

**Fakultät für Physik und Astronomie  
Ruprecht-Karls-Universität Heidelberg**

Bachelorarbeit in Physik  
vorgelegt von Roman Martel  
geboren in Tokmak (Kirgistan)

**August 2012**



# Werkzeuge zur Untersuchung des Hintergrundes von fluoreszierenden Kernspurdetektoren

Diese Bachelorarbeit wurde von Roman Martel ausgeführt am  
Deutschen Krebsforschungszentrum in Heidelberg  
unter der Betreuung von  
Prof. Dr. Oliver Jäkel  
und  
Prof. Dr. Peter Bachert



## Kurzfassung

Gegenüber der konventionellen Strahlentherapie besitzt die Strahlentherapie mit Schwerionen eine Reihe von Vorteilen, die in starken Schwankungen der Energiedeposition in kleinskaligen Bereichen begründet sind. Der auf Aluminiumoxidkristallen basierende fluoreszierende Kernspurdetektor, FNTD, hat sich als vielversprechend herausgestellt, um Untersuchungen in diesen Skalen ( $\mu\text{m}$ ) zu führen. Jedoch zeigt dieser auch in unbestrahlten Bereichen eine signifikante Fluoreszenz („Hintergrund“). Untersuchungen zur Größe und Variation dieses Hintergrundes waren bisher durch die umständliche Anwendung der zu diesem Zweck einsetzbaren Routinen beschränkt. Deshalb wurden in dieser Bachelorarbeit Softwarewerkzeuge zur automatisierten Untersuchung hierfür bereitgestellt, überprüft und an einem Datensatz angewendet. Die bereits implementierten Routinen wurden dabei für den automatisierten Gebrauch angepasst und in eine Skriptsprache übertragen.

Die Überprüfungen ergaben, dass die bereitgestellten Werkzeuge gleiche Ergebnisse liefern wie die zugrundeliegenden Routinen und damit korrekt angepasst und übertragen wurden.

Die Anwendung am Datensatz zeigte, dass das Hauptwerkzeug zur Bestimmung des Hintergrundes mit optimalen Einstellungen den Hintergrund von FNTDs zufriedenstellend bestimmen kann, die nicht oder mit leichten Teilchen bis einschließlich  $^{24}\text{Mg}$  bestrahlt wurden. Bei FNTDs, die mit schwereren Teilchen bestrahlt wurden, scheitert das Werkzeug jedoch aus Gründen, die zwar eingeschränkt aber nicht geklärt werden konnten.

## Abstract

Compared to the conventional radiotherapy the heavy ion radiotherapy offers a series of benefits, which are founded in strong fluctuations of the energy deposition in small-scaled regions. The fluorescent nuclear track detector, FNTD, based on aluminium oxide crystals turned out to be promising to make investigations in these scales ( $\mu\text{m}$ ). However this detector exhibits a significant fluorescence (“background”) also in unirradiated regions. Investigations of size and variation of this background were so far restricted by the laborious application of the routines applicable for this purpose. Therefore in this bachelor thesis tools for the automated investigation for this purpose were provided, tested and applied on a dataset. The already implemented routines were thereby adjusted to the automated usage and transferred to a scripting language.

The tests showed that the provided tools yield the same results as the underlying routines and were therefore correct adjusted and transferred.

The application on the dataset exhibited, that the main tool for the assignment of the background with optimal settings satisfactorily assigns the background of unirradiated and with light particles through  $^{24}\text{Mg}$  irradiated FNTDs. But for FNTDs irradiated with heavier particles the tool fails due to reasons, which could be limited but not resolved.



