



Universität Karlsruhe (TH)  
Institut für Prozeßrechentechnik, Automation und Robotik(IPR)

Diplomarbeit

# **Fokussieren-basierte Rekonstruktion von Mikroobjekten**

Bearbeiter    Cand. Inform. Jan Wedekind

Referent      Prof. Dr.-Ing. H. Wörn  
Korreferent   Prof. Dr.rer.nat. U. Brinkschulte  
Betreuer      Dipl.-Inform. Axel Bürkle

Zeitraum     1.12.2001 – 31.05.2002

### Zusammenfassung

Mit einem Lichtmikroskop und einem motorisiertem Z-Tisch kann man Fokussereien von Mikroobjekten erstellen. Durch Maximierung eines lokalen Schärfemaßes lassen sich die Höhenkarte der Objektoberfläche und künstliche Bilder mit erweiterter Schärfentiefe erzeugen. Wengleich Maximierung eines Schärfemaßes ohne spezielle Bildverarbeitungs-Hardware nicht echtzeitfähig ist, so wird sie doch bereits in der Forensik und der Halbleitertechnik angewandt.

Die in der Literatur vorgestellten Algorithmen weisen allerdings systematische Fehler auf. Während der hier vorgestellte Bildverarbeitungs-Algorithmus im Vergleich ein mehrfaches an Rechenzeit benötigt, unterdrückt er dagegen wirksam diese Fehler und liefert unter gleichen Bedingungen ein besseres Ergebnis.



Abbildung 1: Bild mit 2 mm Schärfentiefe von einem Ammoniten mit ca. 1.5 mm Durchmesser

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>3</b>
<b>1 Einleitung</b>	<b>7</b>
1.1 Motivation . . . . .	7
1.2 Meßaufbau . . . . .	7
1.3 Aufgabenstellung . . . . .	8
1.4 Weitere Anwendungen . . . . .	9
<b>2 Stand der Technik</b>	<b>11</b>
2.1 Messen ohne Verwendung von Fokussereien . . . . .	11
2.1.1 Lasertriangulation . . . . .	11
2.1.2 Konfokales Laserscanningmikroskop . . . . .	12
2.2 Auf Fokussereien basierende Meßverfahren . . . . .	13
2.2.1 Maximieren eines Schärfemaßes . . . . .	13
2.2.2 Vergleich der Unschärfe . . . . .	14
2.3 Laplace-Pyramiden . . . . .	15
2.4 Bilder mit erweiterter Schärfentiefe . . . . .	16
2.5 Inverse Faltung . . . . .	16
2.5.1 Fourieroptik . . . . .	16
2.5.2 Dünn besetzte Matrizen . . . . .	17
2.5.3 Maximum-Likelihood Schätzer . . . . .	18
2.5.4 Blinde inverse Faltung . . . . .	19
2.5.5 Anwendbarkeit . . . . .	21
<b>3 Neuer adaptiver Algorithmus</b>	<b>24</b>
3.1 Maximierung eines Schärfemaßes . . . . .	24
3.2 Projektion von Hilfsmustern . . . . .	25
3.3 Systematischer Fehler . . . . .	29
3.4 Normierte Grauwertvarianz . . . . .	30
3.5 Bestimmung der Höhenkarte . . . . .	32
3.6 Komplexität des Algorithmus . . . . .	33
3.7 Vergleich mit nicht-adaptiver Filterung . . . . .	34
<b>4 Auflösungsverbesserung</b>	<b>35</b>
4.1 Zusätzlicher Vergleich der Unschärfe . . . . .	35
4.2 Interpolation des Schärfemaßes . . . . .	37

<b>5</b>	<b>Implementierung</b>	<b>39</b>
5.1	Verwendete Software . . . . .	39
5.1.1	Betriebssystem . . . . .	39
5.1.2	Übersetzer . . . . .	39
5.1.3	Programmbibliotheken . . . . .	39
5.1.4	Programmierwerkzeuge . . . . .	39
5.1.5	Textverarbeitung . . . . .	40
5.2	Die Benutzeroberfläche . . . . .	40
5.2.1	Das Menü . . . . .	40
5.2.2	Die Unterfenster . . . . .	42
5.2.3	Einstellung der Parameter . . . . .	43
5.2.4	Kalibrierungs-Dialog . . . . .	43
5.3	Die Programmbibliothek . . . . .	45
5.3.1	Konzepte . . . . .	45
5.3.2	Beispiel zur Berechnung der Höhenkarte . . . . .	46
5.3.3	Beispiel zur Berechnung des Tiefenbildes . . . . .	46
<b>6</b>	<b>Praktische Ergebnisse</b>	<b>48</b>
6.1	Visueller Vergleich . . . . .	48
6.2	Vertikale Auflösung . . . . .	48
6.2.1	Test mit simulierter Fokusserie . . . . .	48
6.2.2	Test mit realem Objekt . . . . .	52
6.3	Laterale Auflösung . . . . .	54
6.4	Benötigte Rechenzeit . . . . .	54
6.5	Beobachtungen . . . . .	55
<b>7</b>	<b>Ausblick</b>	<b>57</b>
7.1	Verbesserungen am Meßaufbau . . . . .	57
7.2	Verbesserungen am Algorithmus . . . . .	57
7.3	Schärfemaß mit Wavelets . . . . .	58
7.4	Erweiterungen . . . . .	59
<b>8</b>	<b>Danksagung</b>	<b>61</b>
<b>A</b>	<b>Anhang</b>	<b>62</b>
A.1	Inverse Faltung . . . . .	62
A.1.1	Diskrete Fouriertransformation . . . . .	62
A.1.2	Darstellung von Faltung mit Matrizen . . . . .	62
A.1.3	Eigenraum zirkulanter Matrizen . . . . .	63
A.1.4	EM-Algorithmus . . . . .	63
A.1.5	E-Schritt hier . . . . .	64
A.2	Povray-Dateien . . . . .	66
A.3	Mexican Hat Wavelet . . . . .	67
A.4	Varianz . . . . .	68
A.5	Methode kleinster Quadrate . . . . .	68
A.5.1	Gauß-Markov Modell und Parameterschätzung . . . . .	68
A.5.2	Regressions-Parabel . . . . .	69
A.5.3	Regressions-Ebene . . . . .	69
	<b>Literaturverzeichnis</b>	<b>70</b>

*INHALTSVERZEICHNIS*

5

**Index**

**74**

**Erklärung:**

Hiermit erkläre ich, daß ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe angefertigt habe. Die verwendeten Quellen sind im Text kenntlich gemacht und im Literaturverzeichnis vollständig angegeben.

Datum

Unterschrift

# Kapitel 1

## Einleitung

### 1.1 Motivation

Am Institut für Prozeßrechentechik, Automation und Robotik (IPR) der Universität Karlsruhe wird an der Entwicklung einer flexiblen Mikromanipulationsstation zur automatischen Montage von Mikrosystemen gearbeitet. Während bereits die Manipulation von Mikroobjekten eine technische Herausforderung darstellt, so ist die Automatisierung noch viel schwieriger.

Bild 1.1 zeigt einen der eingesetzten Mikroroboter. Am Robotergreifer sind Dehnungsmeßstreifen angebracht, mit denen der Kontakt zu Mikroobjekten wahrgenommen werden kann. Um ein Mikroobjekt greifen zu können, muß aber dessen Position bestimmt werden. Außerdem muß nach einem Montagevorgang nachgeprüft werden, ob der angestrebte Zustand erreicht wurde.

Dieses Problem läßt sich durch Bildverarbeitung lösen. Mikroskopie-Bilder haben allerdings eine sehr geringe Schärfentiefe. Es gibt aber Algorithmen, die aus einer Serie von Bildern ein Bild erweiterter Schärfentiefe und sogar eine Höhenkarte der betrachteten Mikroszene erzeugen. In der Literatur findet man Bilderkennungsalgorithmen, die in einer solchen Höhenkarte Objekte erkennen können (siehe z.B. [JH96]). Dies wäre also ein möglicher Weg, um Modelle von Mikroobjekten zu erzeugen und diese auch in einer Mikroszene wiederzufinden.

Diese Diplomarbeit behandelt den Bildverarbeitungsschritt, der aus einer Serie von Mikroskop-Bildern eine Höhenkarte erzeugt. Obwohl es bereits viel Literatur zu diesem Thema gibt, wiesen alle im Rahmen dieser Diplomarbeit untersuchten Algorithmen selbst unter kontrollierten Bedingungen systematische Fehler auf. Der in den Kapiteln 3.4 und 3.5 dieser Diplomarbeit vorgestellte Algorithmus unterdrückt wirksam diese systematischen Fehler, ohne daß Kompromisse eingegangen werden müssen.

Der Quellcode des entwickelten Computerprogramms ist frei verfügbar (siehe Seite 77).

### 1.2 Meßaufbau

Um Bilder aufzunehmen stand das leistungsfähige *Leica DM RXA* Mikroskop zur Verfügung (siehe Abb. 1.2). Dieses Mikroskop verfügt über einen elektro-

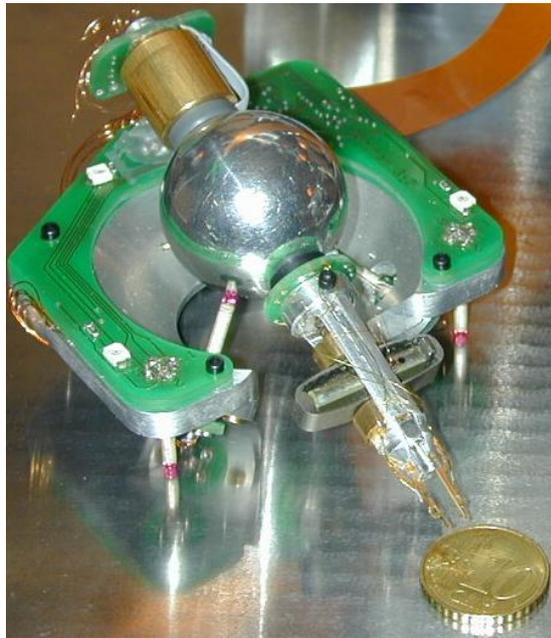


Abbildung 1.1: Mikroroboter MINIMAN 3-II

nisch steuerbaren Objektivrevolver mit einem  $2.5\times$ -, einem  $5\times$ - und einem  $20\times$ -Objektiv.

Die CCD-Kamera und der Framegrabber liefern ein Bild mit  $768 \times 576$  Pixel. Die Grauwerte haben eine Tiefe von 8 Bit. Zusammen mit dem  $20\times$ -Objektiv erhält man eine Auflösung von ca.  $0.75 \mu\text{m}/\text{Pixel}$  (vgl. auch Bild des Kalibrierungsgitters 3.1).

Die Zwei-Kanal-Beleuchtung ermöglicht es, Objekte von unten durch den Glastisch (Durchlicht) oder von oben durch das Objektiv (Auflicht) zu beleuchten. Außerdem können mit einem Filtereinsatz Muster auf das Objekt projiziert werden.

Das Mikroskop hat einen mit einem Piezo-Motor angetriebenen Z-Tisch und ist außerdem mit einem entsprechenden X-Y-Tisch nachgerüstet worden. Damit läßt sich der Objektträger mit einer Schrittweite von  $0.1 \mu\text{m}$  bewegen.

Zur Bildverarbeitung stand ein Dual Pentium III mit 1GHz-Prozessoren, 500MByte Hauptspeicher und Linux-Betriebssystem zur Verfügung.

### 1.3 Aufgabenstellung

Mithilfe des in Kap. 1.2 dargestellten Aufbaus soll die Oberfläche eines Mikroobjektes rekonstruiert werden. Dafür wird von dem Mikroobjekt eine Serie von Bildern gemacht, wobei man die Fokalebene sukzessive über das Mikroobjekt streichen läßt. Mit dem Z-Tisch können die Abstände der fokussierten Ebenen genau gewählt werden.

Eine einfache Software bewegt wiederholt den Tisch und nimmt jedesmal ein Mikroskopbild auf. Dieser dreidimensionale Datensatz wird im folgenden Fokus-



Abbildung 1.2: Leica DM RXA Mikroskop

serie genannt. Aus dieser Fokusserie sollen ein Höhenrelief sowie ein scharfes Bild der gesamten Objektfläche rekonstruiert werden (siehe Abb. 1.3).

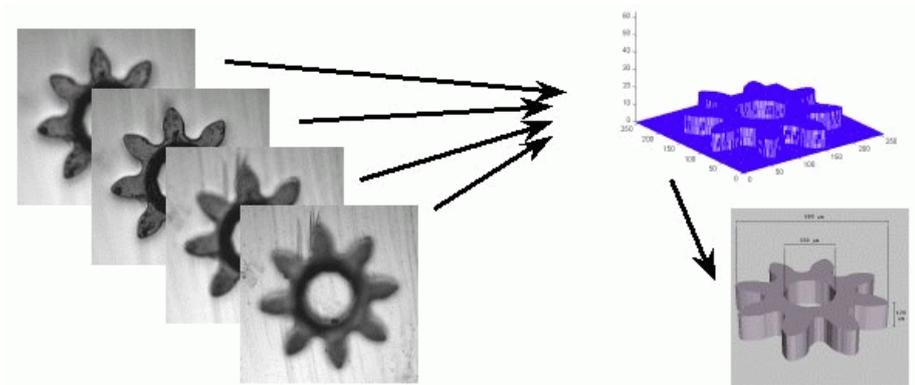


Abbildung 1.3: Aufgabenstellung

## 1.4 Weitere Anwendungen

Ein häufiges Problem ist die geringe Schärfentiefe von Mikroskopbildern. Computerprogramme, die aus einer Fokusserie ein scharfes Bild erzeugen, sind weit verbreitet. Am Mechanical Engineering Lab in Tsukuba (Japan) wurde ein System entwickelt, das aus dem Videobild einer dynamisch fokussierenden Linse ein Videobild mit hoher Schärfentiefe in Echtzeit erzeugt [OOT<sup>+</sup>00]. Die hierfür benötigte hohe Rechenleistung wird von Spezialhardware erbracht.

Optische Meßverfahren werden bereits zur Qualitätsprüfung in der Halbleit-

terindustrie eingesetzt. Fokussierens-basierte Meßtechnik verwendet handelsübliche Komponenten und ist daher kostengünstig einsetzbar [Sch97].

Bei der Identifikation von Schußwaffen in der Forensik werden Fokussierens-basierte Verfahren eingesetzt, um z.B. den Schlagbolzeneindruck auf der Patronenhülse oder Verfeuerungsspuren auf der Projektiloberflächen zur weiteren Bearbeitung zu erfassen[Leó99].

# Kapitel 2

## Stand der Technik

Zur 3-D Rekonstruktion von Mikroobjekten im allgemeinen und speziell aus Fokussereien gibt es bereits eine Menge verschiedener Ansätze.

### 2.1 Messen ohne Verwendung von Fokussereien

#### 2.1.1 Lasertriangulation

Man kann einen Laserstrahl oder eine Laserebene (Laserstrahl, der durch zylindrische Linse aufgefächert wird) auf das zu rekonstruierende Objekt richten. Zeitgleich wird ein Bild des Objekts mit einer CCD-Kamera aufgenommen. Wenn man die Position des entstehenden Schnittpunktes bzw. der entstehenden Schnittlinie im Kamerabild bestimmen kann, lassen sich dreidimensionale Positionen von Punkten auf der Oberfläche des Objekts durch Triangulation bestimmen. Die Oberfläche des Objekts kann so abgetastet werden (siehe Abb. 2.1).

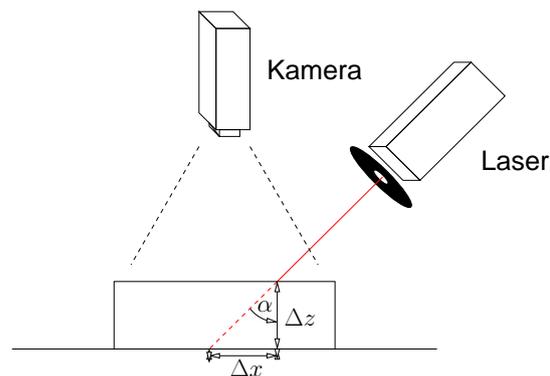


Abbildung 2.1: Rekonstruktion einer Oberfläche mittels Lasertriangulation

Dieses Verfahren funktioniert relativ schnell und lässt sich allgemein bei jedem nicht-durchsichtigen Mikroobjekt anwenden, dessen Oberfläche diffuse Reflexion aufweist. Bei metallischen Oberflächen kann es allerdings u.U. zu stören-

den Reflektionen kommen. Lasertriangulation funktioniert zuverlässig, wenn diese Bedingungen gegeben sind. Die Dicke des Laserbündels setzt dem Auflösungsvermögen allerdings eine Grenze. Da das Bild der Laserlinie außerdem hinreichend scharf sein muß, ist ausreichende Schärfentiefe vonnöten. Die anwendbare Vergrößerung ist daher begrenzt.

Unter Verwendung einer Kamera mit  $5 \mu\text{m}/\text{Pixel}$  Bildauflösung und einer Laserlinienbreite von  $120 \mu\text{m}$  wurde eine Standardabweichung von  $8.57 \mu\text{m}$  in der Höhenbestimmung (der Höhe  $\Delta h$ ) ermittelt. Dabei war die laterale Auflösung  $10 \mu\text{m}$ [Kie01].

### 2.1.2 Konfokales Laserscanningmikroskop

Sogenannte *konfokale Laserscanningmikroskope*<sup>1</sup> tasten mit einem Laserstrahl das Objekt ab. Artikel [SBGH98] zeigt einen speziellen Aufbau, der dort *4-Pi confocal microscope* genannt wird. Der Aufbau ist schematisch in Abbildung 2.2 wiedergegeben.

Der Laserstrahl wird mit einem Strahlteiler gespalten. Die beiden Strahlenbündel werden mit Linsen auf einen punktförmigen Bereich konzentriert, dessen Ausdehnung auch noch durch die entstehende Interferenz begrenzt wird. Das gestreute Licht wird mit einem Detektor gemessen. Dieser Aufbau eignet sich für transparente Objekte wie zum Beispiel Frosch-Embryos[SBGH98]. Durch

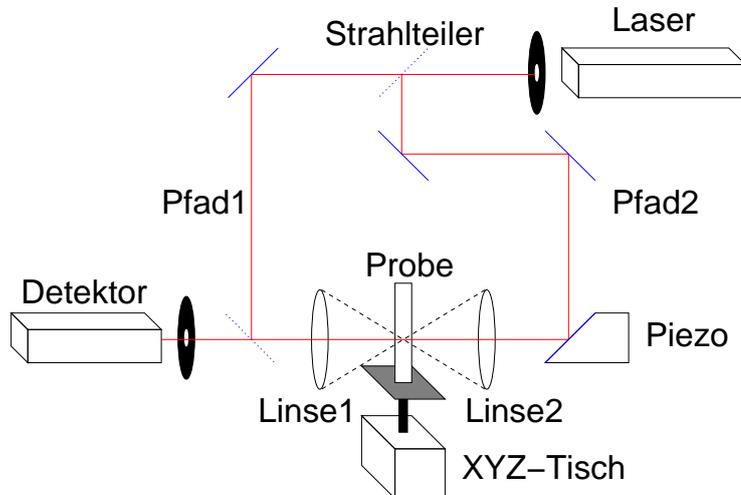


Abbildung 2.2: Schema des 4-Pi confocal microscope[SBGH98]

Abtasten mit dieser Vorrichtung lassen sich dreidimensionale Bilder gewinnen, die selbst ohne Nachbearbeitung schon eine hohe Qualität aufweisen.

Durch Nachbearbeitung der mit einem CSM<sup>2</sup> gewonnenen Bilddaten lassen sich Strukturen im Bereich von 50-200 nm (!) auflösen[Fra00]. Siehe hierzu auch Kapitel 2.5, Seite 16.

<sup>1</sup>engl.: confocal scanning microscope (CSM)

<sup>2</sup>confocal scanning microscope, deutsch:konfokal abtastendes Mikroskop

## 2.2 Auf Fokussereien basierende Meßverfahren

### 2.2.1 Maximieren eines Schärfemaßes

Ein weitverbreitetes Verfahren zur Schätzung der Höhe ist die Maximierung eines Schärfemaßes. Dafür werden z.B. die Gradientenbeträge herangezogen, die man durch Anwenden des Sobel-Operators erhält. Man kann dieses Schärfemaß dann für das Gesamtbild oder über lokale Bereiche des Bildes zusätzlich mitteln.

Die ermittelte "Schärfe" eines Bildes läßt sich mit einem Algorithmus, der die Lage der Fokalebene in einer Schleife anpaßt, zwecks Autofokussierung optimieren. Der Autofokus von Photoapparaten arbeitet zum Beispiel auf diese Art. Vergleichbare Algorithmen zur automatischen Fokussierung werden auch in der Mikroskopie verwendet (siehe z.B. [RBV98]).

Zur Rekonstruktion von Oberflächen dagegen wird ermittelt, in welchem Bild der Fokusserei ein lokaler Bereich das höchste Maß an Schärfe erreicht. Die entsprechende Lage der Fokalebene bestimmt dann die Höhe der Objektoberfläche für diesen lokalen Bereich. Aus den "scharfen" Bildteilen läßt sich ein Gesamtbild zusammensetzen, das überall "scharf" ist.

In [Bas95] wurden verschiedene (lokale) Schärfemaße zur Oberflächenrekonstruktion herangezogen. Unter anderem wurde

$$S_{SOBEL}\{g_z\}(x, y) = \sqrt{[(\mathcal{I}_x \otimes g_z)(x, y)]^2 + [(\mathcal{I}_y \otimes g_z)(x, y)]^2}$$

als lokales Schärfemaß verwendet. Dabei ist  $\otimes$  der Faltungsoperator und  $\mathcal{I}_x$  und  $\mathcal{I}_y$  sind die zum Sobeloperator zugehörigen Filtermatrizen, mit denen das Bild  $g$  gefiltert wird:

$$\mathcal{I}_x = \frac{1}{4} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad \text{und} \quad \mathcal{I}_y = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Die Höhenkarte  $d$  wird dann durch Maximierung des jeweiligen Schärfemaßes  $S\{\cdot\}$  ermittelt:

$$d(x, y) = \underset{z \in \{0, \dots, N-1\}}{\operatorname{argmax}} (S\{g_z\}(x, y)) \quad (2.1)$$

Im Rahmen der Doktorarbeit [Leó99] von F. P. León am MRT<sup>3</sup> der Universität Karlsruhe wurden unter anderem durch Maximierung eines Schärfekriteriums Oberflächen von Pistolenkugeln und Patronenhülsen rekonstruiert. In der Diplomarbeit [Bas95] findet sich ein qualitativer Vergleich verschiedener Schärfemaße, die eingesetzt wurden, um die Höhenkarte aus der Fokusserei zu gewinnen. Der Sobeloperator schnitt beim Vergleich der verschiedenen Filter eher schlecht ab. Wesentlich bessere Ergebnisse wurden laut [Bas95] mit dem Kriterium *lokaler Strukturgehalt* erzielt:

$$S_{LS}\{g_z\}(x, y) = \frac{\sum_{p=-m}^m \sum_{q=-n}^n g_z^2(x+p, y+q)}{\left[ \sum_{p=-m}^m \sum_{q=-n}^n g_z(x+p, y+q) \right]^2}; \quad m, n \in \mathbf{N}_0$$

<sup>3</sup>Institut für Meß- und Regeltechnik

In [Leó99] wird die Grauwertvarianz in einer  $5 \times 5$  Umgebung als Schärfemaß verwendet<sup>4</sup> Die *lokale Grauwertvarianz* hat sich auch im Rahmen dieser Arbeit als gutes Kriterium erwiesen.

[ZZW00] schlägt ein Schärfemaß vor, das sich aus Momenten verschiedener Ordnung zusammensetzt und unabhängig vom Bildinhalt sein soll. Das Schärfemaß wurde in [ZZW00] allerdings nicht auf lokale Bereiche des Bildes angewendet. Dieses Schärfemaß wurde hier nicht getestet.

In [Rön96] wurde eine Standardabweichung der Höhenkarte von  $12 \mu\text{m}$  erreicht. Dabei war der Abstand der Fokusebenen der Bildsequenz  $64 \mu\text{m}$ . Die kleine Standardabweichung, die unter der Z-Auflösung der Fokuserie liegt, wurde dadurch erreicht, daß zusätzlich die Lage des Schärfemaximums zwischen zwei Bildern der Fokuserie geschätzt wurde.

