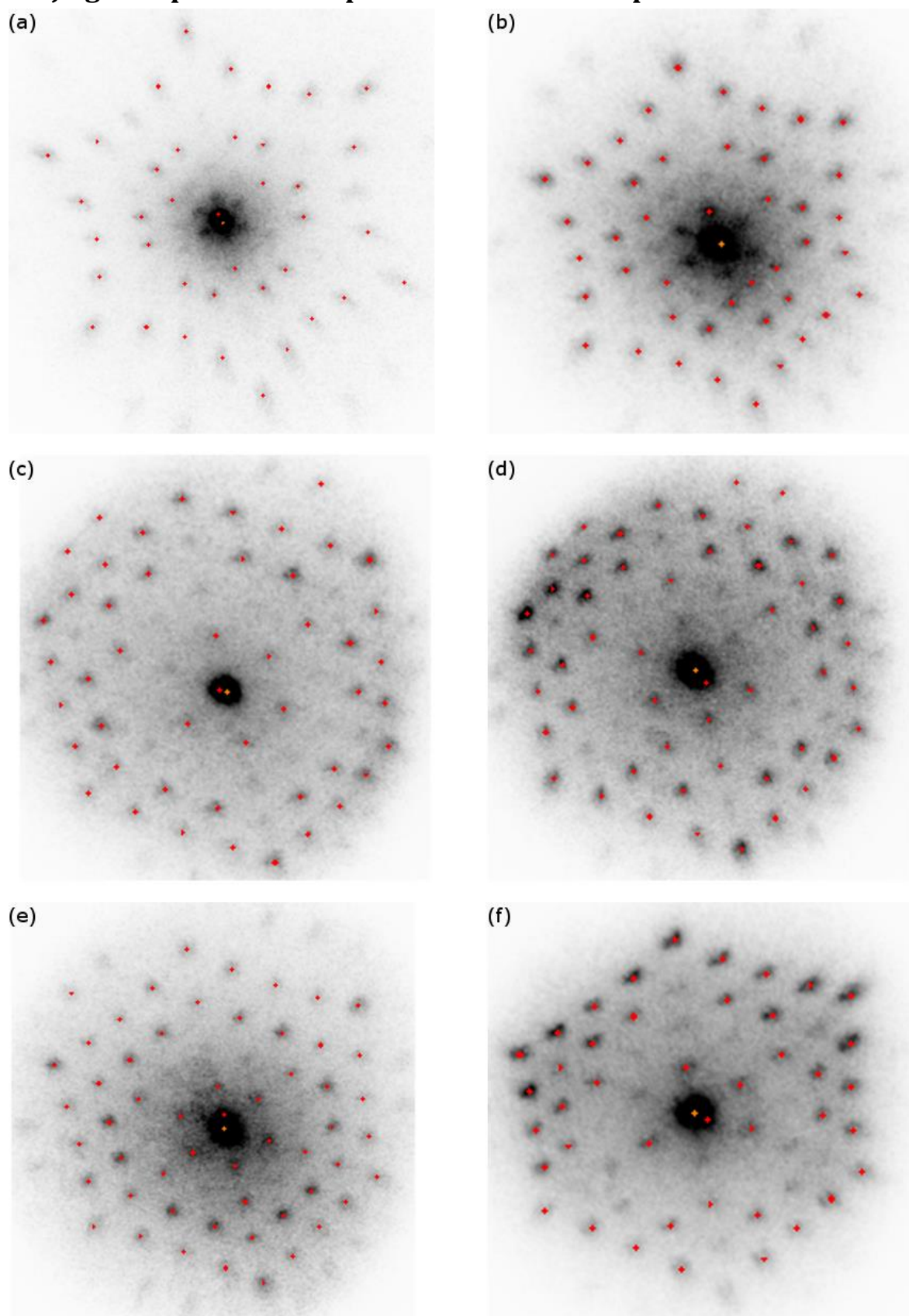
def FFT(array):
    f = np.fft.fft2(array)
    fshift = np.fft.fftshift(f)
    rows, cols = array.shape
    crow,ccol = rows / 2 , cols / 2
    x, y = np.mgrid[-crow:crow, -ccol:ccol]

    FWTM = 70
    c = FWTM / (2 * np.sqrt(2 * math.log1p(10)))
    g = np.exp(-(x**2+y**2)/(c**2))
    z = 1 - g

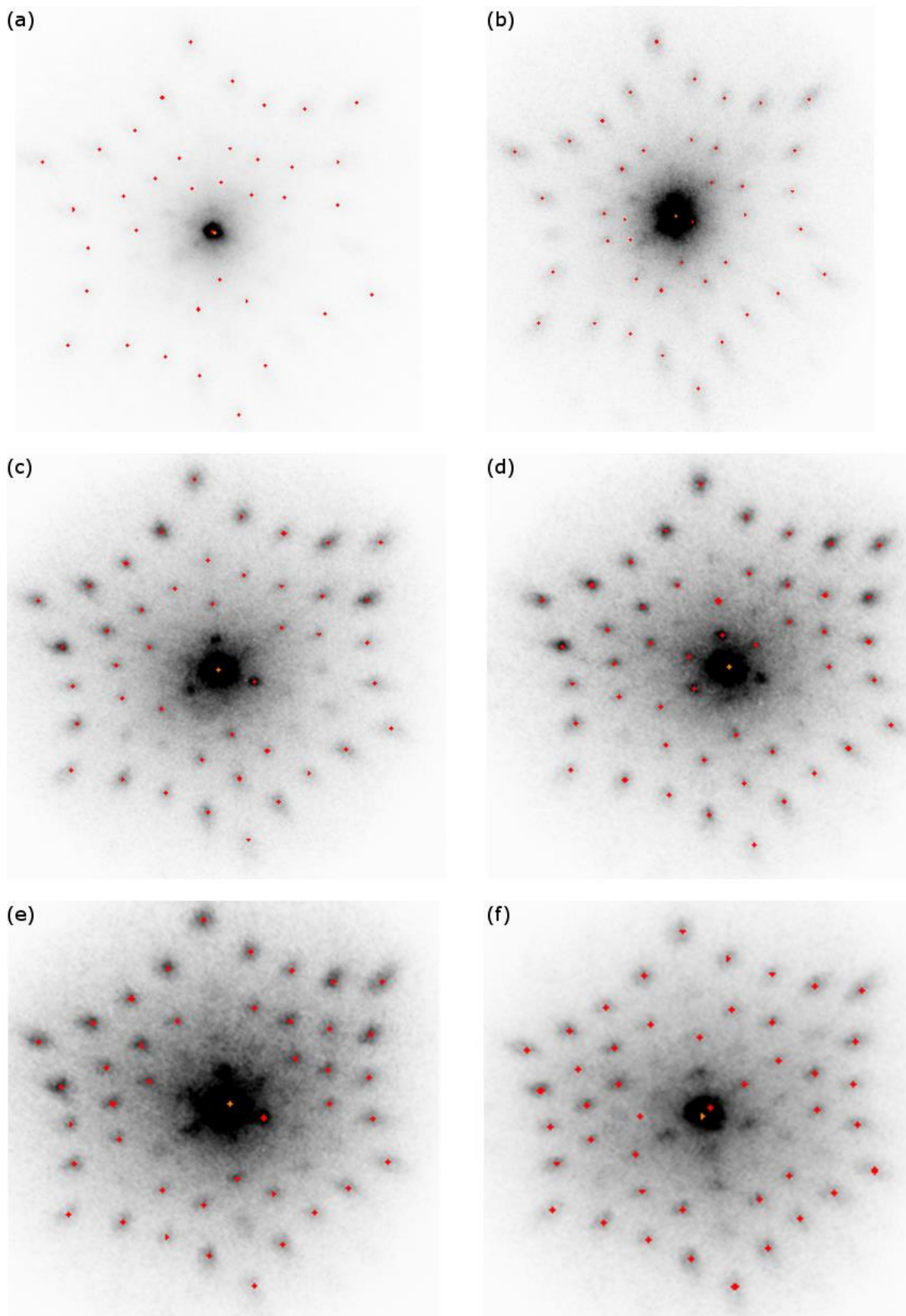
    decon = fshift * z
    img_decon = np.fft.ifft2(decon)
    img_decon = np.abs(img_decon)
    spot_image = ndimage.gaussian_filter(img_decon, 5)
    #convert to int
    spot_int = np rint((spot_image - np.min(spot_image)) * (255/(np.max(spot_image)