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
<b>2. Methoden und Materialien</b>	<b>3</b>
2.1. Fundamentale Größen . . . . .	3
2.1.1. Ionisierende Strahlung . . . . .	3
2.1.2. Dosis . . . . .	3
2.2. FNTD (fluorescent nuclear track detector) . . . . .	4
2.2.1. Einleitung . . . . .	4
2.2.2. Hintergrund . . . . .	5
2.3. Bestrahlungseinrichtungen . . . . .	6
2.3.1. Max-Planck-Institut für Kernphysik (MPIK) . . . . .	6
2.3.2. Heidelberger Ionenstrahl-Therapiezentrum (HIT) . . . . .	6
2.3.3. Center for Nuclear Technologies (DTU Nutech), Roskilde . . . . .	6
2.4. Zeiss LSM 710 . . . . .	6
2.4.1. Konfokale Mikroskopie . . . . .	6
2.4.2. Kontrollparameter . . . . .	7
2.4.3. LSM-Dateiformat . . . . .	8
2.5. Normierung der Countzahl N . . . . .	8
2.6. Spektroskopiedatensatz . . . . .	9
2.7. Software . . . . .	9
2.7.1. Java . . . . .	10
2.7.2. ImageJ . . . . .	10
2.7.3. LSM Reader . . . . .	10
2.7.4. ETH Mosaic Package . . . . .	11
2.7.5. Eclipse . . . . .	15
2.7.6. R . . . . .	15
<b>3. Bereitstellung der Werkzeuge</b>	<b>17</b>
3.1. Vorgehen . . . . .	17
3.1.1. Reduzierung zu Java-Klassen . . . . .	17
3.1.2. Einbindung in R . . . . .	20
3.1.3. Bereitgestellte Werkzeuge . . . . .	21
3.2. Überprüfung . . . . .	22
3.2.1. Einlesen von LSM-Files . . . . .	22
3.2.2. Automatische Erzeugung von Hintergründen . . . . .	24
3.2.3. Teilchenerkennung . . . . .	26

<b>4. Anwendung</b>	<b>31</b>
4.1. Ansatz zur Auswertung des Hintergrundes . . . . .	31
4.2. Bestimmung des optimalen Parameters L . . . . .	33
4.2.1. Einfluss des Ausschneideradius . . . . .	33
4.2.2. Abhängigkeit von L . . . . .	40
4.3. Teilchenabhängige Qualität des Hintergrundes . . . . .	54
<b>5. Zusammenfassung und Diskussion</b>	<b>59</b>
5.1. Werkzeuge . . . . .	59
5.2. Bestimmung des optimalen Parameters L . . . . .	59
5.2.1. Einfluss von $R_{cut}$ . . . . .	59
5.2.2. Unbestrahlte Referenzproben . . . . .	60
5.2.3. Bestrahlte Proben (Teilchenabhängig) . . . . .	60
5.2.4. Bestrahlte Proben (Tiefenabhängig) . . . . .	61
5.3. Teilchenabhängige Qualität des Hintergrundes . . . . .	61
<b>A. Spektroskopiedatensatz</b>	<b>65</b>