Durch **Projektion von Gittern** oder Rastern auf das Objekt kann man das Spektrum der Bilder aktiv verbessern. Die zusätzlichen Bildstrukturen erhöhen die Stabilität des Schärfemaßes. [NN94b] behandelt die optimale Abstimmung von verwendeten Projektionsmuster und Filter aufeinander.

S. K. Nayar hat neben dem eben zitierten Artikel [NN94b] über aktive Beleuchtung auch die Artikel [NN94a] und [NN90] über die Schätzung der Bildtiefe mithilfe eines Laplace-Operators (zweite Ableitung der Grauwertfunktion) veröffentlicht und gute Ergebnisse erzielt. Man beachte, daß der Laplace-Operator im Diskreten große Ähnlichkeit mit einem Wavelet für hohe Frequenzen aufweist.

## 2.2.2 Vergleich der Unschärfe

Verfügt man über zwei Bilder einer Szene, die sich nur in der fokussierten Ebene unterscheiden, kann man durch Vergleich der unterschiedlichen Degradierung der korrespondierenden lokalen Bereiche die Tiefe der beobachteten Oberfläche abschätzen.

[EL93] zeigt einen Überblick über die Methoden, mit denen man eine Höhenkarte durch Vergleich der Unschärfe erhalten kann. Von einem Objekt werden zwei Bilder gemacht, die sich nur in der Wahl der Fokalebene unterscheiden. Zu einem lokalen Bereich des Objektes (dargestellt durch die zweidimensionale Grauwertfunktion  $s$ ) erhält man zwei Bereiche  $g_1$  und  $g_2$  in den beiden Bildern, die durch Faltung mit den lokalen Punktantworten  $h_1$  und  $h_2$  entstehen:

$$g_1 = s \otimes h_1, g_2 = s \otimes h_2$$

Da es sich um lineare und (lokal) verschiebungsinvariante Filterung handelt, gibt es  $h_3$  mit:

$$h_1 \otimes h_3 = h_2$$

[EL93] zeigt weiterhin verschiedene Schätzer für  $h_3$ . Die lokale Höhe hängt direkt mit der "Breite" von  $h_3$  zusammen. Durch Parametrisierung von  $h_3$  läßt sich also mithilfe der zwei Bilder eine Höhenkarte direkt schätzen.

In [EL93] wird gezeigt, daß Matrix-basierte Methoden der inversen Filterung im Fourierbereich vorzuziehen sind, da die Schätzung des Spektrums für einen lokalen Bereich problematisch ist. Da die diskrete Fouriertransformation

---

<sup>4</sup>Da sich die Szene hier schlecht vollständig ausleuchten ließ, wurde die eingesetzte Fokuserie außerdem aus mehreren Fokuserien mit verschiedenen Beleuchtungsbedingungen zusammengesetzt.

nur eine periodische Faltung erlaubt, werden bei Faltungsoperationen Fehler in den Randbereichen von  $h_3$  verursacht. Die Wahl einer günstigen Fensterfunktion ist problematisch. Bei Matrix-basierten Methoden wird die inverse Faltung durch ein Gleichungssystem durchgeführt. Natürlich ist auch hier eine Parametrisierung von  $h_3$  angebracht.

In [Fav99] ist ein Beispiel von S. Nayar gezeigt: Zwei Bilder mit verschiedener Fokussierung einer Szene sind gegeben. Durch lokalen Vergleich der beiden Bilder wurde eine Höhenkarte erzeugt, die sehr stabil ist. Die Auflösung der Höhenkarte konnte in dem Beispiel von [Fav99] hoch gewählt werden, weil das Bild überall hohe Ortsfrequenzen enthielt. Die Methode, mit der diese Bilder erzeugt wurden, ist in [Fav99] nicht erklärt.

## 2.3 Laplace-Pyramiden

In dem Buch [Hab95] finden sich Verfahren, die mit sogenannten Laplace-Pyramiden arbeiten. Dabei erhält man zu einem Bild mit  $(2^n + 1) \times (2^n + 1)$  Pixel Kantenlänge durch mehrfache Filterung und Reduktion der Auflösung eine "Pyramide" (Gauß-Pyramide) von Bildern mit abnehmender Genauigkeit. Die Differenz zweier benachbarter Ebenen der Gauß-Pyramide ergibt eine Ebene der Laplace-Pyramide.

Die Gauß-Pyramiden-Ebene mit der kleinsten Auflösung (von  $1 \times 1$ ) zusammen mit der Laplace-Pyramide erlaubt die verlustlose Rekonstruktion des Originalbildes. Ähnlich wie bei der *Multi Auflösungs-Analyse*<sup>5</sup> mit Wavelets geben die Ebenen der Pyramide ein Frequenzspektrum für jeden Bereich des Bildes an.

In der Studienarbeit von W. Diebel [Die96] wurden die Laplace-Pyramiden genutzt, um aus einer Fokusserie ein Bild mit erweiterter Schärfentiefe zu erhalten. Das Verfahren arbeitet wie folgt:

1. Erzeuge Laplace-Pyramiden zu allen Bildern der Fokusserie
2. Alloziere die Zielpyramide.
3. Für jedes Pixel jeder Pyramidenebene der Zielpyramide:
  - (a) Suche das betragsmäßig größte Element an der entsprechenden Position in allen Pyramiden der Fokusserie.
  - (b) Trage den Wert (mit Vorzeichen) in die Zielpyramide ein.
4. Transformiere die Zielpyramide zurück, um das Bild mit erweiterter Schärfentiefe zu erhalten.

Der Algorithmus funktioniert sehr gut. Vorhandenes Kamerarauschen wird allerdings auch maximiert. Dies stellt bei einer guten Kamera aber kein Problem dar. Der Algorithmus erzeugt jedoch keine Höhenkarte. Die in [Hab95] angegebene Methode müßte außerdem noch erweitert werden, wenn man sich nicht auf Bilder mit  $(2^n + 1) \times (2^n + 1)$  Pixel beschränken will.

---

<sup>5</sup>engl.: multi-resolution analysis

## 2.4 Bilder mit erweiterter Schärfentiefe

Aus einer Fokusserie und der zugehörigen Tiefenkarte läßt sich ein Bild mit erweiterter Schärfentiefe erzeugen. Im folgenden soll es *Tiefenbild* genannt werden.

Um ein Tiefenbild  $v$  zu erhalten, genügt es, ein Bild aus Pixel der jeweils scharfen Bilder der Fokusserie  $\{g_0, \dots, g_{N-1}\}$  zusammenzusetzen. Die Höhenkarte  $d$  gibt zu jedem Pixel den Bildindex des jeweils scharfen Bildes an (Abb. 2.3):

$$v(x, y) = g_{d(x,y)}(x, y)$$

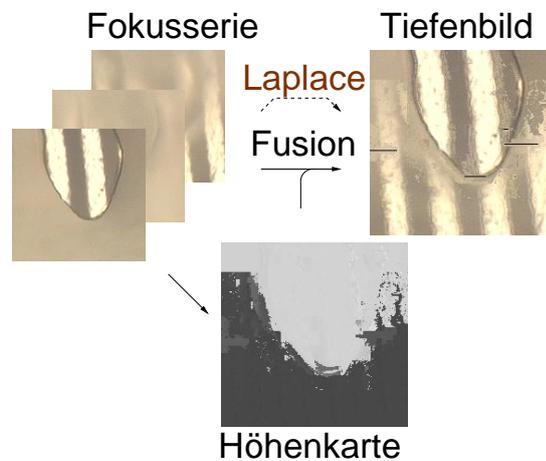


Abbildung 2.3: Fusion von Bildern der Fokusserie zu einem Tiefenbild

Mit den in Kapitel 2.3 vorgestellten Laplace-Pyramiden läßt sich das Tiefenbild auch direkt (d.h. ohne Bestimmung einer Höhenkarte) erzeugen. Die Erzeugung von Tiefenbildern ist wesentlich einfacher als die Berechnung von Höhenkarten, da selbst mit einer falschen Höhenkarte oft noch ein relativ gutes Tiefenbild erzeugt werden kann.

Man kann die zum Erzeugen des Tiefenbildes verwendete Fokusserie frei wählen. Es ist also z.B. möglich, eine Fokusserie unter Verwendung eines projizierten Gitters zu erstellen, um eine gute Höhenkarte zu erzeugen, später aber dann eine Fokusserie ohne Beleuchtungsmuster verwenden, um das Tiefenbild zusammenzusetzen.

## 2.5 Inverse Faltung

### 2.5.1 Fourieroptik

Im Rahmen dieser Diplomarbeit wurde versucht, Mikroobjekte durch dreidimensionale inverse Faltung der Fokusserie zu rekonstruieren. Dies ist allerdings nicht gelungen. Dennoch soll hier kurz darauf eingegangen werden.

Alle bilderzeugenden Systeme haben eine Punktantwort, die eine gewisse Ausdehnung hat. Bei der Lochkamera z.B. wird diese durch die Größe des Loches bestimmt, bei der Röntgenkamera durch die Ausdehnung des Strahlers. Selbst perfekte optische Systeme weisen immer mehr oder weniger ausgeprägte Beugungsmuster auf (natürliche Unschärfe) [Hec99]. Auch das Abbildungsverhalten des *confocal scanning microscope* (siehe Kap. 2.1.2) läßt sich durch seine dreidimensionale Punktantwort charakterisieren.

Wenn die Degradierung des Bildes sich durch ein linearen verschiebungsinvarianten Filter beschreiben läßt, kann man das unverfälschte Bild des Objektes mit Methoden der **Fourieroptik** schätzen. Das Buch *Digital Image Restoration* von B.R. Hunt [AH97] ist eine sehr gute Einführung in dieses Gebiet.

### 2.5.2 Dünn besetzte Matrizen

Die Degradierung des Bildes  $g$  wird durch die Faltung der Objektfunktion  $f$  mit der Punktantwort  $h$  des bilderzeugenden Systems beschrieben:

$$g = f \otimes h \quad (2.2)$$

Diese Faltung läßt sich auch mit Vektoren und (dünn besetzten) Matrizen darstellen. Dabei werden das Bild  $g$  und die Objektfunktion  $f$ , die **zunächst** der Einfachheit halber **eindimensional** seien, durch die Vektoren  $\vec{g}$  und  $\vec{f}$  dargestellt. Der Filter  $h$  wird durch die Matrix  $H$  dargestellt. Dabei werden die Zeilen von  $H$  aus Elementen von  $h$  gebildet, so daß  $H$  konstante Diagonalen hat ( $H$  hat *Toeplitz*-Form).

Die Faltung wird dabei zu einer Matrizenmultiplikation. Die Formel (2.2) läßt sich dann folgendermaßen schreiben:

$$\vec{g} = H\vec{f} \quad (2.3)$$

Wenn  $\vec{g}$  und  $H$  gegeben sind, läßt sich  $\vec{f}$  mit

$$\vec{f} = H^{-1}\vec{g} \quad (2.4)$$

berechnen, vorausgesetzt  $H^{-1}$  existiert. (2.3) bzw. (2.4) nehmen besonders einfache Gestalt an, wenn  $H$  eine Diagonalmatrix ist. In diesem Fall stellt die Multiplikation mit  $H$  bzw.  $H^{-1}$  eine elementweise Multiplikation bzw. Division mit den Diagonalelementen dieser Matrix dar. Es lohnt sich daher, die Eigenraumtransformation von  $H$  zu betrachten:  $H$  ist annäherungsweise zirkulant, da die von der Hauptdiagonalen entfernten Elemente annähernd Null sind (siehe auch Anhang A.1.2). Alle zirkulanten Matrizen lassen sich aber mit Fourier-Kern Matrizen  $\mathcal{F}$  in den Eigenraum transformieren (vgl. Anhang A.1.3 und Formel (A.3)):

$$\begin{aligned} Q_H &= \mathcal{F}^{-1}H\mathcal{F} \\ &= \text{diag}(\Delta_1, \Delta_2, \dots, \Delta_N) \end{aligned}$$

Die Elemente  $\Delta_1, \Delta_2, \dots, \Delta_N$  der Hauptdiagonalen von  $Q_H$  stellen die diskrete Fouriertransformation von  $h$  dar (siehe Formel (A.2)). Transformiert man Gleichung (2.4) in den Eigenraum von  $H$ , so erhält man mit  $F = \mathcal{F}^{-1}\vec{f}$  und  $G = \mathcal{F}^{-1}\vec{g}$ :

$$F = Q_H^{-1}G \quad (2.5)$$

Da  $F$  und  $G$  die Elemente der diskreten Fouriertransformationen  $\mathbf{F}$  und  $\mathbf{G}$  von  $\vec{f}$  und  $\vec{g}$  enthalten und  $Q_H$  eine Diagonalmatrix ist, lautet die entsprechende Gleichung im (diskreten) Fourierbereich:

$$\mathbf{F}_i = \mathbf{G}_i / \Delta_i; \quad i \in \{0, 1, \dots, N-1\} \quad (2.6)$$

Die hier aufgezeigten Parallelen von Matrizenmultiplikation, Faltung und Multiplikation im Fourierbereich werden im nächsten Kapitel gebraucht werden.

### 2.5.3 Maximum-Likelihood Schätzer

Da  $H$  i.a. einige Eigenwerte hat, die annähernd Null sind, ist Formel (2.6) anfällig gegenüber kleinen Fehlern von  $\vec{g}$ . Eine bessere Lösung ist daher, davon auszugehen, daß  $\vec{g}$  nicht fehlerfrei gemessen werden kann. Der Fehler wird dabei durch den Vektor  $\vec{n}$  modelliert (Rauschen):

$$\vec{g} = H\vec{f} + \vec{n}$$

$\vec{f}$  läßt sich einfach mit einem Maximum-Likelihood Schätzer bestimmen, wenn man die Gaußverteilungsannahme mit Formel (2.7) macht. Dabei bezeichnet  $N$  die Dimension der Vektoren. Die Kovarianzmatrix  $\mathcal{C}$  setzt sich aus den Kovarianzmatrizen  $\mathcal{C}_f$  und  $\mathcal{C}_n$  zu  $\vec{f}$  und  $\vec{n}$  zusammen ( $\vec{f}$  und  $\vec{n}$  seien statistisch unabhängig voneinander), wie in Gleichung (2.8) angegeben ist. Der Einfachheit halber wurde außerdem angenommen, daß neben  $\vec{n}$  auch  $\vec{f}$  o.B.d.A. mittelwertfrei ist.

$$p\left(\begin{pmatrix} \vec{f} \\ \vec{n} \end{pmatrix}\right) = \frac{1}{(2\pi)^N \sqrt{|\mathcal{C}|}} e^{((\vec{f}^\top \vec{n}^\top) \mathcal{C}^{-1} (\vec{f}^\top \vec{n}^\top)^\top)} \quad (2.7)$$

$$\mathcal{C} = \begin{pmatrix} \mathcal{C}_f & \mathcal{O} \\ \mathcal{O} & \mathcal{C}_n \end{pmatrix} \quad (2.8)$$

Ersetzt man in Gleichung (2.7)  $\vec{n} = \vec{g} - H\vec{f}$  und wendet den Logarithmus an, so erhält man die Log-Likelihood Funktion:

$$LL(\vec{f}) = -K \begin{pmatrix} \vec{f} \\ \vec{g} - H\vec{f} \end{pmatrix}^\top \begin{pmatrix} \mathcal{C}_f & \mathcal{O} \\ \mathcal{O} & \mathcal{C}_n \end{pmatrix}^{-1} \begin{pmatrix} \vec{f} \\ \vec{g} - H\vec{f} \end{pmatrix} \quad (2.9)$$

$$= -K [\vec{f}^\top \mathcal{C}_f^{-1} \vec{f} + (\vec{g} - H\vec{f})^\top \mathcal{C}_n^{-1} (\vec{g} - H\vec{f})] \quad (2.10)$$

Dabei ist  $K$  ein von  $\vec{f}$  unabhängiger Faktor. Wenn man  $\mathcal{C}_f$  und  $\mathcal{C}_n$  auch durch zirkulante Matrizen annähert, kann man alle Matrizen in Gleichung (2.10) durch Multiplikation mit Fourier-Kern Matrizen in Diagonalgestalt bringen und dann die Gleichung im Fourierbereich ausdrücken:

$$LL(\mathbf{F}) = -K \sum_{i=0}^{N-1} \left[ \frac{\mathbf{F}_i^* \mathbf{F}_i}{\mathbf{S}_{f_i}} + \frac{1}{\mathbf{S}_{n_i}} (\mathbf{G}_i - \Delta_i \mathbf{F}_i)^* (\mathbf{G}_i - \Delta_i \mathbf{F}_i) \right]$$

( $\mathbf{S}_f$  und  $\mathbf{S}_n$  sind die (diskreten) Fouriertransformierten von  $\mathcal{C}_f$  und  $\mathcal{C}_n$ ).  $LL(\mathbf{F})$  wird durch

$$\hat{\mathbf{F}}_i = \frac{\Delta_i^* \mathbf{G}_i}{|\Delta_i|^2 + \mathbf{S}_{n_i} / \mathbf{S}_{f_i}}; \quad i \in \{0, 1, \dots, N-1\} \quad (2.11)$$

maximiert. Man beachte, daß für  $\forall i \in \{0, 1, \dots, N-1\} : \mathbf{S}_{ni} = 0$  ( $\hat{=}$ kein Rauschen) Gleichung (2.11) sich wieder zu Gleichung (2.6) vereinfacht. Des weiteren ist wegen der hier angewendeten Gaußverteilungsannahme der Maximum-Likelihood-Schätzer auch gleich dem Wiener-Filter (bester linearer erwartungstreuer Schätzer)[Sti00].

Im Buch [AH97] werden auch noch andere Möglichkeiten beschrieben, das Rauschen zu unterdrücken. Da der Wiener-Filter dazu neigt Strukturen zu verischen, empfiehlt es sich oft, andere schwache Zwänge<sup>6</sup> in Betracht zu ziehen.

Die Vorgehensweise läßt sich auch auf zwei- oder dreidimensionale Probleme übertragen. Statt zirkulanter Matrizen werden dann Block-zirkulante Matrizen[AH97] eingesetzt; statt Fourier-Kern Matrizen werden Block-Fourier-Kern Matrizen verwendet.

### 2.5.4 Blinde inverse Faltung

Die Schätzung der Objektfunktion mit Formel (2.11) setzt die Kenntnis der Punktantwort  $\Delta$  voraus. Es gibt jedoch Algorithmen, die auch ohne gegebene Punktantwort auskommen. Die Annahme, daß das Bild des Objekt durch ein lineares verschiebungsinvariantes System degradiert ist, reicht nämlich u.U. schon aus, um die Objektfunktion zu schätzen. Wenn man zusätzlich noch bestimmte Randbedingungen für das Aussehen der Punktantwort vorgibt, kann man einen stabilen iterativen Algorithmus entwerfen, der die Objektfunktion **und** die Punktantwort schätzt. Algorithmen dieser Art fallen unter die Überschrift *blinde* inverse Faltung<sup>7</sup>, da die Punktantwort “blind” (ohne Kenntnis der Objektfunktion) geschätzt wird.

[KL91] ist ein (späterer) Artikel von A.K. Katsaggelos und K.T. Lay, in dem das Problem der blinden inversen Faltung formuliert wird. Ähnlich wie mit Gleichung (2.7) werden die Objektfunktion  $\vec{f}$  und das Kamerarauschen  $\vec{n}$  als Ergebnisse eines gauß’schen Zufallsprozesses aufgefaßt. Dabei ist hier allerdings auch die Punktantwort ein Parameter.

$$\vec{g} = H\vec{f} + \vec{n}, \vec{f} \sim N(\vec{m}_f, C_f), \vec{n} \sim N(0, C_n)$$

Eine einfache Maximierung mit der Log-Likelihood Funktion ist hier nicht mehr möglich, da keine geschlossene Lösung für die Lage des Maximums im Parameterraum gefunden werden kann. Ein iterativer Algorithmus ist daher erforderlich.

In diesem speziellen Fall (Maximum-Likelihood Ansatz und keine geschlossene Lösung) bietet sich der sogenannte *EM*<sup>8</sup>-Algorithmus an. Die im Anhang A.1.4 angegebenen Eingangsgrößen für den Algorithmus werden folgendermaßen gewählt:

- Der “unvollständige” Datensatz ist das Bild  $\vec{g}$ .
- Als “vollständiger” Datensatz  $\vec{z}$  wurde in [KL91] das Bild  $\vec{g}$  zusammen mit der unbekanntem Objektfunktion  $\vec{f}$  genommen ( $\vec{z} := \begin{pmatrix} \vec{f} \\ \vec{g} \end{pmatrix}$ ). Die Projektionsfunktion ist dann einfach  $t(\vec{z}) = \vec{g}$ .

<sup>6</sup>engl.: soft constraints

<sup>7</sup>engl.: blind deconvolution

<sup>8</sup>expectation maximization

- Als Verteilung wurde eine multivariate Gaußverteilung angenommen:

$$p_z(\vec{z}; \theta) = \frac{1}{\sqrt{(2\pi)^{2N^2} \mathcal{C}}} e^{-\frac{1}{2} \vec{z}^\top \mathcal{C}^{-1} \vec{z}}$$

Die Gaußverteilung sei der Einfachheit halber o.B.d.A. mittelwertfrei.

Wie in Kap. 2.5.3 sind  $\mathcal{C}_f$  und  $\mathcal{C}_n$  die Kovarianzmatrizen zu  $\vec{f}$  und  $\vec{n}$ . Mit dem Modell  $\vec{g} = H\vec{f} + \vec{n}$  erhält man dann die aus vier Untermatrizen zusammengesetzte Kovarianzmatrix  $\mathcal{C}$ :

$$\mathcal{C} = \begin{pmatrix} \mathcal{C}_f & \mathcal{C}_f H^\top \\ H \mathcal{C}_f & H \mathcal{C}_f H^\top + \mathcal{C}_n \end{pmatrix} \quad (2.12)$$

Der Parametersatz ist also  $\theta = \{\mathcal{C}_f, \mathcal{C}_n, H\}$ .

- Die bedingte Verteilungsdichtefunktion  $p(\vec{z}|\vec{g}; \theta)$  ist eine bedingte Normalverteilung (siehe Formeln (A.9) und (A.10)).

Für den **E-Schritt** des *EM*-Algorithmus wird zunächst  $\ln p_z(\vec{z}; \theta)$  berechnet. Um den Erwartungswert für gegebenes  $\vec{g}$  zu erhalten, reicht es dann, in dieser Formel die Vorkommen von  $\vec{f}$  und  $\vec{f}\vec{f}^\top$  durch ihre Erwartungswerte zu ersetzen (siehe hierzu Anhang A.1.5).

Im **M-Schritt** wird  $U(\theta, \theta_p) = E_z\{\ln p_z(\vec{z}; \theta)|\vec{g}; \theta_p\}$  dann optimiert:

$$\theta_{p+1} = \operatorname{argmax}_\theta U(\theta, \theta_p)$$

Durch Ableiten und Null-Setzen der Fouriertransformierten(!) von  $U(\theta, \theta_p)$  (Formel (A.11)) erhält man die gesuchte Iterationsvorschrift im Fourierbereich:

$$\begin{aligned} \mathbf{S}_f^{(p+1)}(k, l) &= \mathbf{S}_{f|g}^{(p)}(k, l) + \frac{1}{N^2} |\mathbf{M}_{f|g}^{(p)}(k, l)|^2 \\ \Delta^{(p+1)}(k, l) &= \frac{1}{N^2} \frac{\mathbf{G}(k, l) (\mathbf{M}_{f|g}^{(p)})^*(k, l)}{\mathbf{S}_{f|g}^{(p)}(k, l) + \frac{1}{N^2} |\mathbf{M}_{f|g}^{(p)}(k, l)|^2} \\ \sigma_n^2 &= \frac{1}{N^2} \sum_{kl} \left\{ |\Delta^{(p+1)}(k, l)|^2 \left( \mathbf{S}_{f|g}^{(p)}(k, l) + \frac{1}{N^2} |\mathbf{M}_{f|g}^{(p)}(k, l)|^2 \right) \right. \\ &\quad \left. + \frac{1}{N^2} (|\mathbf{G}(k, l)|^2 - 2\operatorname{Re}[\mathbf{G}^*(k, l) \Delta^{(p+1)}(k, l) \mathbf{M}_{f|g}^{(p)}(k, l)]) \right\} \end{aligned} \quad (2.13)$$

Dabei sind  $\mathbf{M}_{f|g}^{(p)}$  und  $\mathbf{S}_{f|g}^{(p)}$  die Fouriertransformierten zum bedingten Mittelwert  $\vec{m}_{f|g}$  und zur bedingten Kovarianz  $\mathcal{C}_{f|g}$  (Formeln (A.9) und (A.10)).  $\vec{m}_{f|g}^{(p)}$  ist das geschätzte Bild.