        - np.min(spot_image)))) .astype(np.uint8)

    #centerspot
    con = fshift * g
    img_con = np.fft.ifft2(con)
    img_con = np.abs(img_con)
    #convert to int
    center_int = np rint((img_con - np.min(img_con)) * (255/(np.max(img_con)
        - np.min(img_con)))) .astype(np.uint8)
    bool_ = center_int > 10
    center_spot= center_int * bool_
    return spot_int, center_spot
```

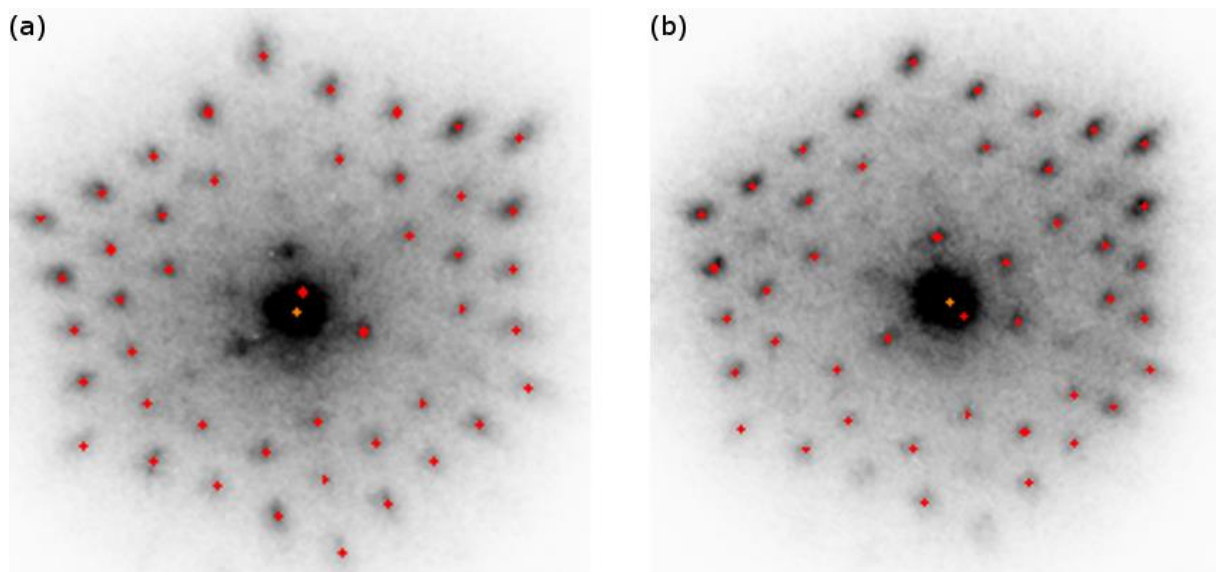
## Bijlage B Spotdetectie op verschillende LEED patronen



Figuur B-1: LEED patronen van het Si(111)-(7x7) oppervlak met toegepaste spot detectie. (a) Ingestelde drempelwaarde