

# **Entwicklung einer Softwarebibliothek zur automatisierten Analyse von Strahlabbildern an ELSA**

Samuel Kronenberg

Bachelorarbeit in Physik  
angefertigt im Physikalischen Institut

vorgelegt der  
Mathematisch-Naturwissenschaftlichen Fakultät  
der  
Rheinischen Friedrich-Wilhelms-Universität  
Bonn

September 2020

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate kenntlich gemacht habe.

Bonn, .....  
Datum

.....  
Unterschrift

1. Gutachter: Prof. Dr. Klaus Desch
2. Gutachter: Dr. Daniel Elsner

# Inhaltsverzeichnis

---

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>3</b>
2.1	Strahldynamik . . . . .	3
2.2	Anpassung mathematischer Modelle an Daten . . . . .	6
<b>3</b>	<b>Framegrabber-Architektur an ELSA</b>	<b>7</b>
3.1	Bisherige Architektur . . . . .	7
3.2	Geplante Architektur . . . . .	7
<b>4</b>	<b>Beschreibung der Funktionalitäten der Bibliothek</b>	<b>11</b>
4.1	Automatisiertes Detektieren multipler Strahlprofile . . . . .	12
4.2	Automatische Startwertbestimmung . . . . .	13
4.3	Ein- und zweidimensionale Anpassungen . . . . .	14
4.4	Beschreibung der Schnittstelle . . . . .	15
<b>5</b>	<b>Evaluation der Anpassungsgüte</b>	<b>17</b>
5.1	Statistische Untersuchung mit simulierten Daten . . . . .	17
5.2	Messung am Beschleuniger LAB . . . . .	22
<b>6</b>	<b>Fazit</b>	<b>25</b>
	<b>Literatur</b>	<b>27</b>
<b>A</b>	<b>Anhang</b>	<b>29</b>
A.1	Minimalbeispiel zur Nutzung der Bibliothek . . . . .	29
A.2	Statistische Untersuchung mit simulierten Daten . . . . .	30
	<b>Abbildungsverzeichnis</b>	<b>39</b>
	<b>Danksagung</b>	<b>41</b>



## Einleitung

---

Zentraler Bestandteil dieser Bachelorarbeit ist die automatisierte, softwaregestützte Auswertung von Strahlabbildern vor allem zur Anwendung an der Bonner Elektronen-Stretcher-Anlage (kurz: ELSA). ELSA ist ein dreistufiger Elektronenbeschleuniger, der einen quasi-kontinuierlichen Elektronenstrahl mit Energien von bis zu 3,2 GeV speichern oder extrahieren kann. Die Software zur Auswertung der Strahlabbilder wird eingesetzt, um die zentralen Eigenschaften des Strahls (vor allem Position und Ausdehnung) zu bestimmen. Hierfür wird der Strahl auf einen fluoreszierenden Schirm bzw. die in einem Magneten entstehende Synchrotronstrahlung direkt auf eine CCD-Kamera geleitet. Der fluoreszierende Schirm wird von einer Kamera gefilmt, sodass auf den Bildern das aktivierte Schirmmaterial zu erkennen ist. Aus den Eigenschaften der Strahlprofile (Breite, Höhe und Intensität) auf den Schirmen können die Eigenschaften des Elektronenstrahls bestimmt werden. Ziel in dieser Bachelorarbeit ist es, eine möglichst präzise Auswertung der Strahlprofile zu ermöglichen und diese Auswertung im Hinblick auf eine mögliche Automatisierung zu optimieren. Hierzu soll eine Software entwickelt werden, die unabhängig vom verwendeten Kamerasystem ist und somit für alle Beobachtungspunkte vergleichbare Ergebnisse liefert.

Hierbei lag der Fokus darauf, bestehende Probleme der bisherigen Auswertungssoftware zu lösen, wobei zunächst die Plattformunabhängigkeit im Mittelpunkt stand. Am Beschleuniger werden mehrere unterschiedliche Systeme verwendet, um die Strahlschirme zu beobachten, in der Vergangenheit gab es für jedes dieser Systeme eine eigene Auswertungssoftware. Durch Plattformunabhängigkeit lässt sich außerdem Wartungs- und Erweiterungsaufwand reduzieren. Ein Ziel war zudem die Verbesserung der Anpassungsergebnisse und vor allem die Erhöhung der Robustheit der Algorithmen auch bei Bildern, die die bisherige Software nur mit Einschränkungen verarbeiten konnte (z.B. überbelichtete Bilder bzw. im Bereich des Strahlprofils gesättigte Bilder).

Die genutzte Programmiersprache ist C++, da diese bereits für das Kontrollsystem der Beschleunigeranlage ELSA genutzt wird. Die komplette Neuentwicklung der Software bietet somit ebenfalls die Möglichkeit, moderne Methoden der Programmiersprache auszunutzen, um die definierten Hauptziele zu erreichen.



## Theoretische Grundlagen

### 2.1 Strahldynamik

Für den Betrieb eines Teilchenbeschleunigers und die Versorgung der angeschlossenen Experimente mit den beschleunigten Teilchen ist es essentiell, die Eigenschaften des Teilchenstrahls innerhalb des Beschleunigers zu kennen bzw. bestimmen und verändern zu können. Hierfür wird der Teilchenstrahl meist im mitbewegten Koordinatensystem eines „Sollteilchens“ beschrieben (vgl. [1]). Dieses Teilchen bewegt sich auf einer festgelegten Sollbahn durch alle Elemente des Beschleunigers. Hiermit kann die Bewegung aller Teilchen in Relation zum Sollteilchens bzw. zur Sollbahn beschrieben werden.

Hierfür wird ein spezielle Nomenklatur des Koordinatensystems verwendet (s. Abb. 2.1):

- $s$ : beschreibt die Koordinate tangential zur Sollbahn.
- $x$ : beschreibt die Koordinate senkrecht zur Sollbahn, die horizontal im Raum orientiert ist.
- $z$ : beschreibt die Koordinate senkrecht zur Sollbahn, die vertikal im Raum orientiert ist.

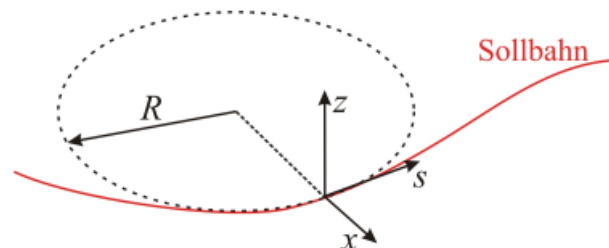


Abbildung 2.1: Koordinatensystem mit der verwendeten Nomenklatur zur Beschreibung von transversalen Teilchenbewegungen. (Quelle: [2])

### 2.1.1 Kenngrößen des Elektronenstrahls

Der Teilchenstrahl in einem Beschleuniger wird über verschiedene Kenngrößen charakterisiert. Wichtig sind hier vor allem die Emittanz  $\epsilon$  und die Amplituden- bzw. Beta-Funktion  $\beta(s)$ , aus diesen Größen ergibt sich unter anderem die Strahlbreite  $\sigma(s)$ .

Der Strahl wird im Phasenraum  $(x, x')$  bzw.  $(z, z')$  beschrieben (vgl. [3]), wobei  $x'$  bzw.  $z'$  jeweils die erste Ableitung der Ortskoordinate nach  $s$  bezeichnet. Ein Teilchen bewegt sich auf der sogenannten Phasenraumellipse, diese beschreibt alle möglichen transversalen Orte eines Teilchens an einer bestimmten Stelle  $s$ , sowie die Winkelrelation zum Orbit (vgl. Abb. 2.2). Orientierung und Fläche der Phasenraumellipse werden u.a. durch  $\epsilon$  und  $\beta(s)$  bestimmt. So ergibt sich der maximale Wert für  $x$  als:  $x_{max} = \sqrt{\epsilon \cdot \beta(s)}$ , dies entspricht der Einhüllenden des Teilchenensembles, wenn dieses nur aus Teilchen mit gleicher Emittanz besteht.

Da die Teilchen in Beschleunigern kanonischen Bewegungsgleichungen unterliegen, gilt das Liouville'sche Theorem (vgl. [4]). Dieses besagt, dass die Fläche der Phasenraumellipse konstant bleibt. Die Fläche der Phasenraumellipse entspricht  $F = \pi \cdot \epsilon$ .

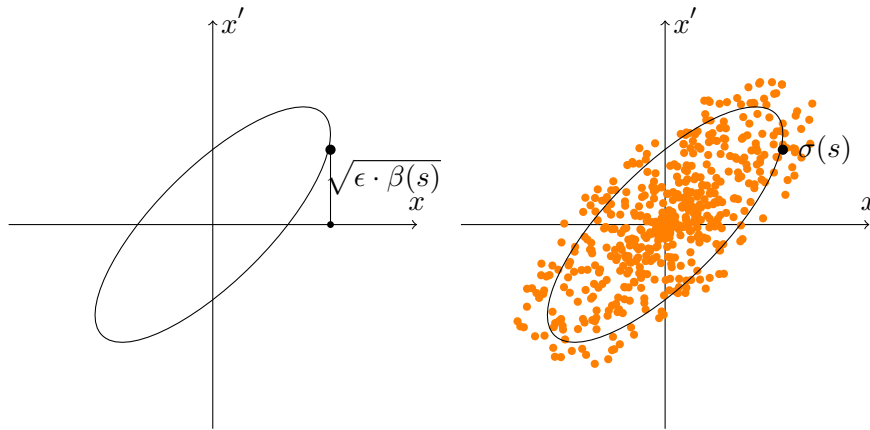


Abbildung 2.2: Die Phasenraumellipse eines Teilchens und eines Ensembles von Teilchen im Phasenraum  $(x, x')$ .

Ein Strahl in einem Teilchenbeschleuniger besteht aus Teilchen verschiedener Emittanzen. Die Strahlbreite  $\sigma(s)$  wird daher definiert als der Abstand von der Strahlachse, an der die Ladungsverteilung auf den Wert  $\rho_0 \cdot \exp(-\frac{1}{2})$  abgefallen ist. In [3] ist die Strahlbreite dann gegeben als:

$$\sigma(s) = \sqrt{\epsilon_{STA} \cdot \beta(s)}$$

Hierbei beschreibt  $\epsilon_{STA}$  die Emittanz eines Teilchens, das gerade  $\sigma(s)$  vom Orbit entfernt ist. Der Wert  $\epsilon_{STA}$  wird genutzt, um einen Strahl als Ensemble einer Vielzahl von Teilchen zu beschreiben. Hierbei wird der Index  $STA$  häufig weggelassen.

### 2.1.2 Bedeutung der Kenngrößen für einen Teilchenbeschleuniger

Es ist aus verschiedenen Gründen wichtig, die Emittanz in einem Teilchenbeschleuniger zu kennen, im Folgenden werden einige Beispiele aufgeführt. In zirkularen Beschleunigern (z.B. Speicherring von ELSA) wird die Beschleunigerstruktur sehr oft von den Teilchen durchlaufen.



Daher können schon kleine Abweichungen vom Orbit auf lange Sicht zu einem totalen Verlust des Strahles führen. Eine weitere Kenngröße eines Beschleunigers ist die Admittanz, sie entspricht der Fläche der größten Phasenraumellipse, die der Beschleuniger akzeptiert, dies ist in Abb. 2.3 dargestellt.

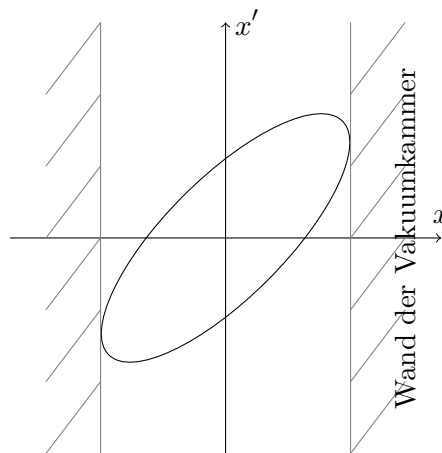


Abbildung 2.3: Die größte Phasenraumellipse, die ein Beschleuniger akzeptiert, liegt gerade innerhalb der gegebenen Apertur. Die Fläche dieser Ellipse wird als Admittanz bezeichnet.

Für den erfolgreichen Betrieb eines Beschleunigers ist eine Admittanz nötig, die deutlich größer ist als die Emittanz des Strahls, damit möglichst wenige Teilchen verloren gehen. Ferner entspricht eine höhere Emittanz auch einer höheren (Winkel-)Divergenz. An den Hadronenphysikexperimenten an ELSA wird ein Strahl mit möglichst kleiner Divergenz benötigt.

### 2.1.3 Bestimmung der Messgrößen mittels Strahlschirmen

Strahlschirme werden genutzt, u.a genutzt um die Strahlbreite  $\sigma$  und daraus bei bekanntem  $\beta(s)$  die Emittanz  $\epsilon$  zu bestimmen. Hierfür gibt es zwei wichtige Möglichkeiten.

**Strahlschirme im Teilchenstrahl** Es können Schirme mit fluoreszierenden Materialien in den Strahl eingebracht werden. Damit kann die Form des Elektronenstrahls dargestellt werden. Jedoch handelt es sich hierbei um eine destruktive Nachweismethode, die daher nicht im Speichermodus des Kreisbeschleunigers verwendet werden kann. Mittels einer Kamera kann das Strahlprofil auf dem Schirm aufgenommen werden, wobei eine höhere Intensität der Fluoreszenz einer höheren Ladungsdichte im Elektronenstrahl entspricht. Die Abbildung auf dem Strahlschirm entspricht also einem Schnitt durch den Elektronenstrahl. Damit kann die Strahlbreite  $\sigma$  in beiden transversalen Dimensionen bestimmt und bei bekannten  $\beta$ -Funktionen zusätzlich die Emittanz  $\epsilon$  in diesen Ebenen berechnet werden.