In [KL91] finden sich zwei Beispielphotos, an denen die Rekonstruktion mit dem Algorithmus demonstriert wird. Diese Methode wird auch im dreidimensionalen angewandt, um die 3-D Daten von konfokalen Laserscanningmikroskopen aufzubereiten. In [MC99] wird der *EM*-Algorithmus in Kombination mit numerischer Optimierung auf 3-D Daten angewandt.

Der Algorithmus in der obigen Form schätzt sowohl die  $N^2$  (im zweidimensionalen Fall) Elemente der Punktantwort als auch die  $N^2$  Werte der Kovarianzmatrix und ist damit stark überparametrisiert. Um ein stabiles Verhalten zu erreichen, muß die Punktantwort parametrisiert werden. I.a. wird nach jedem Schritt des *EM*-Algorithmus eine Regularisierung durchgeführt: Die Punktantwort  $\Delta^{(p)}$  wird angepaßt, so daß sie bestimmten Bedingungen bzw. einem parametrischen Modell genügt[KL91]:

- Achsensymmetrie
- begrenzte Ausdehnung
- Integral über Punktantwort gleich 1.0
- Alle Werte  $\geq 0.0$

### 2.5.5 Anwendbarkeit

Im Rahmen dieser Diplomarbeit wurde versucht, dreidimensionale inverse Faltung auf Fokussereien anzuwenden. Dazu wurde der *EM*-Algorithmus mit 2-D Bildern getestet. Zur Berechnung der diskreten Fourier-Transformationen wurde die *fftw*<sup>9</sup>-Bibliothek verwendet, die unter [FJ] frei verfügbar ist.

Abbildung 2.4 zeigt ein Bild und das entsprechende, mit einem Gauß-Filter degradierte Testbild. Mit dem iterativen Algorithmus (2.13) wurde dann das Originalbild geschätzt. Die Schätzungen des Originalbildes  $\vec{m}_{fg}^{(p)}$  nach den ersten 6 Iterationsschritten sind in Abb. 2.5 zu sehen. Nach wenigen Schritten erreicht der Algorithmus sein bestes Ergebnis. Danach entstehen Artefakte. Dies liegt daran, daß die Ausdehnung der Punktantwort nicht begrenzt wurde (vgl. Abb. 2.6). Alle anderen oben erwähnten Regularisierungen wurden angewandt.



Abbildung 2.4: Originalbild und mit LSI-Filter degradiertes Testbild

Die Experimente haben gezeigt, daß dieser iterative Algorithmus bereits für 2-D Bilder sehr rechenintensiv ist. Die Komplexität der Fouriertransformation,

<sup>9</sup>Fastest Fourier Transform in the West



Abbildung 2.5: Iterative Restauration mit dem EM-Algorithmus

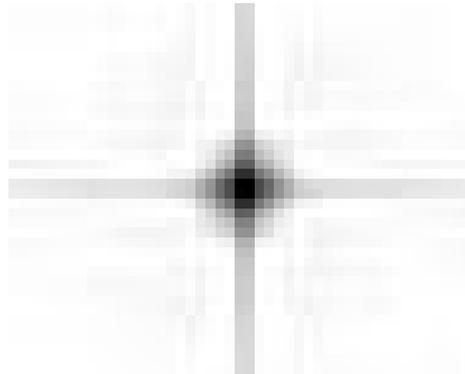


Abbildung 2.6: Die geschätzte Punktantwort des EM-Algorithmus

die die Komplexität des Algorithmus bestimmt, beträgt  $O(N^2 \log(N^2))$ ; d.h. der Aufwand wächst überproportional mit der Zahl der Bildpixel.

Des weiteren ist es nötig, die Lösung für die Punktantwort möglichst weit einzuschränken, um einen stabilen Algorithmus zu erhalten. Die Randbereiche des Bildes weisen größere Fehler auf, da die LSI<sup>10</sup>-Filterung durch eine zyklische Faltung angenähert wird.

Blinde inverse Faltung wird zwar bereits erfolgreich zur Aufbereitung der 3-D Daten von konfokalen Laserscanningmikroskopen (siehe Kap. 2.1.2) verwendet. Bei Fokussierien ist das Problem allerdings, daß die Punktantwort ein Doppelkegel und in Z-Richtung nicht beschränkt ist! Die Annäherung durch zyklische Faltung führt dann zu größeren Fehlern (Das gilt auch für den Wiener-Filter, der bei bekannter/kalibrierter Punktantwort anwendbar ist).

Um stabiles Verhalten zu erreichen, müßte man zudem einen Algorithmus finden, der die Lösung auf eine Oberfläche einschränkt.

*Inverse Filterung* ist nicht anwendbar, solange keine Lösung für diese Probleme besteht.

---

<sup>10</sup>linear shift invariant, deutsch: linear verschiebungsinvariant

## Kapitel 3

# Neuer adaptiver Algorithmus

### 3.1 Maximierung eines Schärfemaßes

Im Kapitel 2.2.1 wurde bereits auf die Maximierung von Schärfemaßen eingegangen. Um die Anwendbarkeit auf Fokussereien allgemeiner Art zu evaluieren, wurde mit dem Schärfemaß *lokaler Strukturgehalt* die Oberfläche eines Mikrogitters geschätzt (Abb. 3.1 zeigt das fokussierte Bild des Mikrogitters). Die Höhenwerte der so erhaltenen Höhenkarte wurden anschließend noch einmal mit einem  $15 \times 15$  Gauß-Filter gefiltert, um das starke Rauschen zu entfernen. Die so geglättete Oberfläche ist in der 3-D Ansicht 3.2 dargestellt.

Im Idealfall wäre die Oberfläche plan. Auch die Verwendung anderer Schärfemaße wie z.B. der *lokalen Grauwertvarianz* führt zu ähnlich schlechten Ergebnissen.

Außerhalb des Mikrogitters ist die Höhenkarte sehr instabil, da dort fast keine Oberflächenmerkmale vorhanden sind. Dies ist ein Problem, daß allgemein viele Verfahren haben, die auf Schärfe/Unschärfe der Bilder basieren: Das unscharfe Bild einer texturarmen eintönigen Region unterscheidet sich kaum vom dem entsprechenden fokussierten Bild.

Ein weiteres Problem offenbart sich aber, wenn man sich die Rekonstruktion genauer anschaut. In Abb. 3.4 erkennt man, daß selbst in Bildbereichen, die über Struktur verfügen, Fehler auftreten. Abbildung 3.3 zeigt einen Schnitt durch die Fokusserei des Mikrogitters, in dem die Höhenlinie rot eingezeichnet ist. Es ist deutlich zu erkennen, daß der Fehler systematischer Art ist: Statt den am besten fokussierten Bildbereich auszuwählen, zieht das lokale Schärfemaß (hier: *lokaler Strukturgehalt*) die "Kanten" im dreidimensionalen Datensatz vor.

Die systematischen Fehler haben sich in der in [Rön96] als Beispiel verwendeten Fokusserei z.B. aber kaum bemerkbar gemacht. Dies liegt allerdings daran, daß die Helligkeitsfunktion der Oberflächen überall hohe Ortsfrequenzen aufwies. Das ist aber im allgemeinen nicht der Fall.

Wenn man z.B. Gitter auf das Objekt projiziert, um das Spektrum der Bilder zu verbessern, stellen diese systematischen Fehler ein großes Problem dar. Die Größe von Gitter und Filter lassen sich zwar aufeinander abstimmen (vgl. [NN94b]); da die Gittergröße aber wegen Beugungseffekten und Aliasing-

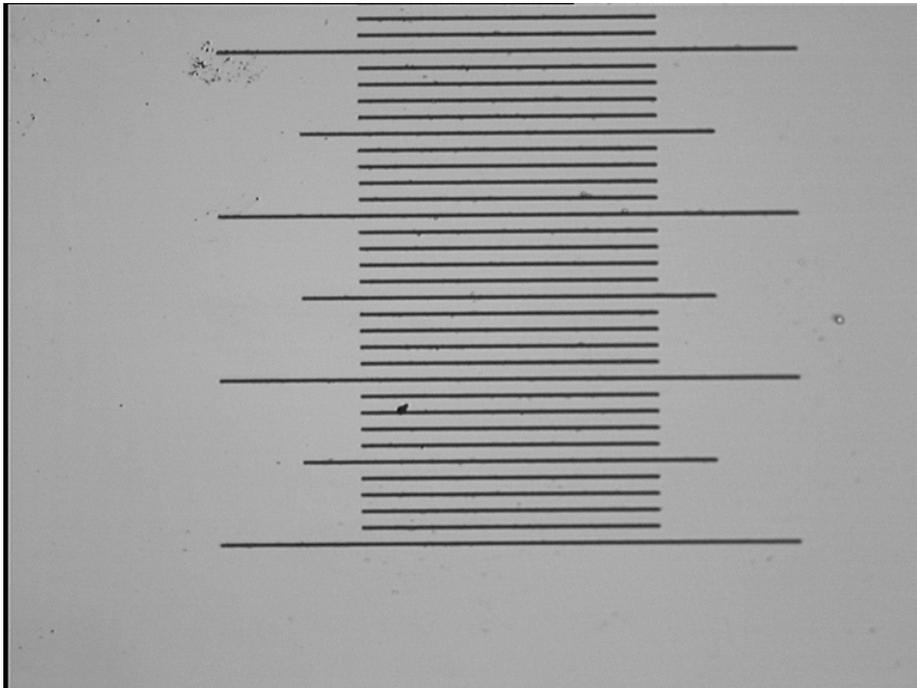


Abbildung 3.1: Oberfläche des Kalibrierungsgitters im Fokus (Spaltbreite  $10\ \mu\text{m}$ )

Effekten nach unten begrenzt ist, wird man dann bestimmte Objektdetails nicht mehr auflösen können.

In Kapitel 3 wird auf die Korrektur des systematischen Fehlers eingegangen. Im nächsten Kapitel geht es zunächst um das Einbringen von zusätzlicher Struktur durch Projektion von Hilfsmustern.

## 3.2 Projektion von Hilfsmustern

Man stelle sich eine Oberfläche vor, die das Licht an jeder Stelle gleichmäßig streut. Wenn keine Zustrahlung oder Abschattung durch Unstetigkeiten in der Oberfläche auftreten, so sieht der Beobachter des Mikroskopbildes eine strukturlose Oberfläche, ungeachtet welche Fokalebene gewählt wurde.

Selbst unter günstigeren Umständen finden sich in den meisten Fokuserien Bildregionen, in denen solche ungünstigen Umstände auftreten. Statt sich jedoch auf genügende Oberflächentextur der Objekte zu verlassen, kann man (sofern man ein geeignetes Mikroskop hat), Lichtmuster auf das Objekt projizieren.

Die Optik der Zwei-Kanal-Beleuchtung des verwendeten Mikroskops (Kap. 1.2) ermöglicht es, ein Filter mit einem Muster einzusetzen, welches dann auf das Objekt projiziert wird. Der Strahlengang eines Mikroskops ist in Abb. 3.5 dargestellt. Beim *Leica*-Mikroskop wird im Bereich des Okulars über ein Prisma das Licht für die Auflichtbeleuchtung in den Strahlengang gespiegelt. Die Auflichtbeleuchtung verfügt über einen eigenen Strahlengang, in dem das zu projizierende Muster eingesetzt werden kann. Die Filter wurden einfach durch

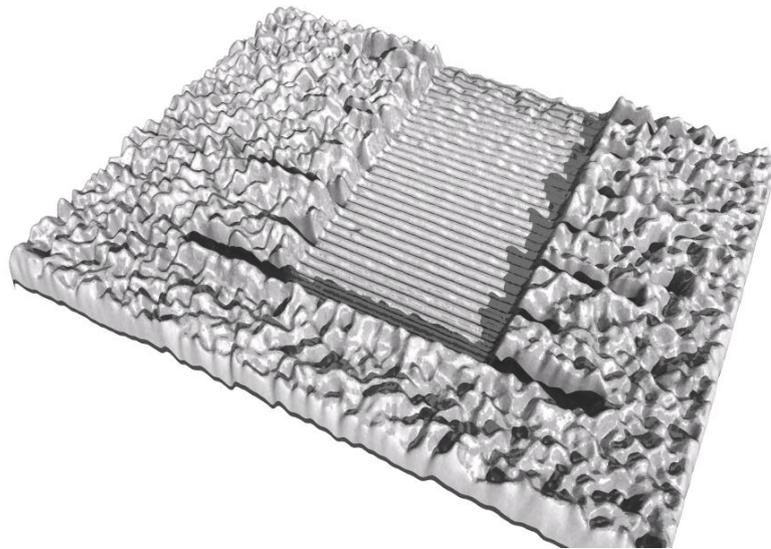


Abbildung 3.2: Geglättete und texturierte Höhenkarte aus der Maximierung des lokalen Strukturgehaltes

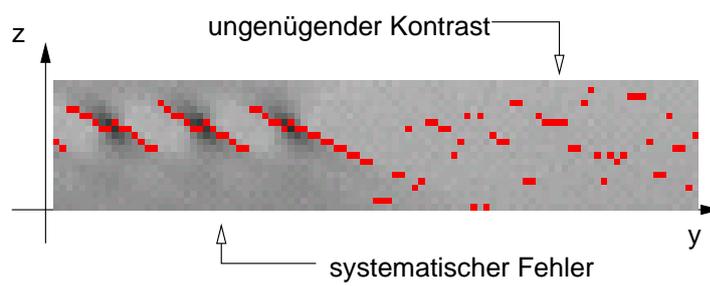


Abbildung 3.3: Systematischer Fehler der Höhenkarte

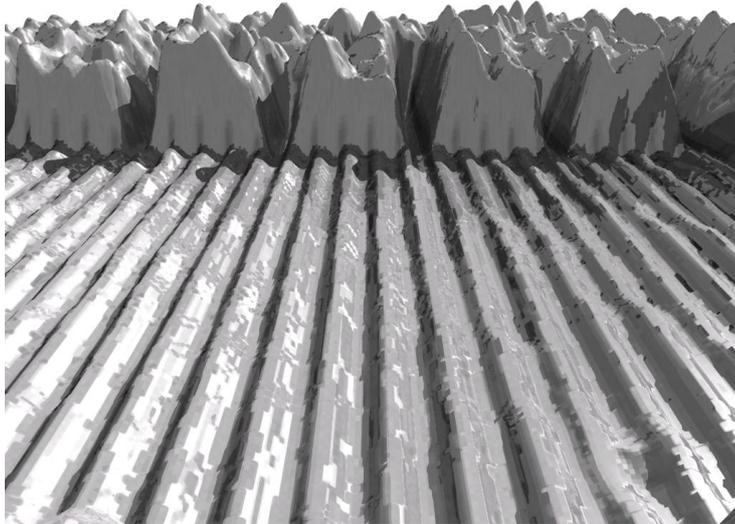


Abbildung 3.4: Systematischer Fehler bei der Maximierung des lokalen Strukturgehaltes

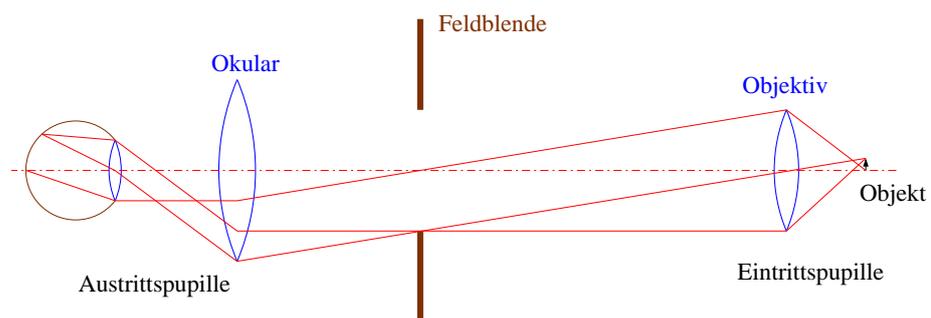
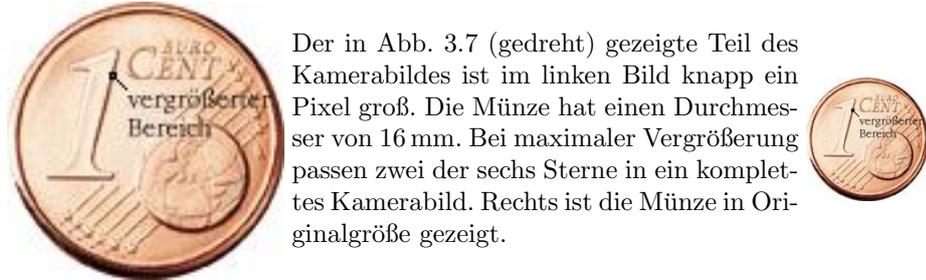


Abbildung 3.5: Strahlengang im Mikroskop[Hec99] (vereinfacht). Es ist keine Beleuchtungsvorrichtung eingezeichnet.

Bedrucken einer Folie mit einem Laserdrucker hergestellt. Auf diese Art wurden ein Gitter und ein Schachbrettmuster hergestellt, wobei das Schachbrettmuster vorzuziehen ist, da es nicht nur Ortsfrequenzen in einer Richtung erzeugt.



Der in Abb. 3.7 (gedreht) gezeigte Teil des Kamerabildes ist im linken Bild knapp ein Pixel groß. Die Münze hat einen Durchmesser von 16 mm. Bei maximaler Vergrößerung passen zwei der sechs Sterne in ein komplettes Kamerabild. Rechts ist die Münze in Originalgröße gezeigt.

Abbildung 3.6: 1-Cent Münze[ECB02]

Abbildung 3.7 zeigt die Tiefenbilder eines Teils einer 1-Cent Münze (siehe Abb. 3.6) einmal ohne und einmal mit projiziertem Schachbrettmuster. Die Felder der Projektion des Schachbrettmusters haben eine Seitenlänge von ca.  $13 \mu\text{m}$ . Die Höhenkarten sind in Abbildung 3.8 gezeigt. Die Höhenkarten wurden durch Maximierung der lokalen Grauwertvarianz (Filtergröße  $3 \times 3$ ) erzeugt.

Man kann deutlich sehen, daß durch die Verwendung des Hilfsmusters die Zahl der Ausreißer zurückgeht. Des weiteren wird die Höhenkarte wesentlich ruhiger. In Abbildung 3.8 kann man jedoch rechts sehen, daß die Höhenkarte Artefakte aufweist, die vom verwendeten Schachbrettmuster herrühren. Die Erklärung hierfür ist, daß systematische Fehler an den Rändern der Felder des Schachbrettmusters auftreten (siehe Kap. 3.1).

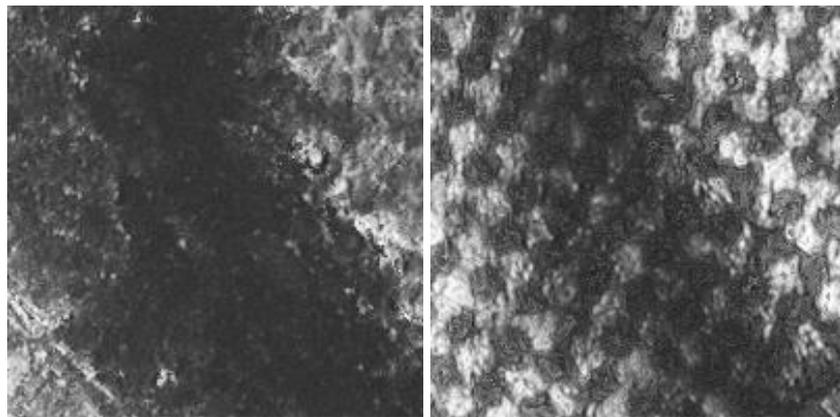


Abbildung 3.7: Tiefenbild des vergrößerten Bereichs ( $148 \mu\text{m} \times 148 \mu\text{m}$ ) der 1-Cent Münze ohne und mit projiziertem Schachbrettmuster

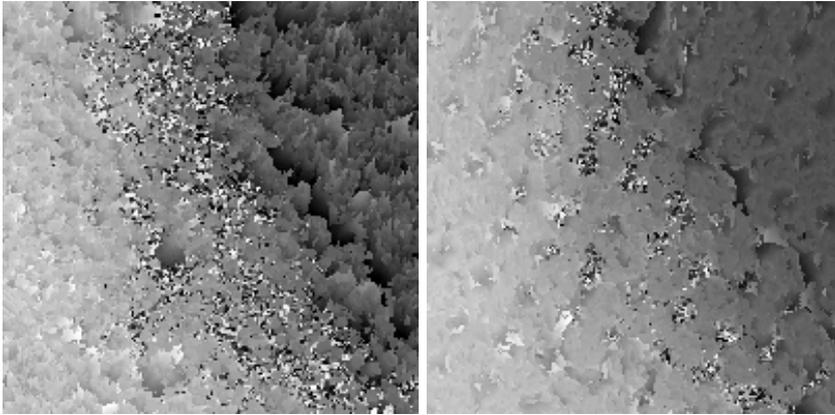


Abbildung 3.8: Höhenkarten erzeugt aus Fokussiererei ohne Hilfsmuster (links) und aus Fokussiererei mit projiziertem Schachbrettmuster (rechts)

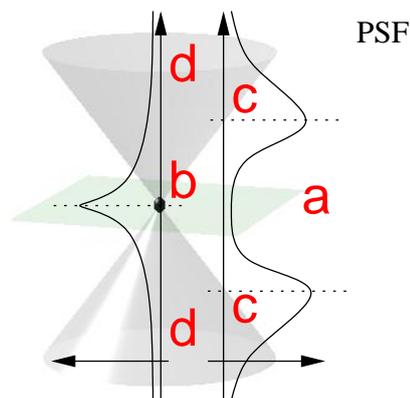


Abbildung 3.9: Qualitativer Verlauf des Schärfemaßes bei der Fokussiererei eines Punktes.

### 3.3 Systematischer Fehler

In Kapitel 3.1 wurde bereits auf die systematischen Fehler hingewiesen, die bei einfacher Maximierung mit einem Schärfemaß auftreten.

Abbildung 3.9 verdeutlicht, wie die systematischen Fehler entstehen. Die Abbildung zeigt schematisch das dreidimensionale Aussehen einer Fokussiererei einer kleinen Kugel (b). Der linke Graph zeigt schematisch den Verlauf eines Schärfemaßes für das Pixel, welches zentral über der Abbildung der Kugel liegt. Hier erreicht das Schärfemaß das Maximum an der erwarteten Stelle, da in den Bereichen (d) die Kontraste des Bildes niedriger ausfallen als in (b).

Das gilt allerdings nicht für Pixel, die in der Nähe des Kugelabbildes liegen (rechter Graph). Im Bereich (a) ist der Kontrast Null, wenn dort keine weiteren Objekttexturen vorhanden sind. Daher treten an den Stellen (c) dann zwei Maxima auf, von denen eines letztendlich die Höhenkarte verunstaltet.

Im Buch [Sch97] wird u.a. dieses Problem angesprochen. Die in [Sch97] vorgeschlagene Lösung sieht folgendermaßen aus: Ein Gütemaß bewertet folgende drei Eigenschaften des Schärfemaximums:

- Krümmung der Schärfemaß-Kurve am Maximum bzw. Ausdehnung des Maximums
- Verhältnis Maximum zu Mittelwert
- Abweichung von den Nachbarfeldern in der Höhenkarte ( $z - \text{median}\{z\}$ )

Mithilfe dieser drei Kriterien wird beurteilt, ob das Maximum plausibel ist. Ggf. wird das Ergebnis zurückgewiesen und die Höhenkarte ist an dieser Stelle undefiniert.

Wenn man das Schärfemaß allerdings lokal mittelt, würde man nicht nur an dem Punkt (b) in Abb. 3.9, sondern auch in einer Umgebung dieses Punktes ein ausgeprägtes Maximum erhalten. Nach einer lokalen Mittelung würden allerdings u.U. Objektdetails verloren gehen. Das Problem besteht also in einer günstigen Wahl der Filtergröße.