# Abbildungsverzeichnis

2.1.	<i>Beispiel für einen FNTD im Größenvergleich zu einer 1-Cent-Münze. . .</i>	4
2.2.	<i>Hintergrund in einem unbestrahlten (links) und mit Röntgenstrahlung bestrahlten (rechts) FNTD. . . . .</i>	5
2.3.	<i>Schema des Lichtweges des Zeiss LSM 710; Nachdruck aus [1] . . . . .</i>	7
2.4.	<i>Links: Originales Bild A, Mitte: Das erzeugte Hingergrundbild B, Rechts: Vom Hintergrund korrigiertes Bild (A-B); Nachdruck aus [11] . . . . .</i>	11
2.5.	<i>A: Originalbild mit sliding window. B: Histogramm der Pixelwerte innerhalb eines sliding windows. C: Hintegrund im ersten Schritt: Die Maxima der Histogramme werden in die Mitte des sliding window gesetzt. D: Komplettierung des Hintergrundbildes durch bilineare Interpolation der in C bestimmten Pixel. E: Geglättetes Hintergrundbild nach Faltung mit einer Gaußfunktion ; Nachdruck aus [11] . . . . .</i>	13
3.1.	<i>Vergleich zwischen dem Ergebniss von read.lsm.image (links) und dem originalen Bild aus ImageJ, das gedreht und gespiegelt wurde (rechts) für 1H, 3 MeV/u, Bild 14 . . . . .</i>	23
3.2.	<i>Vergleich zwischen dem Hintergrund von get.background (links) und dem des Background Subtractors (rechts) für 1H, 3 MeV/u, Bild 14 . . . . .</i>	24
3.3.	<i>Hintergründe für verschiedene L-Werte von 1H, 3 MeV/u, Bild 14 . . . . .</i>	25
3.4.	<i>Hintergründe für kleine L-Werte von 1H, 3 MeV/u, Bild 14 . . . . .</i>	26
3.5.	<i>Vergleich der detektierten Teilchen von find.tracks (links) und Particle Tracker (rechts) bei 1H, 3 MeV/u, Bild 14 . . . . .</i>	27
3.6.	<i>Einfluss der Particle Tracker Parameter für 1H, 3 MeV/u, Bild 14. Links oben sind zu sensible, rechts oben zu grobe und in der Mitte unten optimale Parameter dargestellt. . . . .</i>	28
3.7.	<i>Abweichung der optimalen Parameter bei (12C, 4 MeV), Bild 6 . . . . .</i>	29
4.1.	<i>Schema zur Auswertung der Hintergründe. . . . .</i>	32
4.2.	<i>Mittelwerte der manuellen Hintergründe über <math>R_{cut}</math> . . . . .</i>	36
4.3.	<i>Maxima der manuellen Hintergründe über <math>R_{cut}</math> . . . . .</i>	37
4.4.	<i>Standardabweichung der manuellen Hintergründe über <math>R_{cut}</math> . . . . .</i>	38
4.5.	<i>Beispiele für Proben mit unterschiedlichen Verläufen der Mittelwerte bei hohen <math>R_{cut}</math>-Werten. Dargestell sind jeweils die manuellen Hintergrundbilder mit <math>R_{cut} = 30</math>. Oben: Ansteigende Mittelwerte, Mitte: Schwach abfallende Mittelwerte, Unten: Stark abfallende Mittelwerte . . . . .</i>	39
4.6.	<i><math>\mu_p</math> in Abhängigkeit von L für unbestrahlte Proben . . . . .</i>	42
4.7.	<i><math>f_p</math> in Abhängigkeit von L für unbestrahlte Proben . . . . .</i>	43

---

4.8.	<i>Probe rm1003_BG mit entsprechenden Hintergrundbilder für <math>L = 2, 20, 50</math>. In der Mitte unten ist ein großer Fleck zu sehen, der für die schlechten Prüfwerte dieser Probe verantwortlich ist. Außerdem sieht man einige kleine Felcken im Bild.</i>	44
4.9.	<i><math>\mu_p</math> in Abhängigkeit von <math>L</math> für bestrahlte Proben</i>	48
4.10.	<i><math>f_p</math> in Abhängigkeit von <math>L</math> für bestrahlte Proben</i>	49
4.11.	<i>Differenz zwischen Maximum und Mittelwert der angepassten Countraten der automatischen Hintergründe in MHz über <math>L</math></i>	50
4.12.	<i>Differenz zwischen Maximum und Mittelwert der angepassten Countraten der automatischen Hintergründe in MHz über <math>L</math> für (<math>^{56}\text{Fe}</math>, 5 MeV)</i>	52
4.13.	<i>Differenz zwischen Maximum und Mittelwert der angepassten Countraten der automatischen Hintergründe in MHz über <math>L</math> für (<math>^{131}\text{Xe}</math>, 9 MeV)</i>	53
4.14.	<i>Vergleich einiger Bilder mit ihren optimalen Hintergrundbildern</i>	56
4.15.	<i><math>\mu_p</math> und <math>f_p</math> aufgetragen über die Mittelwerte der angepassten Countraten der manuellen Hintergrundbilder.</i>	57
A.1.	<i>Überblick über die Bilder des Spektroskopiedatensatzes</i>	66
A.-1.	<i>Überblick über die Hintergrundbilder des Spektroskopiedatensatzes mit <math>L=10</math></i>	70