**Synchrotronlichtmonitore** Geladene Teilchen, die in einem Magnetfeld abgelenkt werden, emittieren Synchrotronstrahlung. Ein Teil dieser Strahlung liegt im sichtbaren Spektrum. Durch direkte Abbildung auf eine Kamera kann aus den Eigenschaften der Abbildung der Synchrotronstrahlung auf die Eigenschaften des Quellpunkts, die sehr genau der Dimension des Elektronenstrahls entsprechen, geschlossen werden (vgl. [5]).

## 2.2 Anpassung mathematischer Modelle an Daten

Das Anpassen einer Funktion an eine Menge von Datenpunkten stellt ein Optimierungsproblem dar. Optimiert wird die Abstandssumme (auch Residuensumme) von der anzupassenden Funktion und der Menge der Datenpunkte. Diese Vorgehensweise entspricht dem mathematischen Standard Optimierungsprobleme als Minimierungsprobleme zu formulieren.

In der Anpassung von Funktionen an Datenpunkte ist das Verfahren der kleinsten Quadrate der Standard und wird auch in dieser im weiteren Verlauf beschriebenen Bibliothek genutzt.

### 2.2.1 Minimierungsprobleme

Bei der Anpassung einer Funktion an eine Menge von Daten handelt es sich um ein kontinuierliches Problem, das es unmöglich macht alle Konfigurationen auszuprobieren und die optimale Konfiguration zu finden. Es ist also nötig eine zu optimierende Funktion  $g(x)$  zu definieren und deren Minimum zu finden. Hierfür wird die Ableitung dieser Funktion betrachtet, da ein Extremwert der Funktion einer Nullstelle der ersten Ableitung entspricht:

$$\frac{d}{dx}g(x) \stackrel{!}{=} 0.$$

Wurde eine Nullstelle der Ableitung gefunden, kann die Funktion in der Umgebung dieser Stelle untersucht werden, um festzustellen, ob es sich um ein Minimum handelt (vgl. [6]). Da bei größeren Problemen die Ableitung häufig nicht einfach zu bestimmen ist, nutzen einige Algorithmen die Eigenschaften der Jakobi- und Hessematrizen aus, um eine numerische Bestimmung zu erreichen. Eine genauere Beschreibung findet sich bei [7].

### 2.2.2 Methode der kleinsten Fehlerquadrate

Als zu optimierende Funktion  $g(x)$  eignet sich die Summe aller quadrierten Residuen - also das Quadrat des Abstands  $r(x_j) = d_j - f(x_j)$ , wobei  $d_j$  den Wert an der Stelle  $x_j$  repräsentiert und  $f(x_j)$  den Funktionswert der angepassten Funktion an der gleichen Stelle. In diesem Fall nimmt die Optimierungsfunktion die folgende Form an (vgl. [7]):

$$g(x_j) = \frac{1}{2} \sum_{j=1}^m r^2(x_j)$$

Dies hat den Vorteil, dass das Minimum einer quadratischen Funktion eindeutig bestimmt werden kann.

Für die Anpassung einer Funktion, die von zwei Variablen abhängt, z.B. zur Beschreibung der Helligkeitsverteilung eines Bildes, muss also das Residuum für jeden Pixel berechnet werden. Dies ergibt schon bei einem Bild der Größe  $(200 \times 300)\text{px}$   $6 \cdot 10^4$  Datenpunkte, die bei jeder Iteration berechnet werden müssen. Die Anzahl dieser Punkte steigt quadratisch mit der Kantenlänge des Bildes.

Damit die benötigte Rechendauer gering bleibt, gibt es verschiedene Algorithmen, die das Minimum in möglichst wenigen Iterationen finden. Im Rahmen dieser Bachelorarbeit wird der Levenberg-Marquardt-Algorithmus verwendet (weitere Informationen: [7]).

---

## Framegrabber-Architektur an ELSA

---

Zur Beschreibung der Architektur und der einzelnen Verarbeitungsprozesse wird die Hard- und Software in Ebenen eingeteilt. Die Kameras an den einzelnen Strahlschirmen bilden die sogenannte Kameraebene. Die Anbindung an Diagnose-PCs erfolgt entweder direkt oder über sogenannte *Framegrabber*. Auf den Diagnose-PCs findet dann die Software zur Auswertung der Bilder und das Bestimmen der gewünschten Parameter der aufgenommenen Strahlprofile statt.

### 3.1 Bisherige Architektur

Zu Beginn dieser Arbeit werden vier verschiedene Modelle von *Framegrabbern* genutzt. Diese sind teilweise aufgrund von Inkompatibilitäten der eigentlichen „Digitalisierungskarte“ mit modernen Betriebssystemen an verschiedene PCs angeschlossen. Die *Framegrabber* unterscheiden sich auch in der Form der Datenausgabe. Daher existiert aktuell für jeden *Framegrabber* eine eigenständige Auswertungssoftware. Die Aufgaben, welche mittels der Auswertungssoftware gelöst werden, sind die Aufarbeitung der digitalisierten Bilder (z.B. durch Untergrundkorrektur oder Berechnung des Durchschnitts über mehrere Bilder), das Anpassen von Funktionen an diese Bilder und das Senden der Anpassungsergebnisse an das Kontrollsystem. Die verschiedenen Versionen der Auswertungssoftware unterscheiden sich z.T. nicht nur in der Verarbeitung der von den *Framegrabbern* gelieferten Daten, sondern auch in Art und Umfang der zur Verfügung stehenden Auswertungsmethoden. Die Ergebnisse werden dann von den einzelnen Rechnern in das Kontrollsystem eingespeist. Dieser Datenfluss ist in Abb. 3.1 dargestellt.

Eine große Einschränkung der bisherigen Software besteht darin, dass an den Bildern keine echte zweidimensionale Anpassung durchgeführt werden kann. In der aktuellen Software stehen nur Anpassungen zur Verfügung, die auf Projektionen des zweidimensionalen Bildes auf zwei jeweils eindimensionale Profile (horizontal und vertikal) basieren. Durch die Projektionen gehen Informationen verloren, wodurch die Anpassungsgüte sinkt.

### 3.2 Geplante Architektur

Die oben genannten Einschränkungen sollen durch ein neues Design der Auswertungssoftware behoben werden, das die bisherige Software in mehrere Module aufteilt. Dadurch lassen sich

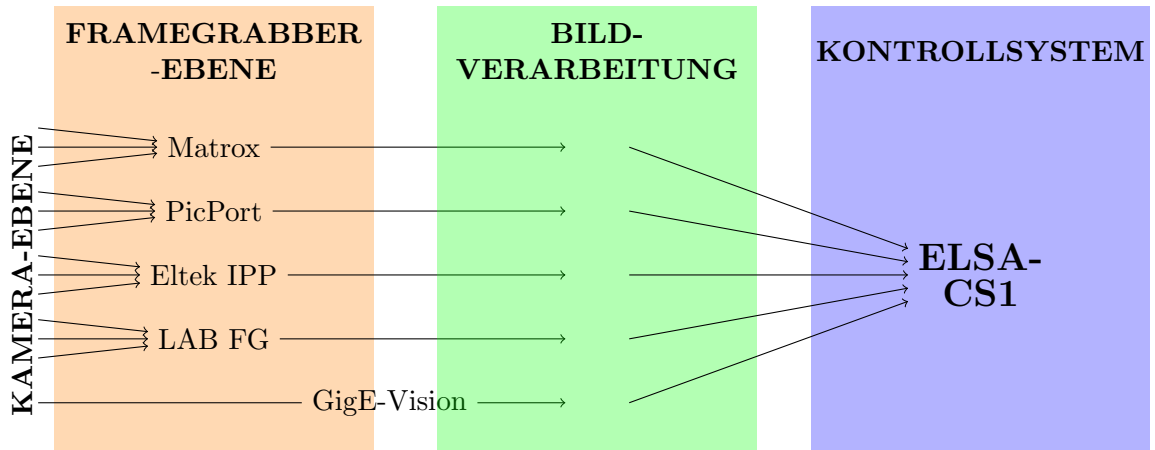


Abbildung 3.1: Bisherige Architektur der Framegrabber und der angeschlossenen Auswertungssoftware.

zukünftige Modernisierungen in Hard- und Software einfacher integrieren. Hierzu wird der Datenfluss in drei Blöcke aufgeteilt, die mit Schnittstellen verbunden sind. Hierbei werden folgende Blöcke gebildet:

- **Kamera und *Framegrabber*:** Anbindung der einzelnen Kameras an geeignete Framegrabber zur Aufbereitung der Bildinformationen bzw. direkt an die Auswertungssoftware,
- **Bildverarbeitung und -analyse:** Bearbeiten der Bilder, Korrektur von Verzerrung und Bildfehlern, Analyse und Bestimmung der gewünschten Parameter,
- **Kontrollsystem:** Sammeln und Überwachen aller Beschleunigerparameter.

Der entsprechende Datenfluss ist zum Vergleich mit dem aktuellen Datenfluss in Abb. 3.2 dargestellt.

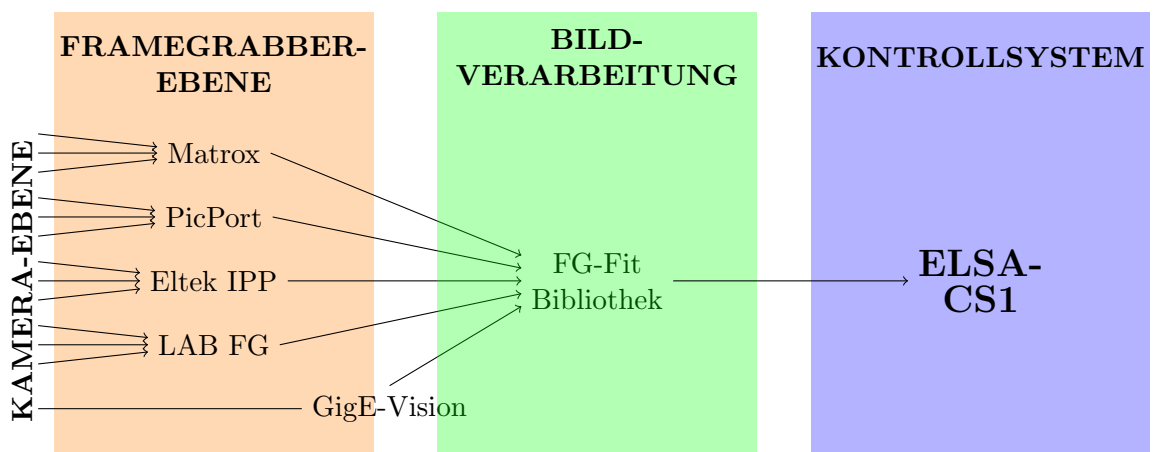


Abbildung 3.2: Geplante Architektur der Framegrabber und der angeschlossenen Auswertungssoftware.

### 3.2.1 Anforderungen an die Auswertungssoftware

Auf Grundlage der Überlegungen zur Umstrukturierung der Auswertung der Schirmbilder werden grundlegende Anforderungen an die Auswertungssoftware formuliert. Diese werden um einige Punkte erweitert, die die Qualität der erhaltenen Daten, sowie die Pflfegbarkeit der Software verbessern sollen. Hiermit werden folgende Anforderungen festgelegt:

1. Unabhängigkeit von *Framegrabber*-Modellen,
2. Möglichkeit sowohl ein- als auch zweidimensionale Anpassungen auszuführen,
3. einfache Erweiterbarkeit durch die Endnutzer,
4. Durchführung grundlegender Korrekturen (z.B. zur Kompensation von Verkippung).

Zudem werden bisher genutzte oder für die Zukunft als essentiell betrachtete Modelle implementiert, hierzu zählen: Ein- und zweidimensionale Gauß-Funktion (zusätzlich mit einem konstanten Untergrund)

$$f(x, z) = I_0 \cdot \exp \left( - \left( \frac{(x - \mu_x)^2}{2 \cdot \sigma_x^2} + \frac{(z - \mu_z)^2}{2 \cdot \sigma_z^2} \right) \right) + n,$$

zweidimensionale Gauß-Funktion mit Rotationswinkel

$$g_r(x, z) = I_0 \cdot \exp \left( - (a(x - \mu_x)^2 + 2b(x - \mu_x)(z - \mu_z) + c(z - \mu_z)^2) \right) + n, \quad (3.1)$$

$$a = \frac{\cos^2 \alpha}{2\sigma_x^2} + \frac{\sin^2 \alpha}{2\sigma_z^2}, \quad b = -\frac{\sin^2 2\alpha}{4\sigma_x^2} + \frac{\sin^2 2\alpha}{4\sigma_z^2}, \quad c = \frac{\sin^2 \alpha}{2\sigma_x^2} + \frac{\cos^2 \alpha}{2\sigma_z^2},$$

und die eindimensionale Fehlerfunktion für Spezialanwendungen:

$$f_{\text{erf}}(x) = \frac{A}{2} \cdot \text{erf}(m \cdot (x - x_0)) + \frac{A}{2},$$

An einigen optischen Aufbauten können mehrere Strahlprofile innerhalb des Bildes liegen, auch diese sollen mit der neuen Auswertungssoftware auswertbar sein.

### 3.2.2 Design der Auswertungsbibliothek

Ziel der Arbeit ist es den Teil der Software, der zur Auswertung der Strahlprofile dient, in einer Bibliothek für alle *Framegrabber* zur Verfügung zu stellen. Im Gegensatz zum ersten Entwurf, der eine Trennung der Bibliothek in Bildbearbeitung und Bildkorrektur sowie Modellanpassung vorsah, wurde entschieden, die Bibliothek nicht inhaltlich aufzuteilen und einfache Bildbearbeitungsschritte als Teil des Anpassungsalgorithmus zu integrieren. Der Grund für diese Entscheidung war, dass diese grundlegenden Bildkorrekturen essentiell für das Bestimmen geeigneter Startparameter für die eigentliche Anpassung sind. Das Kompensieren der Verkippung, die durch die Abbildung auf einen geneigten Schirm entsteht, wird in die *Framegrabber*-Ebene verschoben, da diese Werte aufbauspezifisch und konstant sind. Zudem werden die Daten an dieser Stelle ohnehin in das in dieser Bibliothek verwendete Format konvertiert, daher kann dort auch die individuelle Kompensation leicht erfolgen.

