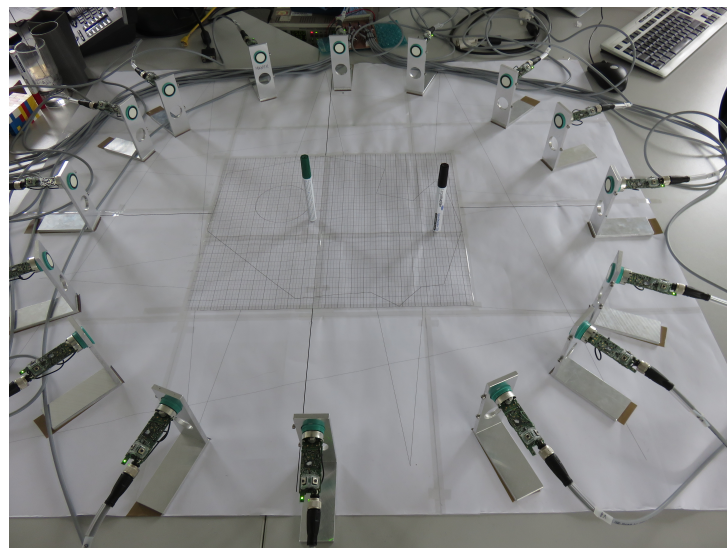




HECTOR SEMINAR



Objekterkennung mit Ultraschall



Abschlussdokumentation der Kooperationsphase 2016/17

Durchgeführt am Institut für Prozessdatenverarbeitung und Elektronik (IPE)
Karlsruher Institut für Technologie (KIT)
Betreut durch Wei Yap Tan

Kai Bartnick

Sergey Lelyakin

Inhaltsverzeichnis

Abstract	2
1 Einleitung	3
2 Material und Methoden	4
2.1 Ultraschall	4
2.2 System zur Objekterkennung mit Ultraschall	4
2.3 Ansteuerungssoftware	8
3 Charakterisierung	9
3.1 Punktspreizfunktion	9
3.2 Messverfahren	11
3.3 Ergebnisse	12
4 Verbesserung der Bilder	15
4.1 Ausgangssituation	15
4.2 Verfahren	15
4.3 Ergebnisse	17
5 Diskussion	22
5.1 Fazit und Diskussion	22
5.2 Ausblick	22
6 Danksagung	24
7 Quellen	25
8 Anhang	26
8.1 Abkürzungsverzeichnis	26
8.2 weitere Ergebnisse	26
8.3 Operationen auf Matrizen in MATLAB	27
8.4 Quelltexte	28
9 Selbständigkeitserklärung	33

Abstract

Airborne ultrasound sensors are widely spread in industrial production. Nowadays, they are mostly used for simple distance detection or to decide whether there is an object, even though there is more potential in ultrasound.

Wei Yap Tan, from the Institute for Data Processing and Electronics (IPE) at Karlsruhe Institute of Technology (KIT), is designing and testing a system, consisting of 16 airborne ultrasound sensors, which are distributed around a certain region of interest (ROI). This system is able to capture two-dimensional pictures of objects in the ROI. In contrast to optical systems, it is independent of the colour or the transparency of the objects which are supposed to be detected. The sensors' positions were optimized to achieve a maximum ROI coverage and a minimum full width at half maximum value (FWHM) of the point spread function (PSF).

In our work we characterized the detecting system with a special focus on the FWHM value of the PSF. As a consequence of the characterization, a correction procedure of the images taken by the system is proposed and tested. The procedure, we developed in our work, uses the distribution of brightness, which is known after the characterization to increase the contrast of the image, especially for two objects.

1 Einleitung

In der medizinischen Diagnostik wird Ultraschall als bildgebendes Verfahren genutzt, zum Beispiel in der Brustkrebsfrüherkennungstechnologie 3D ultrasound computer tomography (3D USCT), die am Institut für Prozessdatenverarbeitung und Elektronik (IPE) des Karlsruher Institut für Technologie (KIT) entwickelt wird. Das Verfahren benutzt eine dreidimensionale Anordnung von Ultraschallsensoren um ein Wasserbecken. Dies ermöglicht eine dreidimensionale Aufnahme der weiblichen Brust, um eine frühzeitige Diagnose von Brustkrebs zu ermöglichen.

In der Industrie hingegen wird Ultraschall bisher nur für wenige Zwecke eingesetzt, zum Beispiel um Entfernungen zu messen oder das Vorhandensein von Objekten zu detektieren. Zur Objekterkennung werden bislang bildgebende Verfahren eingesetzt. Dabei bietet Ultraschall diesen gegenüber eine Reihe von Vorteilen:

Im Gegensatz zu optischen Systemen wird Ultraschall von Objekteigenschaften wie Farbe oder Transparenz nicht beeinflusst. Außerdem tritt im Kontrast zu Röntgen-Apparaten keine Strahlenbelastung auf.

In seiner Dissertation arbeitet Wei Yap Tan daran, aufgrund dieser Vorteile ein System zur Objekterkennung mit Ultraschall zu entwickeln. Dazu ordnet er handelsübliche Industrie-Ultraschallsensoren kreisförmig um ein bestimmtes Areal, die sogenannte Region of Interest (ROI) an. Innerhalb dieser Fläche können einzelne oder auch mehrere Objekte detektiert werden.

Das System ermöglicht es, Aufnahmen von der ROI zu erstellen. Sie zeigen die Position sowie Form und Größe der Objekte, die sich in der ROI befinden.

Durch eine Optimierung der Sensoranordnung ist es Wei Yap Tan gelungen den Bildkontrast und die Halbwertbreite der Punktspreizfunktion¹ zu optimieren. Im Rahmen der Optimierung entwickelte er Vorhersagen für die Halbwertsbreite der Punktspreizfunktion.

Ziel unserer Arbeit ist es, die getroffenen Vorhersagen zur Punktspreizfunktion experimentell zu überprüfen sowie weitere Charakteristiken zu ermitteln, um Eigenschaften des Systems zu bestimmen. Mithilfe der so gewonnenen Eigenschaften sollen die Aufnahmen des Systems verbessert werden.

¹ein Maß für die Auflösung des Systems (Kapitel 3.1, Seite 9)

2 Material und Methoden

2.1 Ultraschall

Schall ist eine longitudinale Druckwelle in einem Medium, in der Regel Luft. Stößt sie auf ein Hindernis, so wird sie von diesem reflektiert. Die Geschwindigkeit der Welle hängt ab von der Beschaffenheit des Mediums, in dem sie sich ausbreitet. Sie liegt für Luft, je nach Temperatur und Druck, bei den auf der Erde üblichen atmosphärischen Verhältnissen zwischen $335 \frac{m}{s}$ und $340 \frac{m}{s}$. In Wasser hingegen beträgt sie ca. $1510 \frac{m}{s}$. Kennt man nun die Schallgeschwindigkeit in dem Medium, in dem sich eine Welle ausbreitet, so kann man, wenn man die Zeit zwischen Aussenden der Welle und dem Eintreffen ihrer Reflexion misst, die Entfernung zu dem Hindernis bestimmen:

$$s = v * t$$

Für Hindernisse, deren Größe weniger als $\frac{\lambda}{4}$ beträgt (Wellenlänge λ), ist die Reflexion allerdings praktisch nicht messbar. Für eine Tonhöhe von 440 Hz (Standard für das a der vierten Oktave) liegt die maximale Größe des erkennbaren Hindernisses demnach bei

$$\begin{aligned}\lambda &= \frac{v}{f} \\ \lambda &= \frac{340 \frac{m}{s}}{440 \text{ Hz}} = 0,772 \text{ m} \\ \frac{\lambda}{4} &= 0,19318 \text{ m}\end{aligned}$$

Dies ist eine Genauigkeit, die für sehr viele Anwendungszwecke nicht verwendbar ist, da eine Abweichung von fast 20 cm, zum Beispiel beim Bau eines Hauses, fatal sein kann. Selbst an der menschlichen Hörgrenze von 20 000 Hz liegt die Genauigkeit bei nur

$$\frac{\lambda}{4} = \frac{340 \frac{m}{s}}{20000 \text{ Hz}} = 0,00425 \text{ m} = 4,25 \text{ mm}$$

was oftmals immer noch zu ungenau ist. Zur technischen Verwendung dieser Entfernungsmethode werden also noch höhere Frequenzen benötigt. Solcher Schall wird Ultraschall genannt, da er vom Menschen nicht hörbar ist. Mit der von uns verwendeten Frequenz von 205 000 Hz beträgt die theoretische Genauigkeit der Entfernungbestimmung

$$\frac{\lambda}{4} = \frac{340 \frac{m}{s}}{205000 \text{ Hz}} \approx 0,000415 \text{ m} = 0,415 \text{ mm}$$

Somit lässt sich mit Ultraschall die Entfernung zu einem Objekt genau bestimmen. Dieses Prinzip spielt für die durchgeführte Arbeit eine grundlegende Rolle

2.2 System zur Objekterkennung mit Ultraschall

Wei Yap Tan am Institut für Prozessdatenverarbeitung und Elektronik (IPE) am Karlsruher Institut für Technologie (KIT) hat ein System entwickelt, das dieses Prinzip nutzt zur Erfassung der Position und Form mehrerer Objekte in einem bestimmten Areal, der sogenannten Region

of Interest (ROI). Das oben beschriebene Prinzip hat den Nachteil, dass nur die erste Reflexion der Ultraschallwelle detektiert wird. Alle Nachfolgenden werden ignoriert, da lediglich die Entfernung zum ersten Objekt in diesem, einfachen, Fall eine Rolle spielt. Auch ist die Richtung, in der das Hindernis sich befindet, unbekannt. Um also die genaue Position mehrerer Objekte zu bestimmen werden mehrere Sensoren benötigt. In unserem Fall wurden Anordnungen mit 14 und 16 Sensoren verwendet.

In der Theorie funktioniert das angewendete Verfahren folgendermaßen: jeder Sensor sendet, einer nach dem anderen, ein Signal aus. Dieses wird von den in der ROI befindlichen Objekten reflektiert und von allen Sensoren aufgenommen. Durch Laufzeitmessung lässt sich nun für jedes Paar von Sensoren berechnen, wie groß die Summe der Entfernungen von ihnen zum Objekt ist. für jedes Sensorenpaar lässt sich somit eine Ellipse bestimmen, welche alle möglichen Positionen des Objektes darstellt. Hat man nun mehrere Ellipsen ermittelt, so werden sie sich in dem Punkt überschneiden, wo sich das beobachtete Objekt befindet. Diese Technik wird als Synthetic Aperture Focusing Technique (SAFT) bezeichnet.