# Tabellenverzeichnis

4.1.	<i>Übersicht der Prüfwerte für unbetrahlte Proben. Aufgelistet sind der insgesamt kleinste und größte Prüfwert und die Mittelwerte der Prüfwerte für <math>L=2</math>, <math>L=10</math> und <math>L=50</math>.</i>	41
4.2.	<i>Übersicht der Prüfwerte für betrahlten Proben. Aufgelistet sind der insgesamt kleinste und größte Prüfwert und die Mittelwerte der Prüfwerte für <math>L=2</math>, <math>L=10</math> und <math>L=50</math>.</i>	45
4.3.	<i>Übersicht der Prüfwerte für betrahlte Proben bei optimalen Parametern. Betrachtet wurde dabei das mittlere Bild aus jeder Probe.</i>	54
A.1.	Spektroskopiedatensatz mit Name, Energie $E$ und Teilchenart der Probe, den Bildernummern des verwendbaren Bereiches, mittlerem Bild, optimalen Particle Tracker Parametern Radius $\omega$ in Pixel, Cutoff $T_s$ und Percentile $r$ in % und den Parametern für manuellen und automatischen Hintergrund $R_{cut}$ und $L$ in Pixel.	65
A.2.	Metadaten der mittleren Bilder des Datensatzes. Eingetragen ist für jede Probe die <i>dwell time</i> $\tau$ in [ $\mu\text{s}$ ], die <i>number of rescans</i> $R$ , die <i>laser power</i> $p$ in %, die mittleren und maximalen Countzahlen $\bar{N}$ und $\max(N)$ , die mittleren und maximalen Nettocountraten $\bar{\eta}_{net}$ und $\max(\eta_{net})$ in [MHz] und die mittlere und maximale angepasste Nettocountrate $\bar{\eta}_{net}^{adj}$ und $\max(\eta_{net}^{adj})$ in [MHz].	74



# 1. Einleitung

Die Strahlentherapie mit Protonen und Ionen besitzt eine Reihe von Vorteilen gegenüber der konventionellen Strahlentherapie. Diese sind unter anderem das inverse Dosisprofil, eine bessere relative biologische Wirksamkeit (RBE) und der günstigere Sauerstoffverstärkungsfaktor. Begründet sind diese Vorteile in starken Schwankungen der Energie-deposition in kleinskaligen Bereichen. Detektoren, die auf diesen Skalen ( $\mu\text{m}$ ) arbeiten können, sind jedoch immer noch sehr begrenzt.

Ein neuer, auf Aluminiumoxidkristallen basierender fluoreszierender Kernspurdetektor, FNTD (engl. fluorescent nuclear track detector), entwickelt von der Crystal Growth Division von Landauer Inc. , Stillwater/OK, wird als vielversprechender Kandidat hierfür in unserer Forschungsgruppe *Heavy Ion Therapy* intensiv untersucht.

Eine Eigenschaft der FNTDs ist jedoch, dass diese auch in unbestrahlten Bereichen eine signifikante Fluoreszenz in der Größenordnung des zu messenden Signals („Hintergrund“) aufweisen.

Die Untersuchungen zur Natur, Größe und Variation dieses Hintergrundes wurde bisher jedoch durch die umständliche Anwendung der zu diesem Zweck prinzipiell einsetzbaren Routinen stark beschränkt. Das wesentliche Ziel dieser Arbeit war es daher, Softwarewerkzeuge zur automatisierten Bestimmung der Größe und Variation dieser Hintergründe bereitzustellen. Dazu wurden im ersten Schritt die bereits implementierten Werkzeuge für den automatisierten Gebrauch angepasst und in eine von unserer Gruppe verwendete Skriptsprache übertragen.

Im zweiten Schritt wurde die Funktionsweise überprüft und an einem Datensatz beispielhaft angewendet. Dabei wurden die Eigenschaften der Werkzeuge in Abhängigkeit von den Parametern der zugrundeliegenden Algorithmen untersucht, um die optimalen Parametereinstellungen zu erhalten.

In Kapitel 2 werden die verwendeten Methoden und Materialien eingeführt. Darunter sind fundamentale Größen, die FNTDs, die Bestrahlungseinrichtungen, in denen die FNTDs bestrahlt wurden, der Spektroskopiedatensatz, an dem die Werkzeuge getestet wurden, und die Software, auf deren Grundlage die Werkzeuge entwickelt wurden.

Weiter folgt in Kapitel 3 eine Beschreibung, wie die Werkzeuge entwickelt wurden, sowie einige Tests, ob sie korrekt übertragen und angepasst wurden.