Grundsätzlich ist es mit der neuen Architektur sowohl möglich die zur Verfügung stehenden Funktionen an die Dimensionen des eigentlichen Bildes (in pixel-Werten) als auch an die tatsächliche physikalische Dimension, unter Berücksichtigung des Abbildungsmaßstabs, anzupassen.



---

## Beschreibung der Funktionalitäten der Bibliothek

---

Im Folgenden wird auf die Kernfunktionalitäten der Auswertungsbibliothek eingegangen. Die Bibliothek ist in C++ geschrieben. Die Funktionalitäten stehen jedem C++-Programm zur Verfügung. Eine Übergabe der Daten an Programme, die in einer anderen Programmiersprache entwickelt wurden, ist zunächst nativ nicht vorgesehen. Bei zukünftigen Bedarf kann eine Schnittstelle zu einer anderen Programmiersprache implementiert werden.

Die Eingangsdaten für die Bibliothek sind Bilder der Kameras als zweidimensionale Pixelmatrizen. Die Bilder werden als Datentyp `cv::Mat` aus der Bibliothek `OpenCV` (vgl. [8]) gespeichert. Alle hier verwendeten Bilder werden in Graustufen mit einem Byte pro Pixel aufgenommen, dies bedeutet, dass jedem Pixel ein Pixelwert zwischen 0 und 255 zugeordnet wird, der die Helligkeit des Pixels repräsentiert<sup>1</sup>. Hierbei repräsentiert der Pixelwert 0 die untere Helligkeitsgrenze (schwarz) und 255 die obere Helligkeitsgrenze (weiß). Das Bild wird dann verarbeitet, um automatisiert Strahlprofile zu identifizieren. Alle gefundenen Strahlprofile und die bestimmten Anfangswerte für die gefundenen Profile werden in einem Objekt der Klasse `PeakProperties` gespeichert. Die Anfangswerte stellen eine grobe Voranalyse von Intensität, Position und Breite des Profils dar. Die Startwerte werden dann in Objekt der Klasse `FitMat` abgelegt. Dieses Objekt beinhaltet außerdem die Matrix mit Pixelwerten, an die die Anpassung erfolgen soll. Die hier erwähnten Schritte werden im Folgenden detaillierter erklärt.

Es ist möglich die Anpassung in Einheiten der Pixeldimensionen des Bildes oder in physikalischen Einheiten durchzuführen. Wenn physikalische Einheiten genutzt werden sollen, müssen diese als zwei weitere zweidimensionale Pixelmatrizen übergeben werden, von denen eine die  $x$ -Koordinaten und die andere die  $z$ -Koordinaten eines jeden Pixels der Bildmatrix enthält. Für den Fall einer eindimensionalen Anpassung erfolgt die Übergabe der Koordinaten in Form einer  $(1 \times N)$ - bzw.  $(N \times 1)$ -Matrix.

---

<sup>1</sup> Durch Besonderheiten der Kamera oder Leistungsverluste in den verwendeten Kabeln kann dieser Wert geringer sein.

## 4.1 Automatisiertes Detektieren multipler Strahlprofile

Für die zweidimensionalen Anpassungen von Gaußfunktionen wurde eine automatische Startwertbestimmung implementiert (zur eigentlichen Bestimmung der Startwerte s. Abs. 4.2). Diese ist in der Lage mehrere Strahlprofile auf einem Bild zu finden. Um diese Funktionalität umzusetzen, wird die Konturerkennung der C++-Bibliothek *OpenCV* verwendet. Zunächst wird ein Gaußscher Weichzeichner<sup>2</sup> auf das Bild angewendet, um mögliche Störpixel zu reduzieren. Anschließend wird ein binärer Filter mit einer konfigurierbaren Filterschwelle angewendet, der alle Pixel mit Pixelwerten unterhalb der Schwelle auf den Pixelwert 0 und alle Werte ab dem Schwellwert auf den Maximalwert von 255 setzt. Hierbei gibt es zwei Möglichkeiten diesen binären Filter anzuwenden:

- manuell: Der Schwellenwert kann von außen gesetzt und angepasst werden.
- automatisch: Der optimale Schwellwert wird automatisch durch Nutzung der *OpenCV*-Bibliothek<sup>3</sup> bestimmt.

Auf das so erzeugte Schwarz-Weiß-Bild wird die Konturerkennung angewendet und ein Begrenzungsrechteck bestimmt. Davon ausgehend sind mehrere Vorgehensweisen möglich:

- „Zerschneiden“ des Bildes in mehrere Submatrizen, von der jeder ein Strahlprofil enthält und Anpassung einer Funktion an jede einzelne Submatrix,
- Bestimmen der Anzahl der Strahlprofile und Anpassung einer Superposition der gleichen Anzahl von Funktionen oder
- eine Mischung beider Varianten, die basierend auf dem Abstand zweier benachbarter Strahlprofile „schneidet“ und dann die Anzahl der Strahlprofile in den Submatrizen bestimmt, um wenn nötig eine multiple Anpassung durchzuführen.

Diese Auswahlmöglichkeiten sind zudem in Abb. 4.1 graphisch dargestellt.

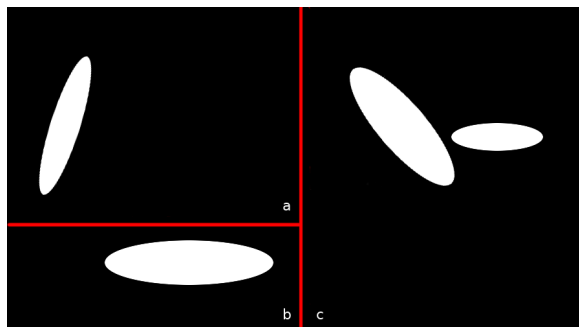


Abbildung 4.1: Skizze zur Veranschaulichung der Möglichkeiten zur „Zerschneidung“ der Matrix, wenn mehrere Strahlprofile auf einem Bild zu erkennen sind. In *a* und *b* ist jeweils ein Profil pro Submatrix, in *c* sind zwei dicht beieinander liegende Profile dargestellt.

In Abb. 4.1 ist ein bereits gefiltertes Bild skizziert, auf dem 4 Strahlprofile zu erkennen sind.

---

<sup>2</sup> Radius: 13 px

<sup>3</sup> Hierbei findet das in OpenCV implementierte Otsu-Schwellwertverfahren Anwendung. Eine Erläuterung dieses Verfahrens findet sich in [9].



Die Strahlprofile in den Submatrizen  $a$  und  $b$  sind dabei ausreichend voneinander entfernt und können getrennt von einander betrachtet werden. Im Gegensatz dazu sind die beiden Strahlprofile in Submatrix  $c$  so nah aneinander, dass sie überlappen. Dementsprechend müssen sie als Superposition zweier Profile betrachtet werden.

Jedes gefundene Strahlprofil wird als ein Objekt der Klasse `PeakProperties` abgelegt. Später werden aus diesen Objekten Objekte der Klasse `FitMat` erzeugt, die neben den Startwerten auch die gewünschte Submatrix enthalten. Entsprechend lassen sich in einem Objekt der Klasse `FitMat` auch die Startwerte für mehrere Strahlprofile speichern.

## 4.2 Automatische Startwertbestimmung

Um eine genaue Anpassung in möglichst wenigen Iterationen zu erhalten, ist es essentiell, für die zu bestimmenden Eigenschaften Startwerte in der korrekten Größenordnung zu ermitteln und zu verwenden. Im Falle der zweidimensionalen Anpassung sind diese Eigenschaften Intensität  $I_0$ , Position  $(\mu_x, \mu_z)$  und Breiten  $\sigma_x, \sigma_z$  (vgl. Gleichung 3.1). Aus den gefundenen Begrenzungsrechtecken können Startwerte zur Durchführung der Anpassungen bestimmt werden. Diese Startparameter werden zunächst im zugehörigen Objekt der `PeakProperties`-Klasse abgelegt. Somit stehen sie dann für die Anpassung im `FitMat`-Objekt zur Verfügung. Das jeweilige Begrenzungsrechteck ist ein Objekt der Klasse `cv::BoundRect` und hat als Eigenschaften: Höhe ( $h$ ) und Breite ( $w$ ) des Rechtecks sowie die  $x$ - und  $z$ -Koordinate der linken oberen Ecke  $(x_{10}, z_{10})$ . Aus diesen Informationen kann bereits näherungsweise die Position des Mittelpunkts  $\mu$  der Gauß-Kurve bestimmt werden. Dies erfolgt sowohl in der  $x$ - als auch in der  $z$ -Ebene:

$$\mu_x = x_{10} + \frac{w}{2} \quad \mu_z = z_{10} + \frac{h}{2} \quad (4.1)$$

Die maximale Intensität  $I_0$  wird abgeschätzt als Intensität an der Stelle  $(\mu_x, \mu_z)$  auf dem mit einem Gauß-Filter bearbeitetem Bild:

$$I_0 = I_{\text{filtered}}(\mu_x, \mu_z)$$

Aus diesen Informationen können dann auch Startwerte für die Größen  $\sigma_x$  und  $\sigma_z$  bestimmt werden. Hierfür muss die Gaußfunktion umgestellt werden:

$$\begin{aligned} I(x) &= I_0 \cdot \exp\left(-\frac{(x - \mu_x)^2}{2 \cdot \sigma_x^2}\right), \\ \Rightarrow \ln(I(x)) &= \ln(I_0) - \frac{(x - \mu_x)^2}{2 \cdot \sigma_x^2}. \end{aligned} \quad (4.2)$$

Bekannt ist in der Auswertung bereits  $I_0$  sowie die Dimensionen des Begrenzungsrechtecks. Da vorher ein binärer Filter angewendet wurde, entspricht der Schwellwert  $I_{\text{thresh}}$  dem Wert  $I(x)$ , wobei  $x$  definiert werden kann als:

$$x = \mu_x - \frac{w}{2} = x_{10}$$

Werden die bekannten Größen in Gleichung 4.2 eingesetzt ergibt sich bei Nutzung der Gleichung 4.1:

$$\ln(I_{\text{thresh}}) = \ln(I_0) - \frac{(x_{10} - \mu_x)^2}{2 \cdot \sigma_x^2}$$

$$\sigma_x = \frac{w}{2 \cdot \sqrt{2} \sqrt{\ln\left(\frac{I_{\text{thresh}}}{I_0}\right)}} \quad (4.3)$$

Die Bestimmung von  $\sigma_z$  erfolgt analog. Die Startwerte für den konstanten Untergrund und Rotationswinkel sind nicht trivial zu bestimmen. Basierend auf den Ergebnissen der in Abschnitt 5 beschriebenen Untersuchungen hat die Wahl der Startwerte für Rotationswinkel und Intensitätsoffset keinen Einfluss auf die Anpassungsgüte, daher werden diese jeweils auf 0 gesetzt.

### 4.3 Ein- und zweidimensionale Anpassungen

Für die Anpassungen der verschiedenen Modelle an die Daten wird die C++-Bibliothek *Gnu Scientific Library*[10] (im Folgenden kurz: GSL) genutzt. Als Anpassungsalgorithmus wird der Levenberg-Marquardt-Algorithmus verwendet. Nativ sind die Anpassungsmethoden der GSL für eindimensionale Anpassungen konzipiert. Zur eindimensionalen Anpassung stehen zur Zeit eine Gauß-Funktion, eine Gauß-Funktion mit Offset und eine Fehlerfunktion zur Verfügung. Der Aufruf zur Anpassung einer dieser Funktionen lautet:

```
int fit_to_data(ELSA::FrameGrab::GaussFitResults &fit_results,
               fit_function f, ELSA::FrameGrab::FitMat &fm, ELSA::FrameGrab
               ::BaseVectors &BV);
```

Hierbei handelt es sich bei `GaussFitResults` um eine neu entwickelte Klasse, die die einzelnen Anpassungs-Parameter für eine Gauß-Funktion enthält. `fit_function` setzt die anzupassende Funktion fest.

Der Aufruf für eine zweidimensionale Anpassung ist identisch. Als Anpassungsfunktionen stehen zur Verfügung:

- zweidimensionale Gaußfunktion:  

$$g(x, z) = I_0 \cdot \exp\left(-\left(\frac{(x-\mu_x)^2}{2\cdot\sigma_x^2} + \frac{(z-\mu_z)^2}{2\cdot\sigma_z^2}\right)\right) + n,$$
- zweidimensionale Gaußfunktion mit *Flattop*:  

$$f(g(x, z)) = \begin{cases} g(x, z), & g(x, z) \leq 255 \\ 255, & \text{sonst} \end{cases},$$
- zweidimensionale Gaußfunktion mit *Flattop* und Rotationswinkel:  

$$f_r(g_r(x, z)) = \begin{cases} g_r(x, z), & g_r(x, z) \leq 255 \\ 255, & \text{sonst} \end{cases} \quad \text{mit } g_r(x, z) \text{ aus Gleichung 3.1,}$$
- Superposition von  $n$  Profilen mit *Flattop* und Rotationswinkel:  

$$f_{\text{mul}}(x, z) = \sum_{i=1}^n (f_{r,i}(x, z))$$

#### Notwendigkeit der Gauß-Funktion mit Flattop

Die Anpassung einer Gaußfunktion an ein übersättigtes Bild liefert keine gute Repräsentation der Daten, dieses Problem soll gelöst werden. Ein übersättigtes Bild kann entstehen, da die Helligkeit in Graustufen angegeben wird. Jede Intensität  $I$ , die höher ist als die Intensität  $I_{255}$  - die niedrigste Intensität, der ein Graustufenwert von 255 zugeordnet wird - entspricht in

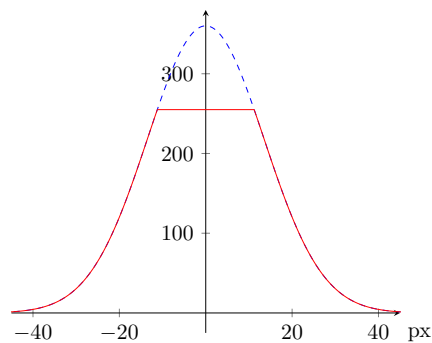


Abbildung 4.2: Eine Gaußfunktion mit  $I_0 > I_{255}$  und ihre Darstellung in einem Graustufenbild mit 1 Byte pro Pixel.