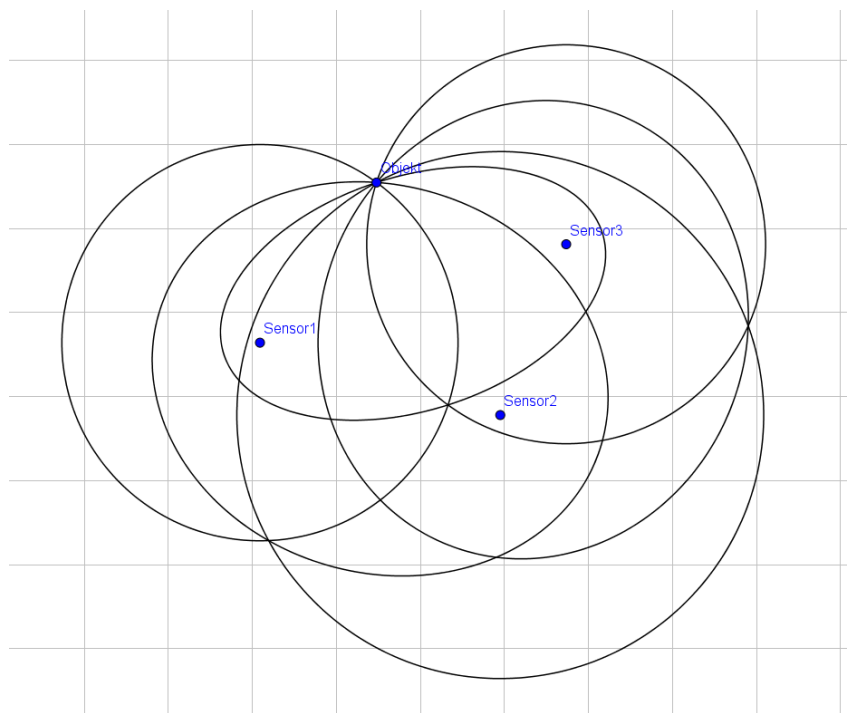


Abb. 1: Theoretisches Beispiel für die Bestimmung der Objektposition

In der Praxis gibt es mehrere Einschränkungen, die es zu beachten gilt. Zum Einen deckt ein Sensor nicht die gesamte Fläche ab, sondern nur einen annähernd kegelförmigen Ausschnitt von ca. 30° .

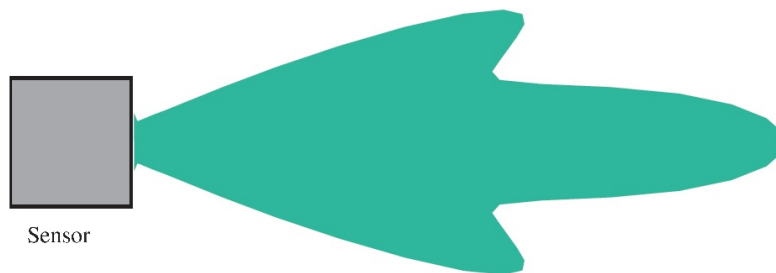


Abb. 2: Angenommener Abdeckungsbereich des Sensors

Zum Anderen ist das Signal, das ein Sensor von einem gegenüberliegenden Sensor erhält, nicht aussagekräftig, da das Echo, das vom Objekt erzeugt wurde, von ihm selbst blockiert wird. Ein Signal, das ein Sensor von dem ihm gegenüberliegenden Sensor bekommt, ist damit höchstwahrscheinlich ein direktes Signal, das nicht von einem Objekt reflektiert wurde und somit für die Messung irrelevant ist. Diese zwei Probleme werden umgangen, indem nur diejenigen Sensorenpaare betrachtet werden, die, von der Mitte des Koordinatensystems aus gesehen, höchstens 30° voneinander entfernt sind.

Ein weitaus schwierigeres Problem ist die Ungenauigkeit der Sensoren. Die gemessene Laufzeit hat immer eine Abweichung durch Messfehler. Daher werden sich die ermittelten Ellipsen nicht alle in einem Punkt kreuzen, sondern in unterschiedlichen Punkten, die dem gesuchten Punkt nahe sind. Um die Position des Objekts dennoch zu bestimmen, wird jede Ellipse "verwischt". Anstatt eine genaue Entfernung anzuzeigen, wird eine Wahrscheinlichkeitsverteilung um die gemessene Entfernung angenommen. Durch Überlagern all dieser "verwischten" Ellipsen bekommt man für jeden Punkt der ROI einen Wert für die Wahrscheinlichkeit eines Objektes an diesem Punkt. Da die Gesamtfläche, die die Objekte in der ROI überdecken, unbekannt ist, ist diese Wahrscheinlichkeit relativ zu den anderen Punkten, keine absolute Wahrscheinlichkeit. Die Werte können also aussagen, dass an einem bestimmten Punkt die Wahrscheinlichkeit für ein Objekt doppelt so groß ist, wie an einem anderen, sind aber nicht in der Lage, anzugeben, dass diese Wahrscheinlichkeit zum Beispiel 50 % beträgt.

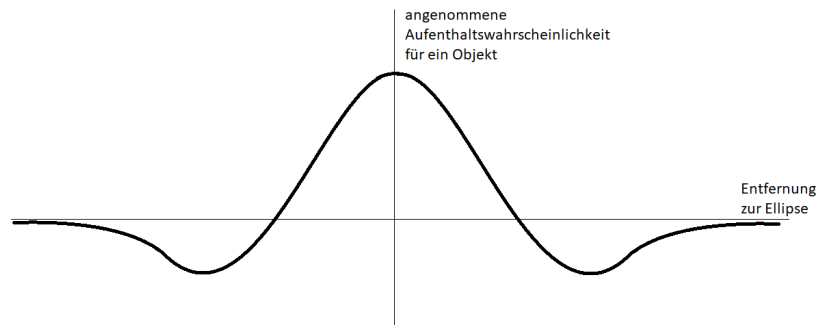


Abb. 3: Verteilung der Wahrscheinlichkeit um die Ursprungsellipse

Eine weitere Maßnahme, die die Aufnahmefähigkeit des Systems verbessern soll, ist, dass die Funktion, die die "Verschwommenheit" der Ellipsen bezeichnet, in einer bestimmten Entfernung zur Ursprungsellipse negative Werte annimmt. Dadurch soll der Kontrast zwischen zwei nahe beieinander gelegenen Punkten verstärkt werden.

Die ermittelte Verteilung relativer Wahrscheinlichkeiten wird im die Messungen verarbeitenden Programm aufgetragen auf ein Raster (Matrix) mit den Dimensionen 1201x1201 (anfänglich 601x601). Die Daten werden somit digitalisiert und können nun vom Computer einfacher verarbeitet werden. Im Folgenden wird durch eine nicht weiter erläuterte Operation der Kontrast zwischen Orten mit hohen Wahrscheinlichkeitswerten und denen mit kleinen verschärft.

Als problematisch erweist sich ebenfalls Schallgeschwindigkeitsvarianz in Abhängigkeit von der Umgebungstemperatur. Der Unterschied in der Schallgeschwindigkeit innerhalb des alltagsüblichen Temperaturbereichs von ca. $5 \frac{m}{s}$ ist zwar nur klein, aber dennoch groß genug, um die Ergebnisse deutlich zu verschlechtern. Dieses Problem wird dadurch umgangen, dass man die Umgebungstemperatur von einem Thermometer abliest und sie von Hand in das Programm eingibt. Da die Korrelation zwischen Temperatur und Schallgeschwindigkeit bekannt ist, kann das Programm letztere ausrechnen und anpassen.

Schließlich besteht das Problem der Ungenauigkeit der Sensorpositionen. Das System kann die Position von Objekten nur bestimmen, wenn es die Position der Sensoren kennt. Diese aber können sich durch zufällige, vom Menschen oder anderen störenden Umwelteinflüssen verursachten, Verschiebungen, verändern. Um die Positionen zu bestimmen, muss das System kalibriert werden. Die Methode, die es hierfür anwendet, basiert ebenfalls auf Ultraschall. Wenn keine Objekte in der ROI vorhanden sind, können gegenüberliegende Sensoren die Entfernung zueinander bestimmen. Die Sensorpositionen werden als die Positionen angenommen, deren Entfernungen am besten auf die gemessenen Entfernungen passen. Sie werden durch ein schrittweises Annäherungsverfahren ermittelt. Diese Methode kann jedoch nur die relativen Positionen der Sensoren zueinander bestimmen. Eine verschobene oder gedrehte Version der realen Sensorpositionen würde genauso als passend akzeptiert werden. Um die tatsächlichen Positionen festzustellen, wird eine zweite Messung durchgeführt, wobei dieses Mal Objekte in der ROI an dem Programm bekannten Stellen platziert werden. Durch deren gemessene Positionen kann die Verschiebung und

Drehung der Sensorpositionen korrigiert werden.

2.3 Ansteuerungssoftware

Passend zu dem beschriebenen System entwickelte Wei Yap Tan bereits eine Ansteuerungssoftware in C#. Für die Berechnungen greift diese auf MATLAB[®]-Funktionen zurück. Die Software ermöglicht es, Aufnahmen mit dem System zu erstellen. Auch die Kalibrierung kann direkt in der Anwendung durchgeführt werden. Des Weiteren ist es möglich, nur die eigentliche Aufnahme zu erstellen, ohne daraus ein Bild zu errechnen. Die Signalverarbeitung sowie die Errechnung des Bildes erfolgt gegebenenfalls in MATLAB.

Dateien werden von der Anwendung in dem von MATLAB lesbaren .m-Format gespeichert. Dies ermöglicht eine direkte und schnelle Weiterverarbeitung, insbesondere von mehreren Aufnahmen. Die fertigen Bilder werden in einer MATLAB-Datei als Matrix gespeichert.