### 3.4 Normierte Grauwertvarianz

Im letzten Kapitel wurde gezeigt, daß die Wahl der Filtergröße das Ergebnis der Maximierung des Schärfemaßes stark beeinflusst. Es empfiehlt sich daher, einen adaptiven Algorithmus zu implementieren, der die Filtergröße variiert. Der hier vorgestellte Algorithmus berechnet ein Schärfemaß für unterschiedliche Filtergrößen. Um Rechenzeit zu sparen, wird bei wachsender lokaler Mittelung auch gleichzeitig die Ortsauflösung gesenkt (s.u.). Als Schärfemaß wurde die normierte Grauwertvarianz (siehe unten S. 30) verwendet. Dazu muß für mehrere Bildauflösungen die Grauwertvarianz bestimmt werden.

Um die lokale Grauwertvarianz für die verschiedenen Auflösungsstufen zu berechnen, wird zunächst das jeweilige Bild der Fokusserie elementweise quadriert (siehe Abb. 3.10). Danach werden das Bild und die quadrierten Grauwerte mit dem Filter  $h$  lokal gemittelt.  $h$  wurde so gewählt, daß das mittlere Pixel betont wird. Außerdem ist  $h$  separierbar. Es sind natürlich auch andere Filter denkbar. Aus diesen beiden Feldern läßt sich die lokale Grauwertvarianz zu jedem Bildpixel nach der Formel (A.13) effizient berechnen.

Das Bild und die quadrierten Grauwerte werden dann mehrmals mit  $s$  gefiltert und die Auflösung der Felder wird reduziert. Für  $s$  wurde einfach der gleitende Mittelwert<sup>1</sup> gewählt; die  $6 \times 6$ -Filtermatrix ist in Bild 3.10 gezeigt. Die Filterung wird nur für je eins von vier Pixeln im Bild durchgeführt, so daß gleichzeitig die Auflösung reduziert wird. Analog zum ersten Schritt werden die Grauwertvarianzen berechnet. Somit erhält man mit moderatem Rechenaufwand eine Reihe von 3-D Datensätzen von Schärfewerten mit abnehmender x-y-Auflösung. Mit abnehmender x-y-Auflösung steigt gleichzeitig die Größe des lokalen Bereiches, auf den sich das Schärfemaß bezieht.

Da helle Objektstrukturen meistens eine höhere Varianz aufweisen als dunkle, wurde die lokale Grauwertvarianz durch den lokalen Mittelwert  $\bar{x}$  dividiert. Dieser Wert soll im folgenden (*lokale*) *normierte Grauwertvarianz* genannt werden.

<sup>1</sup>engl.: moving average (MA)

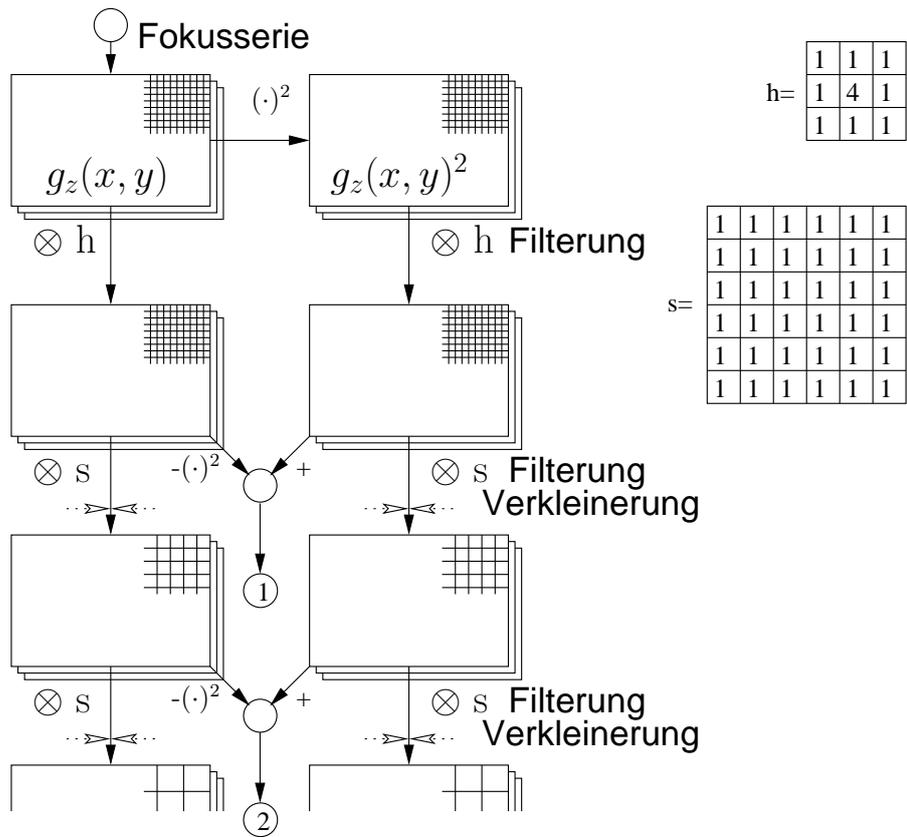


Abbildung 3.10: Datenfluß bei Berechnung der Standardabweichung für verschiedene Auflösungsstufen

### 3.5 Bestimmung der Höhenkarte

Die Bestimmung der Höhenkarte baut auf der Reihe von 3-D Datensätzen von Schärfewerten auf, die, wie im Kapitel 3.4 erklärt, berechnet werden können. Die im folgenden erklärte Bestimmung der Höhenkarte kann aber natürlich auch mit einem anderem Schärfemaß ausgeführt werden, wenn es in unterschiedlichen Auflösungen vorliegt.

Im ersten Schritt werden zu jeder Auflösungsstufe eine Höhenkarte und die Maxima durch einfache Maximierung bestimmt (siehe Abb. 3.11). Da das Schärfemaß zu jeder Auflösungsstufe vorliegt, erhält man dann eine Reihe von Höhenkarten mit abnehmender Auflösung, die fusioniert werden müssen.

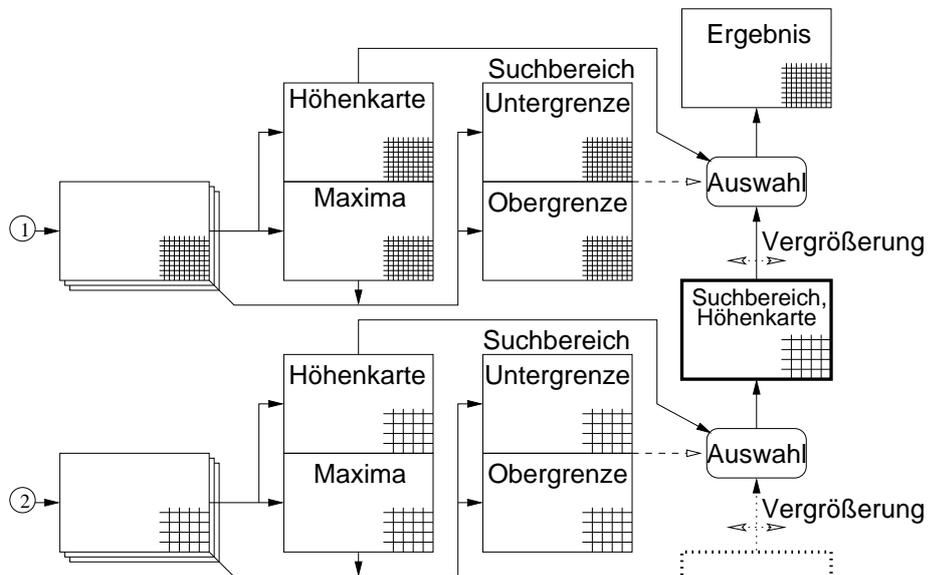


Abbildung 3.11: Datenfluß bei der Schätzung der Höhenkarte

Mit einem Faktor  $\gamma \in (0, 1]$ , der ein Parameter dieses Algorithmus darstellt, wird dann im zweiten Schritt zu jedem Pixel ein Intervall bestimmt, welches im folgenden *Suchbereich* genannt werden soll. Der Suchbereich wird, wie in Abb. 3.12 dargestellt, bestimmt: Der Verlauf des Schärfemaßes zu einem Pixel (einer bestimmten Auflösungsstufe) wird betrachtet und das minimale Intervall wird ermittelt, so daß der Wert "Maximalwert multipliziert mit  $\gamma$ " außerhalb dieses Intervalls nicht mehr überschritten wird.

Wenn der lokale Bereich, aus dem das Schärfemaß gebildet wird, zu klein wird, dann teilt sich das Maximum aus Abb. 3.12 in zwei Maxima auf, die auseinander wandern. Man erwartet dann, daß der Suchbereich wieder größer wird. Dieses Verhalten des Suchbereiches wird bei der im folgenden erklärten Fusion der Höhenkarte verwendet.

Mithilfe der so ermittelten *Suchbereiche* werden die Höhenkarten fusioniert: Beginnend mit dem Suchbereich und der Höhenkarte der größten Auflösung (3 Werte/Pixel) wird das gesuchte Ergebnis schrittweise verfeinert und gleichzeitig der Suchbereich verkleinert (siehe Abb. 3.11 dick eingerahmter Kasten). Dabei

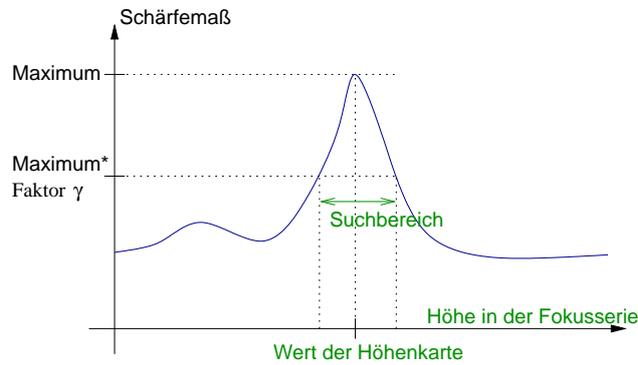


Abbildung 3.12: Definition des Suchbereiches

werden die Ergebnis-Höhenkarte und das Feld mit den Suchbereichen zunächst auf die nächsthöhere Auflösung vergrößert.

Von der feineren Höhenkarte werden dann in das Ergebnis-Feld nur die Pixel übernommen, bei denen der zugehörige Suchbereich in dem Suchbereich der niedrigeren Auflösung enthalten ist. Wird ein Pixel übernommen, wird auch der entsprechende Suchbereich mit übernommen. D.h. es werden immer nur Werte der feineren Höhenkarte übernommen, wenn gleichzeitig auch der Suchbereich verkleinert (oder zumindest beibehalten) werden kann.

Etwaige Ausreißer werden bei dieser Vorgehensweise wirksam unterdrückt. Außerdem tritt der systematische Fehler kaum noch auf, da in diesem Fall nicht eines der beiden Maxima genommen wird, die in der hohen Auflösung auftreten, weil der Suchbereich dann bereits in der niedrigeren Auflösung verkleinert wurde.

### 3.6 Komplexität des Algorithmus

Die Fokusserie habe die Dimensionen  $K \times L \times M$  und damit  $KLM = N$  Elemente.

Um ein  $K \times L$  Bild der Fokusserie zu filtern, wird für jeden Bildpunkt eine feste Anzahl an Operationen benötigt (selbst dann wenn man nicht-separierbare Filter verwendet). Der Aufwand, um ein Bild der Größe  $K \times L$  zu filtern, ist also  $O(KL)$ .

O.B.d.A. seien  $K$  und  $L$  im folgenden gleich einer Potenz von 2 ( $\exists p \in \mathbf{N}_0 : K = L = 2^p$ ). Die Anzahl der Operationen, um eine Schärfemaß-Pyramide zu einem Bild der Größe  $K \times L$  zu erstellen, ist dann proportional zu

$$\sum_{i=0}^{\infty} \frac{KL}{2^i 2^i} = KL \sum_{i=0}^{\infty} \left(\frac{1}{4}\right)^i = \frac{1}{1 - \frac{1}{4}} KL = \frac{4}{3} KL.$$

D.h. die Komplexität der Berechnung einer Schärfemaß-Pyramide ist auch in  $O(KL)$ .

Es werden  $M$  Pyramiden zweimal berechnet. Jede Pyramide wird zweimal berechnet, weil nicht alle Schärfewerte in den Hauptspeicher des Rechners pas-

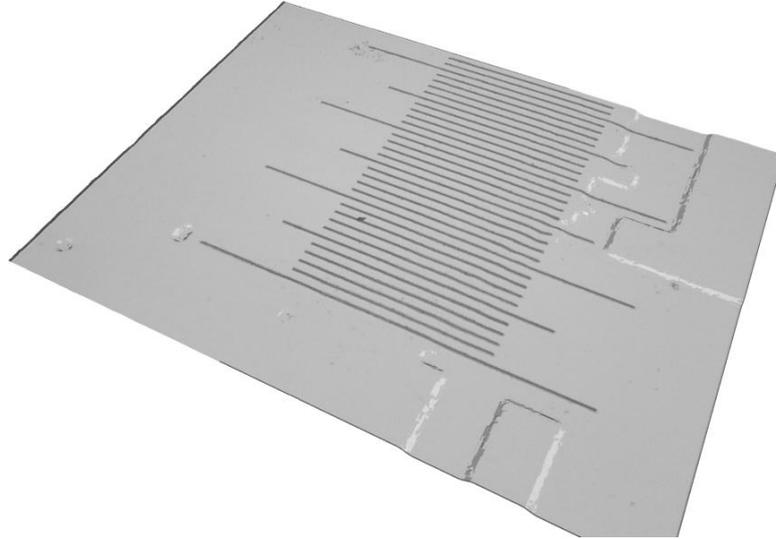


Abbildung 3.13: Geglättete Höhenkarte aus lokaler Grauwertvarianz mit adaptiver Filtergröße

sen. Dieser konstante Faktor ändert aber nichts an der Komplexität. Die Komplexität um alle Schärfewerte zu berechnen ist also  $O(N)$ .

Die Bestimmung der Maxima und der Suchbereiche kann mit einer konstanten Zahl an Operationen pro Pyramiden-Element bewerkstelligt werden (also Komplexität  $O(N)$ ). Die Bestimmung der Ergebnis-Pyramide hat analog zu der Berechnung einer Schärfemaß-Pyramide die Komplexität  $O(KL)$ .

Insgesamt ist die Komplexität also  $O(N)$ . Eine bessere Komplexität kann nicht mehr erzielt werden, ohne einen Teil der Daten der Fokussiererei zu ignorieren.

### 3.7 Vergleich mit nicht-adaptiver Filterung

Wendet man diesem adaptiven Algorithmus auf die Fokussiererei des Mikroglitters an, erhält man eine sehr stabile Höhenkarte. Die Werte der Höhenkarte wurden mit einem  $15 \times 15$  Gauß-Filter gefiltert. Die texturierte Höhenkarte ist in Abb. 3.13 zu sehen. Man vergleiche das Ergebnis mit Abb. 3.2! Der für die Bestimmung der Suchbereiche gewählte Faktor  $\gamma$  war hier 0.8 und es wurden 8 verschiedene Auflösungsstufen betrachtet. Die *Povray*-Datei mit der Abbildung 3.13 erstellt wurde, findet sich im Anhang A.2. Die Qualität der Ergebnisse wird im Kapitel 6 näher untersucht werden.

# Kapitel 4

## Auflösungsverbesserung

### 4.1 Zusätzlicher Vergleich der Unschärfe

Um die Quantisierung der Höhenwerte über die Auflösung der Fokussiererei hinaus zu verbessern, kann man durch Vergleich der Unschärfe (siehe Kap. 2.2.2) den Verlauf der Höhenkarte *zwischen* den Bildern der Fokussiererei abschätzen.

Das Bild eines Punktes wächst nach Artikel [ZZW00] proportional mit dem Abstand der Fokalebene zum Punkt. Wenn man das Bild des Punktes durch eine Kreisscheibe annähert, erhält man für die Punktantwort:

$$h_z(x, y) = \begin{cases} \frac{1}{\pi(zk)^2} & \sqrt{x^2 + y^2} \leq |zk| \\ 0 & \text{sonst} \end{cases}$$

Mit einem fokussierten und einem unscharfen Bild einer Ebene kann man die Punktantwort kalibrieren (siehe Kap. 5.2.4). Wenn der Abstand der Fokalebene des Bildpaares bekannt ist, kann man das Ergebnis der Kalibrierung auf Fokussiererei mit anderem Fokalebene-Abstand übertragen.

In Artikel [EL93] wurde ein einfaches Verfahren zur *Vergleich der Unschärfe*<sup>1</sup> vorgestellt (siehe Kap. 2.2.2). Dabei werden am Anfang eine Menge von verschiedenen  $h_{j\Delta z}$  ausgerechnet. Danach wird der zweite der beiden korrespondierenden Bildbereiche  $g_1$  und  $g_2$  ausgewählt und mit allen  $h_{j\Delta z}$  gefaltet. Das Ergebnis wird dann mit dem Bildbereich  $g_1$  verglichen. Umgekehrt wird versucht,  $g_1$  durch Faltung mit den verschiedenen  $h_{j\Delta z}$  auf  $g_2$  abzubilden.

Das  $h_{j\Delta z}$  mit dem das beste<sup>2</sup> Ergebnis erzielt wurde, gibt die Lage der Höhenkarte zwischen den Bildern an (siehe Formel (4.1)). Je nachdem ob  $g_1$  oder  $g_2$  degradiert wurde, befindet sich dann die Höhenkarte näher an der zu  $g_2$  oder  $g_1$  gehörigen Fokalebene.

Der Algorithmus dazu sieht folgendermaßen aus:

1. Die vom Algorithmus aus Kap. 3 bestimmte Position der Höhenkarte sei  $z = d(x, y)$  mit  $z \in \{0, \dots, N - 1\}$ . Dabei sei  $c$  das Verhältnis der Fokalebene-Abstände der aktuellen und der zur Kalibrierung verwendeten Fokussiererei. Dann werden die lokalen Bereiche  $g_1$  und  $g_2$  aus den Bil-

---

<sup>1</sup>engl.: depth from defocus

<sup>2</sup>Summe der Beträge der Grauwert-Differenzen minimal

dern mit den Fokalebene  $z - 1$  und  $z + 1$  zum Vergleich herangezogen, so daß die Höhenkarte *zwischen* diesen beiden Fokalebene verläuft.

2. Es gilt: wenn Höhenkarte lokal bei  $\approx z + 1$ :  $g_1 \approx g_2[\otimes]h_{-2/c}$   
 wenn Höhenkarte lokal bei  $\approx z - 1$ :  $g_2 \approx g_1[\otimes]h_{+2/c}$  . Mit  
 wenn Höhenkarte lokal bei  $\approx z$  :  $g_1 \approx g_2$

$$z' = z + \frac{1}{M} \operatorname{argmin}_{j \in \{-M, \dots, +M\}} \left\| \begin{cases} g_2 - g_1[\otimes]h_{2j/(cM)} & j < 0 \\ g_1 - g_2[\otimes]h_{2j/(cM)} & j \geq 0 \end{cases} \right\| \quad (4.1)$$

lassen sich die Quantisierungsstufen in  $z$ -Richtung in  $M$  feinere Stufen zerlegen. Die Faltung wird nach [EL93] direkt ausgeführt und die Größe des Bildbereiches des "schärferen" Bildes wird je nach Größe der Punktantwort  $h$  ausgedehnt, um die Randeffekte mit einzubeziehen; daher auch das Symbol  $[\otimes]$ . Abb. 4.1 verdeutlicht den Sachverhalt.

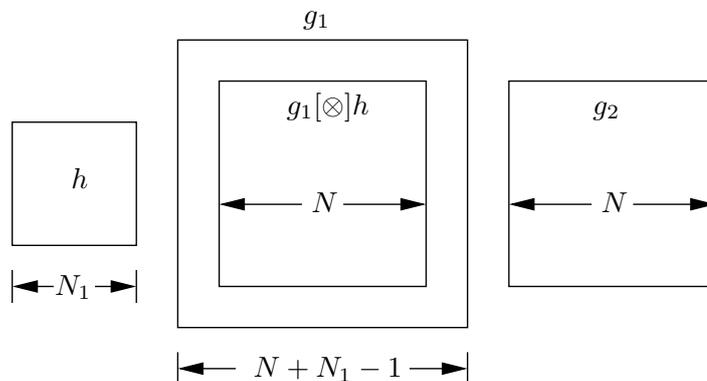


Abbildung 4.1: Unterschiede in der Größe der lokalen Bereiche [EL93]

Die Filtergröße des adaptiven Algorithmus wurde übernommen und bestimmt hier die Größe des zu einem Pixel betrachteten Bildbereiches, damit ausreichend Objektdetails garantiert sind. Die Berechnung wurde für alle Pixel durchgeführt, was sehr ineffizient ist. Allerdings konnte man hier schon sehen, daß die Ergebnisse relativ schlecht ausfielen, es sei den, das Bild des Objektes wies feinkörnige Strukturen auf.

Ein weiteres Problem stellte die Annahme, daß die Höhenkarte in einem lokalen Bereich plan ist, dar. Wenn sich die Höhe der Objektoberfläche innerhalb des lokalen Bereiches merklich ändert, wird das Verfahren instabil und auch anfällig gegenüber systematischen Fehlern. Dies liegt daran, daß der zu minimierende Term in Formel 4.1 in diesem Fall kein ausgeprägtes Minimum aufweist. Daher ist diese Art der Verfeinerung der Höhenkarte i.a. nicht gut anwendbar.

Da der Vergleich der Unschärfe auf der Annahme beruht, daß das Bild mit der Punktantwort  $h$  degradiert wird, dürfen auch keine Hilfsmuster auf das Objekt projiziert werden: Da das Hilfsmuster auf defokussierten Bereichen des Bildes unscharf projiziert und dann ein zweites mal degradiert wird, hat es eine eigene Punktantwort. D.h. das Hilfsmuster und die Texturen der Objektoberfläche werden durch unterschiedliche Filter degradiert!

## 4.2 Interpolation des Schärfemaßes

Bei der Maximierung wird nach Formel (2.1) der diskrete Wert  $\hat{z}$  gewählt, der das Schärfemaß  $S\{g_z\}(x, y)$  maximiert. Theoretisch liegt  $S\{g_z\}(x, y)$  in kontinuierlicher Form vor, d.h. auch für reelle Werte von  $z$ . In [Sch97] findet sich ein Algorithmus, der für festes  $x$  und  $y$  die nicht-quantisierte Lage  $\hat{z}'$  des Maximums von  $S\{g_z\}(x, y)$  abschätzt:

1. Bestimme diskretes  $\hat{z}$ , für das  $S\{g_z\}(x, y)$  maximal wird.
2. Berechne weitere Punkte der Schärfemaß-Kurve, die in der Nachbarschaft des Maximums liegen ( $(\hat{z} - 1, S\{g_{\hat{z}-1}\}(x, y))$ ,  $(\hat{z} + 1, S\{g_{\hat{z}+1}\}(x, y))$ ,  $(\hat{z} - 2, S\{g_{\hat{z}-2}\}(x, y))$ ,  $(\hat{z} + 2, S\{g_{\hat{z}+2}\}(x, y))$ , ...).
3. Approximiere den Verlauf der Schärfemaß-Kurve durch eine Regressions-Parabel (gestrichelte Kurve in Bild 4.2). Die Regressions-Parabel wird mit der Methode der kleinsten Quadrate bestimmt (siehe A.5.2).
4. Die Lage des Maximums der Regressionsparabel ist der nicht-quantisierte Schätzwert für  $\hat{z}'$ .

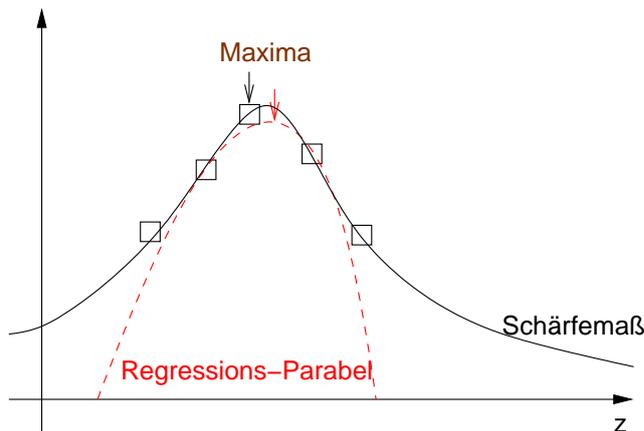


Abbildung 4.2: Diskretes Maximum und Maximum der Regressionsparabel

Diese Art der Auflösungsverbesserung läßt sich einfach mit dem adaptiven Algorithmus verbinden: Der adaptive Algorithmus wählt die Filtergröße des Schärfemaßes und den Wert für  $\hat{z}$ . Die Regressionsparabel wird dann für das ausgewählte Schärfemaß bestimmt. Die Rekonstruktion des Mikrogitters (Bild 3.13) wurde durch diesen Algorithmus verfeinert und ist in Abbildung 4.3 gezeigt. In Kapitel 6 werden die Ergebnisse evaluiert werden.