Graustufen ebenfalls einem Wert von 255.

Dies führt zu dem in Abb. 4.2 skizzierten Abschneiden des Gauß-Profiles. Eine Gaußfunktion mit einer solch hohen Intensität erscheint als „Gaußstumpf“, das Bild wird als übersättigt bezeichnet.

Um an einem übersättigten Bild eine sinnvolle Anpassung durchführen zu können, ist es nötig, dass die anzupassende Funktion dem Datenverlauf entspricht. Die Funktion mit *Flattop* existierte bisher nicht, mit dieser Funktion ist es nun möglich auch im übersättigten Fall ein sinnvolles Anpassungsergebnis zu erhalten.

### Umsetzung von zweidimensionalen Anpassungen mit der GSL

Es wurde bereits beschrieben, dass die GSL nicht nativ für zweidimensionale Anpassungen entwickelt ist. Es musste also ein Algorithmus gefunden werden, der das zweidimensionale Problem so umformuliert, dass es mit der GSL lösbar ist. Eine solche Methode muss das Problem eindimensional formulieren ohne Informationen einer echten zweidimensionalen Anpassungen zu verlieren. Hierfür wird die Pixelmatrix zeilenweise in einen einzigen Vektor überführt. Gleiches passiert mit den Pixelmatrizen, die die Koordinaten enthalten.

Diese Vektoren werden dann als Daten bzw. unabhängige Variablen an die GSL übergeben. Die GSL erwartet außerdem eine Funktion, die in jeder Iteration die Residuen des Modells - mit den in dieser Iteration verwendeten Parametern - zu den tatsächlichen Messdaten berechnet. Aufgrund der beschriebenen Speicherung der Koordinaten sowie der Pixelwerte, kann nun für jeden Punkt das Residuum berechnet werden.

## 4.4 Beschreibung der Schnittstelle

Zur Nutzung der Bibliothek aus der an das Kontrollsystem angeschlossenen Software ist eine definierte Schnittstelle nötig. Hierbei gibt es die Möglichkeit, die Schnittstelle auf einen Aufruf zu begrenzen oder mehrere Funktionen als Schnittstelle zu definieren. Im Laufe der Entwicklung fiel die Entscheidung auf die letztere Variante, da auch diese übersichtlich und nur mit wenigen Aufrufen verbunden ist und für den Nutzer besser nachzuvollziehen ist, wie die Daten verarbeitet werden. Somit sind die drei wichtigen Aufrufe, die die Schnittstelle zur Bibliothek darstellen:

### Peak-Detection

```
std::vector<FitMat> peak_detection(const cv::Mat &src, double
    threshold_value, const bool &sliceMat=false, const bool &
    multipleFit=false, const bool &autoThresh=true);
```

Dieser Aufruf führt die beschriebene automatische Detektion der Strahlprofile durch. Benötigt wird die Pixelmatrix als `cv::Mat` und einen Schwellwert für den binären Filter - dieser wird auch benötigt, wenn die automatische Schwellwertbestimmung genutzt werden soll.

Die weiteren Argumente sind Boolesche Werte, die bestimmte Funktionalitäten aktivieren bzw. deaktivieren. Im Einzelnen bedeuten diese Variablen:

1. `sliceMat`: Die Pixelmatrix wird, wie in Abb. 4.1 dargestellt, in Submatrizen unterteilt, die eine möglichst kleine Anzahl an Strahlprofilen beinhalten,
2. `multipleFit`: Liegen mehrere Strahlprofile in einer Submatrix, werden diese in dem gleichen `FitMat`- Objekt abgelegt. Diese Funktion sollte nur mit der Anpassungsfunktion zur Anpassung mehrerer Strahlprofile verwendet werden,
3. `autoThresh`: Automatische Bestimmung des Schwellwerts für den binären Filter.

### Basis-Vektoren

```
BaseVectors(cv::Mat &dim1, cv::Mat &dim2, FitMat &fm);
BaseVectors(cv::Mat &mat, FitMat &fm, bool const &oneDimFit);
```

Dieser Aufruf erzeugt die Basisvektoren für die Anpassung. Hierbei dient der obere Aufruf zur Nutzung von physikalischen Koordinaten. Die erste Pixelmatrix muss dabei die Informationen der Horizontalen und die zweite Pixelmatrix die Informationen der Vertikalen enthalten. Der untere Aufruf kann zur Anpassung an Pixelkoordinaten für die zweidimensionale Anpassung (`oneDimFit = false`) oder für eine eindimensionale Anpassung an physikalische Koordinaten (`oneDimFit = true`) genutzt werden.

### Datenanpassung der Modelle

```
int fit_to_data(ELSA::FrameGrab::GaussFitResults &fit_results,
    fit_function f, ELSA::FrameGrab::FitMat &fm, ELSA::FrameGrab
    ::BaseVectors &BV, const double upper_limit=255);
```

Dieser Aufruf führt die eigentliche Anpassung mittels der bestimmten Startparameter an die entsprechende (Sub-)Matrix durch. Die Ergebnisse werden in dem Objekt der Klasse `GaussFitResults` abgelegt. Hierbei ist für jeden Parameter der Anpassungswert und der reine Anpassungsfehler bestimmt aus der Kovarianzmatrix gespeichert. Entspricht der maximale Ausgabewert der Kamera aufgrund von Verlusten in den Kabeln oder Besonderheiten der Kamera nicht 255, kann dieser hier angepasst werden. Der gesetzte Wert beeinflusst die Höhe des *Flattops* bei allen Anpassungen, die ein *Flattop* enthalten.

Ein Minimalbeispiel zur Nutzung der Schnittstelle und der Bibliothek ist in Anhang A.1 aufgeführt.

---

## Evaluation der Anpassungsgüte

---

### 5.1 Statistische Untersuchung mit simulierten Daten

Um die Verhaltensweisen und die Güte der einzelnen Anpassungen untersuchen zu können, wurde ein Generator für Bilder entwickelt<sup>1</sup>, die den echten Aufnahmen der Strahlschirme ähneln. Es wird ein gaußförmiger Strahlfleck berechnet, dessen Mittelpunkt  $(\mu_x, \mu_z)$ , Standardabweichungen  $\sigma_x, \sigma_z$ , Intensität  $I_0$  und Rotationswinkel  $\alpha$  verändert werden kann. Unter anderem wurde eine statistisch verteilte Abweichung jedes Pixels vom Idealwert als Rauschen implementiert, sowie die Existenz von Pixeln, die beschädigt sind und entweder kein Signal oder ein zufälliges Signal liefern.

Mithilfe dieses Generators werden einige zur Verfügung stehende Anpassungen miteinander verglichen, sowie der Einfluss verschiedener Parameter auf Anpassungsgüte und benötigter Ausführungszeit untersucht.

Eine Auswahl möglicher Bilder, die vom Generator erzeugt wurden, sind in Abb. 5.1 dargestellt. In Abb. 5.2 ist zudem eine dreidimensionale Darstellung eines solchen Bildes, sowie die daran berechnete Anpassung dargestellt.

#### 5.1.1 Vergleich von Gaußfunktion und Gaußfunktion mit Flattop bei Strahlprofilen ohne Rotationswinkel

Die neu eingeführte Gauß-Funktion mit *Flattop* soll mit der normalen Gaußfunktion verglichen werden. Hierbei wird das Verhalten bei gesättigten und nicht gesättigten Bildern getrennt beobachtet. Es wurden insgesamt 5000 Bilder mit beiden Anpassungen untersucht. Es wurde jeweils die Abweichung des angepassten Werts für einen Parameter ( $I_0, \sigma_{vert}$  oder  $\sigma_{hor}$ ) und des simulierten Werts untersucht. Hierbei wird die Notation  $\Delta I_0 = I_{0,Anpassung} - I_{0,simuliert}$  genutzt. Die Ergebnisse dieser statistischen Untersuchung sind in Tab. 5.1 aufgeführt.

An die Häufigkeitsverteilungen der einzelnen Parameter wurden jeweils Poisson-Verteilungen angepasst. Der Fehler auf den Mittelwert entspricht somit der Standardabweichung der Poissonverteilung. Die Ergebnisse der Untersuchung sind zudem graphisch in Anhang A.2 dargestellt. Die aufgeführten Untersuchungsergebnisse zeigen, dass beide Funktionen im nicht-gesättigten Fall vergleichbare Ergebnisse liefern. Deutlich sind die Unterschiede im Fall

---

<sup>1</sup> nicht Bestandteil dieser Arbeit. Bereitgestellt durch: D. Proft

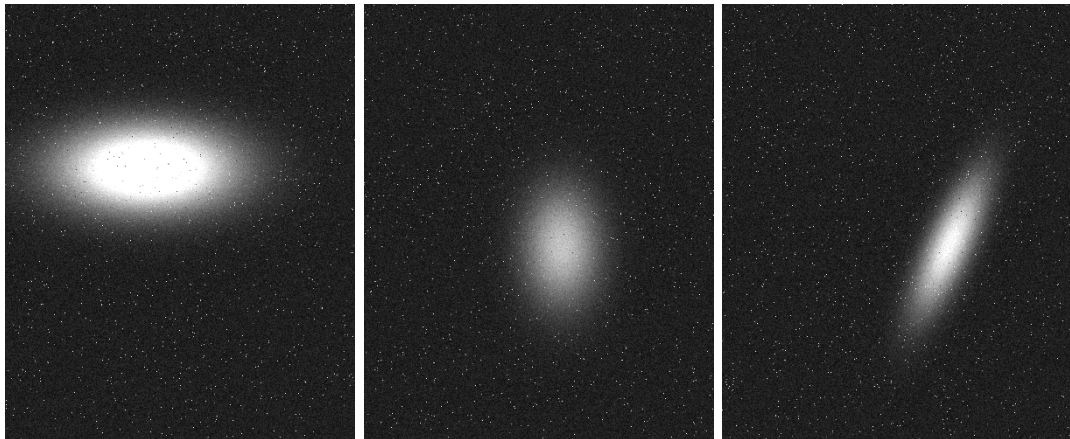


Abbildung 5.1: Eine Auswahl von Bildern, die vom Generator erzeugt wurden. Sie unterscheiden sich in Intensität, Breiten, Mittelpunkt und Rotationswinkel.

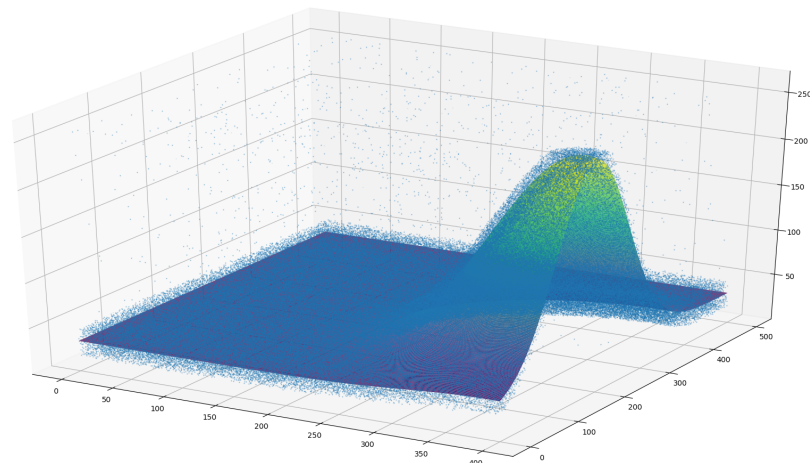


Abbildung 5.2: Dreidimensionale Darstellung eines durch den Bildgenerator erstellten Bildes. Zu erkennen sind dabei die Datenpunkte (blau) und die berechnete Anpassung (als Fläche). Es sind die deutlichen Ausreißer zu erkennen, die zerstörte Pixel der Kamera simulieren sollen.

eines gesättigten Bildes. Die Ergebnisse der Anpassung einer Gauß-Funktion folgen hier nicht mehr erkennbar einer Wahrscheinlichkeitsverteilung und weichen im Mittel deutlich von den simulierten Daten ab, dahingegen bleiben die Ergebnisse der Gauß-Funktion mit *Flattop* weiter poissonverteilt. Die Abweichung der einzelnen Parameter nimmt auch nur minimal zu. Der klare Unterschied in der Anpassungsgüte der beiden Funktionen bei Verwendung eines übersättigten Bildes zeigt sich exemplarisch bei Betrachtung der Anpassungsergebnisse für die Größe  $\sigma_{vert}$ . Die Ergebnisse der beiden Anpassungen für diese Größe im übersättigten Fall ist in Abb. 5.3 dargestellt.

Auffällig ist die Abweichung der Intensität  $I_0$  im nicht-gesättigten Fall von ca.  $-0,81 \pm 0,42\%$ . Als Ursache für diese systematische Abweichung sind die sog. „kaputten Pixel“ ermittelt worden. Liegen diese im Bereich des Strahlprofils und sind die Pixelwerte dieser Pixel im

	Gauß-Funktion		Gauß-Funktion mit <i>Flatlop</i>	
	nicht gesättigt	gesättigt	nicht gesättigt	gesättigt
$\Delta I_0/I_0/\%$	$-0,81 \pm 0,42$	-13,28	$-0,81 \pm 0,42$	$-1,02 \pm 0,37$
$\Delta \sigma_{vert}/\sigma_{vert}/\%$	$0,05 \pm 0,48$	5,88	$0,03 \pm 0,41$	$0,12 \pm 0,79$
$\Delta \sigma_{hor}/\sigma_{hor}/\%$	$0,05 \pm 0,47$	7,09	$0,03 \pm 0,40$	$0,12 \pm 0,34$

Tabelle 5.1: Relative Abweichung der Anpassungsergebnisse von den simulierten Werten

Mittel kleiner als das berechnete Gauß-Profil an dieser Stelle, hat dies die negative Abweichung zur Folge. Dies ist auch in den folgenden Untersuchungen zu beobachten.