MATLAB ist ein Programm zur Erfassung und numerischen Analyse von Daten. Es besteht aus einer Programmiersprache, einer Entwicklungsumgebung und einer interaktiven Konsole. Es ist speziell für die Arbeit mit naturwissenschaftlichen Daten sowie für numerische Methoden ausgelegt. Ein besonderer Vorteil liegt dabei in einer Vielzahl von möglicher Matrix-Operationen. Auf Matrizen werden mathematische Operationen wie Multiplikation, Division oder auch elementweise Division unterstützt. Daneben sind Methoden der Bildverarbeitung, wie zum Beispiel Interpolationen standardmäßig vorhanden. Daher spielt das Programm für diese Arbeit eine zentrale Rolle.

3 Charakterisierung

3.1 Punktspreizfunktion

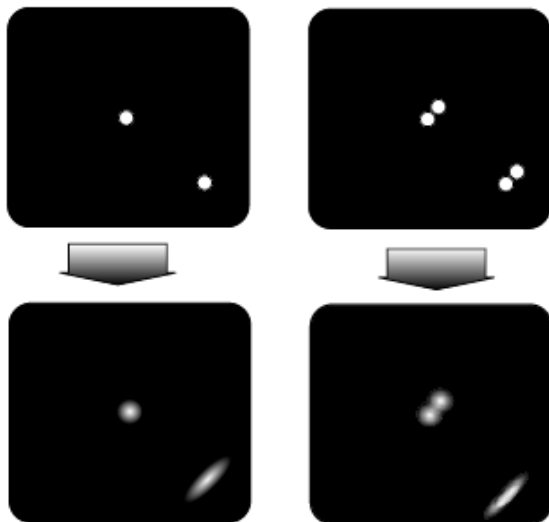


Abb. 4: Wirkung der PSF (unten) auf eine Anordnung von Punktstreuern (oben), zitiert aus [1]

Informationsverlust vor.

In der Bildgebung ist die Punktspreizfunktion (PSF) eine wichtige Eigenschaft eines bildgebenden Systems. Die Punktspreizfunktion gibt an, wie das System ein punktförmiges Objekt, einen sogenannten Punktstreuer, abbildet.

In Abbildung 4 wird die Bedeutung der Punktspreizfunktion für die Bildgebung verdeutlicht. In der oberen Zeile sieht man jeweils die Verteilung der Punktstreuer, wie sie tatsächlich ist, die untere Zeile zeigt das Bild, welches ein fiktives Beispiel-System aufnimmt. Man sieht, dass die Punktspreizfunktion sich in der Regel darin äußert, dass die Punktstreuer größer dargestellt werden und die Umrisse unschärfer werden. Auch eine Verzerrung, hier jeweils am rechten unteren Bildrand erkennbar, kann durch die Punktspreizfunktion auftreten. In jedem Fall liegt ein

Besondere Bedeutung gewinnt die Punktspreizfunktion bei der Abbildung mehrerer, dicht beieinander liegender Objekte. Wie in Abbildung 4, rechts unten gezeigt, besteht die Gefahr, dass die Objekte soweit gespreizt werden, dass sie nicht mehr unterscheidbar sind und als ein Objekt dargestellt werden.

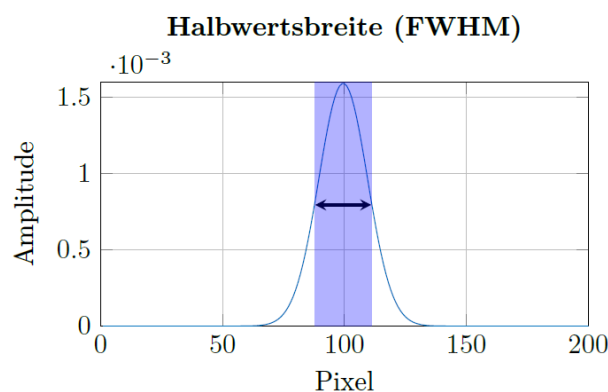


Abb. 5: Halbwertsbreite (FWHM) der PSF, zitiert aus [2]

Gemessen wird in der Regel die Halbwertsbreite der Punktspreizfunktion. Diese gibt, wie in Abbildung 5 gezeigt, an, in welcher Breite die Helligkeits-Amplitude der Abbildung des Punkt-

streuers größer als die halbmaximale Amplitude ist. Würde die Punktspiezfunktion keine Änderungen bewirken, wäre die Amplitudenverteilung ein einzelner Peak mit einer Breite von 0. Die Punktspiezfunktion sorgt nun dafür, dass die Amplitudenverteilung eine breitere Form mit einer größeren Halbwertsbreite annimmt.

Zur Berechnung der Breite der Punktspiezfunktion gibt es verschiedene Ansätze. Um tatsächlich die Halbwertsbreite der Punktspiezfunktion zu messen ist es sinnvoll, als Erstes das Bild passend "abzuschneiden". Dabei werden alle Werte, die kleiner als die Hälfte des Maximums im Bild sind, auf null gesetzt. In MATLAB ist dies mit dem folgenden Ausdruck einfach umzusetzen:

```
Image(Image<0.5*max(Image(:))) = 0;
```

Nun kann man die Halbwertsbreite der Punktspiezfunktion berechnen. Es gibt eine endliche Anzahl n an Pixeln, das heißt an Matrixeinträgen, die ungleich 0 sind. Die Helligkeit dieser Pixel liegt über der Hälfte des Maximums, das heißt, alle Pixel sind für die Halbwertsbreite relevant. Nun berechnet man für jedes mögliche Paar (i, j) aus zwei Pixeln die euklidische Distanz $d(i, j)$ nach dem Satz des Pythagoras.

$$\begin{aligned}\Delta x &= x_i - x_j \\ \Delta y &= y_i - y_j \\ d(i, j) &= \sqrt{\Delta x^2 + \Delta y^2}\end{aligned}$$

Die Breite der Punktspiezfunktion ist nun der Größte aller berechneter $d(i, j)$ -Werte (Abb. 6).

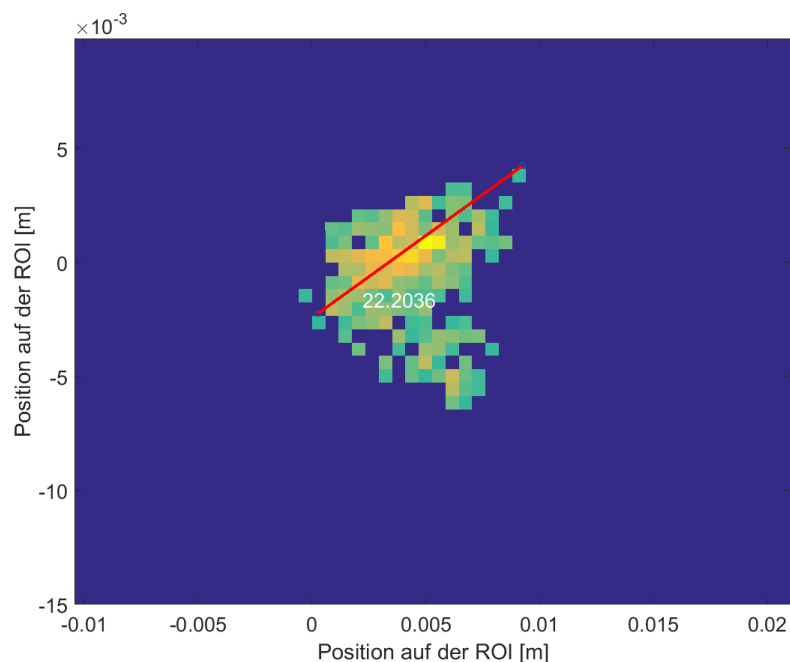


Abb. 6: Funktionsweise der Breitenmessung der PSF. Durch eine Linie verbunden sind die beiden Pixel mit dem größten Abstand (nach dem Abschneiden, 22 Pixel entsprechen 1,1 cm).

3.2 Messverfahren



Abb. 7: Stab zur Charakterisierung

Um die Halbwertsbreite der Punktspreizfunktion zu bestimmen bräuchte man theoretisch einen Punktstreuer. Da die Auflösungsgrenze des Systems aber in etwa bei $\frac{\lambda}{4} \approx 0,4 \text{ mm}$ liegt, wird in dieser Arbeit ein dünner Metalldraht, der auf einer Metallplatte montiert ist, verwendet (Abb. 7).

Die Breite der Punktspreizfunktion ist von der Position des Stabes auf der ROI abhängig. Sie wird in Abhängigkeit von der Position in sogenannten Karten aufgetragen. Diese geben für jede Position auf der ROI die Breite der Punktspreizfunktion an. Diese Karten werden erzeugt, indem man ein Raster mit einem Gitterabstand von 2 cm auf die ROI legt. Nun wird mit dem System an jedem Gitterpunkt ein Bild erstellt; aus diesem kann jeweils die Breite der Punktspreizfunktion bestimmt werden.

Aus diesen Aufnahmen können weitere Charakteristiken gewonnen und ebenfalls in Karten dargestellt werden. Eine, die von besonderem Interesse ist, ist die Verteilung der Helligkeit im System. Als "Helligkeit" wird hierbei die Amplitude der Messwerte bezeichnet. Auch diese Größe lässt sich aus der Aufnahme des Metallstabes ermitteln. Dabei kann man die Helligkeit auf zwei Arten messen: Der einfachste Ansatz ist, die Helligkeit eines Bildes durch den maximalen Wert im Bild zu beschreiben. Da diese Maxima aber starken zufälligen Schwankungen unterworfen sein können, müssten sehr viele Messungen durchgeführt werden, um statistisch aussagekräftige Daten zu erhalten. Betrachtet man jedoch nicht nur den maximalen Wert im Bild, sondern summiert stattdessen alle Matrix-Einträge, das heißt, man berechnet ein Integral des Bildes, so gleichen sich zufällige Schwankungen größtenteils aus und die Ergebnisse gewinnen mehr Aussagekraft.