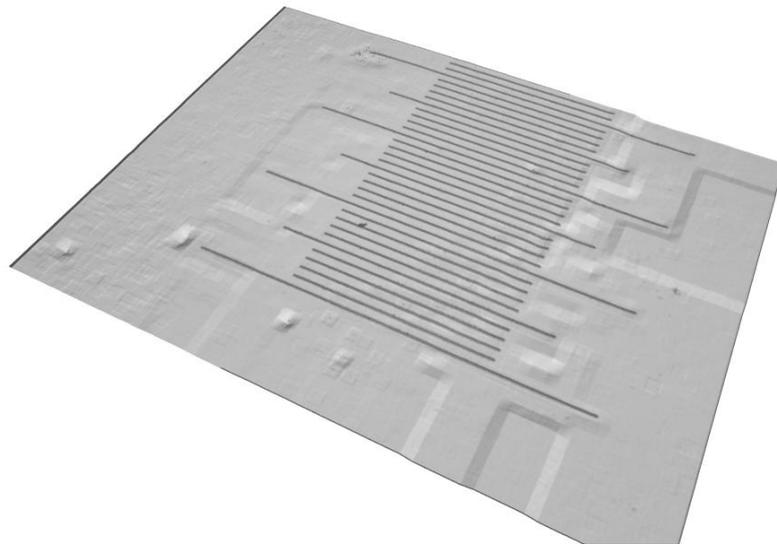


Abbildung 4.3: Die Rekonstruktion aus Bild 3.13 mit zusätzlicher Interpolation des Schärfemaßes

# Kapitel 5

## Implementierung

### 5.1 Verwendete Software

#### 5.1.1 Betriebssystem

Als Betriebssystem wurde *Linux* verwendet, da es unter diesem Betriebssystem eine Menge frei verfügbarer Bibliotheken und Übersetzer gibt. Das Programm wurde später auch nach *Windows NT* portiert.

#### 5.1.2 Übersetzer

Die Programme wurden in der Programmiersprache *C++* geschrieben und mit dem *GNU-C++*-Übersetzer übersetzt.

#### 5.1.3 Programmbibliotheken

Die Benutzeroberfläche wurde mit der *KDE/Qt*-Bibliothek und dem *Qt-designer* entwickelt (frei erhältlich unter [Qt]). Da es mittlerweile auch eine freie Version für *Windows* gibt, kann das Programm mit seiner Benutzeroberfläche auch unter *Windows* ausgeführt werden.

Zum Ausführen der für die Methode kleinster Quadrate benötigten Matrizen- und Vektorrechnung wurde die *newmat*-Bibliothek [Dav] verwendet, die frei für *Linux* und *Windows* verfügbar ist.

#### 5.1.4 Programmierwerkzeuge

Abb. 5.1 zeigt den Zusammenhang zwischen Quellcodes und sekundären Dateien. Teile der Benutzeroberfläche wurden mit dem *Qt-designer* erstellt. Dieses Werkzeug erlaubt es, die Benutzeroberfläche interaktiv zu entwerfen. Das Layout wird in einer *UI*-Datei gespeichert. Aus dieser Datei wird mit dem *UI*-Übersetzer ein *C++*-Quellcode erzeugt, der ein GUI<sup>1</sup>-Fenster definiert, dessen Funktionalität dann nach dem Framework-Prinzip implementiert wird.

Die Entwickler-Dokumentation befindet sich in den *C++*-Header-Dateien. Das *doxygen*-Werkzeug (siehe [vH00]) übersetzt diese Dokumentation automa-

---

<sup>1</sup>graphical user interface; deutsch: graphische Benutzeroberfläche

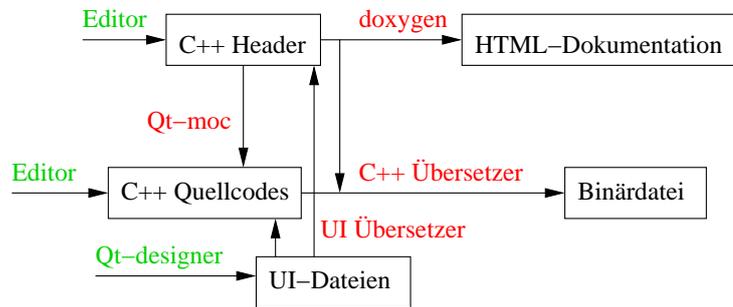


Abbildung 5.1: Zusammenhang der Quellcodes und der Sekundärdateien

tisch in eine HTML-Dokumentation, die mit einer Suchmaschine durchsucht werden kann.

Die in Abbildung 5.1 dargestellten Übersetzungsvorgänge werden mithilfe von *automake/make* durchgeführt. Die Werkzeuge *automake* und *autoconf* erlauben es, das Programm in kurzer Zeit auf einem anderen Rechner zu übersetzen und zu installieren.

### 5.1.5 Textverarbeitung

Dieses Dokument wurde mit  $\text{\LaTeX}$  bzw. *PDFLaTeX* erstellt. Die 3-D Bilder wurden mit dem freien Raytracing-Programm *Povray* [Pov] berechnet. Die Zeichnungen sind mit *xfig* angefertigt worden. Die begleitende Präsentation wurde mit  $\text{\LaTeX}$  und *texpower* erstellt.

## 5.2 Die Benutzeroberfläche

Abbildung 5.2 zeigt das Aussehen der Benutzeroberfläche. Das Hauptfenster erlaubt die Bearbeitung einer Fokusserie<sup>2</sup>. In mehreren Unterfenstern kann man die Fokusserie, die berechneten Höhenkarten und das Tiefenbild inspizieren. Die berechneten Ergebnisse lassen sich zur weiteren Verwendung abspeichern.

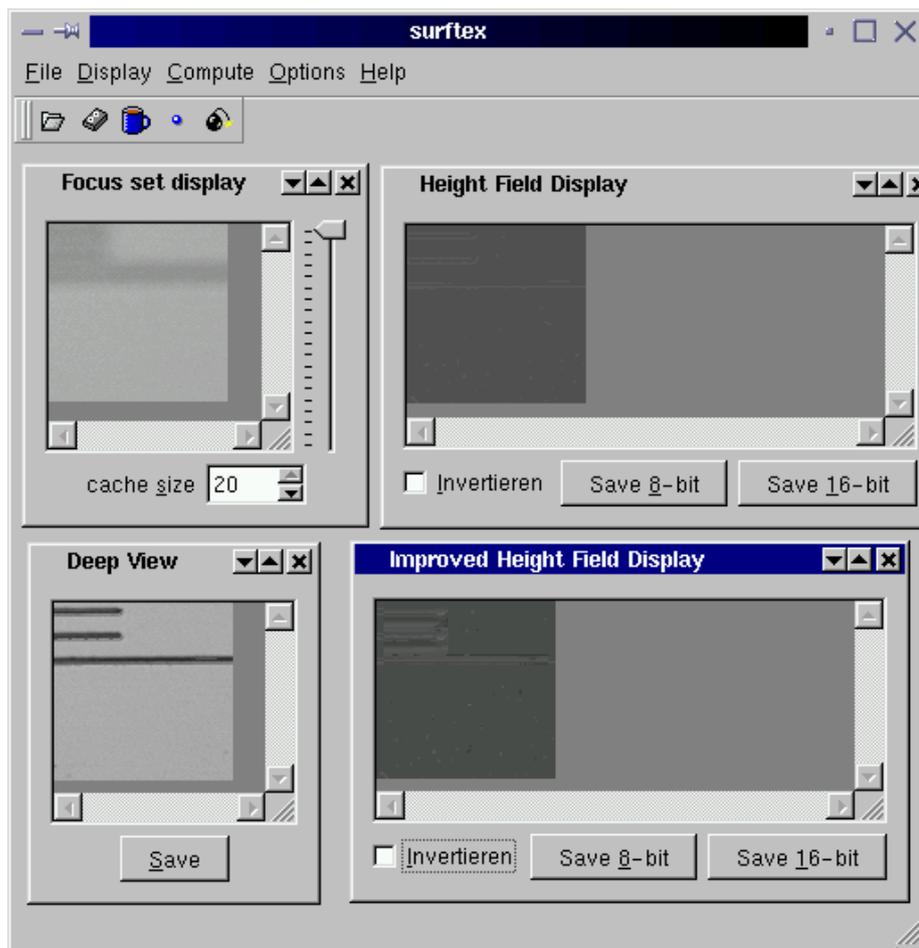
### 5.2.1 Das Menü

Ein Teil der Menüeinträge ist hier mit einer kurzen Beschreibung wiedergegeben.

- **File**
  - **Open Focus Set:** Ein Dateiauswahlfenster erlaubt es, eine Menge von Bilddateien zu selektieren, die eine Fokusserie bilden.
  - **Grab Focus Set:**<sup>3</sup> Das Programm erlaubt es auch, eine Fokusserie direkt mit dem Meßaufbau aufzunehmen. Da der Speicherbedarf relativ hoch ist, wird jede Aufnahme gleich in einer Datei gespeichert.

<sup>2</sup>kein MDI (engl.: multiple document interface)

<sup>3</sup>Dieser Menüpunkt ist nur in der erweiterten (nicht portierbaren) Version des Programms verfügbar, da hierfür das Mikroskop und die Kamera angesprochen werden müssen.

Abbildung 5.2: Die Benutzeroberfläche des Programms *surfex*

- **Load All:** Alle Bilder der mit “Open Focus Set” spezifizierten Fokuserie werden in den Hauptspeicher geladen.

- **Compute**

- **Height Field:** Die Höhenkarte wird durch Maximierung des adaptiven Schärfemaßes berechnet.
- **with Interpolation:** Die Höhenkarte wird mit dem adaptiven Schärfe-  
maß berechnet und zusätzlich durch Interpolation verbessert (siehe  
Kap. 4.2).
- **Improve Height Field:** Eine berechnete Höhenkarte kann durch  
Vergleich der Unschärfe (siehe Kap. 4.1) weiter verbessert werden.
- **Deep View:** Ein Tiefenbild (Bild mit erweiterter Schärfentiefe) wird  
erzeugt.
- **with Import:** Ein Tiefenbild wird mit einer zweiten Fokuserie er-  
zeugt. Diese zusätzliche Option erlaubt es, ein Projektions-Gitter zur  
Verbesserung der Höhenkarte zu verwenden, ohne daß es nachher im  
Tiefenbild wieder auftaucht.

- **Options**

- **Calibrate Defocus:** Um die Höhenkarte durch Vergleich der Un-  
schärfe (siehe Kap. 4.1) zu verbessern ist eine Kalibrierung der Punkt-  
antwort nötig.
- **Configure Height Field:** Die Parameter für den adaptiven Algo-  
rithmus zur Berechnung der Höhenkarte lassen sich hier einstellen.

## 5.2.2 Die Unterfenster

Es gibt vier Unterfenster, die in Abb. 5.2 zu sehen sind.

- Mit dem **Focus Set** Fenster kann man sich die Bilderserie anschauen. Die  
Bilder werden bei Bedarf nachgeladen. Da der Hauptspeicherbedarf einer  
Fokuserie u.U. sehr groß ist, läßt sich die maximale Anzahl an Bildern,  
die im Hauptspeicher gehalten werden, über ein Eingabefeld begrenzen.
- Das **Height Field** Fenster zeigt die berechnete Höhenkarte an. Die be-  
rechnete Höhenkarte läßt sich als 8-Bit Grauwertbild abspeichern. Die be-  
rechnete Höhenkarte kann auch als Rot-Grün Bild für *Povray* exportiert  
werden. Der Rotanteil enthält die höherwertigen und der Grünanteil die  
niedrigen 8 Bits.
- Analog gibt es das Fenster mit dem Titel “**Improved Height Field**”,  
welches die durch Abschätzung der Unschärfe verbesserte Höhenkarte an-  
zeigt.
- Das **Deep View** Fenster zeigt das Bild mit erweiterter Schärfentiefe, wel-  
ches sich ebenfalls abspeichern läßt.

### 5.2.3 Einstellung der Parameter

Vor der Berechnung des Höhenfeldes muß die Zahl der zu betrachtenden Auflösungsstufen eingestellt werden. Der Faktor, der bestimmt, wie stark der Suchbereich für die Höhenkarte eingeschränkt wird (siehe Faktor  $\gamma$  in Kapitel 3.5), wird im selben Fenster eingegeben (Abb. 5.3).

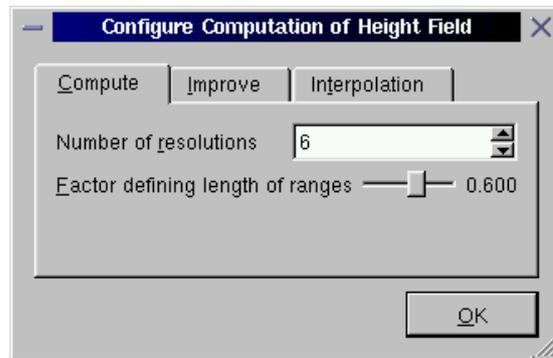


Abbildung 5.3: Die Parameter zur Berechnung der Höhenkarte

Um die Höhenkarte mittels Abschätzung der Unschärfe zu verbessern, wird der Abstand der Fokalebene der Fokusserie benötigt (Abb. 5.4). Außerdem ist die Zahl der Zwischenebenen, die "ausprobiert" werden sollen, einzustellen (Konstante  $M$  in Gleichung (4.1)).

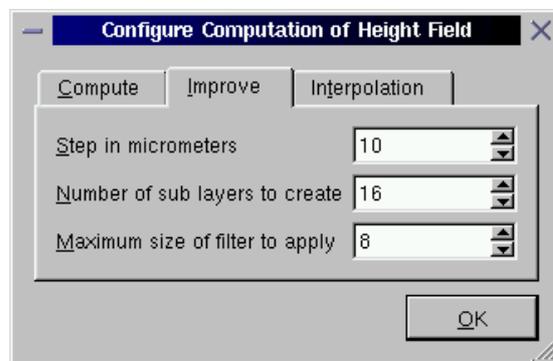


Abbildung 5.4: Die Parameter zur Verbesserung der Höhenkarte

Der letzte Teil des Dialogfensters, der in Abbildung 5.5 dargestellt ist, erlaubt die Einstellung der Zahl der Stützstellen, die zur Interpolation der Schärfemaß-Kurve verwendet werden (siehe Kapitel 4.2). Es kann nur eine ungerade Anzahl eingestellt werden.

### 5.2.4 Kalibrierungs-Dialog

Um die Höhenkarte durch Abschätzung der Unschärfe (Kap. 4.1) zu verbessern, muß die Punktantwort des Mikroskops kalibriert werden. Dazu wird ein

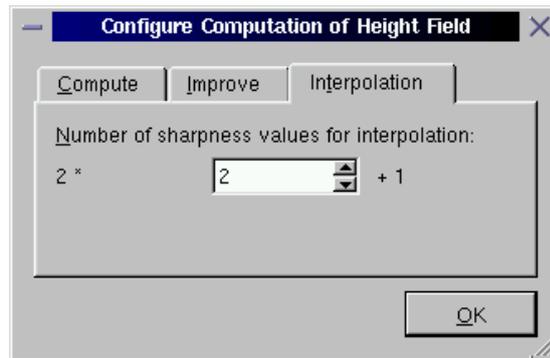


Abbildung 5.5: Die Parameter zur Auflösungsverbesserung durch Interpolation

Bild einer scharf abgebildeten Ebene und ein korrespondierendes unscharfes Bild benötigt. Die Breite der Punktantwort läßt sich dann einstellen. Das zu minimierende Fehlermaß gibt denn Abstand des unscharfen Bildes und des mit der eingestellten Punktantwort gefilterten, scharfen Bildes an.

Um die eingestellte Breite der Punktantwort auf die Fokuserie übertragen zu können, muß des weiteren der Abstand der Fokalebene der beiden Bildern angegeben werden.

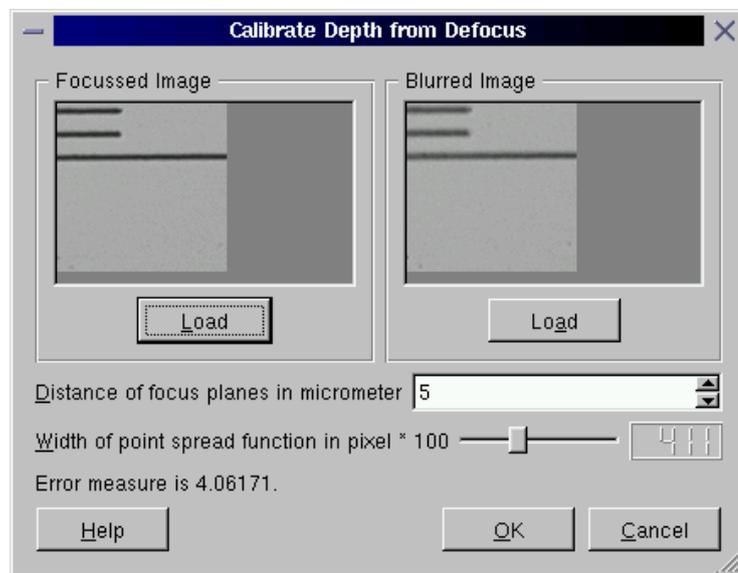


Abbildung 5.6: Das Fenster zur Kalibrierung der Punktantwort

## 5.3 Die Programmbibliothek

### 5.3.1 Konzepte

Zur Implementation insbesondere der Felder wurden die Möglichkeiten von Templates und Operatoren ausgenutzt:

```
NumberND<int> a( 2, 5, 5 ), b( 2, 5, 5 );
...
NumberND<int> c = a - b;
```

Die Laufzeitfehler, die vom Entwickler z.T. nicht ausgeschlossen werden können, werden mithilfe der Ausnahmebehandlung<sup>4</sup> von *C++* bearbeitet:

```
try {
    NumberND<int> a( 2, 3, 5 ); // 2-D Feld der Größe 3x5.
    cout << "3. Dimension = " << a.getDimension( 2 ) << endl;
} catch ( Error e ) {
    cerr << e << endl;
}
```

Die dynamischen n-dimensionalen Felder<sup>5</sup> wurden selbst implementiert, da die Standard-Template-Bibliothek von *C++* derzeit in dieser Richtung noch nichts bietet:

```
NumberND<int> a( 2, 5, 5 ); a.value( 4, 4 ) = 2;
NumberND<float> b; b.convertFrom( a.sqr() );
NumberND<int> c = a;
c.value( 4, 4 ) -= (int)( b.value( 4, 4 ) );
assert( c.value( 4, 4 ) == -2 );
```

Der Zuweisungsoperator ist "faul". D.h. Felder teilen gemeinsame Daten, solange dies möglich ist: In diesem Beispiel wird der Speicher für die Elemente von *c* erst im Funktionsaufruf der vorletzten Zeile alloziert! Nachteilig ist, daß der schreibende Zugriff auf Feldelemente mit der Methode `value(...)` dadurch langsamer wird. Dafür können jedoch Felder Rückgabewert einer Funktion sein, ohne das große Datenmengen kopiert werden müssen.

Wird auf alle Elemente eines Feldes in einer Schleife zugegriffen, so wird das über Template-Methoden mit einem Funktionsobjekt gemacht. Damit wird die `value(...)`-Methode umgangen, ohne daß das Kapselungsprinzip verletzt wird:

```
struct PrintInt: public unary_function<int,void> {
    inline void operator()( const int &x ) const {
        cout << x << endl; }
}
{ NumberND<int> a;
  ...
  a.forEach( PrintInt() );
  ...
```

---

<sup>4</sup>engl.: exception handling

<sup>5</sup>engl.: arrays

### 5.3.2 Beispiel zur Berechnung der Höhenkarte

Der Algorithmus ist in einer Programmbibliothek getrennt von den GUI-Programm abgelegt. Ein einfaches Programm, welches eine Höhenkarte berechnet, sieht dann etwa so aus:

```

...
// Erzeuge Objekt zur Berechnung der Höhenkarte der Größe 'width' *
// 'height'. Die Fokuserie hat 'depth' Bilder. Das Schärfemaß ist
// die lokale normierte Grauwertvarianz. 'factor' stellt die Breite
// der Suchbereiche ein.
MaximizeSharpness maximizeSharpness( width, height, depth,
                                     p = new DeviationPyramid( NUMMESHES ),
                                     factor);
for ( int c=0; c<2; c++ )
  for ( HeightType z=0; z<depth; z++ ) {
    QImage *image = ...;// Hier wird das 'z'te Bild geladen.
    assert( image );
    if ( c == 0 )
      // Im ersten Durchlauf werden die Maxima bestimmt.
      maximizeSharpness.cycleLoopOne( z, image );
    else
      // Im zweiten Durchlauf werden die Suchbereiche bestimmt.
      maximizeSharpness.cycleLoopTwo( z, image );
  };
// Die Ergebnis-Höhenkarte wird berechnet.
NumberND<HeightType> heightField = maximizeSharpness.finish();
delete p;
...

```

Der Algorithmus benötigt die beiden Durchläufe, da nicht alle berechneten Daten im Speicher gehalten werden können. U.U. können nicht einmal alle Bilder der Fokuserie gleichzeitig im Speicher residieren. Daher ist die Schnittstelle so beschaffen, daß das aufrufende Programm die Bilder einzeln bereitstellen muß.

### 5.3.3 Beispiel zur Berechnung des Tiefenbildes

Um ein Tiefenbild zu berechnen wird eine Höhenkarte benötigt. Zur Erzeugung dieses Bildes gibt es das Funktionsobjekt `UpdateDeepView`, mit welchem die fokussierten Bereiche eines Bildes in das Tiefenbild kopiert werden:

```

...
// Erzeuge leeres Bild (QImage von Qt-Bibliothek).
deepImage.create( width, height, 32 );
for ( HeightType z=0; z<depth; z++ ) {
  QImage *image = ...;// Lade z-tes Bild der Fokuserie.
  // Erzeuge Funktionsobjekt.
  UpdateDeepView updateDeepView( deepImage, z, *image );
  // Hier werden die fokussierten Bereiche des Bildes nach
  // 'deepImage' kopiert.
  heightField.forEach( updateDeepView );
}

```

```
};  
...
```

Auch hier muß das aufrufende Programm die Bilder einzeln vorgeben können, damit nicht alle Bilder in den Hauptspeicher geladen werden müssen.

# Kapitel 6

## Praktische Ergebnisse

### 6.1 Visueller Vergleich

Abbildung 6.1 zeigt Höhenkarten des in Abb. 3.6 markierten Bereiches der 1-Cent Münze. Die vier Kombinationen aus ohne/mit projiziertem Hilfsmuster und einfache/adaptive Maximierung wurden ausprobiert. Man kann erkennen, daß die mit dem adaptive Algorithmus und dem Projektionsmuster ermittelte Höhenkarte nahezu keine Ausreißer mehr aufweist. Außerdem weist diese Höhenkarte keine mit dem Schachbrettmuster einhergehenden Artefakte auf.

Bild 6.2 zeigt die Rekonstruktion eines Ammoniten mit ca. 1.5 mm Durchmesser. Da die Höhenkarte nicht in x- und y-Richtung interpoliert wird (siehe auch Kap. 7.2), mußte sie geglättet werden. Trotzdem sind riefenartigen Artefakte zurückgeblieben. Dies ließe sich durch Interpolation aber beheben. Die Rekonstruktion zeigt aber deutlich, daß der Algorithmus in der Lage ist, auch Objektoberflächen mit hoher Steigung und großen Sprüngen zu bewältigen!

In Abbildung 6.3 ist ein Teil einer 1-Cent Münze gezeigt. Die entsprechende nicht-texturierte Rekonstruktion ist im Bild 6.4 zu sehen. Sie zeigt, welche Objektdetails tatsächlich erfaßt wurden. Die Fehler an den Rändern der Struktur sind durch mangelhafte Helligkeit an diesen Stellen zu erklären.

Bild 6.5 zeigt einen kleinen Ausschnitt von einem Stück Schiefer. Mit diesem Bild läßt sich allerdings die Güte des Algorithmus nur schwer beurteilen. Strukturen, deren Größe unter der Kantenlänge des projizierten Schachbrettmusters liegt, konnten hier nur schlecht erfaßt werden, da das Objekt selber wenig Oberflächentextur aufwies.