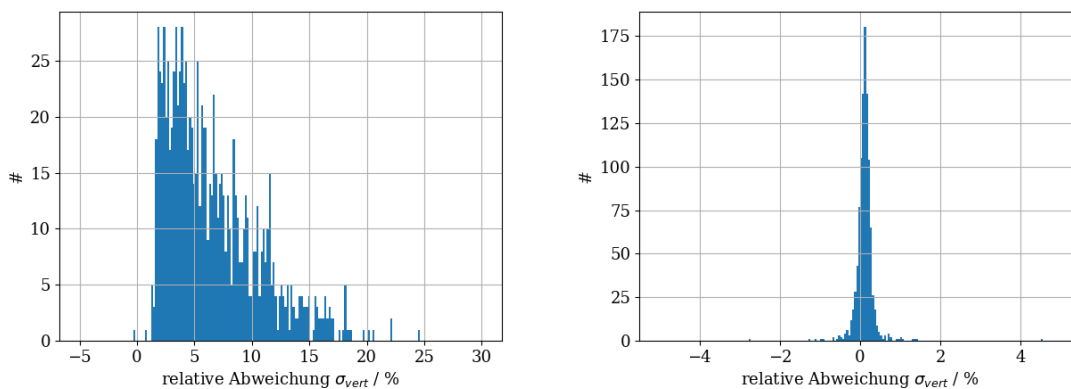


Abbildung 5.3: Abweichung der bestimmten vertikalen Strahlfleckbreite  $\sigma_{vert}$  von der Simulierten bei Verwendung der zweidimensionalen Gauß-Funktion (links) und der zweidimensionalen Gauß-Funktion mit *Flatlop* (rechts) - jeweils im gesättigten Fall.

### 5.1.2 Vergleich von Gaußfunktion und Gaußfunktion mit Flatlop sowie Gaußfunktion mit Flatlop und Rotationswinkel

Eine weitere Untersuchung beschäftigt sich mit der Auswertungsgüte bei rotierten Strahlprofilen. Diese treten am Beschleuniger aus verschiedenen Gründen auf (z.B. gedreht eingebaute Kameras, gedreht verbaute Quadrupole oder parasitäre, longitudinale Magnetfelder). Hierbei soll die Anpassung einer Gaußfunktion mit *Flatlop* und Rotationswinkel mit den bereits vorher untersuchten Anpassungen verglichen werden. Abb. 5.4 zeigt die Verteilung der Rotationswinkel der Strahlprofile in dieser Untersuchung. Es wurden 3000 Bilder mit jeder der Anpassungen ausgewertet.

Die Ergebnisse dieser Untersuchung sind in Tab. 5.2 aufgeführt, eine grafische Aufbereitung findet sich in Anhang A.2. Da die Anpassung mit Rotationswinkel spürbar langsamer ist als die beiden anderen Anpassungen, wurde das Verhalten der Anpassungen in Abhängigkeit des Rotationswinkels ausgewertet. Die Ergebnisse dieser Auswertung sind in Abb. 5.5 dargestellt. Aus dieser Untersuchung folgt, dass die beiden nicht-rotierten Anpassungen bereits bei kleinen Abweichungen von horizontalen und vertikalen Hauptachsen - bei Rotationswinkeln von ca.  $10^\circ$  bzw.  $80^\circ$  - stärker streuende Ergebnisse liefern als die rotierte Anpassung. Liegt der Rotationswinkel im Bereich  $\alpha \in [20^\circ, 70^\circ]$ , streuen die Ergebnisse der nicht rotierten

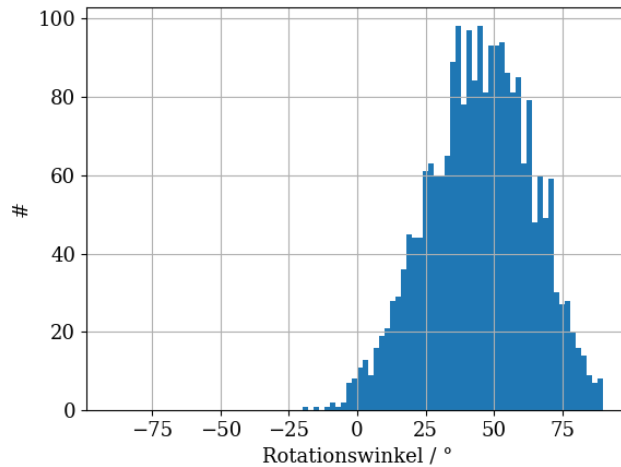


Abbildung 5.4: Die Verteilung der Rotationswinkel der in der Untersuchung der Gaußfunktion mit *Flat*top und Rotationswinkel berücksichtigten Strahlflecken.

	n. ges.	ges.	$\Delta I_0/I_0/\%$	$\Delta\sigma_{vert}/\sigma_{vert}/\%$	$\Delta\sigma_{hor}/\sigma_{hor}/\%$
G.-fkt.	x		-9,66	15,76	-8,11
G.-fkt.		x	-18,17	18,84	-3,12
G.-fkt. + <i>FT</i>	x		-9,56	15,70	-8,16
G.-fkt. + <i>FT</i>		x	-13,76	15,78	-5,70
G.-fkt. + <i>FT</i> & Rtwkl.	x		$-0,76 \pm 0,98$	$0,02 \pm 0,36$	$0,02 \pm 0,40$
G.-fkt. + <i>FT</i> & Rtwkl.		x	$-0,97 \pm 0,46$	$0,10 \pm 0,56$	$0,10 \pm 0,40$

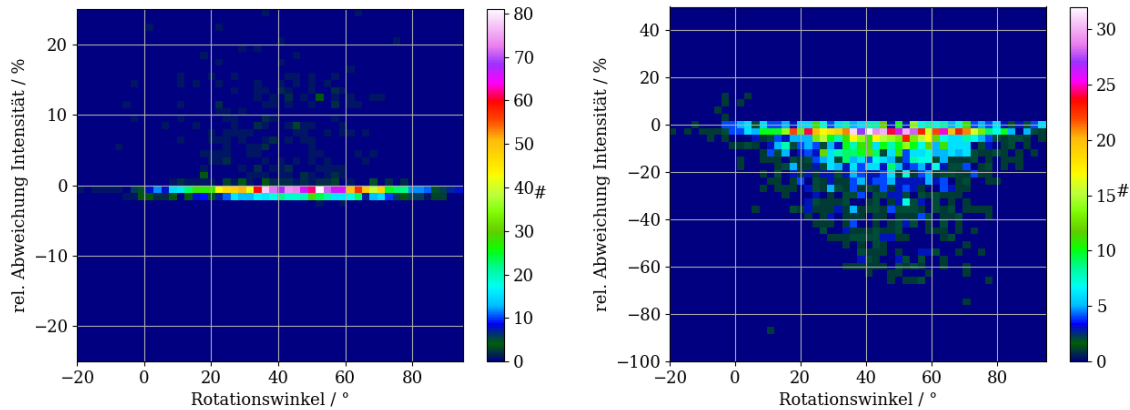
Tabelle 5.2: Relative Abweichung der angepassten Ergebnisse von den simulierten Werten bei Nutzung von rotierten Strahlprofilen.

Anpassungen so stark, dass eine Anpassung keine verlässlichen Ergebnisse mehr liefert. In diesem Bereich sollte entweder auf die langsamere, rotierte Anpassung zurückgegriffen werden oder die Rotation vor der Anpassung kompensiert werden.

### 5.1.3 Untersuchung des Einfluss der Größe *xtol* des Anpassungsalgorithmus auf Ausführungsdauer und Anpassungsgüte

Der Anpassungsalgorithmus der *GSL* verfügt über mehrere Abbruchbedingungen, mit denen festgelegt wird, wann die Anpassung als konvergiert angesehen und damit beendet wird. Relevant sind dabei 3 Abbruchbedingungen: Die Größe *xtol* gibt die Abbruchbedingung bezüglich der relativen Änderung der Parameter zwischen zwei Iterationen, *gtol* ist die Abbruchbedingung bezüglich des relativen Gradienten der Residuensumme und *ftol* ist die Abbruchbedingung bezüglich der relativen Änderung von  $\chi^2$  zwischen zwei Iterationen. Die Wahl der Abbruchbedingungen kann die Anpassungsgüte verringern, dies ist der Fall, wenn die Abbruchbedingungen sehr groß gewählt werden und der Algorithmus zu früh abbricht. Außerdem kann die Ausführungszeit beeinflusst werden, da kleinere Werte für die




 (a) Gaußkt. mit *Flattop* und Rotationswinkel

(b) Gaußfunktion

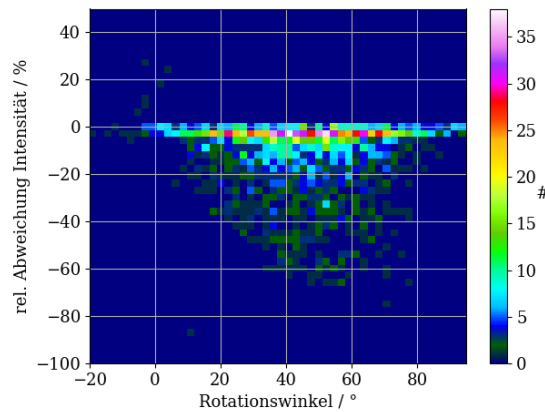

 (c) Gaußkt. mit *Flattop*

Abbildung 5.5: Relative Abweichung der Intensität von den simulierten Werten bei Nutzung der verschiedenen Anpassungen in Abhängigkeit des Rotationswinkels der simulierten Strahlprofile.

Abbruchbedingungen erst nach mehr Iterationen unterschritten werden.

Für die spätere Nutzung ist es sinnvoll die Laufzeit zu minimieren ohne die Anpassungsgüte zu beeinflussen. Hierbei findet sich jedoch keine genaue Angabe in der Dokumentation des Algorithmus, wie die Abbruchbedingungen gewählt werden sollen. Für  $xtol$  wird aufgeführt, dass sich der Wert in der Größenordnung der gewünschten Genauigkeit bewegen soll (vgl. [11]). Daher wurde entschieden einen optimalen Wert selber zu bestimmen. Hierfür wurden Anpassungen an 3000 Bilder durchgeführt. An jedes Bild wurden dabei mehrere Anpassungen durchgeführt, die sich nur in der Wahl der Größenordnung von  $xtol$  unterscheiden.

Die Ergebnisse der relativen Abweichungen von Intensität  $I_0$  und Breiten  $\sigma_{hor}, \sigma_{vert}$  sind in Abhängigkeit des gewählten Werts der Abbruchbedingung in Abb. 5.6 dargestellt. Die Anzahl der benötigten Iterationen in Abhängigkeit des Werts von  $xtol$  wurde ebenfalls untersucht und ist in Abb. 5.7 aufgeführt. Aus den Ergebnissen kann abgelesen werden, dass die Wahl der Abbruchbedingung  $xtol$  im Bereich  $xtol \leq 10^{-3}$  keinen signifikanten Einfluss auf die

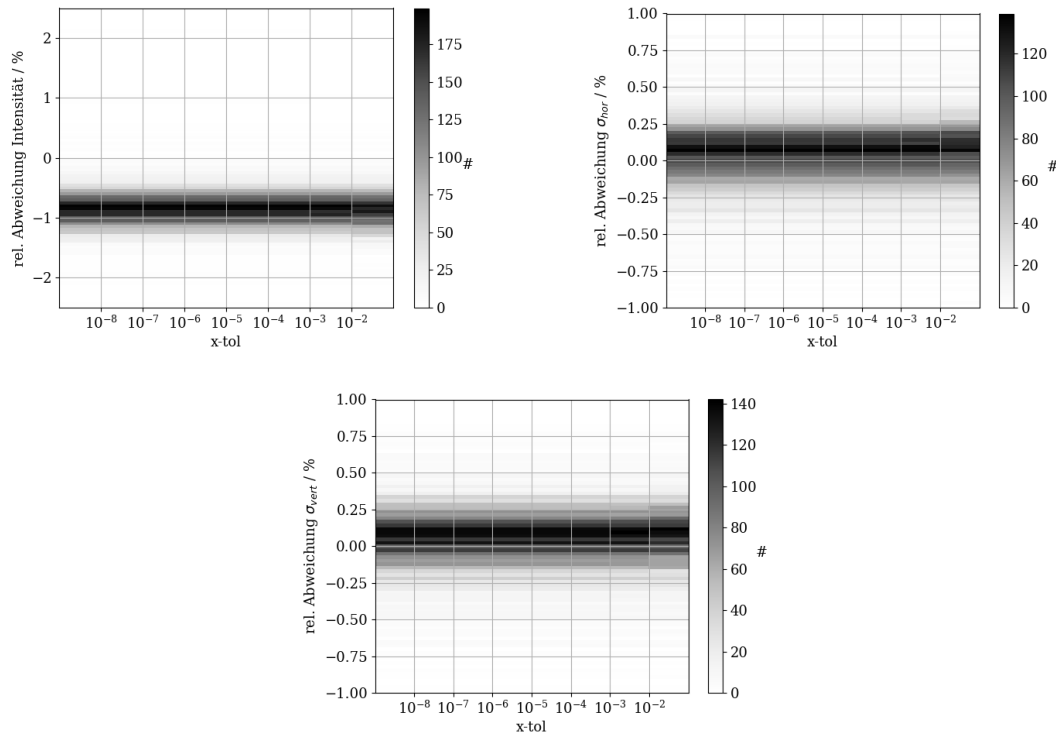


Abbildung 5.6: Abweichung der bestimmten Anpassungsparameter in Abhängigkeit des verwendeten Werts von „xtol“.