Bei der genaueren Analyse der Bilder stellt man fest, dass nicht alle Objekte genau an der Position abgebildet werden, an der sie tatsächlich stehen. Diese Abweichung kann man in einer weiteren Charakteristik erfassen. Dazu berechnet man sowohl in X- wie auch in Y-Richtung die Abweichung Δx beziehungsweise Δy zwischen der Position des Stabes auf der ROI und der Position des Maximums im Bild. Die Gesamtabweichung δ berechnet sich dann gemäß dem Satz des Pythagoras folgendermaßen:

$$\delta = \sqrt{\Delta x^2 + \Delta y^2}$$

Es kann vorkommen, dass in allen Bildern das Maximum um einen bestimmten Wert verschoben ist. Daher ist es sinnvoll zunächst für alle Bilder die Verschiebungen Δx und Δy zu ermitteln. Dann bestimmt man jeweils den Durchschnitt $\overline{\Delta x}$ bzw. $\overline{\Delta y}$ aller Δx - bzw. Δy -Werte; dies ist die globale Verschiebung. Für die Bestimmung der Gesamtabweichung δ subtrahiert man diese Verschiebung von Δx und Δy , bevor man δ bestimmt:

$$\begin{aligned} \Delta x_n &= \Delta x - \overline{\Delta x} \\ \Delta y_n &= \Delta y - \overline{\Delta y} \\ \delta &= \sqrt{(\Delta x_n)^2 + (\Delta y_n)^2} \end{aligned}$$

3.3 Ergebnisse

Die Ergebnisse der Charakterisierung werden in Karten dargestellt. Hierbei steht jeder Punkt auf der Karte entsprechend für den Wert des Maximums, Integrals, der Halbwertsbreite der PSF bzw. der Verschiebung des Maximums von der Soll-Position der Aufnahme an diesem Ort. In jeder Karte sind 290 Messpunkte enthalten und diese Karten wurden für mehrere Messreihen (mit 14 und mit 16 Sensoren) erstellt. Im Folgenden sind die Ergebnisse der letzten Messreihe dargestellt:

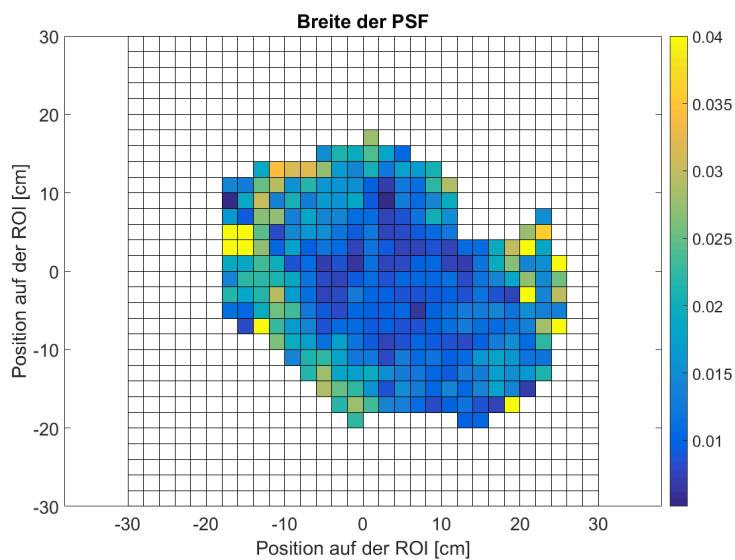


Abb. 8: Halbwertsbreite der PSF

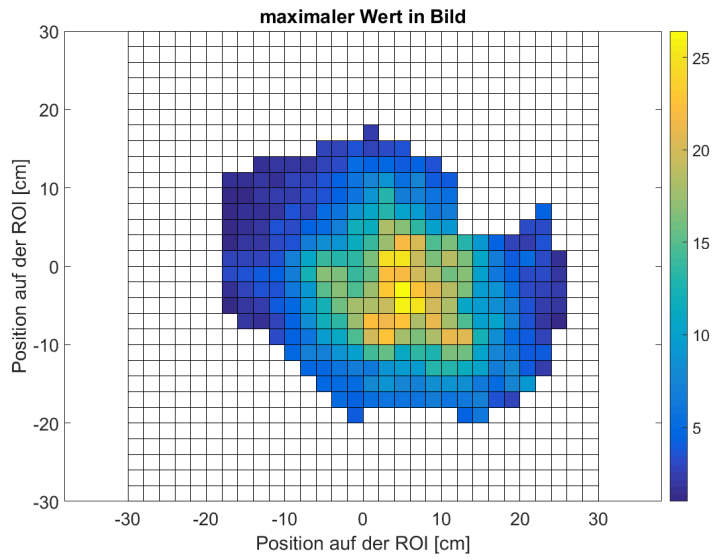


Abb. 9: Verteilung der Helligkeiten (maximaler Wert im Bild)

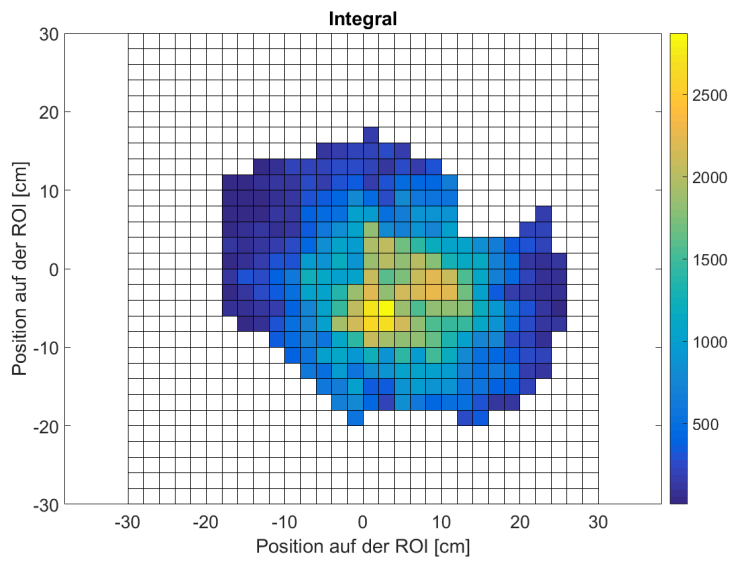


Abb. 10: Verteilung der Helligkeiten (Integral)

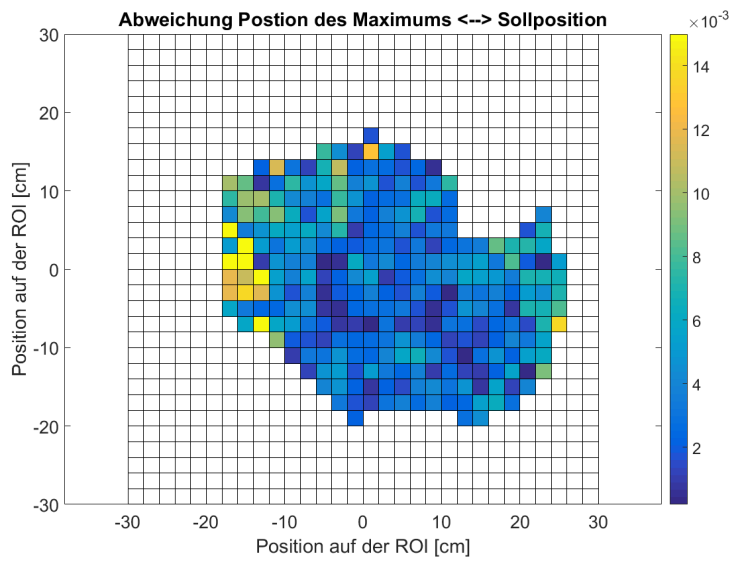


Abb. 11: Abweichung des Maximums von Soll-Position

Zu erkennen ist, dass alle Charakteristiken sich zum Rand der ROI hin verschlechtern. Die für unsere Zielsetzung wichtigste Charakteristik, die Halbwertsbreite der Punktspreizfunktion, zeigt Werte von unter 1 cm und bis zu über 5 cm, abhängig von der genauen Position. Diese Werte sind weitaus schlechter als die theoretisch maximale Genauigkeit von ca. 0,4 mm (Seite 4). Sie stimmen auch nicht mit den von Wei Yap Tan [2] vorausgesagten überein. Die anderen Charakteristiken entsprechend in grober Annäherung den Voraussagen.

4 Verbesserung der Bilder

4.1 Ausgangssituation

Aus der Charakterisierung ist deutlich erkennbar, dass die Helligkeiten im System ungleich verteilt sind. Wie man in Abbildung 10 (Seite 13) erkennen kann, werden Objekte in der Mitte deutlich heller dargestellt als solche, die am Rand liegen. Nimmt man nur ein einzelnes Objekt mit relativ begrenzter Größe auf, so ist dies nicht weiter von Bedeutung. Dies liegt daran, dass MATLAB für den größten Wert der Matrix automatisch die Farbe gelb wählt, während der kleinste Wert in dunkelblau dargestellt wird. Auch am Rand werden Objekte durch deutlich größere Werte repräsentiert als leere Teile der ROI. Daher wählt das Programm die Farbe gelb und der Hintergrund bleibt dunkelblau. Erstellt man jedoch Aufnahmen von mehreren Objekten, so wird diese Ungleichverteilung der Helligkeiten zum Problem. Objekte, die in der Mitte stehen, haben sehr hohe Helligkeitswerte, entsprechend werden ihre Maxima in gelb dargestellt. Objekte, die weiter am Rand stehen, haben deutlich niedrigere Helligkeitswerte. Dementsprechend werden ihre Maxima nur noch hellblau oder sogar dunkelblau dargestellt. Dadurch heben sie sich, kaum vom Hintergrund ab (Abb. 12).

Um dieses Problem zu umgehen wurde in dieser Arbeit ein Verfahren zur Bildkorrektur entwickelt.