van 53 (b) Ingestelde drempelwaarde van 65. (c) Ingestelde drempelwaarde van 49. (d) Ingestelde drempelwaarde van 49. (e) Ingestelde drempelwaarde van 67. (f) Ingestelde drempelwaarde van 50.



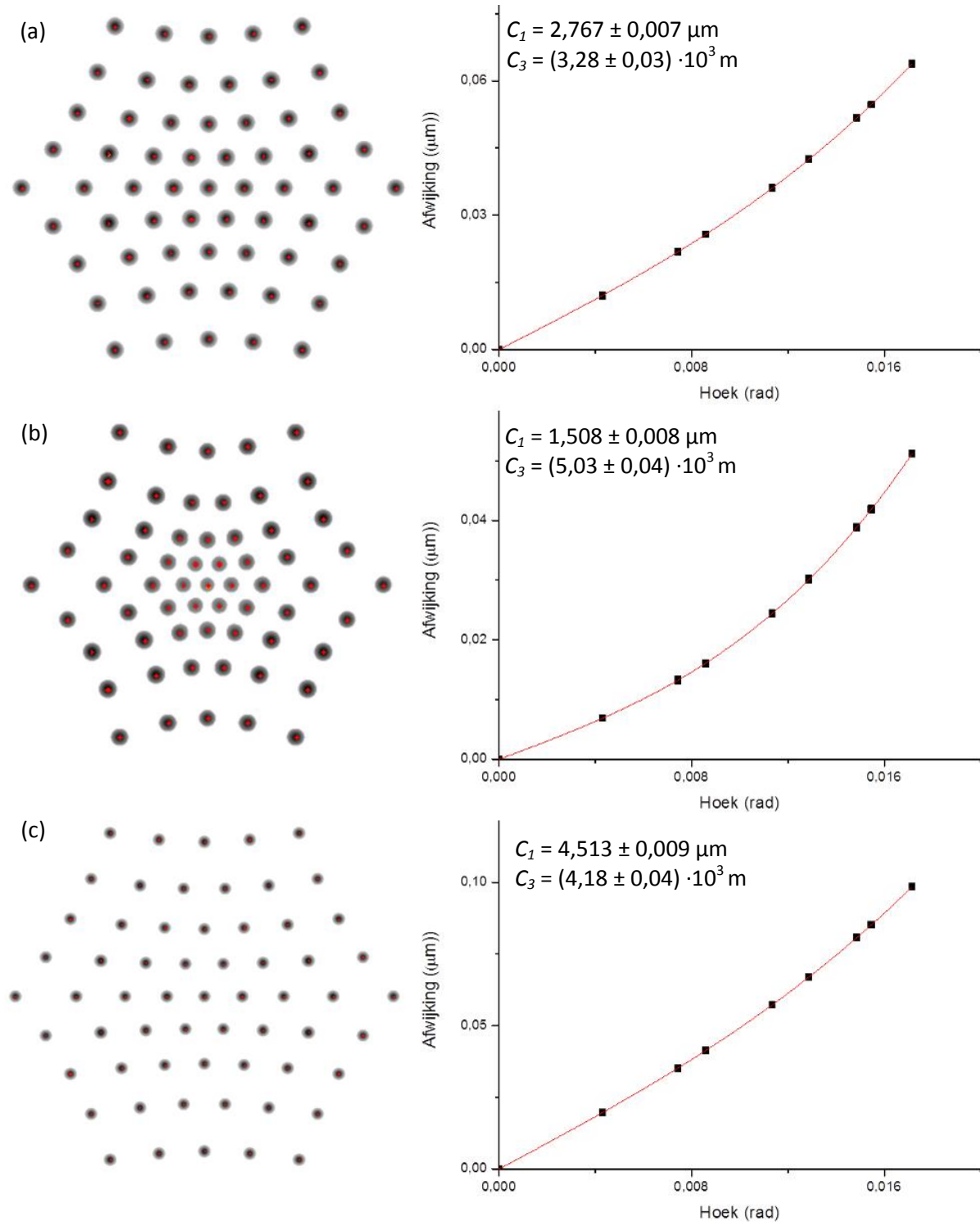
Figuur B-2: LEED patronen van het Si(111)-(7x7) oppervlak met toegepaste spot detectie. (a) Ingestelde drempelwaarde van 7. (b) Ingestelde drempelwaarde van 52. (c) Ingestelde drempelwaarde van 40. (d) Ingestelde drempelwaarde van 43. (e) Ingestelde drempelwaarde van 55. (f) Ingestelde drempelwaarde van 50.



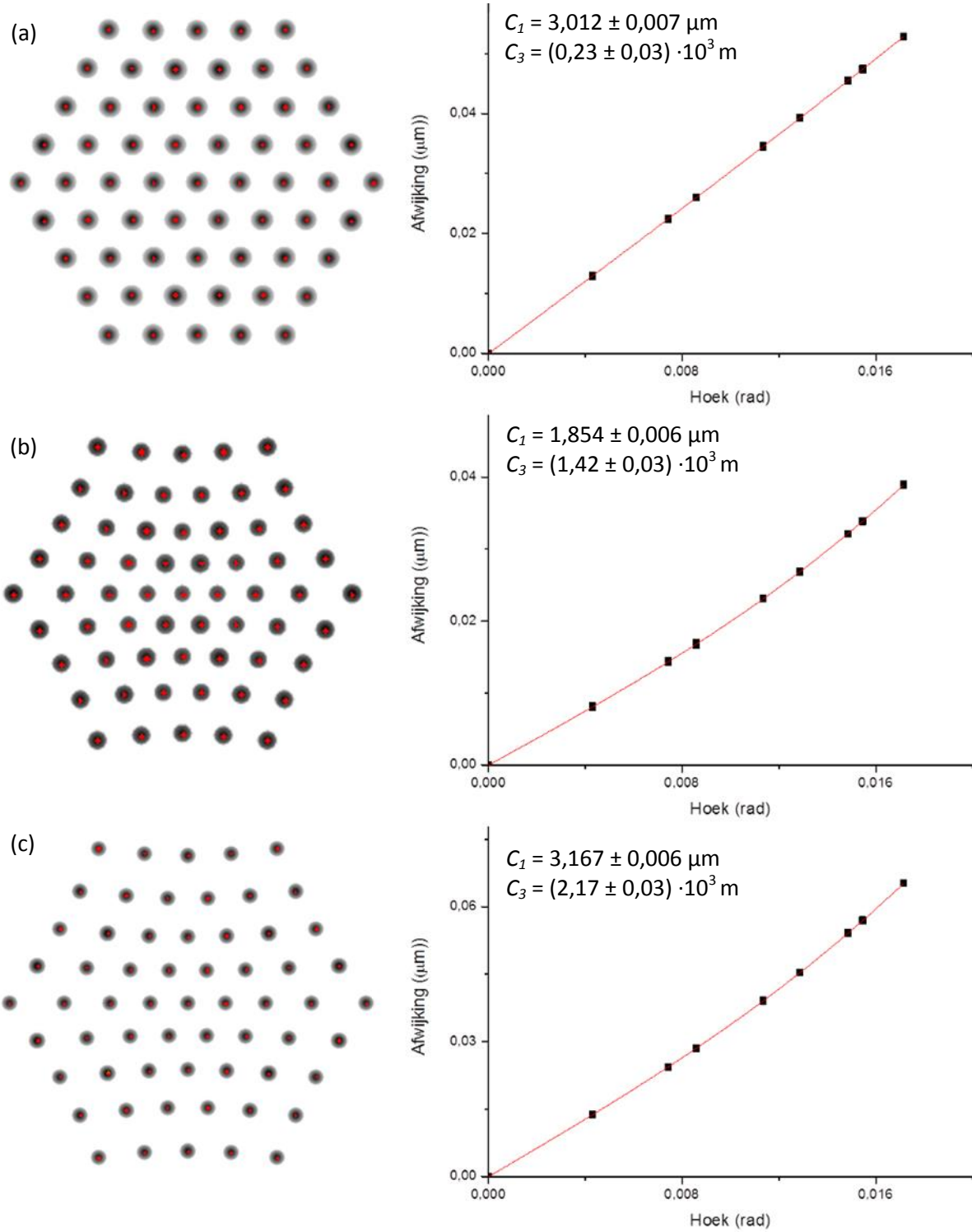