Anpassungsgüte hat, jedoch die Anzahl benötigter Iterationen deutlich beeinflussen kann.

Anhand dieser Ergebnisse wurde für  $xtol$  der Wert  $10^{-3}$  gewählt. Dieser wird als der gewünschte Kompromiss zwischen geringer Ausführungsdauer und hoher Anpassungsgüte angesehen. Für diesen Wert ergibt sich:

$$\Delta I_0 = (-0,86 \pm 0,24) \%, \quad \Delta \sigma_{hor} = (0,06 \pm 0,19) \%, \quad \Delta \sigma_{vert} = (0,07 \pm 0,19) \%.$$

## 5.2 Messung am Beschleuniger LAB

Aufgrund der laufenden Optimierungsmaßnahmen an ELSA konnte die Softwarebibliothek noch nicht an diesem Beschleuniger getestet werden. Tests der Bibliothek in der Bildauswertung fanden daher am Laborbeschleuniger (kurz: LAB) statt. Hierbei wurde die Auswertungssoftware sowohl live getestet als auch Aufnahmen gespeichert, die zur Verifikation späterer Versionen zur Verfügung stehen. Erfahrungen aus diesem Test sind, dass eine Berücksichtigung des Drehwinkels hier zwingend notwendig ist, da die Rotationswinkel der einzelnen Strahlprofile hier teilweise im Bereich von über  $20^\circ$  liegt sowie die Notwendigkeit einer automatisierten Bestimmung eines optimalen Schwellwerts für den binären Filter.

Es konnte festgestellt werden, dass der Algorithmus bei Nutzung eines Ausschnitts des Bildes, dass das Strahlprofil vollständig beinhaltet, eine Anpassungsfrequenz in der Größenordnung der Auslesefrequenz der Kamera liefert. Nach der im Abschnitt 5.1.3 beschriebenen Optimierung

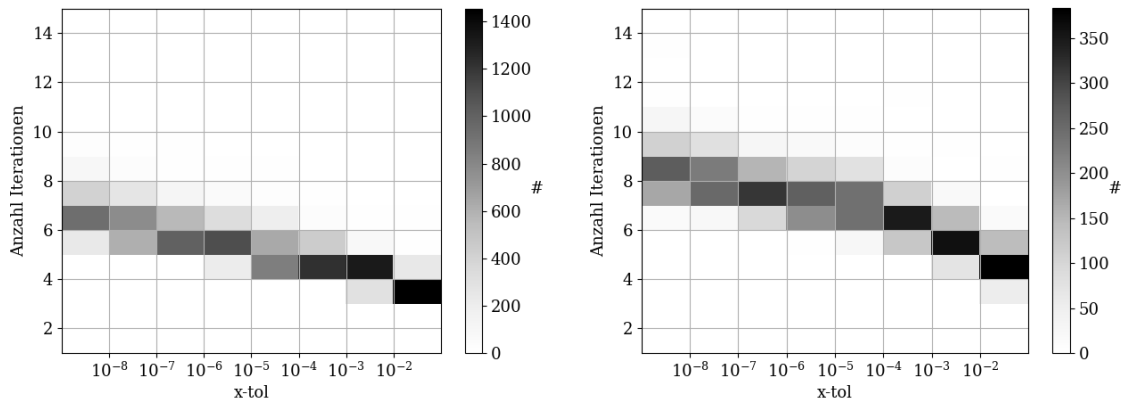


Abbildung 5.7: Anzahl der nötigen Iterationen in Abhängigkeit der Toleranz „x-tol“ für den nicht-gesättigten Fall (links) und den gesättigten Fall (rechts).

wurde die Anpassungsfrequenz nochmal erhöht.

Es wurde festgestellt, dass es ausreicht, wenn nur ein Teil des Strahlprofils im untersuchten Ausschnitt liegt, um die Form des Strahlprofils zu beschreiben.



## Fazit

---

Im Rahmen dieser Bachelorarbeit wurde eine Softwarebibliothek zur automatisierten Analyse der Strahlschirmbilder an ELSA und am LAB entwickelt. Diese Entwicklung ist ein zentraler Teil im Gesamtprojekt der Umstrukturierung der Framegrabber-Architektur und der daran angeschlossenen Software.

Die Hauptziele der neuen Bibliothek bestehen darin, die Auswertung zu zentralisieren und unabhängig von den einzelnen *Framegrabbermodellen* zu machen. Ferner sollten echte zweidimensionale Anpassungen implementiert und die allgemeine Anpassungsgüte verbessert werden. Hierbei wurde ein besonderer Schwerpunkt auf die Auswertungsgüte in bisher nur schlecht auswertbaren Fällen gelegt - insbesondere bei übersättigten oder rotierten Strahlprofilen. Durch die Entscheidung, die Umsetzung der Auswertungssoftware als C++-Bibliothek auszuführen, konnte die geforderte Plattformunabhängigkeit erreicht werden. Zudem war es möglich die Bibliothek in die *Framegrabber*-Software am LAB zu integrieren. Diese ist das erste aktualisierte und modernisierte Programm zur optischen Strahldiagnose und kann in Zukunft als Blaupause für die Entwicklung der zentralen *Framegrabber*-Software an ELSA dienen. Zudem machte es diese Implementierung möglich, die Auswertungssoftware auch an echten Daten zu testen und weitere Verbesserungen zu implementieren.

Die Umsetzung als Bibliothek erleichtert außerdem auch die zukünftige Erweiterung um weitere Anpassungsfunktionen.

Die neu eingeführten echten zweidimensionalen Anpassungen ermöglichen eine deutlich gesteigerte Anpassungsgüte. Bei der Nutzung der Software am LAB wurde unter anderem deutlich, dass eine Anpassung auch dann noch gut die Daten beschreibt, wenn ein Teil des Elektronenstrahls den Strahlschirm nicht erreicht. Dies war mit den vorher genutzten Projektionen nicht möglich. Die eingeführte Gaußfunktion mit sog. *Flattop* als weitere verfügbare Anpassungsfunktion erhöht die Anpassungsgüte bei übersättigten Bildern deutlich. Zudem ermöglicht es die Anpassungsfunktion mit Rotationswinkel auch rotierte Strahlprofile, die z.B. aufgrund einer rotiert eingebauten Kamera entstehen, mit einer hohen Genauigkeit zu vermessen.

In der Bibliothek ist außerdem auch das Detektieren und Anpassen von mehreren Strahlprofilen pro Strahlschirm vorgesehen und realisiert, was eine erweiterte Nutzbarkeit der Software auch an anderen optischen Aufbauten ermöglicht. Neben der Nutzung im Betrieb der Beschleuniger als Teil des Kontrollsystems, kann die Software auch zu anderen Zwecken

eingesetzt werden. So wurde sie z.B. bereits genutzt, um Abbildungseigenschaften von neu entwickelten optischen Aufbauten zu untersuchen [12].

Zudem wurde die Anpassungsgüte statistisch evaluiert, um eine Entscheidungsgrundlage für die spätere Wahl der optimalen Anpassungsfunktion zu bieten. Diese Evaluierung legt nahe, die Gaußfunktion mit *Flattop* als Standardanpassungsfunktion zu wählen und dass bei Rotationswinkel im Bereich  $20^\circ \leq \alpha \leq 70^\circ$  die Anpassungsfunktion mit rotiertem Profil genutzt werden sollte.

Die vorgestellte Bibliothek erfüllt in ihrem aktuellen Entwicklungsstand alle an sie gestellten Anforderungen und liefert einige darüber hinausgehende Möglichkeiten (z.B. Detektion und Anpassung mehrerer Strahlprofile). Sie bietet die geforderte Plattformunabhängigkeit und bietet aufgrund ihrer Struktur auch in der Zukunft eine leichte Erweiterbarkeit. Mögliche zukünftige Anforderungen bedenkend ist die Bibliothek im Hinblick auf eine vollkommen automatisierte Nutzung entwickelt worden und könnte so z.B. auch Daten für die Bestimmung der Strahleigenschaften liefern, ohne dass die einzelnen Aufnahmen der Kamera ständig beobachtet und eine sog. *Area-of-Interest* von einem Benutzer gesetzt und, wenn nötig, aktualisiert werden muss.

# Literatur

---

- [1] S. Khan, *Einführung in die Physik der Teilchenbeschleuniger und Synchrotronstrahlungsquellen*, TU Dortmund, 2011, URL: [http://www.delta.tu-dortmund.de/cms/Medienpool/Khan\\_Bilder/Beschleuniger-Skript-2010.pdf](http://www.delta.tu-dortmund.de/cms/Medienpool/Khan_Bilder/Beschleuniger-Skript-2010.pdf) (besucht am 13.08.2020) (siehe S. 3).
- [2] S. Patzelt, *Eine neue externe Strahlführung für Detektortests an ELSA*, Diplom: Physikalisches Institut der Universität Bonn, Kap. Theoretische Grundlagen, URL: [https://www-elsa.physik.uni-bonn.de/Publicationen/texte/patzelt\\_diplom.pdf](https://www-elsa.physik.uni-bonn.de/Publicationen/texte/patzelt_diplom.pdf) (besucht am 13.08.2020) (siehe S. 3).
- [3] K. Wille, *Physik der Teilchenbeschleuniger und Synchrotronstrahlungsquellen, Eine Einführung*, 1. Aufl., B.G.Teubner, 1992, ISBN: 978-3-519-03087-4 (siehe S. 4).
- [4] H. Wiedemann, *Particle Accelerator Physics*, Graduate Texts in Physics, Springer International Publishing, 2015, ISBN: 9783319183176 (siehe S. 4).
- [5] U. Berges, *Hochauflösende optische Strahldiagnose mittels Synchrotronstrahlung am Beispiel der Synchrotronstrahlungsquelle DELTA*, Dissertation: Institut für Beschleunigerphysik und Synchrotronstrahlung Universität Dortmund, 2000, Kap. Grundlage des optischen Synchrotronlichtmonitors, URL: <https://eldorado.tu-dortmund.de/handle/2003/2318?locale=de> (besucht am 06.09.2020) (siehe S. 5).
- [6] F.-M. Dittes, *Optimierung, Wie man aus allem das Beste macht*, 1. Aufl., Springer Vieweg, 2015, Kap. Schritt für Schritt: deterministische Lösungsverfahren, ISBN: 978-3-642-53888-9 (siehe S. 6).
- [7] J. Nocedal und S. J. Wright, *Numerical Optimization*, second, Springer, 2006, Kap. Least-Squares Problems (siehe S. 6).
- [8] *Open Computer Vision*, 2020, URL: [www.opencv.org](http://www.opencv.org) (siehe S. 11).
- [9] *Image Thresholding, Otsu's Binarization*, 2020, URL: [docs.opencv.org/3.4/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html) (besucht am 08.09.2020) (siehe S. 12).
- [10] *GNU Scientific Library*, 2020, URL: [www.gnu.org/software/gsl](http://www.gnu.org/software/gsl) (siehe S. 14).

- [11] *GNU Scientific Library*,  
*Testing for Convergence*, Nonlinear Least-Squares Fitting, Englisch,  
URL: [www.gnu.org/software/gsl/doc/html/nls.html#testing-for-convergence](http://www.gnu.org/software/gsl/doc/html/nls.html#testing-for-convergence)  
(besucht am 07.09.2020) (siehe S. 21).
- [12] K. Kranz, *Optimierung der Sychrotronlicht Strahlmonitore in der externen Strahlführung zu den Experimentierbereichen E1/E2 an ELSA*,  
Bachelorarbeit: Physikalisches Institut Universität Bonn, 2020, in Vorbereitung  
(siehe S. 26).



---

## Anhang

---

### A.1 Minimalbeispiel zur Nutzung der Bibliothek

```
#include "libfit.h"

int main(int argc, char** argv){
    /*Speicherung der Pixelmatrix in diesem Objekt*/
    cv::Mat image;
    /*Speicherung der Anpassungsergebnisse*/
    ELSA::FrameGrab::GaussFitResults fitResults;

    double threshold = 50;
    auto fitMats = peak_detection(image, threshold, true, false,
        true);
    for (auto &fitMat : fitMats) {
        if(fitMat.FitClearance() == true){
            ELSA::FrameGrab::BaseVectors BV(image, fitMat, false);
            ELSA::FrameGrab::fit_to_data(fitResults, ELSA::FrameGrab
                ::fit_function::gauss_2d_flattop, fitMat, BV);
        }
    }
    return 0;
}
```

## A.2 Statistische Untersuchung mit simulierten Daten

### Vergleich von Gauß-Fkt. und Gauß-Fkt. mit Flattop

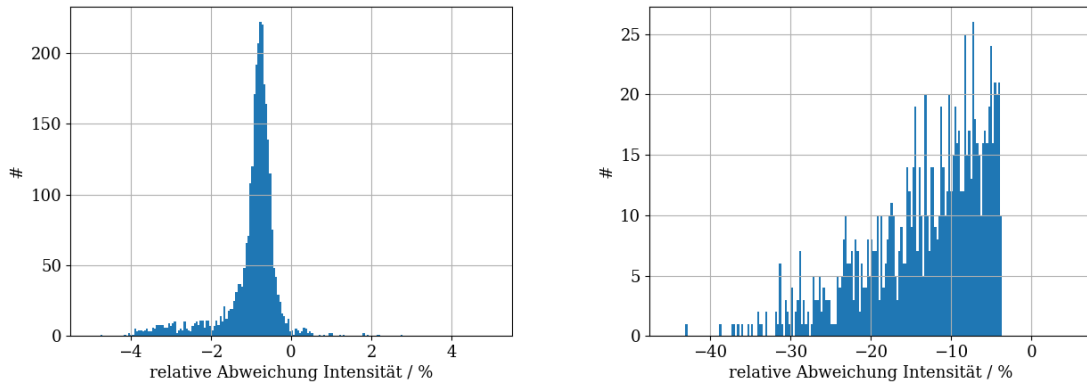


Abbildung A.1: Abweichung der bestimmten Intensität  $I_0$  von der simulierten Intensität bei Verwendung der zweidimensionalen Gauß-Funktion. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