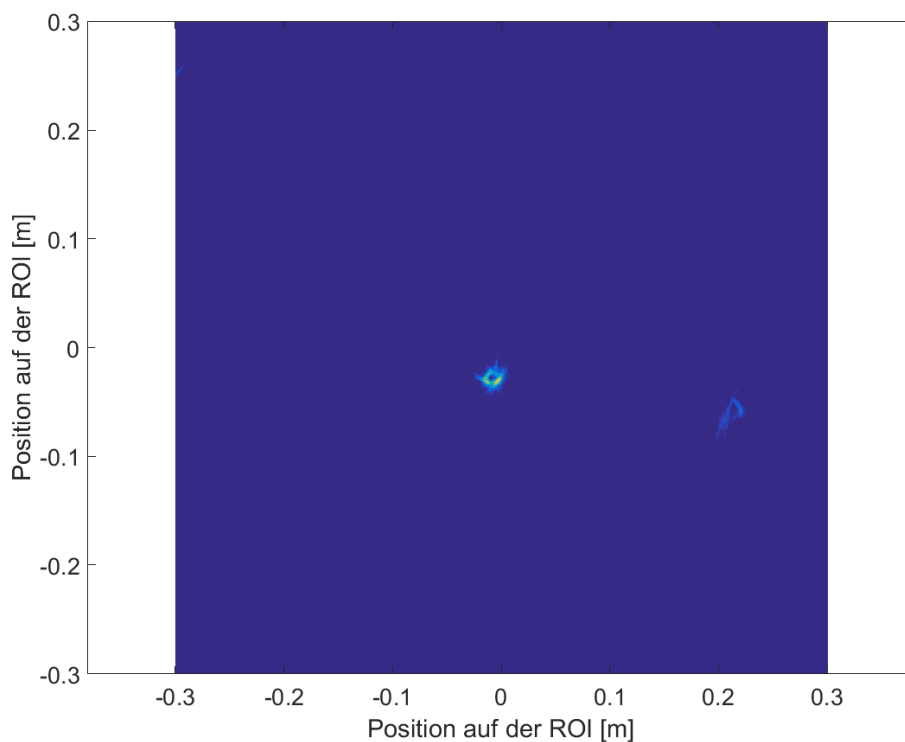


Abb. 12: Aufnahme von zwei Stiften vor der Korrektur

4.2 Verfahren

Ein einfacher Ansatz ist, die Helligkeit aller Punkte, deren Helligkeit größer als ein bestimmter Schwellenwert ist, auf einen festgelegten Wert zu setzen. Dies ist allerdings mit einem großen

Informationsverlust verbunden. Hätte man zwei Objekte A und B, die unterschiedlich beschaffen sind, so wäre es möglich, diese so zu konstruieren, dass A bei einer Aufnahme in der Mitte der ROI in etwa doppelt so hell dargestellt wird wie B. Platziert man nun A und B beliebig in der ROI wäre es möglich, nach einer Korrektur, A von B über die Helligkeit zu unterscheiden. Das oben beschriebene Korrekturverfahren würde dies jedoch verhindern, da tatsächlich relative Helligkeitsunterschiede² nicht erhalten bleiben.

Aus der Charakterisierung ist bekannt, mit welcher Helligkeit das gleiche Objekt an verschiedenen Stellen abgebildet wird. Diese relativen Helligkeiten kann man nun nutzen, um die Helligkeiten eines gegebenen Bildes unter Erhaltung der tatsächlichen relativen Helligkeiten zu korrigieren. Dazu teilt man das Bild B elementweise durch die Helligkeitsverteilung H . Der Matrix-Wert an der Stelle (i, j) im korrigierten Bild B' berechnet sich dann folgendermaßen:

$$B'(i, j) = \frac{B(i, j)}{H(i, j)} \quad (1)$$

Dies führt zu einer Normierung der Helligkeiten. Die hohen Helligkeitswerte in der Mitte werden durch große Werte aus der Helligkeitsverteilung geteilt, kleine Helligkeitswerte am Rand durch kleinere Werte aus der Helligkeitsverteilung. Unterschiede zwischen den realen Helligkeiten der Objekte bleiben so erhalten.

Um gute Ergebnisse zu erzielen, ist eine optimal passende Helligkeitsverteilung erforderlich. Der einfachste Ansatz ist, direkt die ermittelte Helligkeitsverteilung (Abb. 10, Seite 13) zu verwenden. Dieses Verfahren wird im folgenden als Typ 0 bezeichnet. Allerdings handelt es sich bei den Helligkeiten um die Integrale der Aufnahmen zur Charakterisierung. Für die Korrektur werden diese Werte benutzt, um einzelne Helligkeitswerte anzupassen. Da die Breite der Punktspreizfunktion nach außen hin zunimmt, führt dies zu gewissen Ungenauigkeiten. Bei Typ 0 wird das Integral eines Bildes aus der Charakterisierung verwendet, um die Helligkeit eines einzelnen Pixel im zu korrigierenden Bild anzupassen. Da die Punktspreizfunktion nicht konstant ist, ist die Korrektur eines einzelnen Pixel mit einem Integral an manchen Stellen zu stark.

Ein erster Ansatz besteht darin, durch eine elementweise Potenzierung der Helligkeitsverteilung eine bessere Korrektur zu erreichen. Aus verschiedenen Tests hat sich hierbei 0,5 als am besten geeignete Potenz herausgestellt. Dies führt zu einer neuen Helligkeitsverteilung H' . Der Eintrag an der Stelle (i, j) berechnet sich folgendermaßen:

$$H'(i, j) = H(i, j)^{0,5} \quad (2)$$

Zwar führt diese Potenzierung zu einer verbesserten Korrektur-Qualität, doch sind aus der Charakterisierung weitere Daten bekannt, welche noch bessere Ergebnisse ermöglichen.

Dazu verwendet man in einem zweiten Ansatz nun nicht mehr das Integral der Bilder zur Korrektur. Stattdessen teilt man das Integral durch die Anzahl der Pixel, deren Helligkeitswert Teil des Integrals sind. Somit verwendet man zur Korrektur an einer Position P , die durchschnittliche Helligkeit der gesamten Aufnahme des Stabes an P . Dieses Korrekturverfahren wird im folgenden als Typ 1 bezeichnet. Da nun anstelle des Integrals eine Durchschnittshelligkeitswert zur Korrektur verwendet wird, ist eine Potenzierung nicht mehr notwendig.

²Die unechten Helligkeitsunterschiede, die durch das System erzeugt werden, sollen im Gegensatz dazu ausgeglichen werden.

Für die elementweise Division müssen die Dimensionen beider Matrizen (Bild und Helligkeitsverteilung) gleich sein. Die Auflösung des Systems ist deutlich größer, als die Auflösung der Charakterisierung, daher muss die Helligkeitsverteilung interpoliert werden. Verwendet wird eine lineare Interpolation. Ein Eintrag in der interpolierten Matrix wird als der gewichtete Durchschnitt der vier nächsten Matrix-Einträge, für die ein Messwert existiert, berechnet.

4.3 Ergebnisse

Diese Korrektur lässt sich nun zum Beispiel auf das Bild aus Abbildung 12 anwenden, was zu dem in Abbildung 13 gezeigten Ergebnis führt. Der Stift in der rechten Bildhälfte wird so deutlicher gezeigt.

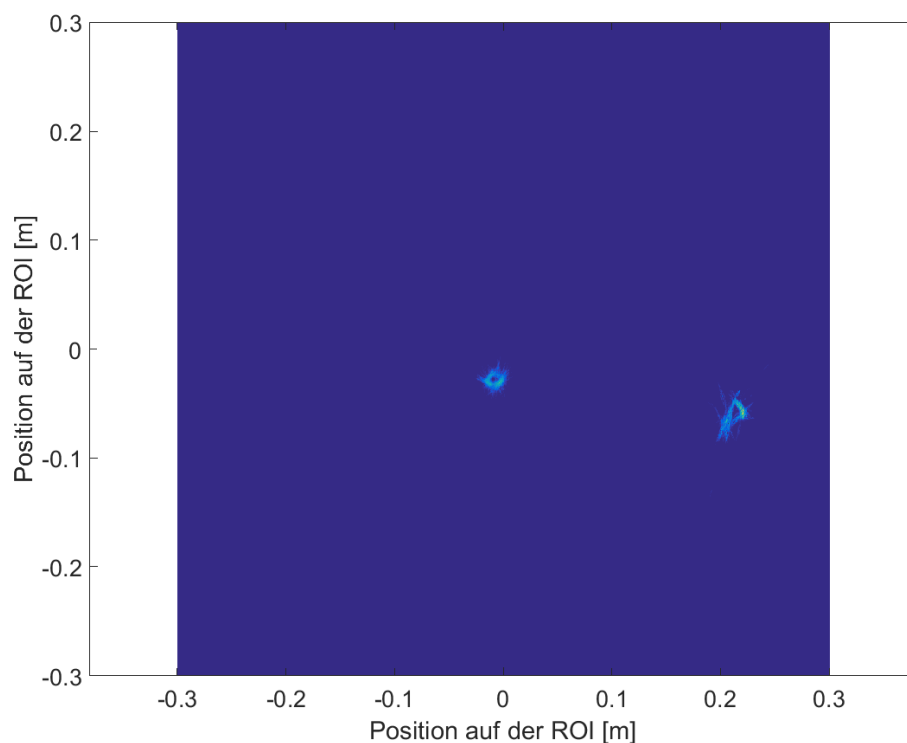


Abb. 13: Abbildung 12 nach der Anwendung der Korrektur (Typ 1)

Im Folgenden soll allgemein evaluiert werden, wie gut die Korrektur tatsächlich arbeitet.

In einem ersten Schritt kann man dazu Aufnahmen eines einzelnen Objektes betrachten. Man platziert dieses auf verschiedenen Positionen in der ROI und erstellt jeweils eine Aufnahme. Betrachtet man die Helligkeitsverteilung, so sollte man mit einem ähnlichen Ergebnis wie bei der Charakterisierung rechnen. Nach einer idealen Korrektur erwartet man eine konstante Helligkeit, unabhängig vom Ort der Aufnahme.

Verteilt man die Messungen regelmäßig über die ROI, so ist es möglich, Karten analog zur Charakterisierung vor und nach der Korrektur zu erzeugen. Als Maß für die Verbesserung kann man die Standardabweichung der Helligkeit vom Durchschnitt über alle Messungen jeweils vor und

nach der Korrektur betrachten. Ideal wäre eine Durchschnittsabweichung von 0, da das gleiche Objekt überall gleich hell dargestellt werden sollte.

Da die absoluten Helligkeiten vor und nach der Korrektur unterschiedlich sind, sollte man die Durchschnittsabweichung vor der Betrachtung normieren, das heißt, jeweils durch den Durchschnittswert der Helligkeiten, vor beziehungsweise nach der Korrektur, teilen.

Positioniert man einen handelsüblichen Filzstift (Abb. 14) auf 33 Messpunkten im Abstand von jeweils 6 cm auf der ROI, erhält man das in Abbildung 15 gezeigte Ergebnis.

Das angewandte Korrekturverfahren wirkt sich positiv auf die Helligkeitsverteilung aus. Vor der Korrektur ist die diese ähnlich wie in Abbildung 10. Nach der Korrektur sind die Helligkeiten deutlich gleichmäßiger verteilt.

Dies äußert sich auch in den normierten Standardabweichungen.