Figuur B-3: LEED patronen van het Si(111)-(7x7) oppervlak met toegepaste spot detectie. (a) Ingestelde drempelwaarde van 57. (b) Ingestelde drempelwaarde van 55.



## Bijlage C Fit van gedetecteerde spots bij gegenereerde LEED patronen

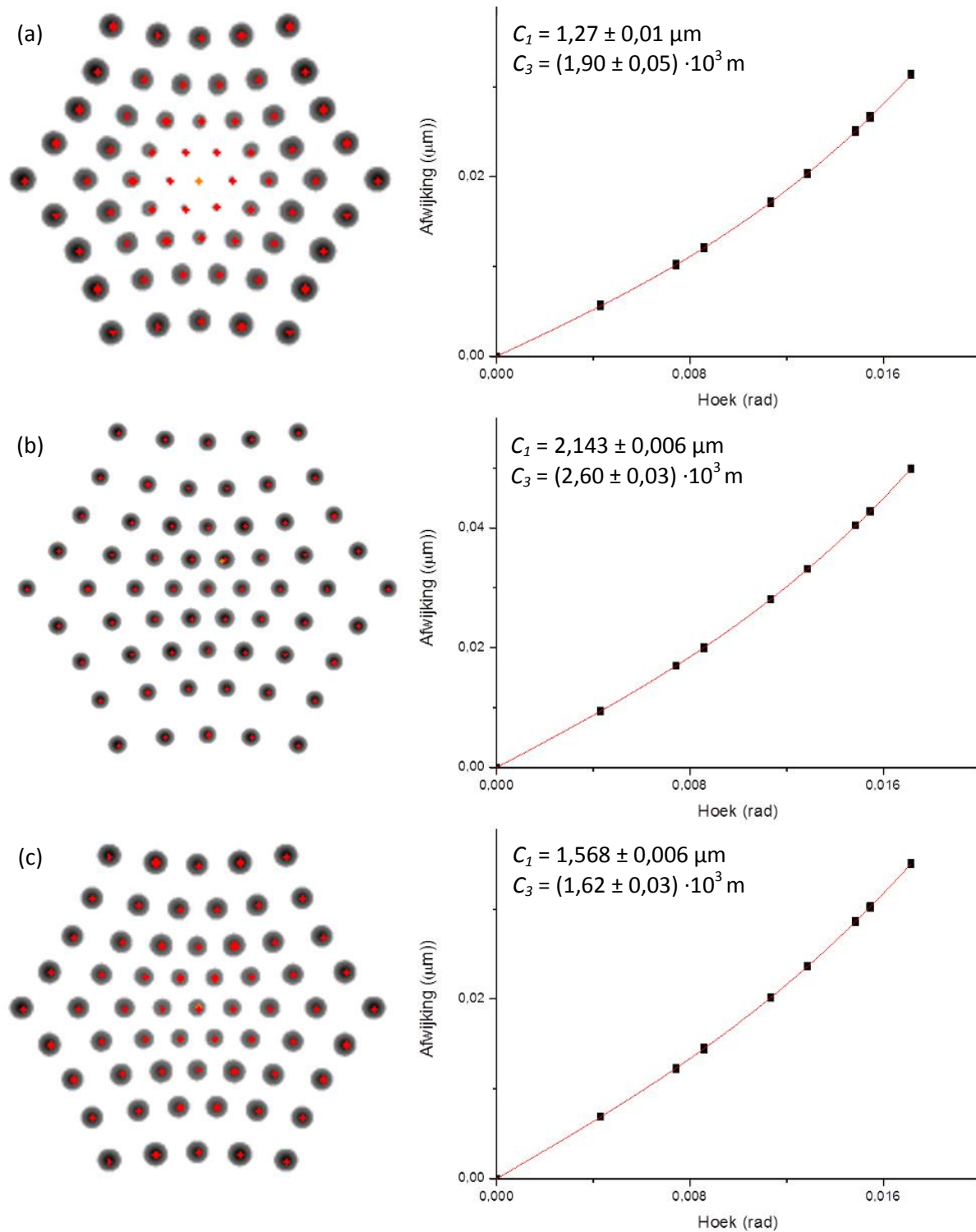


Figuur C-1: Gegenereerde LEED patronen met ingestelde waarden voor defocus en sferische aberraties en een fit in de vorm  $r_s = C_1\alpha + C_3\alpha^3$  door de afwijking van de spots als functie van de hoek met de optische as. (a) Defocus ingesteld op  $3,00 \mu\text{m}$  en derde orde sferische aberratie op  $3400 \text{ m}$ . (b) Defocus ingesteld op  $1,50 \mu\text{m}$  en derde orde sferische aberratie op  $5000 \text{ m}$ . (c) Defocus ingesteld op  $4,50 \mu\text{m}$  en derde orde sferische aberratie op  $4150 \text{ m}$ .



Figuur C-2: Gegeneerde LEED patronen met ingestelde waarden voor defocus en sferische aberraties en een fit in de vorm  $r_s = C_1\alpha + C_3\alpha^3$  door de afwijking van de spots als functie van de hoek met de optische as. (a) Defocus ingesteld op  $3,00 \mu\text{m}$  en derde orde sferische aberratie op  $230 \text{ m}$ . (b) Defocus ingesteld op  $1,85 \mu\text{m}$  en derde orde sferische aberratie op  $1400 \text{ m}$ . (c) Defocus ingesteld op  $3,16 \mu\text{m}$  en derde orde sferische aberratie op  $2130 \text{ m}$ .





Figuur C-3: Gegeneerde LEED patronen met ingestelde waarden voor defocus en sferische aberraties en een fit in de vorm  $r_s = C_1\alpha + C_3\alpha^3$  door de afwijking van de spots als functie van de hoek met de optische as. (a) Defocus ingesteld op  $1,26 \mu\text{m}$  en derde orde sferische aberratie op  $1910 \text{ m}$ . (b) Defocus ingesteld op  $2,14 \mu\text{m}$  en derde orde sferische aberratie op  $2570$ . (c) Defocus ingesteld op  $1,56 \mu\text{m}$  en derde orde sferische aberratie op  $1620 \text{ m}$ .