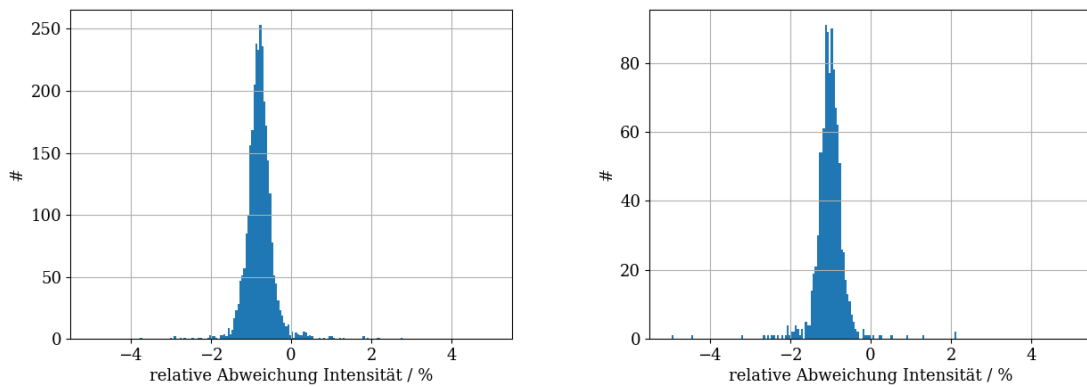


Abbildung A.2: Abweichung der bestimmten Intensität  $I_0$  von der simulierten Intensität bei Verwendung der zweidimensionalen Gauß-Funktion mit *Flattop*. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

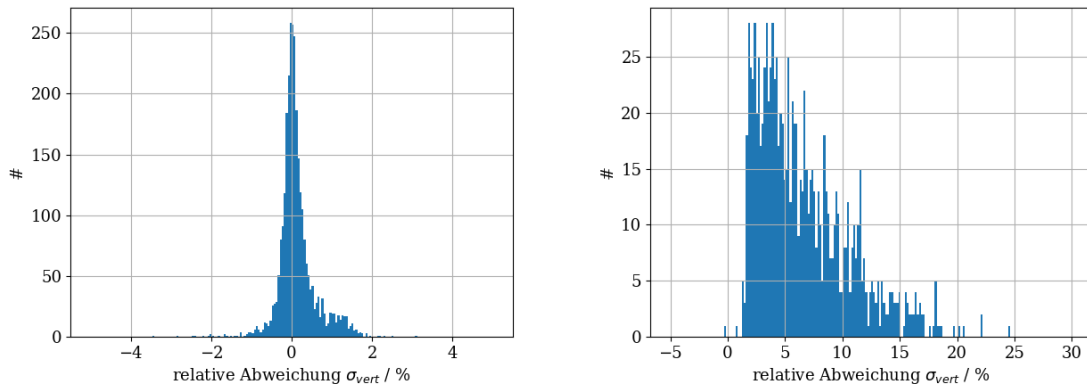


Abbildung A.3: Abweichung der bestimmten vertikalen Strahlfleckbreite  $\sigma_{vert}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

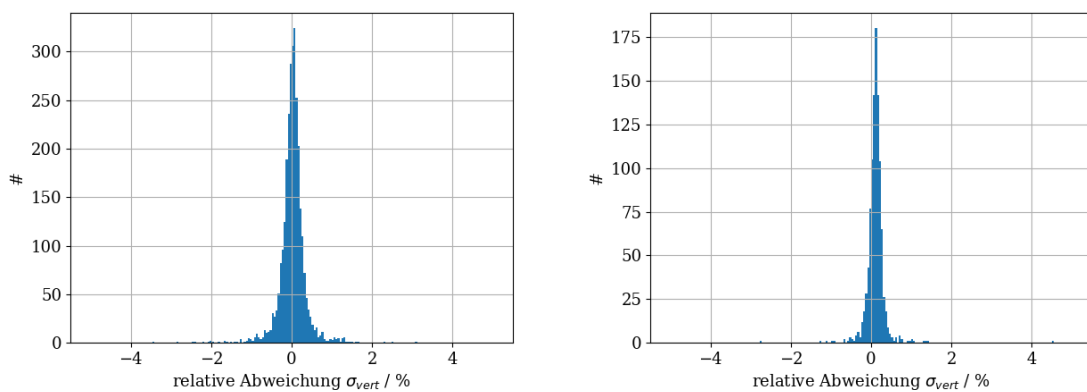


Abbildung A.4: Abweichung der bestimmten vertikalen Strahlfleckbreite  $\sigma_{vert}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion mit *Flattop*. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

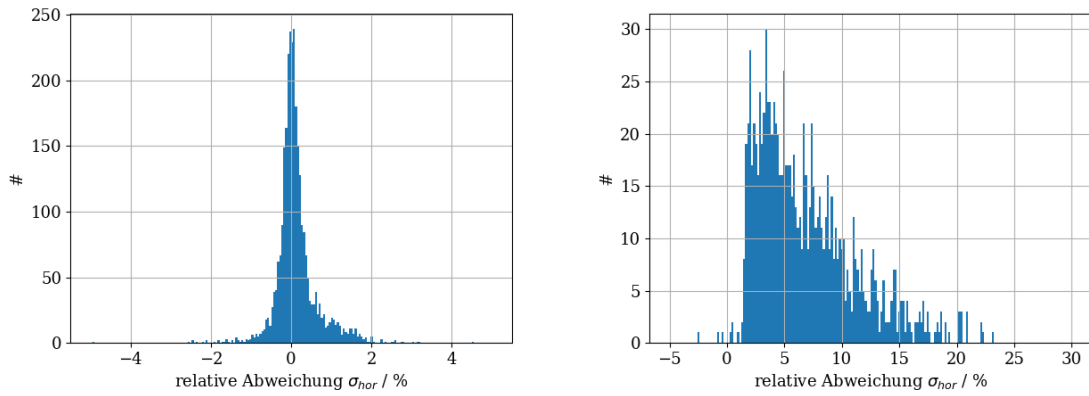


Abbildung A.5: Abweichung der bestimmten horizontalen Strahlfleckbreite  $\sigma_{hor}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

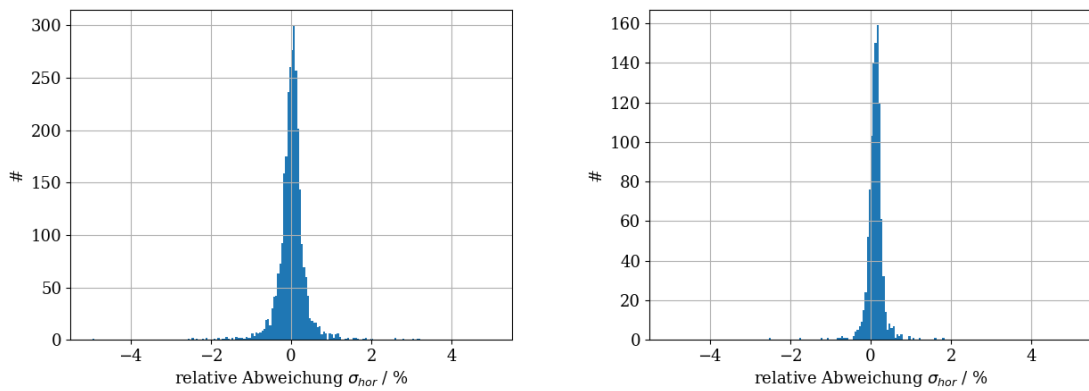


Abbildung A.6: Abweichung der bestimmten horizontalen Strahlfleckbreite  $\sigma_{hor}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion mit *Flattop*. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

### Vergleich von Gauß-Fkt. und Gauß-Fkt. mit Flattop sowie Gauß-Fkt. mit Flattop und Rotationswinkel

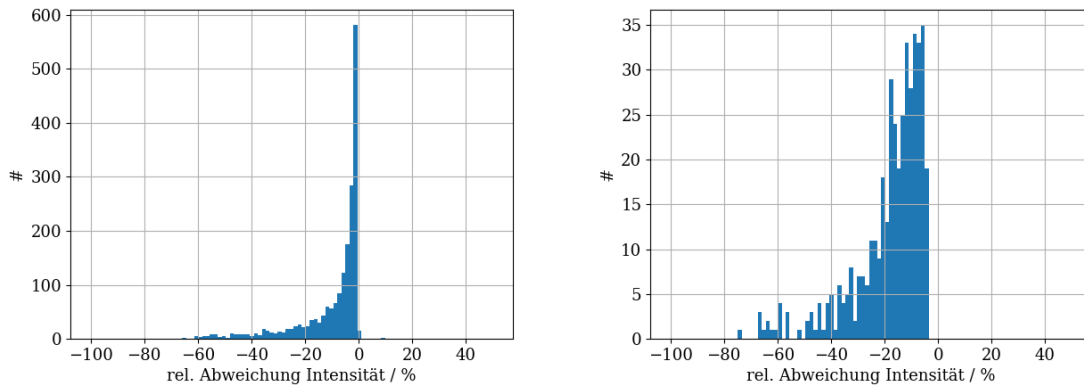


Abbildung A.7: Abweichung der bestimmten Intensität  $I_0$  von der simulierten Intensität bei Verwendung der zweidimensionalen Gauß-Funktion. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

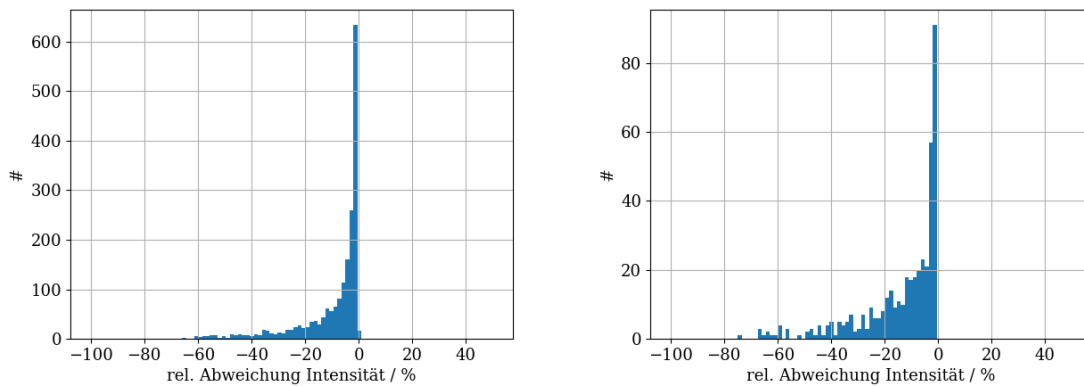


Abbildung A.8: Abweichung der bestimmten Intensität  $I_0$  von der simulierten Intensität bei Verwendung der zweidimensionalen Gauß-Funktion mit *Flattop*. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

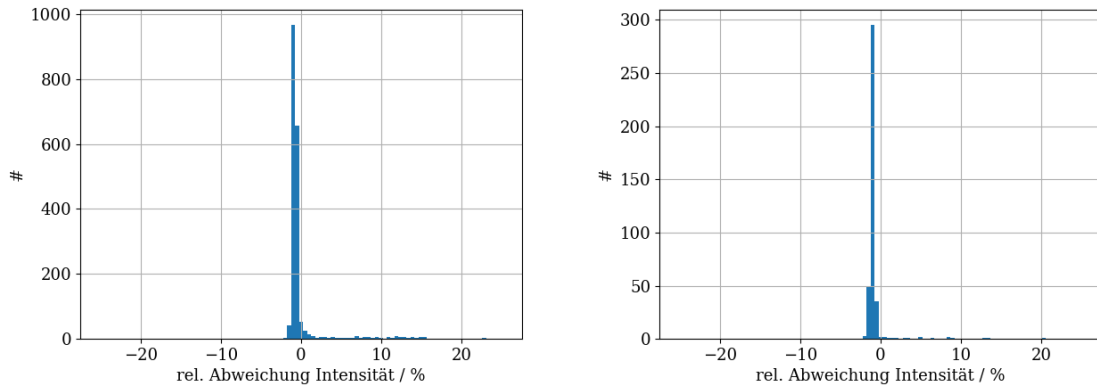


Abbildung A.9: Abweichung der bestimmten Intensität  $I_0$  von der simulierten Intensität bei Verwendung der zweidimensionalen Gauß-Funktion mit *Flattop* und Rotationswinkel. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

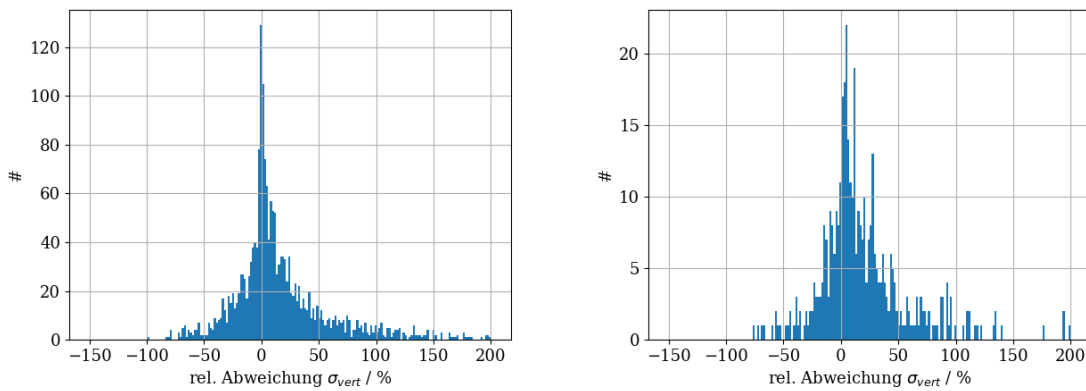


Abbildung A.10: Abweichung der bestimmten vertikalen Strahlfleckbreite  $\sigma_{vert}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