Abb. 14: Filzstift zur Überprüfung der Korrektur

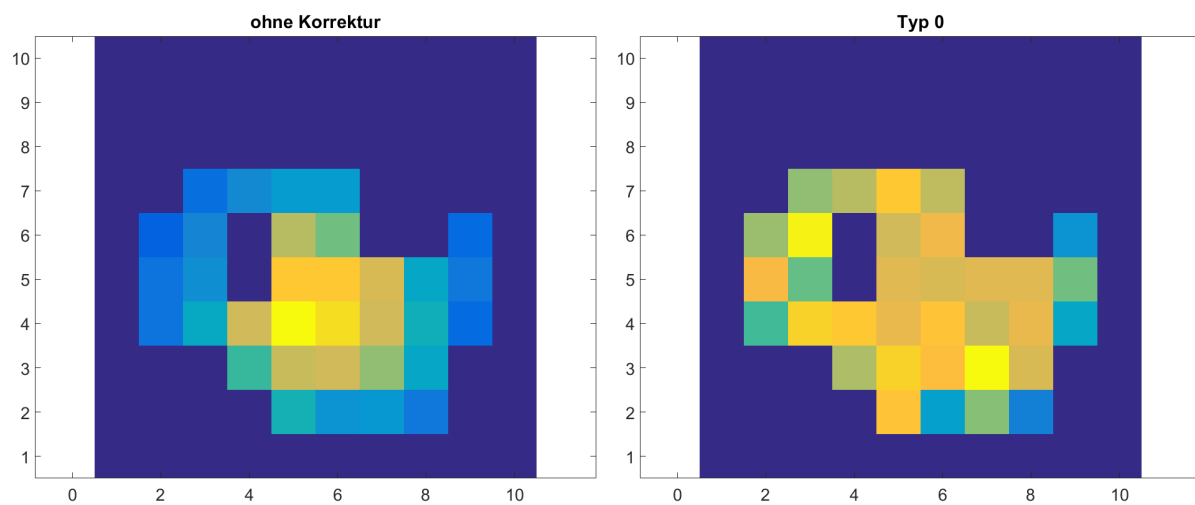


Abb. 15: Karte der normierten Helligkeitsverteilung, vor der Korrektur (links), und nach der Korrektur mit Typ 0, Potenz 0,5 (rechts)

Tab. 1: Normierte Standardabweichung der Helligkeiten vor und nach der Korrektur

vor der Korrektur	Nach der Korrektur	
	Typ 0; Potenz 0,5	Typ 1
0,55633	0,25638	0,30231

Die Verbesserung ist deutlich erkennbar; bei Typ 0 wird die Standardabweichung, im Vergleich zu vor der Korrektur, mehr als halbiert und auch bei Typ 1 wird sie fast auf die Hälfte reduziert. Das gewählte Korrekturverfahren zeigt demnach deutliche Effekte.

Die Korrektur soll jedoch ebenso bei Aufnahmen von zwei oder mehr Objekten angewendet werden. Daher setzt die zweite Messung der Korrektur-Qualität in genau diesem Bereich an. Erstellt man eine Aufnahme, nachdem man zwei gleiche Objekte auf der ROI platziert hat, so sollten diese nach einer Korrektur im Idealfall beide gleich hell erscheinen.

Diese Anforderung kann für eine weitere Evaluation des Korrekturverfahrens verwendet werden. Dazu nimmt man eine Reihe von Aufnahmen, die jeweils zwei gleiche Objekte zeigen, wie zum Beispiel die Aufnahme aus Abbildung 12. Nun misst man die Helligkeit der beiden Maxima. Dazu kann zum Beispiel unser Code aus Listing 3 verwendet werden. Nun bestimmt man das Verhältnis zwischen den beiden Maxima vor und nach der Korrektur.

Abbildung 16 zeigt das Verhältnis zwischen den beiden Maxima für 10 Aufnahmen von zwei Filzstiften (Abb. 14) vor und nach der Korrektur. Das Ergebnis einer idealen Korrektur wäre ein konstantes Verhältnis von 1, hier in Blau dargestellt.

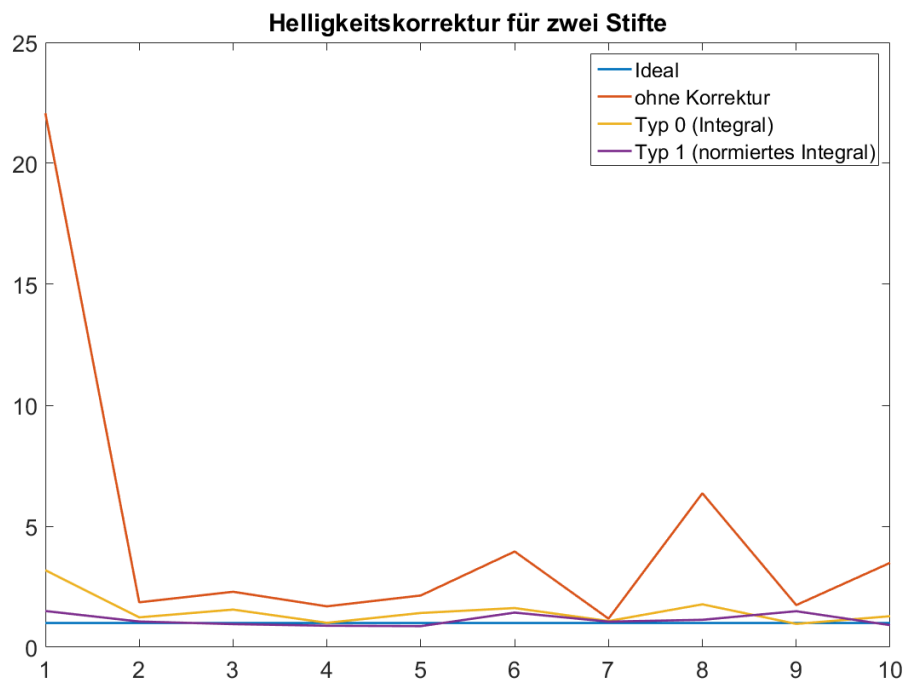


Abb. 16: Helligkeitsverhältnis zwischen zwei Stiften vor und nach der Korrektur

Sowohl durch die Anwendung von Korrekturverfahren Typ 0 als auch Typ 1 werden die ermittelten Werte deutlich verbessert, auch wenn der Ideal-Wert nicht ganz erreicht wird. Berechnet man die durchschnittlichen Verhältnisse über alle 10 Aufnahmen, vor und nach der Korrektur, wird der Effekt noch deutlicher (Tabelle ??).

Tab. 2: Durchschnittliches Helligkeitsverhältnis zweier Stifte vor und nach der Korrektur

vor der Korrektur	Nach der Korrektur	
	Typ 0; Potenz 0,5	Typ 1
4,6776	1,5116	1,1296

Der Idealwert wird auch hier nicht ganz erreicht, doch das durchschnittliche Verhältnis wird auf weniger als ein Drittel reduziert. Hierbei ist die Korrektur für Typ 1 besser als für Typ 0.

Beide Tests zeigen, dass die Korrektur, insbesondere für mittelgroße Objekte, deutlich bessere Werte liefert. Allerdings führt bei Tests mit einem Stift die Korrektur mit Typ 0 zu besseren Ergebnissen, während der Test mit zwei Stiften Korrekturverfahren Typ 1 als besser geeignet zeigt. Für eine Anwendung der Korrektur muss man deshalb, je nach Anwendung, entscheiden, welches Korrekturverfahren (Typ 0 oder 1) man verwendet. Da die Korrektur vor allem bei Aufnahmen mehrerer Objekte einen Vorteil bietet, wird sich in den meisten Fällen das Korrekturverfahren Typ 1 als geeigneter erweisen. Sofern es möglich ist kann in Vorversuchen geprüft werden, welches Korrekturverfahren besser geeignet ist. Andernfalls empfehlen wir die Verwendung von Typ 1. Das Funktionieren des Korrekturverfahrens Typ 1 lässt sich des weiteren aus rationalen Überlegungen besser begründen als die heuristisch gewählte Potenz bei Typ 0.

Einzelne Tests mit größeren Objekten (Durchmesser einige Zentimeter) (Abb. 17 und Abb. 18) zeigen deutlich schlechtere Resultate. Dies liegt im Wesentlichen daran, dass das System Ecken heller als Kanten oder nach innen gestülpte Ecken erkennt. Diese Unterschiede sind bislang in der Charakterisierung nicht erfasst, und können daher auch nicht korrigiert werden.

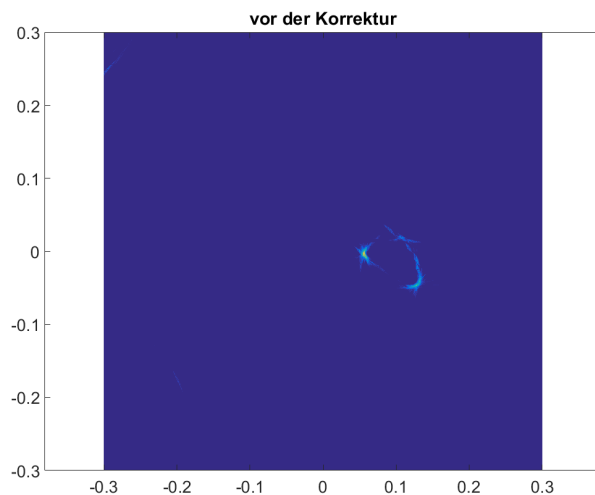


Abb. 17: Großes Objekt vor der Korrektur

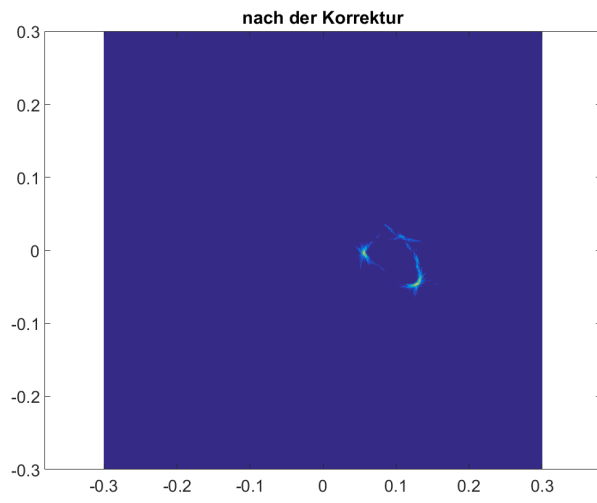


Abb. 18: Das Bild aus Abbildung 17 nach der Korrektur

5 Diskussion

5.1 Fazit und Diskussion

Die zu Projektbeginn formulierten Ziele konnten erreicht werden. Das System wurde charakterisiert und auf Basis der Daten ein Verfahren zur Korrektur der Helligkeiten entwickelt und evaluiert.