## **Bijlage D Grafische gebruikersomgeving**

Om de resultaten van de beschreven algoritmes weer te geven aan de gebruiker is er een grafische gebruikersinterface (GUI) gemaakt. De GUI is geprogrammeerd in de standaard meegeleverde bibliotheek TKinter van Python. In Figuur D-1 is de layout van geprogrammeerde GUI te zien. De GUI is opgedeeld in zes verschillende onderdelen. Deze onderdelen zijn: het canvas, de systeemknoppen, drempelwaarde instellingen, spot manipulatie knoppen, schakelknoppen en bewerkingsknoppen.

Het canvas is het onderdeel in de GUI waar op alle bewerkingen aan het LEED patroon getoond worden. Hierop zullen bijvoorbeeld de spot detectie en Delaunay-triangulaties getoond worden. Voorbeelden van wat op het canvas weergegeven kan worden zijn te vinden in Figuur 5-7

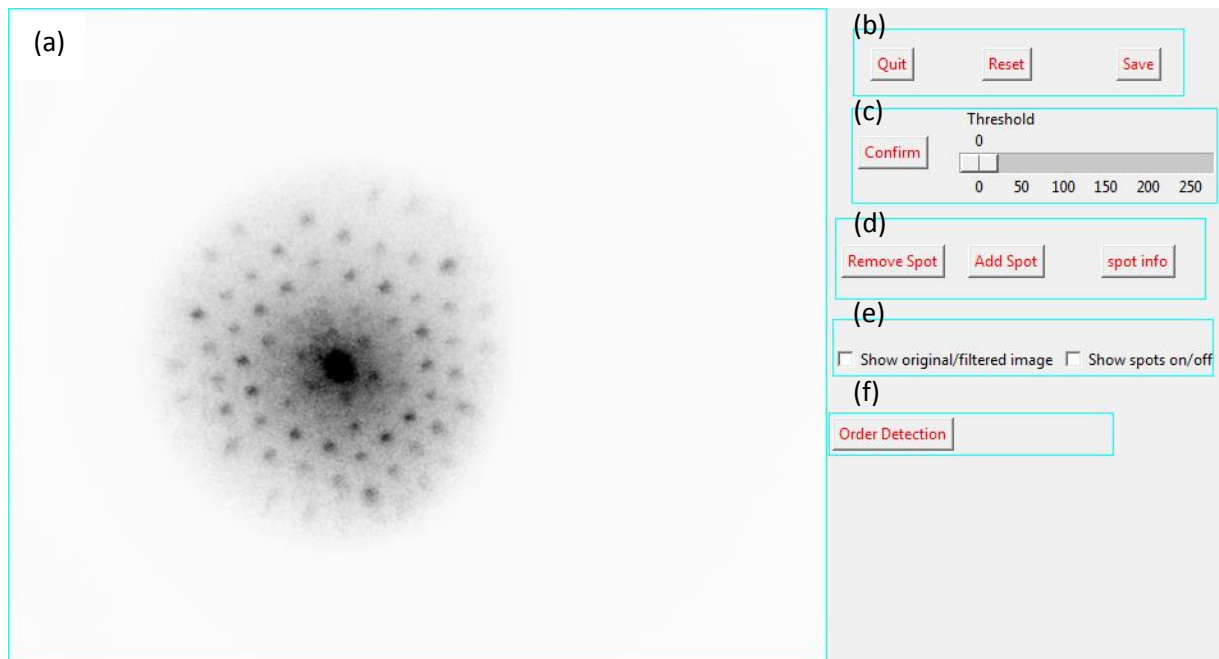
Met de systeemknoppen is het mogelijk om commando's te geven die op het hele programma betrekking hebben. De systeemknoppen bestaan uit, zoals in Figuur D-1(b) is te zien, een Quit-, een Reset- en een Save knop. De Quit-knop sluit het programma af waarbij alle niet opgeslagen informatie wordt verwijderd. De Reset-knop verwijdert alle berekende waardes uit het geheugen en zet alle ingestelde waardes terug op de standaard waardes.

De drempelwaarde instellingen bestaan uit een slider voor het instellen van de drempelwaarde en een Confirm-knop voor het bevestigen van deze waarde waarna de spotdetectie wordt gestart. De drempelwaarde loopt van 0 tot 255 omdat de afbeeldingen waarmee wordt gewerkt 8-bit afbeeldingen zijn.

Met de spot manipulatie knoppen is het mogelijk om verkeerd herkende spots te verwijderen of om cruciale spots die niet zijn herkend op de hoekpunten toe te voegen. De knop spot info geeft toegang tot spot specifieke informatie bijvoorbeeld x- en y-coördinaat in de afbeelding, afstand tot het middelpunt enzovoorts.