### 6.2 Vertikale Auflösung

#### 6.2.1 Test mit simulierter Fokussiererei

Um die Qualität der Ergebnisse zu bewerten, wurde eine Fokussiererei mit *Povray* simuliert. Die Fokussiererei zeigt dabei eine Ebene mit einem Schachbrettmuster. Da geneigte Oberflächen höhere Ansprüche an die Algorithmen stellen, wurde eine Fokussiererei der Größe  $100 \times 100 \times 100$  Pixel von einer geneigten Ebene erstellt. Die Felder des Schachbrettmusters haben eine Größe von ca.  $5 \times 5$  Pixeln. Abbildung 6.6 zeigt das erste, mittlere und letzte Bild der Fokussiererei.

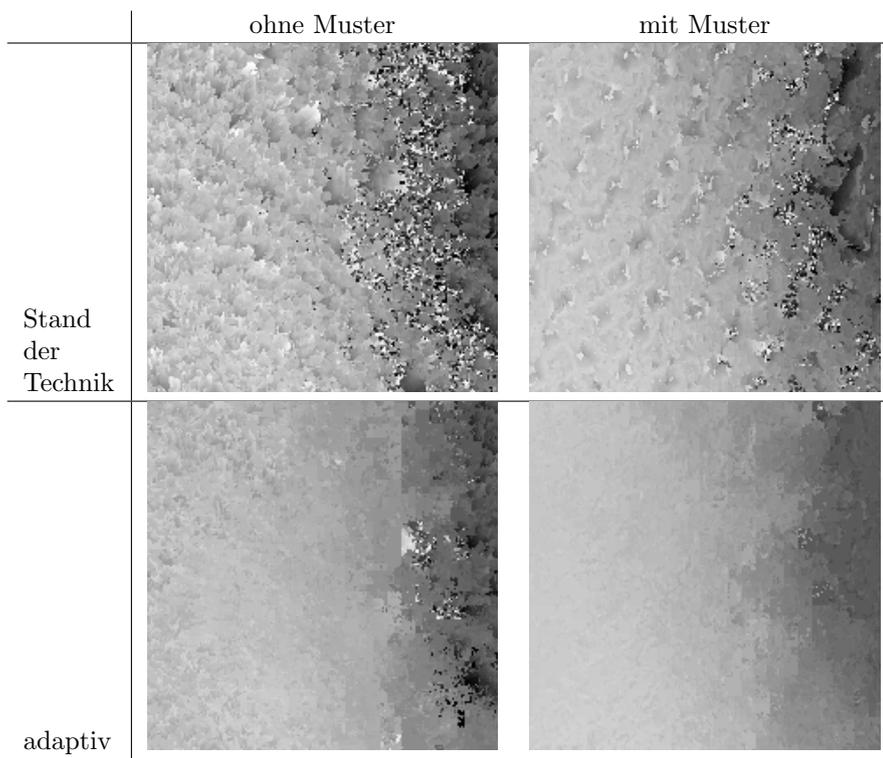


Abbildung 6.1: Visueller Vergleich der Höhenkarten des vergrößerten Bereichs der 1-Cent Münze aus Abb. 3.6



Abbildung 6.2: Rekonstruktion eines Ammoniten mit ca. 1.5 mm Durchmesser

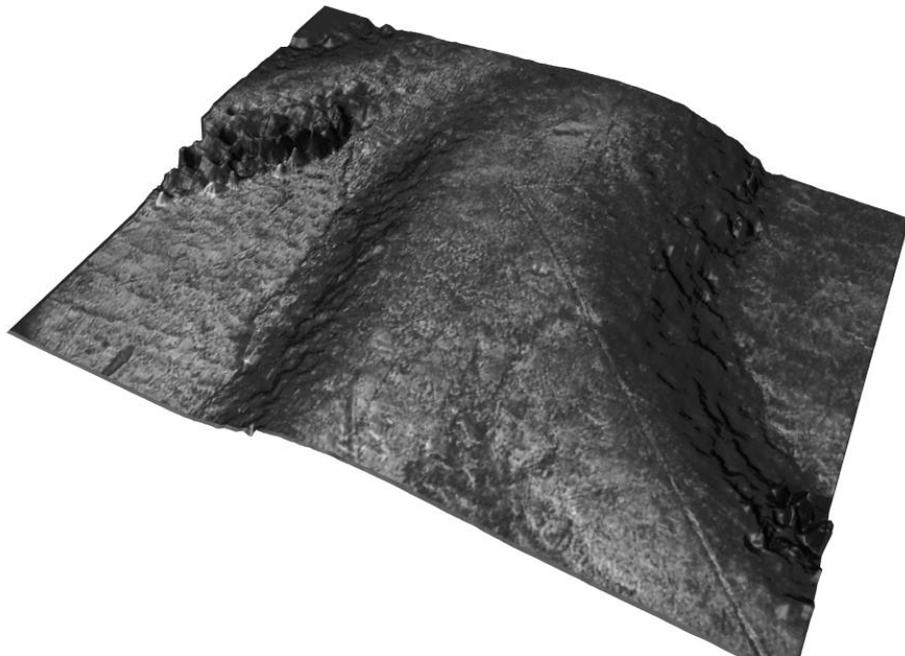


Abbildung 6.3: (Texturierte) Rekonstruktion eines Teils einer 1-Cent Münze

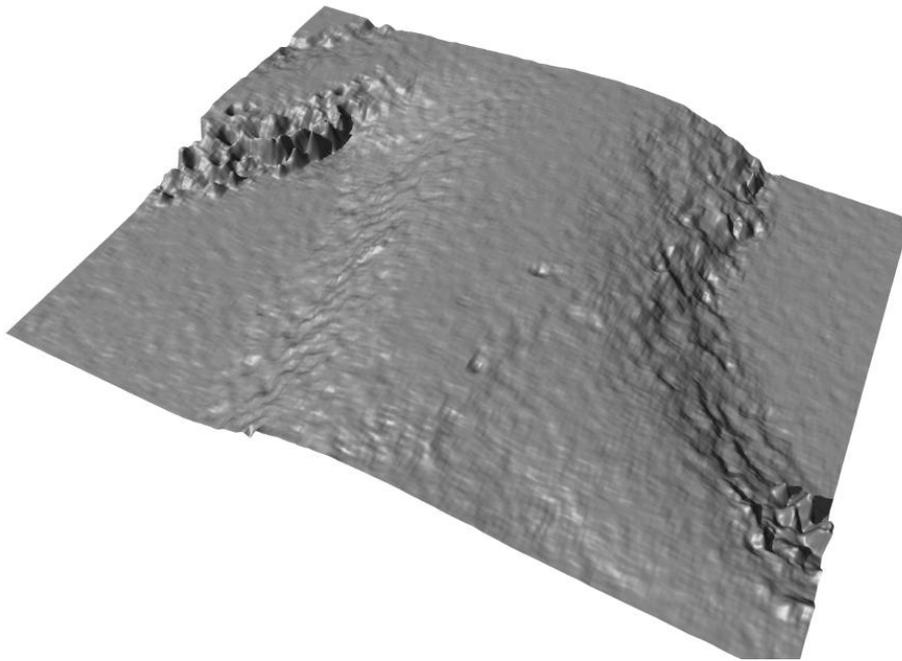


Abbildung 6.4: Nicht-texturierte Rekonstruktion zu Bild6.3

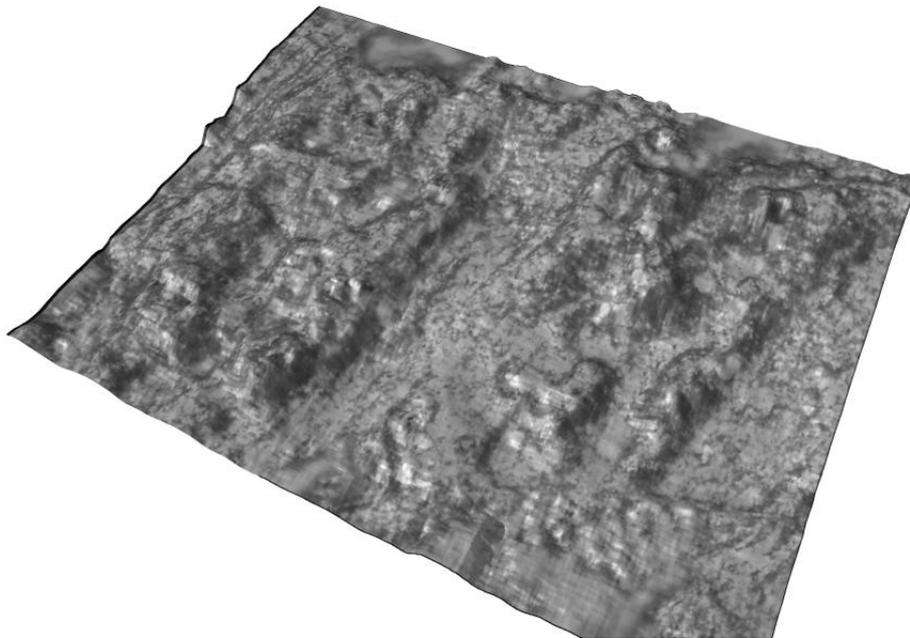


Abbildung 6.5:  $537\mu\text{m} \times 403\mu\text{m}$  Ausschnitt der Oberfläche von einem Stück Schiefer

Die Ausreißer in den Bildern rühren daher, daß *Povray* die Unschärfe mithilfe von zufällig ausgewählten Strahlen berechnet. Je höher die Bildqualität sein soll, desto höher ist die benötigte Rechenzeit. Da die fehlerhaften Pixel als Test für mögliches Kamerarauschen angesehen werden können, wurde die Fokussiererei so belassen.

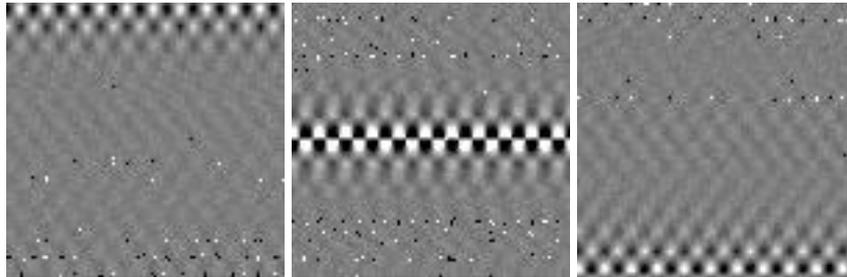


Abbildung 6.6: Erstes, mittleres und letztes Bild der mit *Povray* simulierten Fokussiererei

Mit den verschiedenen Algorithmen wurde jeweils eine Höhenkarte  $d(x, y)$  zu dieser Fokussiererei berechnet. Die resultierenden Höhenkarten sind in Abbildung 6.7 zu sehen. Zu den Höhenkarten wurden anschließend jeweils die Parameter der Gleichung  $a_0x + a_1y + a_2 = z$  der Regressions-Ebene, wie im Anhang A.5.3 beschrieben, bestimmt. Zuletzt wurde das mittlere Fehlerquadrat  $\sigma_d^2$  berechnet, das ein Maß dafür ist, wie stark die Höhenkarte von der Regressions-Ebene abweicht. Die Tabelle in Abbildung 6.8 zeigt die Ergebnisse für die Regressions-Ebene und die Wurzel des Fehlerquadrats. Der Faktor  $\gamma$  für den adaptiven Algorithmus hatte den Wert 0.4 und es wurden 6 Auflösungsstufen betrachtet.

Es ist deutlich zu sehen, daß das Ergebnis der einfachen Maximierung besonders schlecht ausgefallen ist. Durch einen größeren Filter könnte dieses Ergebnis natürlich verbessert werden.

Der adaptive Algorithmus schneidet um mehr als eine Größenordnung besser ab. Der Vergleich der Unschärfe verschlechtert das Ergebnis. Die Interpolation führt zu einer geringfügigen Verbesserung von  $\sigma_d$ .

### 6.2.2 Test mit realem Objekt

Es wurde ein Test mit einem realen ebenen Objekt durchgeführt. Obwohl ein hoher Wert für  $\gamma$  hier zu besseren Ergebnissen führen würde, wurde 0.6 als Wert für den Faktor  $\gamma$  des adaptiven Algorithmus gewählt und es wurden 6 Auflösungsstufen des Schärfemaßes betrachtet. Wählt man z.B.  $\gamma = 0.9$ , wird der adaptive Algorithmus i.a. die groben Auflösungsstufen des Schärfemaßes bevorzugen und dabei größere Objektdetails übersehen. Da die Evaluierung aber auch auf nicht-planare Objekte, bei denen auch die laterale Auflösung eine Rolle spielt, übertragbar sein soll, wurde *kein* hoher Wert für  $\gamma$  gewählt.

Zum Test wurde eine Fokussiererei von einem Teil der Seitenfläche eines Aluminiumprofils unter Verwendung des Schachbrettmusters erstellt. Tabelle 6.9 zeigt die erzielten Varianzen für die einfache Maximierung und den adaptiven Algorithmus (mit und ohne Auflösungsverbesserung durch Vergleich der Unschärfe bzw. Interpolation).

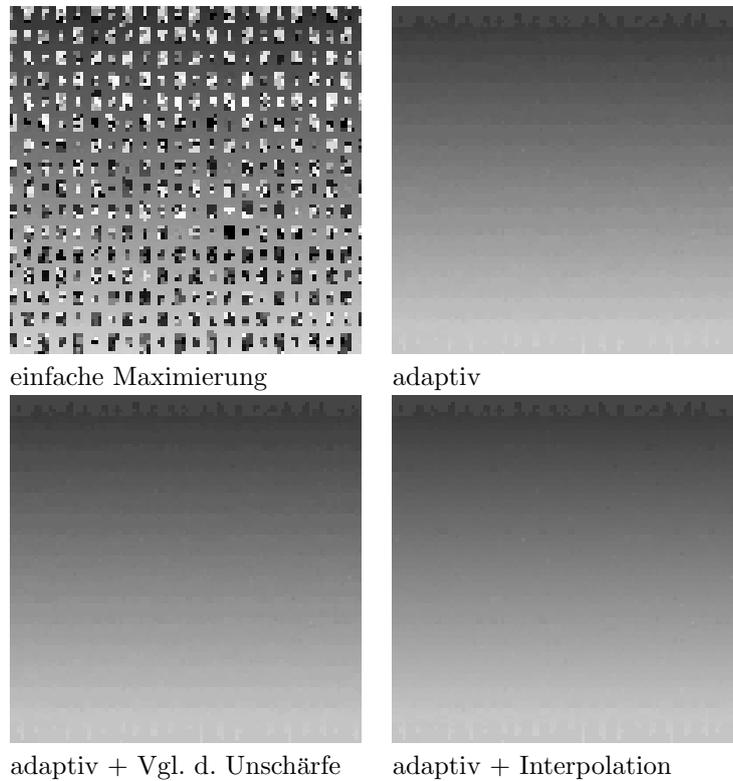


Abbildung 6.7: Mit verschiedenen Methoden ermittelte Höhenkarten zu der simulierten Fokusserie.

Algorithmus	$a_0$	$a_1$	$a_2$	$\sigma_d$
einfach	-0.004002640	0.302939	34.6382	21.6531
adaptiv	0.000294509	0.593865	21.0919	1.41354
adaptiv + Vgl. d. Unschärfe	0.000633765	0.594962	20.9502	1.49563
adaptiv + Interpolation	0.000028820	0.591588	21.1590	1.37037

Abbildung 6.8: Vergleich der Ergebnisse zu der simulierten Fokusserie

Algorithmus	$a_0$	$a_1$	$a_2$	$\sigma_d/2 \mu\text{m}$
einfach	-0.00356323	-0.000872398	25.0365	3.664350
adaptiv	-0.00357069	-0.001075390	25.7811	0.762538
adaptiv + Vgl. d. Unschärfe	-0.00334136	-0.000914712	25.7965	0.970599
adaptiv + Interpolation	-0.00361783	-0.001032650	25.8181	0.675776

Abbildung 6.9: Vergleich der Ergebnisse zu der realen Fokusserie

Rechenzeit in Sekunden für	10 Bilder	100 Bilder
einfach	5.19	45.04
adaptiv	9.47	68.85
adaptiv + Vgl. d. Unschärfe	77.26	153.57
adaptiv + Interpolation	23.42	87.72

Abbildung 6.10: Vergleich der für die verschiedenen Algorithmen benötigten Rechenzeiten

Das Ergebnis, welches mit der einfachen Maximierung erzielt wurde, ist hier wesentlich besser als bei der simulierten Fokusserie, da das reale Objekt auch feinkörnige Texturmerkmale aufweist. Nichtsdestotrotz schneidet auch hier der adaptive Algorithmus besser ab, da das Schachbrettmuster die Unterdrückung der systematischen Fehler nötig macht. Da der Fokalebeneabstand der Fokusserie  $2\ \mu\text{m}$  betrug, ist die erreichte Standardabweichung hier  $1.35\ \mu\text{m}$ !

### 6.3 Laterale Auflösung

Die begleitende laterale Auflösung läßt sich schlecht quantifizieren, da sie von verschiedenen Umständen abhängt. Weist das betrachtete Mikroobjekt keine feinkörnigen Texturen auf, so hängt die erreichbare Auflösung vom verwendeten Hilfsmuster ab (hier Schachbrettmuster mit  $13\ \mu\text{m} \times 13\ \mu\text{m}$  Feldern in der Projektion). Wenn das Objekt feinkörnige Texturen zeigt, so wird der adaptive Algorithmus die Breite und Höhe des verwendeten Schärfemaß-Filters aber auf 3 Pixel senken, was bei dem verwendeten Meßaufbau Objektdetails mit einem Durchmesser von  $2.2\ \mu\text{m}$  sichtbar machen sollte.

### 6.4 Benötigte Rechenzeit

In Tabelle 6.10 sind beispielhaft benötigte Rechenzeiten für die verschiedenen Algorithmen angegeben. Die Leistungsdaten des verwendeten Rechners sind bereits in Kapitel 1.2 angegeben worden. Die Bilder hatten eine Größe von  $768 \times 576$  Pixel. Die Filtergröße für den einfachen Algorithmus war  $3 \times 3$ . Der adaptive Algorithmus hatte 6 Auflösungsstufen zu betrachten. Außerdem wurden bei der Interpolation 11 Schärfewerte verwendet und beim Vergleich der Unschärfe wurden 16 verschiedene Zwischenebenen betrachtet (Parameter  $M$  in Formel (4.1)).

Der Vergleich der Unschärfe benötigt viel Rechenzeit und wurde aber, da er im allgemeinen keine guten Ergebnisse erzielte, nicht weiter optimiert (siehe Kapitel 4.1). Die Interpolation benötigt genauso wie der Vergleich der Unschärfe eine von der Anzahl der Bilder der Fokusserie unabhängige Rechenzeit.

Die für den adaptiven Algorithmus benötigte Rechenzeit ist, wie zu erwarten war, etwas höher. Man darf aber nicht vergessen, daß der verwendete Meßaufbau zum Aufnehmen der Bilder bereits ca.  $0.5\ \text{Sekunden/Bild}$  benötigt. Da der adaptive Algorithmus zunächst einmal die Maxima und in einem zweiten Schritt erst die Suchbereiche bestimmt, kann er aber nicht vollständig “on-the-fly” also während der Aufnahme der Bilder ablaufen.

## 6.5 Beobachtungen

Es wurden verschiedene Beobachtungen gemacht. Dabei sind folgende Probleme beobachtet worden, die die Anwendbarkeit des neuen Algorithmus begrenzen:

- Die Mikroobjekte müssen diffuse Reflektion aufweisen. Mit dem verwendeten Meßaufbau lassen sich geneigte reflektierende Oberflächen (z.B. Metall) nicht ausreichend aufhellen und es kommt zu fehlerhaften Resultaten (siehe z.B. Abb. 7.1 in Kapitel 7). Zu durchsichtigen Objekten wie zum Beispiel Kristallen kann man i.a. keine guten Höhenkarten erhalten. In dem Bild erweiterter Tiefenschärfe fallen diese Probleme allerdings weniger auf (vgl. Abb. 6.11).
- Die Rechenzeit (siehe Kap. 6.4 ist relativ hoch. Allerdings ist sie linear mit der Größe des Datensatzes. Der verwendete Meßaufbau benötigte ohnehin ca.  $0.5 \text{ Sekunden/Bild}$ . Der Umfang der berechneten Daten ist enorm. Die Schärfemaß-Werte werden zweimal benötigt und auch zweimal berechnet, da sie nicht im Hauptspeicher abgelegt werden können. Der Algorithmus ist ohne teure Spezialhardware nicht in Echtzeit einsetzbar. Der in [Sch97] erwähnte Einsatz in der industriellen Qualitätskontrolle für Halbleiterbausteine ist nur stichprobenartig möglich.
- Die Fokussereien sollten keine hohen globalen Helligkeitsunterschiede haben. Wenn man z.B. ein dunkles Objekt auf einem hellen Hintergrund betrachtet, treten an den Rändern auch mit dem adaptiven Verfahren noch systematische Fehler auf, da die Helligkeitsänderung in diesem Fall mit einer großen Unstetigkeit der Oberfläche einhergeht. In der niedrigen Auflösung der adaptiven Maximierung läßt sich die Höhe der Oberfläche dann nicht bestimmen, da keine einheitliche Höhe existiert; in der hohen Auflösung treten wieder die systematischen Fehler auf.
- Besonders gute Ergebnisse wurden bei Oberflächen mit geringer Steigung und feinkörnigen Texturen erzielt.

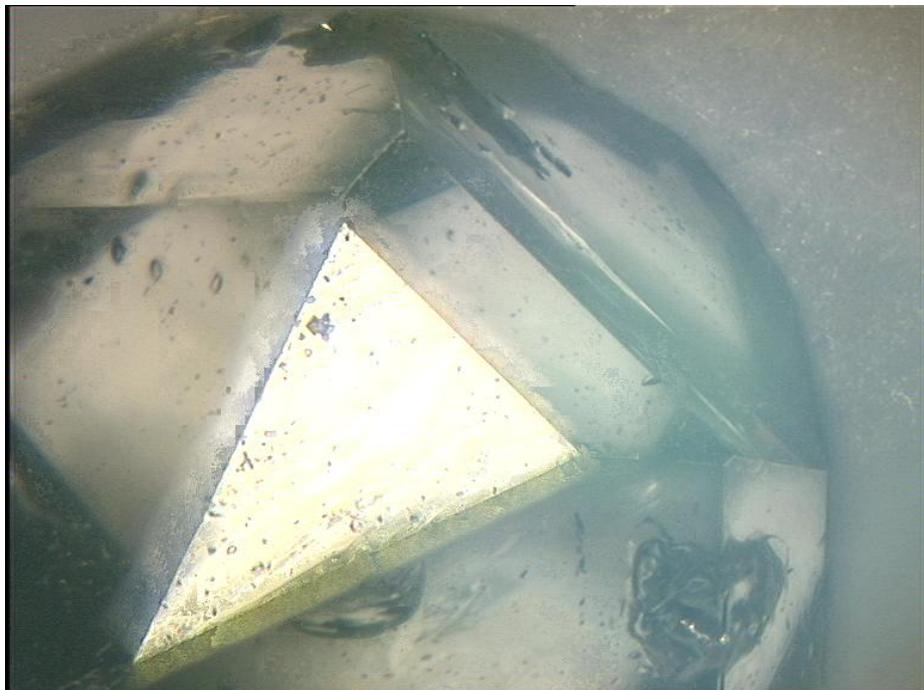


Abbildung 6.11: Tiefenbild eines kleinen Bergkristalls

# Kapitel 7

## Ausblick

Wie bei fast jeder Arbeit, gibt es auch hier noch Möglichkeiten zur Verbesserung und Erweiterung:

### 7.1 Verbesserungen am Meßaufbau

- In [Sch97] wird ein Helligkeitsregler zur Einstellung der Beleuchtung des Bildfeldes vorgestellt. Die Einstellung der Helligkeit muß dann nicht mehr manuell vorgenommen werden. Die Helligkeit des Bildfeldes ist vor allem bei CCD-Kameras mit 8-Bit Grauwerten von zentraler Bedeutung. Sobald Bildbereiche zu dunkel oder zu hell (Sättigung der Kamerapixel) werden, ist eine Abschätzung der Bildtiefe unmöglich, da alle Strukturmerkmale verschwinden.
- Die Bewegungsachse des *Z-Tisches* ist nicht genau parallel zur optischen Achse des Objektivs. Durch eine Korrektur (mechanische oder Software-seitige Kalibrierung) würde die Fokussiererei “symmetrischer”. Dadurch wird das Ergebnis verbessert, da der Verlauf der Schärfemaß-Kurve auch “symmetrischer” wird.