Die Charakterisierung des Systems hat für alle Charakteristiken sowohl für 14 als auch für 16 Ultraschallsensoren Ergebnisse geliefert. Diese weichen allerdings im Bereich der Punktspreizfunktion von den von Wei Yap Tan simulierten Ergebnissen deutlich ab.

Diese (aus [2]), weisen deutlich kleinere Werte für die Punktspreizfunktion der optimierten Anordnung von 16 Sensoren auf und liegen im Bereich einiger Millimeter. Die von uns durchgeführte experimentelle Bestimmungen ergab hingegen Werte im Bereich von einigen Zentimetern. Diese Abweichungen lassen sich vor allem aus dem Unterschied zwischen Simulation und Experiment erklären.

Während in der Simulation zum Beispiel mit beliebig hoher Genauigkeit Laufzeiten detektiert werden können, ist dies im realen System nicht möglich. Die Genauigkeit der Laufzeitdetektion wird unter anderem durch die Taktfrequenz der Messapparatur beeinflusst. Ein weiterer wichtiger Faktor ist die Trägheit der Sensoren. Die Schallwandler haben eine gewisse Latenzzeit, bevor sie ein eingetroffenes Signal an die Auswertung weiterleiten.

Auch die Sendeleistung der Ultraschallsensoren liegt bei einer Simulation auf einem höheren Wert als in der Realität, wo es zum Beispiel vorkommen kann, dass das Leistungsoptimum eines Sensors bei einer etwas anderen Frequenz als der verwendeten liegt.

Ebenfalls könnte es ein Störfaktor sein, dass das verwendete Objekt kein echter Punktstreuer ist. So kann zum Beispiel die Metallplatte unter dem Stab Schall reflektieren, was die Genauigkeit der Erfassung der Position des Objekts verschlechtern könnte. Auch kann es sein, dass der Stab mit einem Durchmesser von ca. 1,5 mm zu klein für das System ist, dessen maximales Auflösungsvermögen bei ca. 0,4 mm liegt, also nur geringfügig kleiner als die Objektgröße. Durch die geringe Größe des Objektes kann es daher unter Umständen nicht richtig erfasst werden.

Mit dem entwickelten Korrekturverfahren konnte der Bildkontrast deutlich verbessert werden (Abschnitt 4.3), trotzdem gibt es noch Potential für Optimierungen. Dies hat verschiedene Ursachen. Die Korrektur arbeitet aktuell auf der Basis einer Charakterisierungsmessung. Diese enthält Messfehler sowie zufällige Schwankungen, die das Messergebnis negativ beeinflussen. Weiterhin kommt es bei größeren sowie bei mehreren Objekten zu Abschattungseffekten, da das eine Objekt verhindert, dass alle Ultraschallsensoren das andere detektieren können. Diese Effekte sind bei der Charakterisierung mit einem Stift bislang nicht berücksichtigt und der Grund, warum die Korrektur mit den verwendeten Filzstiften besser funktioniert als mit ausgedehnteren Objekten.

5.2 Ausblick

Obwohl wir für einige Probleme Lösungsansätze entwickeln konnten, gibt es noch einige Möglichkeiten für weitere Aufgabenstellungen.

So wäre es sinnvoll, die Messungen zur Charakterisierung einige Male zu verschiedenen Zeiten, das heißt mit geringfügig anderen Versuchsbedingungen, zu wiederholen. Sowohl für die Charakterisierung als auch für die Korrektur könnte man dann auf das arithmetische Mittel oder den Median aller Messungen zurückgreifen und so zufällige Schwankungen und Messfehler ausgleichen.

Der zur Charakterisierung verwendete Stab ist aktuell auf einer Metallplatte befestigt, damit er frei in der ROI platziert werden kann. Bei unseren Experimenten sind wir davon ausgegangen, dass der Einfluss der Platte auf das Messergebnis vernachlässigt werden kann. Allerdings ist dies nicht sicher und könnte Gegenstand einer weiteren Untersuchung sein.

Im Rahmen der Optimierung hat Wei Yap Tan die Sensoranordnung so gewählt, dass stets mindestens zwei Punktstreuer erkannt werden können. Die in dieser Arbeit durchgeführte Charakterisierung berücksichtigt allerdings immer nur einen Punktstreuer. Es wäre daher sinnvoll, weitere Tests mit zwei Punktstreuern durchzuführen. Da dies zu sehr vielen Messungen führen würde, wäre es denkbar, eine automatisierte Messvorrichtung zu konstruieren. Diese würde auch die oben vorgeschlagenen mehreren Charakterisierungsmessungen ermöglichen.

Schließlich wäre es interessant zu versuchen, Abschattungseffekte zu korrigieren. Da bekannt ist, wo sich das abschattende Objekt befindet, könnte prinzipiell evaluiert werden, wie stark die Abschattung in verschiedenen Bereichen der ROI ist. Diese Information könnte man dann zum einen dazu nutzen, den abgeschatteten Bereich zu markieren, sodass bekannt ist, in welchem Bereich das System nicht in der Lage ist, dort potentiell befindliche Objekte zu erkennen. Zum anderen könnte man teilweise abgeschattete Bereiche verstärken und somit einen ähnlichen Effekt wie bei der momentan vorhandenen Korrektur erzielen.

6 Danksagung

Wir danken Herrn Wei Yap Tan für die Betreuung und Unterstützung in allen Phasen des Projektes. Unser Dank gilt dem Institut für Prozessdatenverarbeitung und Elektronik (IPE) am Karlsruher Institut für Technologie (KIT), welche uns ermöglicht haben dieses Projekt durchzuführen.

Weiterhin bedanken wir uns bei unseren Kursleitern, Anke Richert und Thomas Knecht, die uns im Rahmen der Kooperationsphase und in den Jahren davor im Hector-Seminar betreut und uns dieses Projekt vermittelt haben.

Der Hector-Stiftung, insbesondere Josefine und Dr. Hans-Werner Hector danken wir für die langjährige Förderung.

7 Quellen

Literatur

- [1] Wei Yap Tan. Optimierung der Lokalisierungsleistung bildgebender Ultraschallsysteme. [Vortrag, Juli 2017].
- [2] Wei Yap Tan. Optimierung der Lokalisierungsleistung bildgebender Ultraschallsysteme. [unveröffentlicht].
- [3] Wei Yap Tan, Till Steiner, Nicole V. Ruiten. Newton's Method based Self Calibration for a 3D Ultrasound Tomography System, Oktober 2015.
- [4] Wei Yap Tan, Till Steiner, Nicole V. Ruiten. Automatic Optimization of Sensor Positioning for an Airborne Ultrasound Imaging System, September 2016.
- [5] Wikipedia. Punktspreizfunktion — Wikipedia, Die freie Enzyklopädie, 2016. <https://de.wikipedia.org/w/index.php?title=Punktspreizfunktion&oldid=156627901> [Online; Stand 9. Juli 2017].

© 2017 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

8 Anhang

8.1 Abkürzungsverzeichnis

Abkürzung	Bedeutung
FWHM	Halbwertsbreite
PSF	Punktspreizfunktion
ROI	Region of Interest
SAFT	Synthetic Aperture Focusing Technique

8.2 weitere Ergebnisse

Im folgenden sind die Ergebnisse der Charakterisierung mit 14 Sensoren dargestellt:

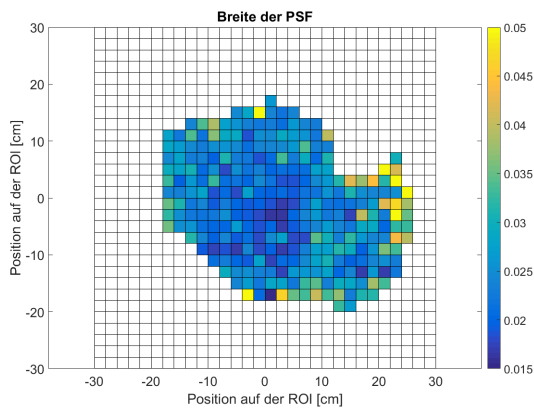


Abb. 19: Halbwertsbreite der PSF

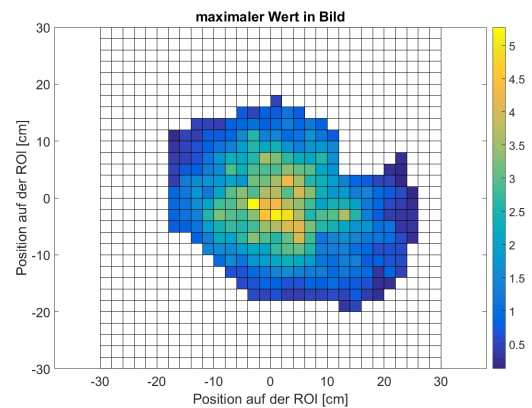


Abb. 20: Verteilung der Helligkeiten (Maximaler Wert im Bild)

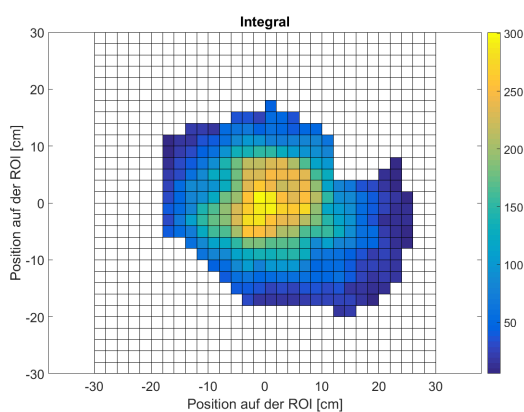


Abb. 21: Verteilung der Helligkeiten (Integral)

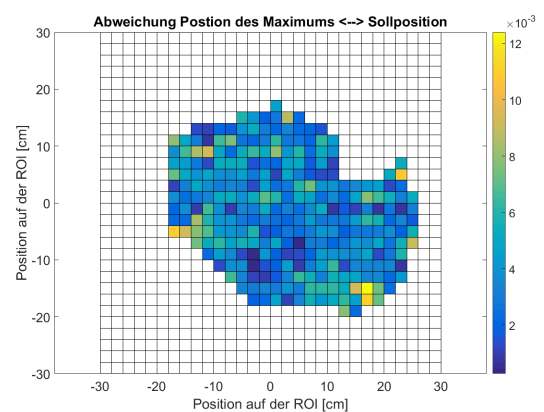


Abb. 22: Abweichung des Maximums von der Soll-Position

8.3 Operationen auf Matrizen in MATLAB

MATLAB bietet viele Möglichkeiten, um mit Matrizen zu arbeiten. Da die gewonnen Messwerte in Form einer Matrix gespeichert werden, sind diese für das Projekt von besonderen Interesse.