In Kapitel 4 werden die Werkzeuge anschließend an einem Datensatz angewendet und deren Abhängigkeiten von den Parametern untersucht.

Zum Schluss werden in Kapitel 5 die wesentlichen Resultate dieser Arbeit zusammengefasst und diskutiert.



## 2. Methoden und Materialien

### 2.1. Fundamentale Größen

#### 2.1.1. Ionisierende Strahlung

Ionisierende Strahlung bezeichnet Teilchen- oder Photonenstrahlung, die aufgrund der ihr zur Verfügung stehenden Teilchen- bzw. Photonenenergie die Möglichkeit besitzt Materie zu ionisieren, d.h. Elektronen aus den Atomhüllen dieser Materie zu lösen.[15]

Sie verursacht in lebenden Organismen Strahlenschäden, wird aber auch aufgrund dieser Wirkung in der Medizin für die kurative oder palliative Behandlung verschiedener Krankheiten angewendet.

Aufgrund ihrer Wirkung ist es möglich die ionisierende Strahlung in speziellen Detektoren nachzuweisen und zu untersuchen.

#### 2.1.2. Dosis

Um die Strahlenwirkungen von ionisierender Strahlung zu quantifizieren, wurde eine Reihe von Dosisbegriffen eingeführt, die für verschiedene Messtechniken oder zur Abschätzung des Strahlenrisikos für Menschen relevant sind. Da die Strahlenwirkungen auf Menschen auf der Absorption der Strahlungsenergie im Gewebe beruht, ist die fundamentale makroskopische physikalische Dosisgröße die *Energiedosis*. [15]

**Energiedosis** Die Energiedosis  $D_{med}$  beschreibt die pro Massenelement absorbierte Energie. Für sie gilt:

Sei  $\rho_{med}$  die Dichte des Absorbermaterials (med = Medium),  $dE_{abs}$  die mittlere bei einer Bestrahlung mit ionisierender Strahlung lokal vom Absorbermaterial absorbierte Energie und  $dm_{med}$  die Masse des bestrahlten Volumenelements  $dV$  des Absorbermaterials.

Dann gilt:

$$D_{med} = \frac{dE_{abs}}{dm_{med}} = \frac{1}{\rho_{med}} \frac{dE_{abs}}{dV}$$

Die SI-Einheit der Energiedosis ist Joule pro Kilogramm ( 1 J/kg = 1 Gy).

Bei Energiedosisangaben muss das Absorbermaterial genannt werden, da für verschiedene Materialien auch die benötigten Energien zur Freisetzung von Elektronen variieren, die maßgeblich für die Absorption der Energie verantwortlich sind.

**Spezifische Energie** Für die Betrachtungen in dieser Arbeit ist die spezifische Energie von besonderer Bedeutung. Für diese gilt nach dem 85. Bericht der Internationalen

Kommission für Strahlungseinheiten und Messung, ICRU, von 2011 [14]:

Die (übertragene) spezifische Energie,  $z$ , ist der Quotient von  $\epsilon$  und  $m$ . Dabei bezeichnet  $\epsilon$  die durch ionisierende Strahlung auf Materie in einem Volumen der Masse  $m$  übertragene Energie.

$$z = \frac{\epsilon}{m}$$

Die SI-Einheit der spezifischen Energie ist Joule pro Kilogramm (  $1 \text{ J/kg} = 1 \text{ Gy}$ ).

## 2.2. FNTD (fluorescent nuclear track detector)

### 2.2.1. Einleitung

Die fluoreszierenden Kernspurdetektoren (FNTD, engl. fluorescent nuclear track detector), wurden ursprünglich von der Crystal Growth Division von Landauer Inc. Stillwater/OK für die optische Datenspeicherung mit hoher Kapazität entwickelt.