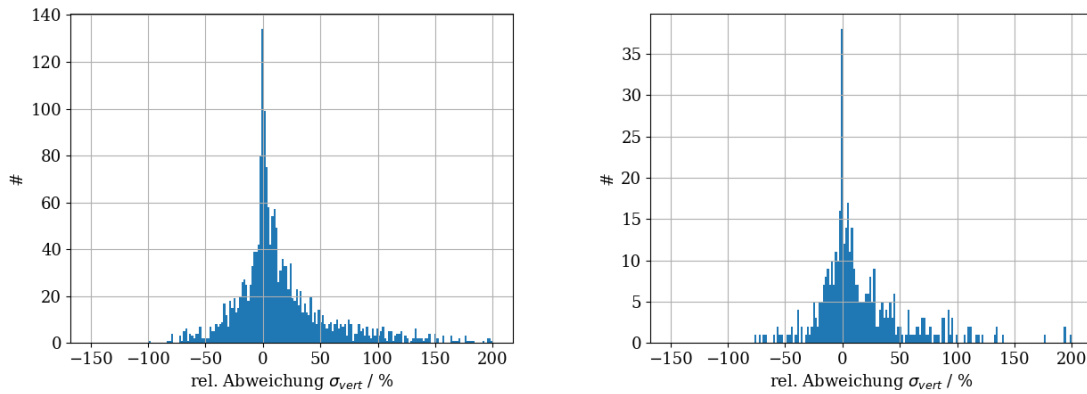


Abbildung A.11: Abweichung der bestimmten vertikalen Strahlfleckbreite  $\sigma_{vert}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion mit *Flattop*. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

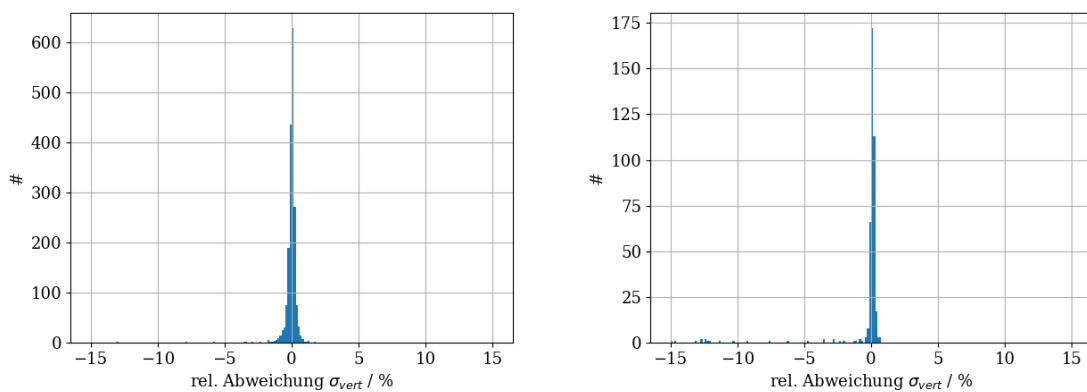


Abbildung A.12: Abweichung der bestimmten vertikalen Strahlfleckbreite  $\sigma_{vert}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion mit *Flattop* und Rotationswinkel. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

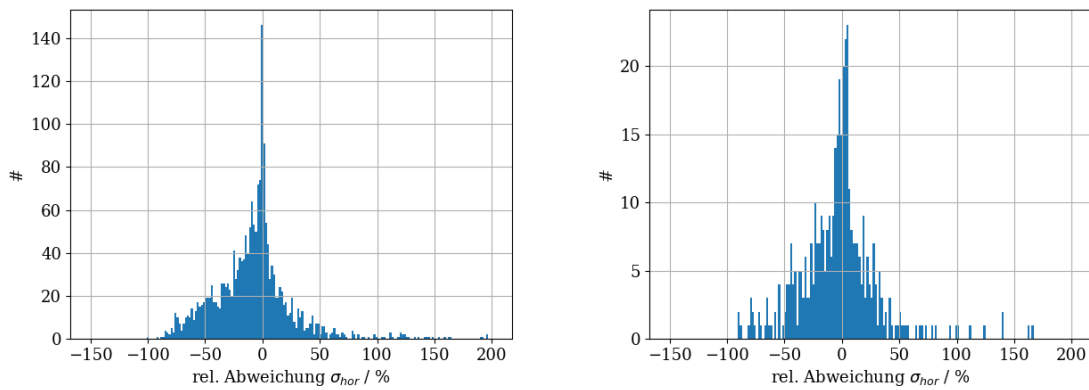


Abbildung A.13: Abweichung der bestimmten horizontalen Strahlfleckbreite  $\sigma_{hor}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.

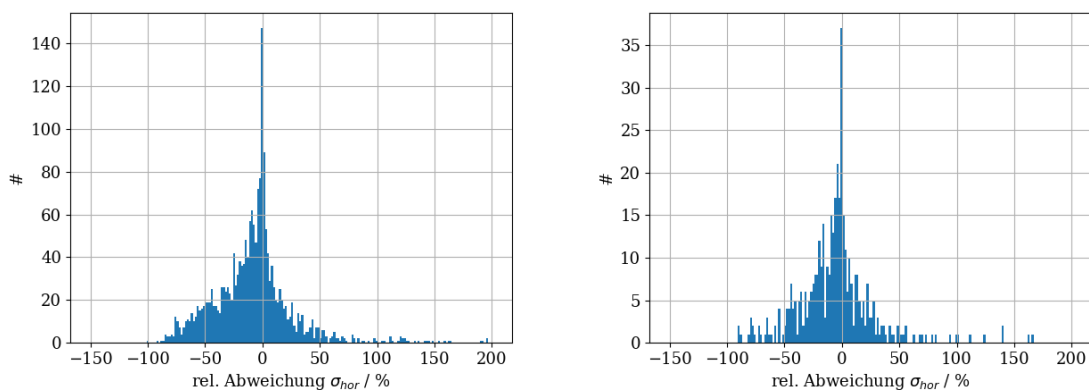


Abbildung A.14: Abweichung der bestimmten horizontalen Strahlfleckbreite  $\sigma_{hor}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion mit *Flattop*. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.



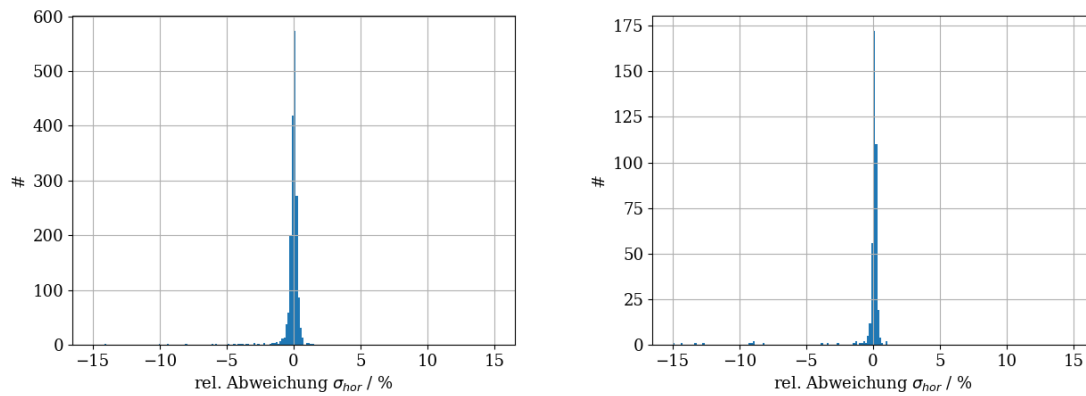


Abbildung A.15: Abweichung der bestimmten horizontalen Strahlfleckbreite  $\sigma_{hor}$  von der Simulation bei Verwendung der zweidimensionalen Gauß-Funktion mit *Flat* und Rotationswinkel. Links im nicht-gesättigten Fall und rechts im gesättigten Fall.



# Abbildungsverzeichnis

---

2.1	Koordinatensystem mit der verwendeten Nomenklatur zur Beschreibung von transversalen Teilchenbewegungen. . . . .	3
2.2	Phasenraumellipse eines einzelnen Teilchens und eines Ensembles von Teilchen	4
2.3	Admittanz eines Beschleunigers . . . . .	5
3.1	Bisherige Architektur der Framegrabber und der angeschlossenen Auswertungssoftware. . . . .	8
3.2	Geplante Architektur der Framegrabber und der angeschlossenen Auswertungssoftware. . . . .	8
4.1	„Zerschneiden“ der Pixelmatrix zur Separierung der Strahlprofile . . . . .	12
4.2	Entstehung „Gauß-Stumpf“ in einem Graustufenbild . . . . .	15
5.1	Beispielsausgabe des Schirmbildsgenerator . . . . .	18
5.2	Dreidimensionale Darstellung eines Generatorbildes . . . . .	18
5.3	Vergleich der Abweichung der vertikalen Strahlfleckbreite $\sigma_{vert}$ im gesättigten Fall für die beiden nicht-rotierten Anpassungen . . . . .	19
5.4	Rotationswinkel der Strahlprofile bei der Untersuchung der Gaußfunktion mit <i>Flattop</i> und Rotationswinkel . . . . .	20
5.5	Relative Abweichung der Intensität von den simulierten Werten bei Nutzung der verschiedenen Anpassungen in Abhängigkeit des Rotationswinkels der simulierten Strahlprofile. . . . .	21
5.6	Relative Abweichung der Anpassungsparameter in Abhängigkeit von „xtol“ .	22
5.7	Anzahl der nötigen Iterationen in Abhängigkeit der Toleranz „xtol“ für den nicht-gesättigten Fall (links) und den gesättigten Fall (rechts). . . . .	23
A.1	Abweichung der bestimmten zur simulierten Intensität $I_0$ für die Gaußfunktion	30
A.2	Abweichung der bestimmten zur simulierten Intensität $I_0$ für die Gaußfunktion mit <i>Flattop</i> . . . . .	30
A.3	Abweichung der bestimmten zur simulierten vertikalen Strahlfleckbreite $\sigma_{vert}$ für die Gaußfunktion . . . . .	31
A.4	Abweichung der bestimmten zur simulierten vertikalen Strahlfleckbreite $\sigma_{vert}$ für die Gaußfunktion mit <i>Flattop</i> . . . . .	31
A.5	Abweichung der bestimmten zur simulierten horizontalen Strahlfleckbreite $\sigma_{hor}$ für die Gaußfunktion . . . . .	32
A.6	Abweichung der bestimmten zur simulierten horizontalen Strahlfleckbreite $\sigma_{hor}$ für die Gaußfunktion mit <i>Flattop</i> . . . . .	32

A.7	Abweichung der bestimmten zur simulierten Intensität $I_0$ für die Gaußfunktion	33
A.8	Abweichung der bestimmten zur simulierten Intensität $I_0$ für die Gaußfunktion mit <i>Flattop</i>	33
A.9	Abweichung der bestimmten zur simulierten Intensität $I_0$ für die Gaußfunktion mit <i>Flattop</i> und Rotationswinkel	34
A.10	Abweichung der bestimmten zur simulierten vertikalen Strahlfleckbreite $\sigma_{vert}$ für die Gaußfunktion	34
A.11	Abweichung der bestimmten zur simulierten vertikalen Strahlfleckbreite $\sigma_{vert}$ für die Gaußfunktion mit <i>Flattop</i>	35
A.12	Abweichung der bestimmten zur simulierten vertikalen Strahlfleckbreite $\sigma_{vert}$ für die Gaußfunktion mit <i>Flattop</i> und Rotationswinkel	35
A.13	Abweichung der bestimmten zur simulierten horizontalen Strahlfleckbreite $\sigma_{hor}$ für die Gaußfunktion	36
A.14	Abweichung der bestimmten zur simulierten horizontalen Strahlfleckbreite $\sigma_{hor}$ für die Gaußfunktion mit <i>Flattop</i>	36
A.15	Abweichung der bestimmten zur simulierten horizontalen Strahlfleckbreite $\sigma_{hor}$ für die Gaußfunktion mit <i>Flattop</i> und Rotationswinkel	37

# Danksagung

---

Ich möchte Herrn Prof. Dr. Klaus Desch dafür danken, dass ich diese Bachelorarbeit übernehmen konnte und dass ich in die Arbeitsgruppe ELSA aufgenommen wurde. Ich habe hier nicht nur meine Kenntnisse und Fähigkeiten im Bereich der Programmierung erweitert, sondern auch einen ersten, kleinen Eindruck von der nötigen Arbeit für einen erfolgreichen Beschleunigerbetrieb erhalten können, den ich hoffentlich noch vertiefen kann. Ich möchte mich bei Herrn Dr. Daniel Elsner bedanken, der bereitwillig das Koreferat übernommen hat und der die Erprobung der Software am LAB ermöglicht hat.

Ganz besonders möchte ich Dennis Proft danken, der mir in den vergangenen vier Monaten nicht nur jederzeit mit Rat und Tat zur Seite stand, sondern auch keine Gelegenheit ausließ meine Kenntnisse in der Programmierung, sei es in C++ oder seien es generelle Konzepte, zu erweitern und zu vertiefen. In zeitaufwendiger mühsamer Arbeit haben wir jeden gut versteckten Fehler finden können.

Einen besonderen Dank auch an Michael Switka für das Korrekturlesen dieser Arbeit.

Bei allen Mitarbeitern der Arbeitsgruppe ELSA möchte ich mich für alle gegebenen Anregungen und Denkanstöße bedanken, besonders bei Frank Frommberger für die Einführung in die Thematik der *Framegrabber* und für die Hilfe bei jeglichen Rückfragen.

Abschließend möchte ich mich bei meinen Kommilitonen in der Arbeitsgruppe - besonders bei Kilian Kranz, mit dem ich in dieser Zeit das Büro teilte - für die gute Atmosphäre und die interessanten Gespräche bedanken.

Auch allen Anderen, die einen Anteil an der erfolgreichen Durchführung dieser Arbeit auch in dieser besonderen Zeit haben, möchte ich herzlich danken.