### 7.2 Verbesserungen am Algorithmus

- Die Auflösungsverbesserung mittels Vergleich der Unschärfe wird im Moment für jedes Pixel durchgeführt, bei dem ausreichend lokale Strukturmerkmale auftreten. Die Filtergröße wird dann bis zu einer bestimmten Obergrenze variiert. Dieser Algorithmus könnte jedoch auch adaptiv gestaltet werden. D.h. man würde bei Bildbereichen, die nur niederfrequente Strukturen aufweisen, einen großen Filter anwenden, dafür aber die Auflösungsverbesserung nur noch für Bildbereiche durchführen.
- Mittels lokaler inverser Filterung könnte man u.U. das Aussehen des Tiefenbildes noch verbessern. Damit könnte man auch zu Bereichen der Objektoberfläche, die in keinem Bild der Fokussiererei vollständig scharf abgebildet sind, eine “scharfe” Oberflächentextur bekommen. Im Moment ist das Tiefenbild an jeder Stelle mindestens so unscharf, wie das beste Bild

der jeweiligen Region. Dies setzt allerdings voraus, daß die Auflösungsverbesserung genaue und verlässliche Ergebnisse liefert.

- Im Moment wird ein gemeinsamer Faktor  $\gamma$  zur Bestimmung der Suchbereiche verwendet. Man könnte hier natürlich auch unterschiedliche Faktoren für die einzelnen Ebenen der Pyramiden verwenden.
- Durch Fusion von Fokussereien mit verschiedenen Beleuchtungsbedingungen, wie in [Leó99] beschrieben, kann man das Ergebnis für die Höhenkarte verbessern. Bild 7.1 zeigt einen Teil des Buchstaben “E” von einer 1-Cent Münze. Die Ränder des “E”s können durch Auflichtbeleuchtung nicht erhellt werden, so daß die Rekonstruktion dort fehlschlägt.
- Es wurden in Kapitel 6 nur die Abweichungen bei ebenen Objekten untersucht. Die optimale Filtergröße bei Objekten, die nicht eben sind, liegt allerdings i.a. niedriger, da hier die laterale Auflösung auch eine Rolle spielt. Des weiteren sind die Maxima des Schärfemaßes zwar besser ausgeprägt, wenn die Vergrößerung des Mikroskops höher gewählt wird. Dafür verändert aber das Licht von defokussierten Regionen immer größere Teile der Bilder der Fokusserei. Es gibt daher eine Vielzahl von denkbaren Testfällen, die untersucht werden können.
- Im Moment werden die Höhenwerte des adaptiven Algorithmus ohne Interpolation in x- und y-Richtung übernommen. D.h., wenn der adaptive Algorithmus eine niedrigere Auflösung auswählt, erhalten mehrere Pixel der Höhenkarte den gleichen Wert. In der Höhenkarte treten daher gelegentlich Kanten auf. Eine Interpolation könnte zum Beispiel mittels Delaunay-Triangulation bewerkstelligt werden. Der Schätzfehler wird dann i.a. geringer ausfallen.

### 7.3 Schärfemaß mit Wavelets

Andere Schärfemaße und Vorfilterungen für die Fokusserei sind denkbar. Die Wavelet-Transformation besitzt interessante Eigenschaften, die sie für ein Schärfemaß besonders tauglich machen könnten und wurde im Rahmen dieser Diplomarbeit untersucht. Die sogenannte *multiresolution analysis*<sup>1</sup> erzeugt bereits eine Art Pyramide. Quadrierung der Werte der Wavelet-Transformation liefert eine Spektralanalyse, die aber im Gegensatz zur Fouriertransformation Ortsinformation enthält. Allerdings stellt z.B. das *mexican hat*-Wavelet (siehe Anhang A.3) niedriger Frequenz ein Tiefpaß-Filter dar (siehe Formel (A.12)). Um ähnlich wie mit der *lokalen Grauwertvarianz* auch Fokussereien, die nur dünn gesät Merkmale aufweisen, welche auch nur hohe Frequenzen aufweisen, erfolgreich analysieren zu können, muß man für die groben Auflösungsstufen der Schärfemaß-Pyramide eines Bildes das Schärfemaß aus verschiedenen Wavelets zusammensetzen:

- Das Schärfemaß muß die Leistung des Wavelets niedriger Frequenz enthalten.
- Das Schärfemaß muß außerdem einen lokalen Mittelwert der Leistung der Wavelets höherer Frequenzen enthalten.

<sup>1</sup>deutsch: Analyse verschiedener Auflösungen

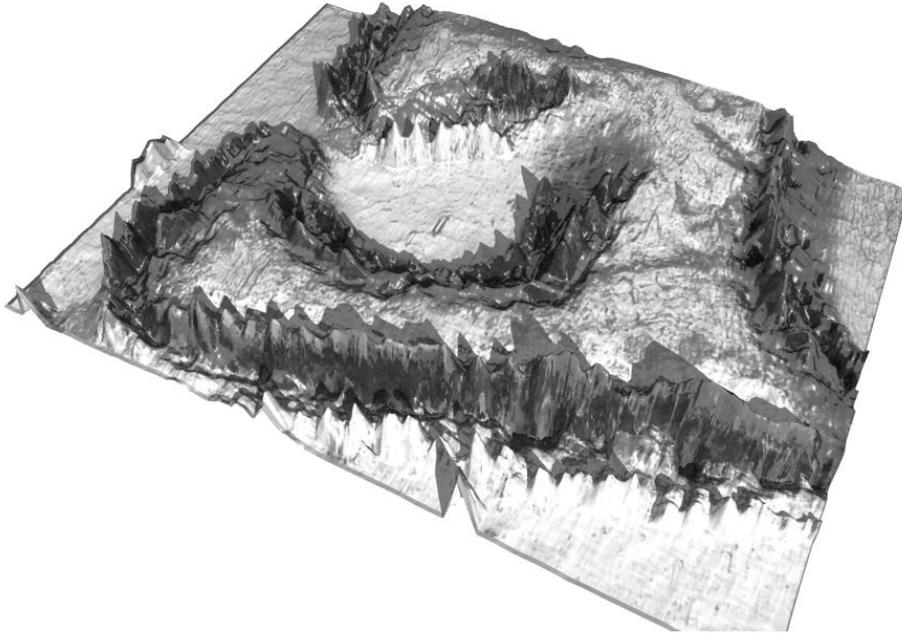


Abbildung 7.1: Rekonstruktion eines Teils des Buchstaben “E” von einer 1-Cent Münze

Für eine Einführung zu dem Thema *Wavelets* wird hier auf das Buch [Dau92] verwiesen.

## 7.4 Erweiterungen

- Durch automatisches Zusammensetzen mehrerer Höhenkarten lassen sich auch größere Objekte mit hoher Detailgenauigkeit rekonstruieren. Abbildung 7.2 zeigt z.B. eine solche Rekonstruktion. Man könnte auch durch Bilden von Vereinigungs- und Schnittmengen 3-D Modelle von Mikroobjekten bilden, wenn man Fokussereien von verschiedenen Ansichten des Objekts erstellt hat.
- Basierend auf Höhenkarten läßt sich Objekterkennung durchführen. In [JH96] wird ein Verfahren vorgestellt, um Objekte mittels Vergleich lokaler Oberflächenmerkmale (sog. Spin-Images) in einer Szene zu finden.

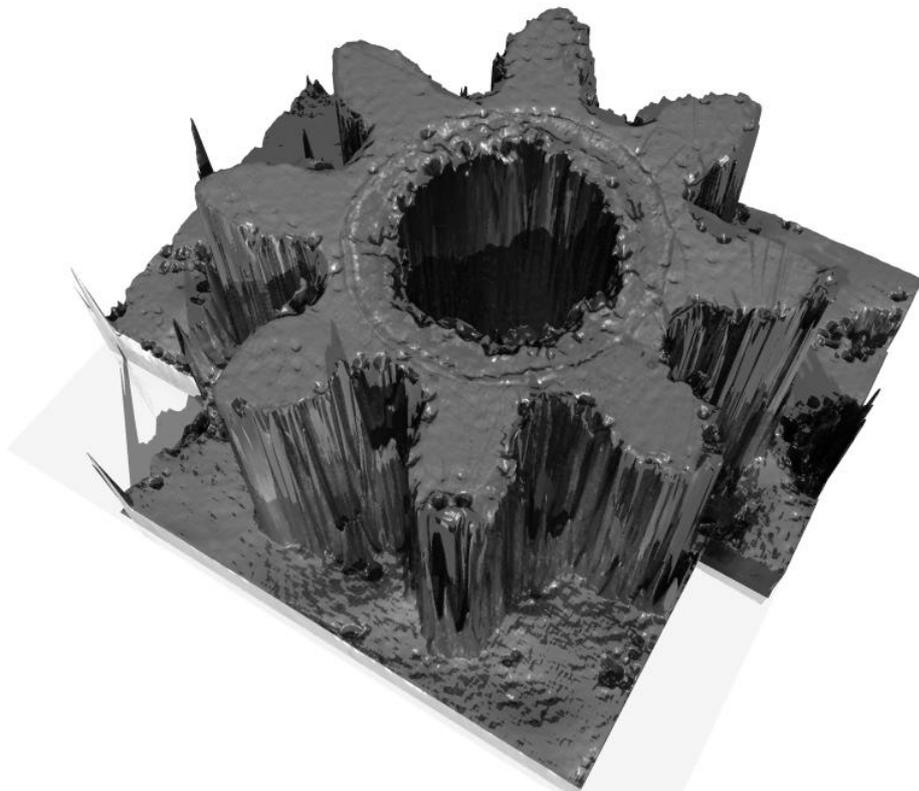


Abbildung 7.2: Aus drei Fokuserien zusammengesetzte Rekonstruktion eines Mikrozahnrades (mit  $\varnothing 0.6 \mu\text{m}$ )

## Kapitel 8

# Danksagung

Danke an Prof. J. Travis, das Miniman-Team aus Sheffield und alle, die meinen Aufenthalt in Sheffield zu einem schönen Erlebnis gemacht haben.  Ich möchte mich bei M. Heizmann vom MRT für die hilfreichen Anregungen bedanken.

Danke an das Miniman-Team, die HiWis und Prof. Wörn am IPR in Karlsruhe für die Zusammenarbeit am Miniman-Projekt, die mir viel Spaß gemacht hat.

Danke auch an den DAAD, der meinen Auslandsaufenthalt in England finanziell ermöglicht hat.

Vielen Dank meinem Betreuer A. Bürkle, für die Unterstützung bei der Diplomarbeit und die Organisation des Auslandsaufenthalt.

Zum Schluß möchte ich meinen Eltern meinen Dank kundtun, deren Erziehung und finanzielle Unterstützung mein Studium ermöglicht haben.

# Anhang A

## Anhang

### A.1 Inverse Faltung

#### A.1.1 Diskrete Fouriertransformation

Sei  $\vec{f} = (f_0, f_1, \dots, f_{N-1})^\top \in \mathbf{R}^N$ . Die diskrete Fouriertransformation  $\mathbf{F} \in \mathbf{C}^N$  von  $\vec{f}$  ist wie folgt definiert:

$$\mathbf{F}_i := \sum_{k=0}^{N-1} f_k e^{-\frac{2\pi i k i}{N}} \text{ mit } \mathbf{i}^2 = -1$$

( $\mathbf{i}$  bezeichnet hier die komplexe Wurzel von  $-1$ .) Dann gilt umgekehrt:

$$f_k = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{F}_i e^{+\frac{2\pi i k i}{N}}$$

#### A.1.2 Darstellung von Faltung mit Matrizen

Sei  $\vec{f}, \vec{h} \in \mathbf{R}^N$ . Die diskrete Faltung von  $\vec{f}$  mit  $\vec{h}$  lässt sich dann durch Multiplikation mit einer Matrix darstellen:

$$\underbrace{\begin{pmatrix} \tilde{g}_0 \\ \tilde{g}_1 \\ \vdots \\ \tilde{g}_{2N-1} \end{pmatrix}}_{\in \mathbf{R}^{2N-1}} = \underbrace{\begin{pmatrix} h_0 & 0 & \cdots & 0 \\ h_1 & h_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ h_{N-1} & & & h_0 \\ 0 & \ddots & & h_1 \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & 0 & h_{N-1} \end{pmatrix}}_{\in \mathbf{R}^{(2N-1) \times N}} * \underbrace{\begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix}}_{\in \mathbf{R}^N} \quad (\text{A.1})$$

Der Ergebnisvektor hat eine andere Größe als  $\vec{f}$ , da die Faltung die ‘‘Ausdehnung’’ von  $\vec{f}$  vergrößert. Die Matrix ist allerdings nicht quadratisch und daher nicht invertierbar.

Falls sich allerdings die ausschlaggebenden Werte der Matrix in der Nähe der Hauptdiagonalen konzentrieren, so läßt sich die Matrix in Gleichung (A.1) auch durch eine zirkulante Matrix<sup>1</sup> annähern[AH97]:

$$\underbrace{\begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{pmatrix}}_{\in \mathbf{R}^N} = \underbrace{\begin{pmatrix} h_0 & h_{N-1} & \cdots & h_1 \\ h_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & & h_{N-1} \\ h_{N-1} & \cdots & h_1 & h_0 \end{pmatrix}}_{\in \mathbf{R}^{N \times N}} * \underbrace{\begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix}}_{\in \mathbf{R}^N}$$

Dies entspricht der Annäherung der Faltung durch eine zyklische Faltung.

### A.1.3 Eigenraum zirkulanter Matrizen

Sei  $\mathcal{A} \in \mathbf{R}^{N \times N}$  eine zirkulante Matrix mit:

$$\mathcal{A} = \begin{pmatrix} a_0 & a_{N-1} & \cdots & a_1 \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & & a_{N-1} \\ a_{N-1} & \cdots & a_1 & a_0 \end{pmatrix}$$

Dann hat  $\mathcal{A}$  die Eigenwerte

$$\lambda_k = \sum_{j=0}^{N-1} a_j e^{-\frac{2\pi k j \mathbf{i}}{N}} \quad (\text{A.2})$$

Man beachte, daß die Eigenwerte der eindimensionalen diskreten Fouriertransformation der Folge  $(a_i)$  entsprechen.

$\mathcal{A}$  wird von einer sogenannten Fourier-Kern Matrix  $\mathcal{F}$  in den Eigenraum transformiert[AH97]:

$$\mathcal{A} = \mathcal{F} Q_A \mathcal{F}^{-1} \text{ mit } \mathcal{F}^{-1} := \left( e^{\frac{2\pi u_{kl} \mathbf{i}}{N}} \right) \text{ und } \mathcal{U} = (u_{kl}) := \begin{pmatrix} 0 & 0 & 0 & \cdots \\ 0 & 1 & 2 & \cdots \\ 0 & 2 & 4 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (\text{A.3})$$

$Q_A$  ist dann eine Diagonalmatrix mit den Eigenwerten  $\lambda_i$  in der Hauptdiagonalen:

$$Q_A = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$$

Ähnliches gilt für Block-zirkulante Matrizen. Diese werden von einer Block-Fourier-Kern Matrix in den Eigenraum transformiert. [AH97]

### A.1.4 EM-Algorithmus

Eine ausführliche Beschreibung des *EM*-Algorithmus findet sich in im Buch[Kay93], auf dessen Inhalt auch dieses Kapitel beruht.

#### Gegeben:

<sup>1</sup>Zur Definition von zirkulant vgl. [Bro00]

- Ein “unvollständiger” Datensatz  $\vec{y}$ , der keinen Ansatz mit Maximum-Likelihood erlaubt.
- Eine Projektionsfunktion  $t$ , die einen “vollständigen” Datensatz  $\vec{z}$  auf einen “unvollständigen” Datensatz  $\vec{y} = t(\vec{z})$  abbildet.
- Die von dem Parametersatz  $\theta$  abhängige Verteilungsdichtefunktion  $p_z(\vec{z}; \theta)$ , die den stochastischen Prozeß, mit dem der Datensatz gebildet wurde, beschreibt.
- Die bedingte Verteilungsdichtefunktion  $p(\vec{z}|\vec{y}; \theta)$  zu  $\vec{z}$  bei bekanntem  $\vec{y}$ .

Dabei würde ein “vollständige” Datensatz  $\vec{z}$  einen Maximum-Likelihood Ansatz für  $\theta$  erlauben. Die Wahl der stochastischen Variable  $\vec{z}$  ist im Prinzip frei.

**Gesucht:**

- Parametersatz  $\theta$ , der den Erwartungswert  $E_z$  der Likelihood Funktion  $p(\vec{z}|\vec{y}; \theta)$  maximiert.

Bei geeigneter Wahl des “vollständigen” Datensatzes (d.h. die unten aufgeführten Schritte lassen sich mathematisch und numerisch lösen), führt der Ansatz mit dem *EM*-Algorithmus zu einem Optimierungsverfahren für  $\theta$ .

Der fertige *EM*-Algorithmus führt in einer Schleife zwei Schritte aus:

- **E-Schritt:** Bestimme den Erwartungswert für die Log-Likelihood Funktion bei gegebenen “unvollständigen” Datensatz  $\vec{y}$  und angenommenen Parametersatz  $\theta_p$ :

$$\begin{aligned} U(\theta, \theta_p) &:= E_z \{ \ln p_z(z; \theta) | y; \theta_p \} \\ &= \int \ln p_z(z; \theta) p(z|y; \theta_p) dz \end{aligned}$$

- **M-Schritt:** Suche  $\theta_{p+1}$ , für das  $U(\theta, \theta_p)$  maximal wird:

$$\theta_{p+1} = \underset{\theta}{\operatorname{argmax}} U(\theta, \theta_p)$$

- Wiederhole diese beiden Schritte ( $p \rightarrow p + 1$ ), bis  $\theta_p$  konvergiert.

Der Vorteil des *EM*-Algorithmus gegenüber anderen Optimierungsverfahren ist, daß die Konvergenz beweisbar ist (man kann zeigen, daß  $U(\theta_{p+1}, \theta_p)$  monoton mit  $p$  wächst). Allerdings konvergiert der *EM*-Algorithmus nicht notwendigerweise gegen das globale Maximum von  $U(\theta, \theta)$ .

### A.1.5 E-Schritt hier

Zunächst ist die Log-Likelihood Funktion

$$\ln p_z(\vec{z}; \theta) = -N^2 \ln(2\pi) - \frac{1}{2} \ln |\mathcal{C}| - \frac{1}{2} \vec{z}^\top \mathcal{C}^{-1} \vec{z} \quad (\text{A.4})$$

zu berechnen. Dazu wird zunächst die Determinante von  $\mathcal{C}$  bestimmt:

$$|\mathcal{C}| = \begin{vmatrix} \mathcal{C}_f & \mathcal{C}_f H^\top \\ H\mathcal{C}_f & H\mathcal{C}_f H^\top + \mathcal{C}_n \end{vmatrix} \quad (\text{A.5})$$

$$= \det(\mathcal{C}_f(H\mathcal{C}_f H^\top + \mathcal{C}_n) - H\mathcal{C}_f \mathcal{C}_f H^\top) \quad (\text{A.6})$$

$$= \det(\mathcal{C}_f \mathcal{C}_n) = |\mathcal{C}_f| |\mathcal{C}_n| \quad (\text{A.7})$$

Die Verdrehung der Matrizenmultiplikation von  $\mathcal{C}_f$  und  $H$  in Gleichung (A.5) ist möglich, da angenommen wird, daß die Matrizen zirkulant sind (i.e. sich mit Fourier-Kern Matrizen diagonalisieren lassen):

$$q\mathcal{C}_f H = \mathcal{F}Q_f Q_H \mathcal{F}^{-1} = \mathcal{F}Q_H Q_f \mathcal{F}^{-1} = H\mathcal{C}_f$$

Wenn man weißes Rauschen voraussetzt, ist  $\mathcal{C}_n$  eine Diagonalmatrix und es gilt:

$$|\mathcal{C}_n| = (\sigma_n^2)^{N^2}$$

. Da die Diagonalmatrix  $Q_f$  die Elemente der Fouriertransformation  $\mathbf{S}_f$  der Kovarianzmatrix  $\mathcal{C}_f$  als Diagonale hat, gilt für die Determinante von  $\mathcal{C}_f$ :

$$|\mathcal{C}_f| = |\mathcal{F}Q_f \mathcal{F}^{-1}| = |Q_f| = \prod_{kl} \mathbf{S}_f(k, l)$$

Außerdem muß in Formel (A.4) die Inverse von  $\mathcal{C}$  bestimmt werden. Im Buch [HJ85] findet sich die Formel, für die Inverse einer zusammengesetzten Matrix:

$$\begin{aligned} \mathcal{A}^{-1} &= \begin{pmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{21} & \mathcal{A}_{22} \end{pmatrix}^{-1} \\ &= \begin{pmatrix} (\mathcal{A}_{11} - \mathcal{A}_{12}\mathcal{A}_{22}^{-1}\mathcal{A}_{21})^{-1} & \mathcal{A}_{11}^{-1}\mathcal{A}_{12}(\mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12} - \mathcal{A}_{22})^{-1} \\ (\mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12} - \mathcal{A}_{22})^{-1}\mathcal{A}_{21}\mathcal{A}_{11}^{-1} & (\mathcal{A}_{22} - \mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12})^{-1} \end{pmatrix} \end{aligned}$$

Anwenden dieser Formel auf (2.12) ergibt:

$$\mathcal{C}^{-1} = \begin{pmatrix} (\mathcal{E} - H^\top(H\mathcal{C}_f H^\top + \mathcal{C}_n)^{-1}H\mathcal{C}_f)^{-1}\mathcal{C}_f^{-1} & -H^\top\mathcal{C}_n^{-1} \\ -\mathcal{C}_n^{-1}H & \mathcal{C}_n^{-1} \end{pmatrix} \quad (\text{A.8})$$

Damit läßt sich  $\ln p_z(\vec{z}; \theta)$  berechnen, wenn  $\vec{z}$  und  $\theta$  bekannt sind. Um den E-Schritt zu vervollständigen muß jetzt noch nach Anhang A.1.4

$$E_z\{\ln p_z(\vec{z}; \theta) | \vec{g}; \theta_p\}$$

bestimmt werden. Dazu wird (A.8) in (A.4) eingesetzt. Außerdem werden die Vorkommnisse von  $\vec{f}$  und  $\vec{f}\vec{f}^\top$  durch ihre Erwartungswerte substituiert [FW88]. Die Erwartungswerte sind der Mittelwert und die Kovarianz der bedingten Normalverteilung<sup>2</sup> (siehe [FKN90]), die auch wieder eine Normalverteilung ist:

$$E\{\vec{f} | \vec{g}\} = \mathcal{C}_f H^\top (H\mathcal{C}_f H^\top + \mathcal{C}_n)^{-1} \vec{g} =: \vec{m}_{f|g} \quad (\text{A.9})$$

$$E\{\vec{f}\vec{f}^\top | \vec{g}\} = \mathcal{C}_f - \mathcal{C}_f H^\top (H\mathcal{C}_f H^\top + \mathcal{C}_n)^{-1} H\mathcal{C}_f =: \mathcal{C}_{f|g} \quad (\text{A.10})$$

<sup>2</sup>engl: conditional (multivariate) normal distribution

Die so gebildete Formel für  $U(\theta, \theta_p) = E_z\{\ln p_z(\vec{z}; \theta) | \vec{g}; \theta_p\}$  ist hier nicht wiedergegeben. Sie wird mit Fourierkernen diagonalisiert bzw. in den Fourierbereich transformiert. Man erhält:

$$\begin{aligned} \ln p_z(z; \theta) = & -N^2 \ln(2\pi) - \frac{N^2}{2} \ln(\sigma_n^2) + \sum_{kl} \left[ -\frac{1}{2} \ln \mathbf{S}_f(k, l) - \frac{1}{2N^2} \mathbf{G}^* \mathbf{G} (\sigma_n^2)^{-1} \right. \\ & + \frac{1}{N^2} \operatorname{Re}\{\mathbf{G}^* \Delta(\sigma_n^2)^{-1} \mathbf{M}_{f|g}^{(p)}\} - \frac{1}{2} \mathbf{S}_f(\Delta \Delta^* (\sigma_n^2)^{-1} + \mathbf{S}_f^{-1}) \\ & \left. - \frac{1}{2N^2} (\mathbf{M}_{f|g}^{(p)})^* (\Delta \Delta^* (\sigma_n^2)^{-1} + \mathbf{S}_f^{-1}) \mathbf{M}_{f|g}^{(p)} \right] \end{aligned} \quad (\text{A.11})$$

Der Übersicht halber wurden die Indizes  $k$  und  $l$  bis auf eine Ausnahme weggelassen.