De schakelknoppen maken het mogelijk om te schakelen tussen twee standen. Op het moment zijn er twee schakelknoppen aanwezig. Met één van de knoppen is het mogelijk om te schakelen tussen het onbewerkte LEED patroon en het bewerkte LEED patroon. De tweede knop maakt het mogelijk om gedetecteerde spots wel of niet te laten zien op het canvas.

De bewerkingsknoppen zijn bedoeld voor het starten van bewerkingen nadat de spots zijn gedetecteerd. Op dit moment is hier alleen een knop voor de start van de orde detectie aanwezig. Wanneer het programma verder wordt ontwikkeld zullen hier ook knoppen voor tilt en astigmatisme detectie bij komen.



Figuur D-1: Grafische gebruikersomgeving. (a) Canvas waarop alle bewerkingen getoond worden. (b) Systeem knoppen. (c) Slider om de drempelwaarde in te stellen. (d) Knoppen voor spot manipulatie. (e) Check-knoppen voor het schakelen tussen standen. (f) Bewerkingsknoppen.

## **Bijlage E De originele opdrachtomschrijving**

Belangrijke vragen bij het bestuderen van materiaaloppervlakken zijn:

- Hoe zijn de atomen aan het oppervlak gerangschikt?
- Wat is het gedrag van de elektronen aan het oppervlak?
- Waar bevinden zich eenheidscel-hoge of atomaire stapranden?
- Op welke manier beïnvloeden bovenstaande zaken elkaar?

In Leiden bestuderen we deze en vele andere vragen op het gebied van de vaste stof fysica met behulp van lage-energie-elektronenmicroscopie (LEEM), een relatief nieuwe techniek. In een LEEM worden oppervlakken met behulp van coherente elektronen afgebeeld bij een energie van een fractie van een eV tot ca 50 eV. In Leiden gaan we daarbij veel verder dan het gebruik van de microscoop voor het onderzoek: veel werk wordt gedaan aan de doorontwikkeling van de LEEM techniek en aan het instrument zelf. De Leidse LEEM (ESCHER) behoort tot de absolute top in de wereld met een record resolutie van 1.4 nm.

In verhouding tot de diffractielimiet is deze resolutie voor een elektronenmicroscoop uitzonderlijk: normaliter wordt de resolutie van een elektronenmicroscoop bepaald door de sferische en chromatische aberraties van de lenzen. Er is een oneindige reeks aberratiecoëfficiënten, gelukkig convergeert de reeks snel: we kunnen het beeld immers zien. Teneinde de lensbeperkingen te doorbreken en de elektronenopbrengst te verhogen maakt ESCHER gebruik van een aberratiecorrector in de vorm van een elektronenspiegel. Door het instellen van de equipotentiaalvlakken voor deze spiegel zijn de chromatische- en de sferische aberraties van het elektronoptische systeem tot de derde orde corrigeerbaar. Momenteel vereist instelling van deze aberratiecorrector een uitstekende uitlijning van de microscoop en een tijdrovende analyse van het diffractiepatroon van een 7x7 gereconstrueerd Si(111) oppervlak.

Het doel van dit project is het automatiseren van deze instellingen en met name van C3, de derde orde sferische aberratiecoëfficiënt. Daartoe dient allereerst begrip gevormd te worden van het afbeeldingsprincipe van een LEEM en in het bijzonder van ESCHER. Vervolgens dienen middels beeldanalyse diffractiedata automatisch geïnterpreteerd te worden. De bedoeling is hiervoor de programmeertaal Python te gebruiken. De software dient uit de positie van de LEED-spots vervolgens parameters C1 (defocus) en C3 te extraheren waaruit met behulp van de optische systeemparameters aberratiecoëfficiënten berekend kunnen worden. Via de Python interface kunnen de berekende aberratiecoëfficiënten rechtstreeks naar de besturingshardware geschreven worden zodat de microscoop met een druk op de knop iteratief gecorrigeerd kan worden.

Preconditie voor het automatisch corrigeren van de microscoop is een goede uitlijning. Dit uitlijnproces bestaat uit het uitvoeren van een aantal deflectorinstellingen. De optimalisatie van de deflectoren kan worden bepaald middels de beeldbeweging als gevolg van de Lorentzkracht op de elektronen in de lenzen. Doel van het project is ook deze correcties geautomatiseerd te kunnen uitvoeren.