Abbildung 2.1 zeigt einen typischen FNTD-Chip, wie sie in unserer Gruppe verwendet werden, im Größenvergleich zu einer 1-Cent-Münze. Die FNTD-Chips bestehen aus einzelnen Aluminiumoxid-Kristallen, dotiert mit Kohlenstoff und Magnesium,  $\text{Al}_2\text{O}_3:\text{C},\text{Mg}$ . Durch besondere Bedingungen bei der Erzeugung der Kristalle enthalten die FNTDs Farbzentren, die unter ionisierender Strahlung durch herausgelöste Elektronen eine radiochromatische Transformation zeigen. Bei Stimulation mit 620 nm zeigen solche transformierten Farbzentren eine starke Fluoreszenz bei 750 nm mit kurzer Lebensdauer ( $75 \pm 5 \text{ ns}$ ). Dies ermöglicht die Auslese mittels eines konfokalen Laser Scanning Mikroskopes, ohne den Zustand der Detektoren zu verändern.

Die Signalintensität hängt in bestimmten Grenzen von der spezifischen Energie (siehe Abschnitt 2.1.2) ab.

Folglich ist es möglich ganze Bahnen von ionisierenden Teilchen, die einen FNTD passiert haben, in diesem dauerhaft zu speichern und beliebig oft auszulesen. Eine genaue Beschreibung der Eigenschaften und Funktionsweise der FNTDs findet sich in [10].

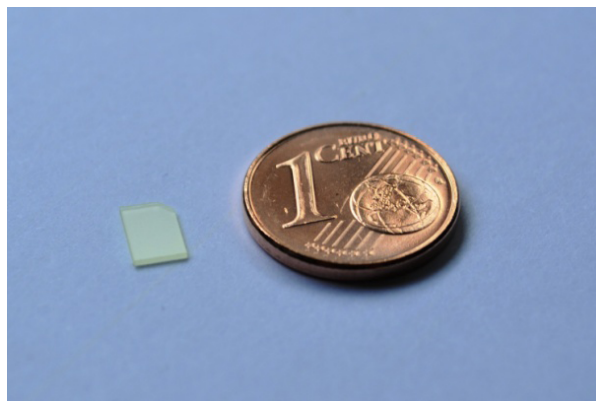


Abbildung 2.1.: *Beispiel für einen FNTD im Größenvergleich zu einer 1-Cent-Münze.*



### 2.2.2. Hintergrund

Die FNTDs zeigen eine Fluoreszenz in unbestrahlten Bereichen und auch wenn sie nicht bestrahlt wurden, welche in der Größenordnung der zu messenden Signale ist. Diese Fluoreszenz wird im Rahmen dieser Arbeit als *Hintergrund* bezeichnet. Der Hintergrund kann von der Größenordnung zwischen verschiedenen FNTDs und sogar innerhalb eines FNTDs deutlich schwanken. Bei letzterem zeigt er zwei verschiedene Formen von Variationen.

**Kleinskalige Variation** Diese besteht aus Strukturen von etwa  $1\ \mu\text{m}$ . Sie bleibt stabil bei mehrmaligen Auslesen, widersteht Erhitzen (z.B. 10 min bei  $680^\circ\text{C}$ ) und optischer Bleichung. Die Ursache für diese Variation ist noch unbekannt. Eine wahrscheinliche Möglichkeit sind Inhomogenitäten in der Dotierung mit Magnesium.

**Grobskalige Variation** Diese zeigt Strukturen in größeren Skalen ( $> 10\ \mu\text{m}$ ). Sie können in einigen Bildern Schwankungen um den Faktor 2 ausmachen.

Abbildung 2.2 zeigt Beispiele für den Hintergrund in einem unbestrahlten und einem mit Röntgenstrahlung bestrahlten FNTD.