## A.2 Povray-Dateien

Das Raytracing<sup>3</sup> Programm *Povray* wurde verwendet, um die Ergebnisse photorealistisch zu visualisieren.

Das Programm *Povray* kann 16-Bit Höhenkarten über Farbbilder importieren. Dabei wird das höherwertige Byte im Rot- und das niederwertige Byte im Grün-Kanal des Farbbildes abgelegt. Außerdem erlaubt *Povray* die Projektion von Texturen auf beliebige 3-D Modelle. Für mehr Information siehe [Pov].

Die *Povray*-Eingabedatei, mit der die 3-D Bilder 3.2, 3.13 und 4.3 erzeugt wurden, sieht folgendermaßen aus:

```
#include "colors.inc"
global_settings { ambient_light White } // Weißes Streulicht
background { color White } // Weißer Hintergrund
camera { // Kameraposition und Blickrichtung
    location < 0.2, 0.5, -0.4 >
    look_at < 0.0, 0.0, 0.0 >
    angle 60
}
light_source { // Lichtquelle
    < 1000, 1000, -1000 >
    color White
    fade_distance 1400
}
height_field {
    png "gitterHF.png" // Die Höhenkarte als 16-Bit PNG-Datei
    smooth // Interpolation
    pigment {
        image_map {
            png "gitterDV.png" // Das Tiefenbild als Textur
            once
            map_type 0
        }
        rotate 90*x
    }
}
```

---

<sup>3</sup>deutsch: Strahlverfolgung

```

finish {
  ambient    0.2
  diffuse    0.8
  reflection 0.3
  specular   0.6
  roughness  0.01
}
translate < -0.5, -0.5, -0.5 >
// Korrekten Maßstab herstellen.
// 0.537: Kamerabildbreite (mm)
// 0.403: Kamerabildhöhe (mm)
// 0.1 : Höhe der Fokussierreihe (mm)
scale < 0.537, 0.1, 0.403 >
}

```

Das Programm *Povray* läßt auch die Simulation von begrenzter Schärfentiefe zu. Daher kann man *Povray* verwenden, um Fokussierreihen zu erzeugen. Die Eingabedatei, um z.B. die Fokussierreihe einer kleinen schwarzen Kugel zu erhalten, sieht so aus:

```

#include "colors.inc"
global_settings { ambient_light White } // Weißes Streulicht
background { color White } // Weißer Hintergrund
camera {
  location      < -10.0, 0.0, 0.0 >
  look_at      < 0.0, 0.0, 0.0 >
  focal_point  < clock - 0.5, 0.0, 0.0 > // Punkt auf der Fokalebene
  aperture 50.0 // Blendenöffnung
  blur_samples 600
  confidence 0.95
  variance 1/256
}
sphere { <0,0,0>, 0.05 // Die schwarze Kugel
  pigment{ color Black }
  finish { ambient 1.0 }
}

```

Die genaue Bedeutung der Schlüsselwörter findet sich im *Povray*-Benutzerhandbuch[Pov]. Das "clock"-Schlüsselwort ist eine Variable, deren Wert monoton von 0.0 bis 1.0 steigt, wenn man eine Serie von Bildern berechnen läßt.

### A.3 Mexican Hat Wavelet

Das *mexican hat*<sup>4</sup>-Wavelet sieht aus, wie ein mexikanischer Hut (siehe Abb. A.1). Die Formel für dieses Wavelet ist:

$$\psi(x) = \frac{2}{\sqrt{3}\pi^{\frac{1}{4}}}(1-x^2)e^{-\frac{1}{2}x^2}$$

Das zugehörige Spektrum ist[GW98]:

$$\widehat{\psi}(\xi) = K\xi^2 e^{-2\pi^2\xi^2} \tag{A.12}$$

---

<sup>4</sup>deutsch.: mexikanischer Hut

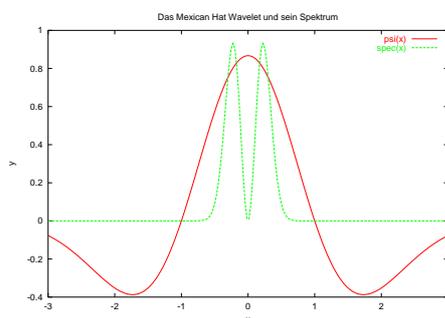


Abbildung A.1: Das Mexican Hat Wavelet (rot) und sein Spektrum (grün)

## A.4 Varianz

Die Varianz  $\sigma$  der Werte  $x_1, x_2, \dots, x_n$  kann aus den Mittelwert der Werte  $\bar{x}$

$$\bar{x} := \frac{1}{n} \sum_{i=1}^n x_i$$

und dem Mittelwert der Quadrate  $\overline{x^2}$

$$\overline{x^2} := \frac{1}{n} \sum_{i=1}^n x_i^2$$

mit der bekannten Formel

$$\sigma = \sum_{i=1}^n (x_i - \bar{x})^2 = \overline{(x - \bar{x})^2} = \frac{1}{n} \sum_{i=1}^n x_i^2 - \left( \frac{1}{n} \sum_{i=1}^n x_i \right)^2 \quad (\text{A.13})$$

berechnet werden.

## A.5 Methode kleinster Quadrate

### A.5.1 Gauß-Markov Modell und Parameterschätzung

Die Methode kleinster Quadrate und das Gauß-Markov Modell werden ausführlich in [Sti00] beschrieben. Die Designmatrix  $\mathcal{H}$  und der Parametervektor  $\vec{a}$  ergeben den Vektor der erwarteten Werte  $\mathcal{H}\vec{a}$ . Beobachtet dagegen wird  $\vec{b}$ . Der Beobachtungsfehler wird durch  $\vec{\epsilon}$  modelliert:

$$\mathcal{H}\vec{a} = \vec{b} + \vec{\epsilon}$$

Dabei wird angenommen, daß  $\vec{\epsilon}$  Realisation eines mittelwertfreien, normalverteilten Zufallsprozesses mit Kovarianz  $\sigma\mathcal{I}$  ist<sup>5</sup> (unkorreliert).

Bei der Methode der kleinsten Quadrate wird

$$J(\vec{a}) = \vec{\epsilon}^\top \vec{\epsilon} = (\mathcal{H}\vec{a} - \vec{b})^\top (\mathcal{H}\vec{a} - \vec{b})$$

<sup>5</sup> $\mathcal{I}$  sei die Einheitsmatrix

minimiert. Mit  $\left. \frac{\delta J(\vec{a})}{\delta \vec{a}} \right|_{\hat{\vec{a}}} \stackrel{!}{=} 0$  erhält man als Schätzwert  $\hat{\vec{a}}$ :

$$\hat{\vec{a}} = (\mathcal{H}^\top \mathcal{H})^{-1} \mathcal{H}^\top \vec{b} \quad (\text{A.14})$$

Dieser Schätzer ist der beste lineare, erwartungstreue Schätzer für das Gauß-Markov Modell [Sti00].

### A.5.2 Regressions-Parabel

**Gegeben** sei eine Menge von Punkten  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ .

**Gesucht** seien die Parameter  $a_0$ ,  $a_1$  und  $a_2$ , so daß

$$J((a_0, a_1, a_2)^\top) = \sum_{i=1}^n (a_0 x_i^2 + a_1 x_i + a_2 - y_i)^2 \text{ minimal wird.}$$

**Lösung:** Man wähle

- $\mathcal{H} = \begin{pmatrix} x_1^2 & x_1 & 1.0 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n & 1.0 \end{pmatrix}$  und
- $\vec{b} = (y_1, y_2, \dots, y_n)^\top$ .

Dann erhält man mit Formel (A.14) den gesuchten Parametervektor  $\hat{\vec{a}} = (\hat{a}_0, \hat{a}_1, \hat{a}_2)^\top$ . Der Extremwert der Regressions-Parabel ist bei  $x = -\frac{\hat{a}_1}{2\hat{a}_0}$  für  $\hat{a}_0 \neq 0$ .

### A.5.3 Regressions-Ebene

**Gegeben** sei die Höhenkarte  $d : \begin{cases} \{1, \dots, K\} \times \{1, \dots, L\} & \rightarrow \mathbf{R} \\ (x, y) & \mapsto d(x, y) \end{cases}$ .

**Gesucht** seien die Parameter  $a_0$ ,  $a_1$  und  $a_2$ , so daß

$$J((a_0, a_1, a_2)^\top) = \sum_{x=1}^K \sum_{y=1}^L (a_0 x + a_1 y + a_2 - d(x, y))^2 \text{ minimal wird.}$$

**Lösung:** Man wähle

- $\mathcal{H} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ \vdots & \vdots & \vdots \\ K & L & 1 \end{pmatrix}$  und
- $\vec{b} = (d(1, 1), d(2, 1), \dots, d(K, L))^\top$ .

Dann erhält man mit Formel (A.14) den gesuchten Parametervektor  $\hat{\vec{a}} = (\hat{a}_0, \hat{a}_1, \hat{a}_2)^\top$ .

Das mittlere Fehlerquadrat ist  $\frac{1}{KL-3} J(\hat{\vec{a}})$ , da drei Parameter frei wählbar sind.

# Literaturverzeichnis

- [AH97] ANDREW, H. C. und B. R. HUNT: *Digital image restoration*. Prentice-Hall signal processing series. Prentice-Hall, Englewood Cliffs, London, 1997.
- [Bas95] BASCHANG, CHRISTOF: *Rekonstruktion von 3D-Bildern aus Fokussierien*. Diplomarbeit, März 1995. Institut für Meß- und Regelungstechnik, Universität Karlsruhe (TH), Prof. Dr.-Ing. F. Mesch. <http://www-mrt.mach.uni-karlsruhe.de/Mitarbeiter/puente/arbeiten/baschang.html>.
- [Bro00] BROOKES, MIKE: *Matrix Reference Manual*. WWW-Seite, 2000. <http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html>.
- [Dau92] DAUBECHIES, INGRID: *Ten Lectures on Wavelets*, Band 61 der Reihe *CBMS-NSF regional conference series in applied mathematics*. Society for Industrial and Applied Mathematics, 1992.
- [Dav] DAVIES, ROBERT: *newmat-library*. WWW-Seite. <http://webnz.com/robert/>.
- [Die96] DIEBEL, WOLFGANG: *Multifokussierung und Objekterkennung im Mikrobereich*. Studienarbeit, August 1996. Institut für Prozeßrechen-technik und Robotik, Universität Karlsruhe (TH), Prof. Dr.-Ing. Rembold.
- [ECB02] *Homepage der Europäischen Zentralbank*. WWW-Seite, 2002. <http://www.euro.ecb.int/>.
- [EL93] ENS, JOHN und PETER LAWRENCE: *An Investigation of Methods for Determining Depth from Focus*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(2):97–108, Februar 1993.
- [Fav99] FAVARO, PAOLO: *Shape from Focus/Defocus*. Technischer Bericht, Washington University, Department of Electrical Engineering, Electronic Signals and Systems Research Lab, One Brookings dr. 1127, St. Louis - MO 63130, USA, 1999? <http://ee.wustl.edu/~fava/DFDTutorial.pdf>.
- [FJ] FRIGO, MATTEO und STEVEN G. JOHNSON: *Fastest Fourier Transform in the West*. WWW-Seite. MIT, Library and more references at <http://www.fftw.org/>.

- [FKN90] FANG, KAI-TAI, SAMUEL KOTZ und KAI-WANG NG: *Symmetric Multivariate and Related Distributions*, Band 36 der Reihe *Monographs on Statistics and Applied Probability*. Chapman and Hall Ltd, 11 New Fetter Lane, London EC4P 4EE, 1990. Proof for the conditional density-function of a multivariate elliptical distribution.
- [Fra00] FRASER, SCOTT E.: *An optical coherence microscope for 3-dimensional imaging in developmental biology*. Optics Express, 6(7):136–146, März 2000. Biological Imaging Center, California Institute of Technology, 1200 East California Boulevard, Pasadena, CA 91125 <http://www.opticsexpress.org/oearchive/source/19250.htm>.
- [FW88] FEDER, MEIR und EHUD WEINSTEIN: *Parameter Estimation of Superimposed Signals Using the EM Algorithm*. IEEE Transactions on Acoustics, Speech and Signal Processing, 36(4):477–489, April 1988.
- [GW98] GASQUET, C. und P. WITOMSKI: *Fourier Analysis and Applications*, Band 30 der Reihe *Texts in Applied Mathematics*. Springer, 1998.
- [Hab95] HABERÄCKER, PETER: *Praxis der Digitalen Bildverarbeitung und Mustererkennung*. Carl Hanser Verlag, München Wien, 1995.
- [Hec99] HECHT, EUGENE: *Optik*. R. Oldenbourg Verlag, Rosenheimer Straße 145, D-81671 München, second Auflage, 1999. This is a translation. There also exists an english original (Addison Wesley: Optics).
- [HJ85] HORN, ROGER A. und CHARLES A. JOHNSON: *Matrix Analysis*. Cambridge University Press, The Pitt Building, Trumpington Street, Cambridge CB2 1RP, 32 East 57th Street, New York, NY 10022, USA, 1985. Formula for inverse of partitioned matrix.
- [JH96] JOHNSON, ANDREW E. und MARTIAL HERBERT: *Matching Oriented Points*. PDF-Dokument, Mai 1996. Carnegie Mellon University, keywords:3-D object recognition, oriented point,structural indexing.
- [Kay93] KAY, STEVEN M.: *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall Signal Processing Series. Prentice Hall International, Inc., A Simon & Schuster Company, Englewood Cliffs, New Jersey 07632, 1993. Description of EM-algorithm.
- [Kie01] KIEFER, MATTHIAS: *3D-Koordinatenbestimmung von Mikroroboter-greifern und Mikrowerkstücken mit Hilfe von Bildverarbeitung und Projektion von optischen Rastern*. Diplomarbeit, Juli 2001. Institut für Prozeßrechenstechnik, Automation und Robotik, Universität Karlsruhe (TH) Prof. H. Wörn und Institut für Theoretische Teilchenphysik, Prof. Dr. H.-M. Staudenmaier <http://wwwipr.ira.uka.de/~kiefer/diplomarbeit.pdf>.
- [KL91] KATSAGGELOS, A.K. und K.T. LAY: *Maximum Likelihood Blur Identification and Image Restoration Using the EM Algorithm*. IEEE Transactions on Signal Processing, 39(3):729–733, März 1991.

- [Leó99] LEÓN, FERNANDO PUENTE: *Automatische Identifikation von Schußwaffen*. Nummer 787 in *Fortschritt-Berichte 8*. VDI Verlag Gmbh, Düsseldorf, 1999. Doktorarbeit am Institut für Meß- und Regeltechnik, Universität Karlsruhe (TH).
- [MC99] MARKHAM, JOANNE und JOSÉ ANGEL CONCHELLO: *Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur*. *Journal of The Optical Society of America A*, 16(10):2377–2391, Oktober 1999. parametric blind deconvolution <http://www.opticsinfobase.org/abstract.cfm?id=1290>.
- [NN90] NAYAR, SHREE K. und YASUO NAKAGAWA: *Shape from Focus: An Effective Approach for Rough Surfaces*. IEEE, 1990.
- [NN94a] NAYAR, SHREE K. und YASUO NAKAGAWA: *Shape from Focus*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831, August 1994.
- [NN94b] NOGUCHI, MINORI und SHREE K. NAYAR: *Microscopic Shape from Focus Using Active Illumination*. IEEE, Seiten 147–152, 1994. Department of Computer Science, Columbia University, New York, NY 10027 <http://citeseer.nj.nec.com/noguchi94microscopic.html>.
- [OOT<sup>+</sup>00] OHBA, K., C. ORTEGA, K. TANIE, G. RIN, R. DANGI, Y. TAKEI, T. KANEKO und N. KAWAHARA: *Real-Time Micro Observation Technique for Tele-Micro-Operation*. In: *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000. Mechanical Engineering Lab., 1-2 Namiki, Tsukuba, Japan.
- [Pov] *Persistence OF Vision RAYtracer*. WWW-Seite. <http://www.povray.org/>.
- [Qt] *Qt cross-platform C++ GUI*. WWW-Seite. Trolltech, Oslo, Norway, <http://www.trolltech.com>.
- [RBV98] ROCA, XAVIER, XAVIER BINEFA und JORDI VITRIÀ: *New auto-focusing algorithm for cytological tissue in a microscopy environment*. *Optical Engineering*, 37(2):635–641, Februar 1998. Universitat Autònoma de Barcelona, Computer Vision Center, Edifici O, 08193 Bellaterra, Spain <http://ojps.aip.org/journal.cgi/dbt?KEY=OPEGAR&Volume=37&Issue=2>.
- [Rön96] RÖNNEBERG, GERRIT: *Mikrostrukturen im Fokus*. *F&M Zeitschrift für Elektronik, Optik und Mikrosystemtechnik*, 104(10):715–720, 1996. Es existiert hierzu ein nicht veröffentlichter ausführlicherer Aufsatz.
- [SBGH98] SCHRADER, MARTIN, KARSTEN BAHLMANN, GÜNTER GIESE und STEFAN W. HELL: *4Pi-Confocal Imaging in Fixed Biological Specimens*. *Biophysical Journal*, 75:1659–1668, Oktober 1998. [http://www.mpibpc.gwdg.de/abteilungen/200/publications/pdf/Biophys.\\_J.\\_75\\_1659-1668.pdf](http://www.mpibpc.gwdg.de/abteilungen/200/publications/pdf/Biophys._J._75_1659-1668.pdf).

- [Sch97] SCHEUERMANN, TORSTEN: *Berührungslose Gestaltvermessung von Mikrostrukturen durch Fokussuche*, Band 13 der Reihe *Wissenschaftliche Schriftenreihe des ICT*. Fraunhofer Institut für Chemische Technologie (ICT), Joseph-von-Fraunhofer-Straße 7, 76327 Karlsruhe, Februar 1997. engl. title: Profilometry of microstructures by depth from focus techniques.
- [Sti00] STILLER, CHRISTOPH: *Bildverarbeitung*. Vorlesung, 2000. Vorlesung über Bildbearbeitung and der Universität Karlsruhe <http://i21www.ira.uka.de/Veranstaltungen/Bildverarbeitung/>.
- [vH00] HEESCH, DIMITRI VAN: *doxygen*. WWW-Seite, 2000. <http://www.stack.nl/~dimitri/doxygen/index.html>.
- [ZZW00] ZHANG, YANI, YING ZHANG und CHANGYUN WEN: *A new focus measure method using moments*. *Image and Vision Computing*, 18(12):959–965, 2000. School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore, Singapore 639798 <http://www.elsevier.nl/gej-ng/10/22/73/40/36/27/abstract.html>.

# Index

- $\vec{a}$ , 68
- Abtasten, 12
- Auflichtbeleuchtung, 25
- Auflösung, 8
  - stufen, 30
- Auflösungsverbesserung, 35–37
- Ausblick, 57–59
- autoconf, 40
- Autofokus
  - von Photoapparaten, 13
- Autofokussierung, 13
- automake, 40
  
- $\vec{b}$ , 68
- Beleuchtung
  - Zwei-Kanal-, 8, 25
- Beugungsmuster, 17
  
- $\mathcal{C}$ , 18
- C++, 39
  - Header-Dateien, 39
- $c$ , 35
- $\mathcal{C}_f$ , 18
- $\mathcal{C}_n$ , 18
  
- $d$ , 13
- deconvolution
  - blind, 19
- Degradierung, 17
  - des Bildes, 17
  - Vergleich der, 14
- $\Delta$ , 17
- Designmatrix, *siehe* Matrix
- Diagonalmatrix, *siehe* Matrix
- Dokumentation
  - HTML-, 40
- doxygen, 39
  
- $\vec{e}$ , 68
- Eigenraum, 17
  
- Eigenwerte, 18
- Ergebnis, 48–55
- exportieren, 42
  
- $F$ , 18
- $\mathbf{F}$ , 18
- $\mathcal{F}$ , 17, 63
- $f$ , 17
- Faltung, 14, 17
  - blinde inverse, 19–21
  - inverse, 16–19
- Fehler
  - maß, 44
  - systematischer, 24, 29–30
- Fenster
  - Haupt-, 40
  - Unter-, 42
- FFTW, 21
- Filter, 25
  - einsatz, 8
- Fokalebene, 8, 13, 14
- Fokussiererei, 8–9
  - simulierte, 48
- Fourier
  - Kern, 17, 18, 63
  - optik, 16–17
  - diskrete -transformation, 62
  
- $\mathbf{G}$ , 18
- $g$ , 13, 17
- Gradientenbeträge, 13
- Grauwertvarianz, 14
  - normierte, 30
  
- $H$ , 17
- $\mathcal{H}$ , 68
- $h$ , 17
- Höhenkarte
  - 16-Bit, 66
- Hauptspeicher, 42

- Hilfsmuster, 25
- Höhenkarte, 13
- Höhenkarte
  - Bestimmung der, 32–33
- Höhenrelief, 9
- i, 62
- Interpolation
  - des Schärfemaßes, 37
- Kalibrierung
  - Dialog, 43–44
- KDE, *siehe* Qt
- Komplexität, 34
- Komplexität, 21, 33–34
- Kreisscheibe, 35
- Laplace
  - Operator, *siehe* Operator
- Laserscanningmikroskop
  - konfokales, *siehe* Mikroskop
- Laserstrahl, 12
- Latex, 40
- seeHilfsmuster, 25
- Likelihood
  - Log-, 18
  - Maximum-, 18
  - Schätzer, Maximum-, 18–19
- Linux, 39
- $LL(\cdot)$ , 18
- Lochkamera, 17
- Log-Likelihood, *siehe* Likelihood
- Matrix
  - basiert, *siehe* Methoden
  - Design-, 68
  - Diagonal-, 63
  - dünn besetzte, 17–18
  - zirkulante, 18, 63
- Maximum-Likelihood, *siehe* Likelihood
- Meßaufbau, 7–8
- Methoden
  - Matrix-basiert, 14, 15
- mexican hat, 67
- microscope
  - confocal scanning-, 12
- Mikroskop, 7
  - konfokales Laserscanning-, 12
- Momenten, *siehe* Moment
- N, 33
- newmat, 39
- Operator
  - Laplace-, 14
- Ortsfrequenzen, 15
- Parameter
  - Einstellung der, 43
- PDFLaTeX, 40
- Povray, 40, 66
  - Eingabedatei, 66
- Projektion, 14
- Punktantwort, 17, 35
- Pyramide
  - Laplace-, 15
- $Q_H$ , 17
- Qt, 39
  - designer, 39
- Rastern, *siehe* Raster
- Rauschen, 18
- Röntgenkamera, 17
- Rot-Grün Bild, 42
- Schärfemaß
  - Pyramide, 33
  - Interpolation des, 37
  - maximieren eines, 13–14
  - Maximierung eines, 13
- Scharfemaß
  - Maximierung eines, 24–25
- Schärfentiefe, 16
- $S_f$ , 18
- $S_n$ , 18
- Sobel-Operator, 13
- Sobeloperator, 13
- Strahlengang, 25
- Strahlteiler, 12
- Suchbereich, 32
- System
  - bilderzeugendes, 17
  - optisches, 17
- texpower, 40
- Tiefenbild, 16
- Tisch
  - X-Y-, 8
  - Z-, 8
- Toeplitz-Form, 17
- Triangulation, 11

Laser-, 11–12

Unschärfe

natürliche, 17

Vergleich der, 14–15, 35–36

$v$ , 16

Varianz, 68

Vergleich, 48

Wavelet, 58

Wiener-Filter, 19

xfig, 40

**Jan Wedekind**

Adresse: . Scheffelstr. 65, 3.OG, D-76135 Karlsruhe  
Email: ..... jan.wedekind@stud.uni-karlsruhe.de  
WWW: ..... <http://www.uni-karlsruhe.de/~unoh>



Dieser Text ist auch in elektronischer Form im *Elektronischem Volltextarchiv (EVA)*<sup>6</sup> der Universitätsbibliothek Karlsruhe erhältlich. Die zugehörige Software ist frei verfügbar und steht unter der GNU-Lizenz. Die Software ist auf meiner persönlichen Internetseite<sup>7</sup> zu finden (Datei *surftex.tar.gz*).

---

<sup>6</sup>URL: <http://www.ubka.uni-karlsruhe.de/eva/>

<sup>7</sup>im Moment: <http://www.uni-karlsruhe.de/~unoh/surftex/>