Der Zugriff auf einzelne Matrix-Elemente erfolgt mittels runden Klammern, die Indizierung der Einträge startet bei eins. Der folgender Code greift auf den Wert in der dritten Zeile der ersten Spalte einer beliebigen Matrix M zu:

```
M(3,1)
```

Will man mehrere Elemente einer Spalte oder Zeile auf einmal abfragen, geschieht dies mittels des ":"-Operators. Mit diesem ist es möglich einen Bereich anzugeben, der aus der Matrix zurückgegeben wird. Alternativ kann er auch ohne Bereich verwendet werden. Dies führt dazu, dass die komplette Zeile oder Spalte zurückgegeben wird. So kann man zum Beispiel durch folgenden Code die gesamte dritte Spalte einer Matrix M auswählen. Man erhält als Rückgabe einen Vektor, technisch eine $n \times 1$ Matrix

```
M(:,3)
```

Verwendet man nur den "Doppelpunkt"-Operator, ohne die Angabe einer zweiten Zahl, so werden alle Elemente der Matrix in eine neue Matrix mit nur einer Zeile gespeichert. Dies ist zum Beispiel für die Bestimmung von Maxima wichtig, da nur so die *max*-Funktion von MATLAB nach globalen Maxima sucht.

Grundlegende Matrix-Operationen wie Matrix-Multiplikation und -Division sind in MATLAB nativ verfügbar. Doch auch weitere Operationen sind vorhanden. So zum Beispiel die elementweise Matrix-Multiplikation und -Division. Möchte man die Matrizen A und B elementweise multiplizieren, so geschieht dies folgendermaßen:

```
A.*B
```

Des weiteren ist die Verwendung von Vergleichsoperatoren im Zusammenhang mit Matrizen ist zulässig. So kann man zum Beispiel effizient testen, welche Werte einer Matrix M größer als ein Schwellenwert s sind. Die erhaltene Matrix ist eine Matrix von logischen Werten. Sie enthält eine 0 an den Stellen, an denen der Eintrag in M die Bedingung nicht erfüllt. An den Stellen, an denen der Eintrag in M die Bedingung erfüllt, ist eine 1 enthalten.

Diese Matrix von logischen Werten kann wiederum verwendet werden, um auf M schreibend zuzugreifen. Der folgende Ausdruck sorgt dafür, dass alle Einträge in M , die kleiner als ein Schwellenwert s sind, auf 0 gesetzt werden.

```
1 res = M < s;  
2 M(res) = 0;
```

8.4 Quelltexte

Listings

1	Ermittlung der Charakteristiken	28
2	Korrekturfunktion	30
3	Ermittlung der Helligkeit der zwei größten Objekte	31

Listing 1: Ermittlung der Charakteristiken

```
1 %clear;
2
3 version = 'mai_juni';
4 ROINumber = '1';
5
6 files = ls;
7 si = size(files);
8 s = si(1);
9
10 prefix = 'bild_scan_kart_';
11
12 %s=33;
13
14 % bei erster Ausfuehrung
15 data = zeros(s-2,9);
16 startIndex = 0;
17
18 %1 --> Breite der PSF
19 %2 --> Max
20 %3 --> Flaeche unter Graph
21 %4 --> gemessenes X an Max
22 %5 --> gemessenes Y an Max
23 %6 --> flaeche/max
24 %7 --> soll-X
25 %8 --> Soll-Y
26 %9 --> anzahl punkte nicht 0
27
28 %load(horzcat('..',filesep,'ROIGrid',ROINumber));
29
30 %startIndex --> Start der Version
31
32 for k = drange(3:s)
33     str = files(k,:)
34     if isempty(strfind(str,prefix))
35         k = k+1;
36         continue;
37     end
38
39     load(str);
40
41
42     Image(Image<0.5*max(Image(:))) = 0;
```

```

43
44     count = length(nonzeros(Image));
45     data(k-2+startIndex,9)= count;
46
47     xL = [];
48     yL = [];
49     i = 1;
50
51     mx = 0;
52     my = 0;
53     cMV = 0;
54
55     flaeche = 0;
56
57     si = size(Image);
58
59     for x = drange(1:si(2))
60         for y = drange(1:si(1))
61             %           uncomment for cutting the picture with the ROI
62             %           Image(y,x) = Image(y,x)*ROIGrid(y,x);
63             flaeche = flaeche + Image(y,x);
64             if(Image(y,x)>0)
65                 xL(i)=x;
66                 yL(i)=y;
67                 i=i+1;
68             end
69             if(Image(y,x)>cMV)
70                 mx = x;
71                 my = y;
72                 cMV = Image(y,x);
73             end
74         end
75     end
76
77     if sum(Image(:)) == 0
78         continue;
79     end
80
81     figure(100); imagesc(xAxis,yAxis,Image'); axis equal; set(gca,'YDir','normal');
82
83     l = length(xL);
84     m=0;
85
86
87     for i = drange(1:l)
88         for j = drange(1:l)
89             x1 = xL(i)-xL(j);
90             y1 = yL(i)-yL(j);
91             d = sqrt(x1*x1+y1*y1);
92             if(d>m)
93                 m=d;
94                 im = i;

```

```

95         jm = j;
96     end
97 end
98 end
99
100 data(k-2+startIndex,1) = m;
101 data(k-2+startIndex,2) = cMV;
102 data(k-2+startIndex,3) = flaeche;
103 data(k-2+startIndex,4) = yAxis(my)*100;
104 data(k-2+startIndex,5) = xAxis(mx)*100;
105 data(k-2+startIndex,6) = flaeche/cMV;
106
107 i = size(prefix,2)+1;
108 strX = str(i);
109 i = i+1;
110 while str(i)~='_'
111     strX = horzcat(strX,str(i));
112     i = i+1;
113 end
114
115 i = i+1;
116 strY='';
117 while str(i)~='_'
118     strY = horzcat(strY,str(i));
119     i = i+1;
120 end
121
122 strX = strX
123 strY = strY
124
125 data(k-2+startIndex,7) = str2double(strX);
126 data(k-2+startIndex,8) = str2double(strY);
127
128
129 end
130
131 startIndex=k-2;
132
133 save(horzcat('daten_version_',version),'files','data');

```

Listing 2: Korrekturfunktion

```

1 %%Korrektur Funktion:
2 %Image: zu Korrigierendes Bild
3 %correctionFile: Matlab-Daei mit den Informationen zum korrigieren
4 % (Korrekturkarte)
5 %
6 %correctionType (optional, Standardwert: 1): Art der Korrektur:
7 %           0: "Integral"
8 %           1: "normiertes Integral" (empfohlen)
9 %power (optional, Standardwert: 1): Potenz auf die Korrekturkarte

```

```

10 %           (--> Staerke der Korrektur)
11 %       fuer 0: 0.5
12 %       fuer 1: 1
13 function [Image] = correct(Image, correctionFile, correctionType, power)
14
15
16 %Setzen von Standardparamteren
17 if(nargin<3)
18     correctionType = 1;
19 end
20 if(nargin<4)
21     if(correcionType == 0)
22         power = 0.5;
23     else if(correctionType==1)
24         power = 1;
25     end
26 end
27 %0.5 wenn correctionType 0
28 %1 wenn correctionType 1
29
30 end
31
32 %laedt die Korrekturkarte
33 load(correctionFile);
34
35 %Auswahl der Korrekturkarte, je nach Type der Korrektur
36 if correctionType == 0
37     correctionMap = correctionMapIntegral;
38 else if correctionType == 1
39     correctionMap = correctionMapNormalizedIntegral;
40 end
41 end
42
43 %Potenzierung der Korrekturkarte
44 correctionMap = correctionMap .^ power;
45
46 %eigentliche Korrektur;
47 %elementweises dividieren
48 Image = Image./correctionMap;
49
50 end

```

Listing 3: Ermittlung der Helligkeit der zwei größten Objekte

```

1 function [maxG1,maxG2] = twoMainMaxima(Image2,useMax)
2     Image3 = Image2;
3
4     Image3(Image3~=0)=1;
5     Image3 = bwlabel(Image3,8);
6
7     maxim = max(Image3(:));

```



```

8
9     groupsSize = zeros(maxim+1,1);
10
11     maxValues = zeros(maxim+1,1);
12
13     for i = drange(0,maxim)
14         Image4 = Image2;
15
16         Image4(Image3~=i)=0;
17
18         groupsSize(i+1) = length(nonzeros(Image4));
19         if useMax == 1
20             maxValues(i+1) = max(Image4(:));
21         else
22             maxValues(i+1) = sum(Image4(:));
23         end
24     end
25
26
27     maxIndex = zeros(3,1);
28     maxGroupSize = zeros(3,1);
29
30     for i = drange(1,maxim+1)
31         g = groupsSize(i);
32
33         j=3;
34         while maxGroupSize(j)<g
35             j=j-1;
36             if j == 0
37                 break;
38             end
39         end
40
41         if j < 2
42             for k = drange(j+1,2)
43                 maxGroupSize(k+1) = maxGroupSize(k);
44                 maxIndex(k+1) = maxIndex(k);
45             end
46
47         end
48
49         if j~=3
50             maxGroupSize(j+1)=g;
51             maxIndex(j+1)=i;
52         end
53
54     end
55
56     maxG1 = maxValues(maxIndex(1));
57
58     maxG2 = maxValues(maxIndex(2));
59 end

```

9 Selbständigkeitserklärung

Hiermit versichern wir, dass wir diese Arbeit unter der Beratung durch Wei Yap Tan selbstständig verfasst haben und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt wurden.

Ort, Datum

Kai Bartnick

Ort, Datum

Sergey Lelyakin