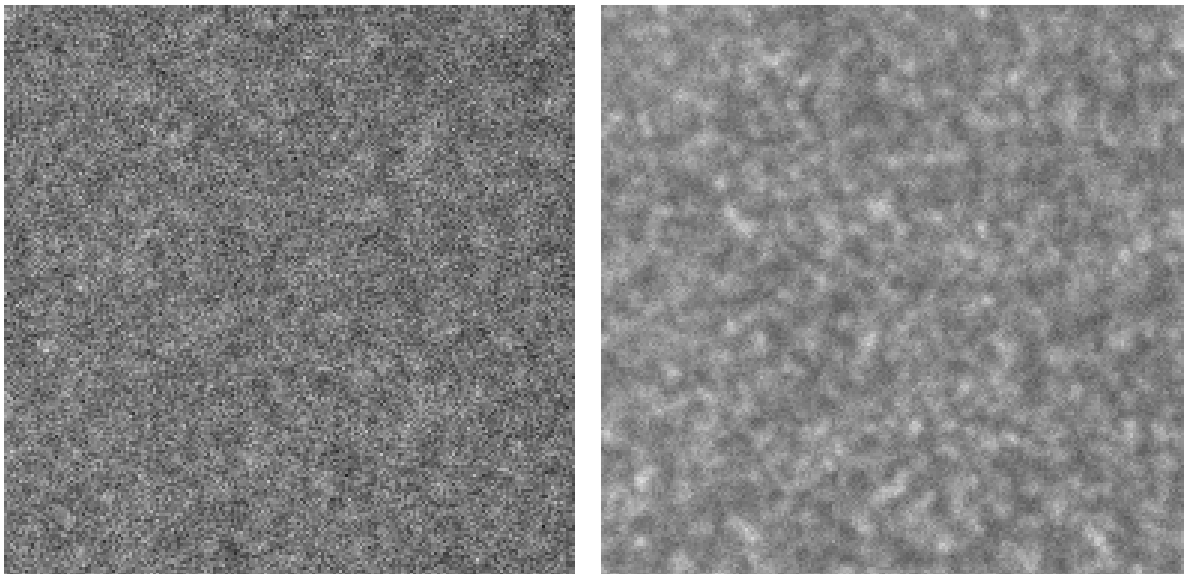


Abbildung 2.2.: *Hintergrund in einem unbestrahlten (links) und mit Röntgenstrahlung bestrahlten (rechts) FNTD.*

In beiden Beispielen kann man die kleinskalige Variation als eine Art von „Rauschen“ erkennen.

Neben diesen Variationen zeigen sich an den Kanten der Detektoren ebenfalls starke Fluktuationen des Hintergrundes.

Diese werden jedoch nicht weiter diskutiert, da diese Bereiche beim Auslesen der FNTDs stets ausgespart werden.

## 2.3. Bestrahlungseinrichtungen

In diesem Abschnitt werden die Einrichtungen genannt, in denen die FNTDs, die in dieser Arbeit behandelt werden, bestrahlt wurden.

### 2.3.1. Max-Planck-Institut für Kernphysik (MPIK)

Das MPIK hat einen 12 MV MP-Tandem van-de-Graaff-Beschleuniger, der mit einer maximal realisierbaren Spannung von bis zu 11 MV mit Ionenstrahlen betrieben werden kann.

Es steht eine große Bandbreite an Ionen in einem weiten Energie- und Fluenzbereich zur Verfügung. [5]

### 2.3.2. Heidelberger Ionenstrahl-Therapiezentrum (HIT)

Das HIT ist eine Ionentherapie-Anlage des Universitätsklinikums Heidelberg.

Es ist die erste kombinierte Therapieanlage in Europa, in der Bestrahlung sowohl mit Protonen als auch mit Schwerionen (Kohlenstoff-, Sauerstoff-, Heliumionen) möglich ist.

Darüber hinaus ist es die weltweit erste Ionentherapie-Anlage, die mit dem Intensitätsmodulierten Raster-Scan-Verfahren ausgestattet ist, welches bisher das präziseste Bestrahlungsverfahren in der dreidimensionalen Bestrahlung von Tumoren ist. [2]

### 2.3.3. Center for Nuclear Technologies (DTU Nutech), Roskilde

Das DTU Nutech ist Dänemarks nationales Kompetenzzentrum für nukleare Technologie.

Dort ist unter anderem die Bestrahlung von Proben mit Gammastrahlung (Cobalt-60) mit Dosisleistungen bis zu einigen 10 Gy/min möglich.

## 2.4. Zeiss LSM 710

Zum Auslesen der FNTDs wurde im Rahmen dieser Arbeit das Zeiss LSM 710 konfokale Laser Scanning Mikroskop zusammen mit der Kontrollsoftware ZEN 2009 verwendet.

In diesem Abschnitt werden die Besonderheiten des LSM 710 und der konfokalen Laser Scanning Mikroskopie behandelt. Eine genaue Beschreibung hierüber findet sich in [1].

### 2.4.1. Konfokale Mikroskopie

Bei einem Laser Scanning Mikroskop rastert ein fokussierter Laserstrahl die Probe punktweise ab. Der Vorteil der konfokalen Laser Scanning Mikroskopie liegt darin, dass hierbei nur Licht eingefangen wird, das aus einer einzelnen Ebene der Probe emittiert wird. Dies wird dadurch erreicht, dass eine zur Fokusebene konjugierte Lochblende das Licht blockiert, welches von Objekten außerhalb der vom Laser abgetasteten Ebene eingestrahlt

wird. Die Daten jedes abgerasterten Punktes der Ebene werden zu einem Bild zusammengesetzt, das damit einen optischen Schnitt durch die Probe darstellt. Durch Verschieben der Fokusebene können mehrere Bilder solcher Schnitte in verschiedenen Tiefen erstellt und zu einem 3D-Bilderstapel zusammengesetzt werden. Das Schema des Lichtweges im LSM 710 ist in Abbildung 2.3 dargestellt.

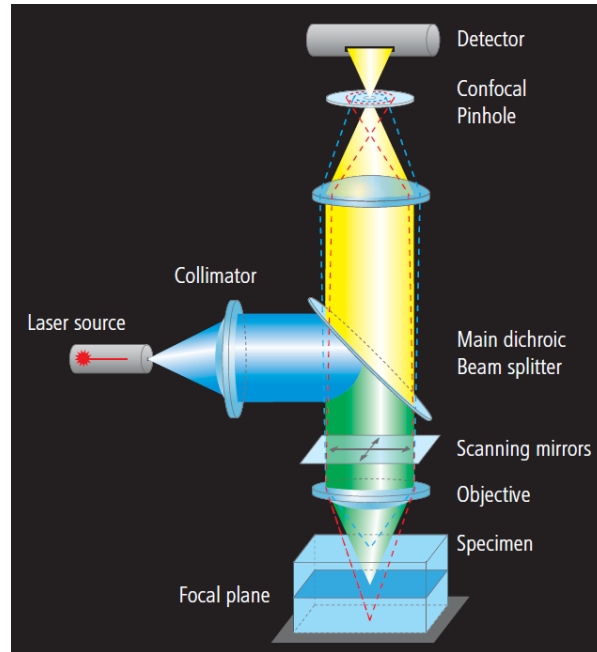


Abbildung 2.3.: Schema des Lichtweges des Zeiss LSM 710; Nachdruck aus [1]

## 2.4.2. Kontrollparameter

Die Kontrollsoftware ZEN ermöglicht bei der Auslese die Manipulation der folgenden für diese Arbeit relevanten Parameter [16]:

- Die *dwell time*  $\tau$  bestimmt wie lange der Laser einen Punkt beleuchtet und beeinflusst damit die gesamte Anzahl an detektierten Photonen (Countzahl)  $N$ . Sie lässt sich variieren von 1,27 bis 177  $\mu\text{s}$ .
- Die *number of rescans*  $R = 1, \dots, 16$  bestimmt wie oft der Laser wiederholt über einen Punkt scannt und hat damit ebenfalls Einfluss auf  $N$ .
- Die *laser power*  $p = 0.1$  bis 100 % bestimmt die relative Anregung des Lasers. Sie beeinflusst ebenfalls  $N$ .
- Der *pinhole diameter*  $d_p = 10$  bis 520  $\mu\text{m}$  regelt die Dicke des Schnittes durch die Probe. Standardmäßig wird  $d_p = 1$  AU gewählt. (AU (Airy unit) ist eine Größe um den *pinhole diameter* zu normieren.)
- Das zur Auslese verwendete Objektiv. Verwendet wurde im Rahmen dieser Arbeit das Objektiv *Plan-Apochromat 63x/1.40 Oil DIC M27*.

### 2.4.3. LSM-Dateiformat

Die 3D-Bilderstapel werden von ZEN in dem eigenen von Zeiss entwickelten *LSM-Dateiformat* mit der Dateierweiterung *.lsm* gespeichert. Dies ist eine Weiterentwicklung des zum Speichern von Bilddaten weit verbreiteten TIFF (Tagged Image File Format), mit eigenen spezifischen „tags“. Es werden sowohl die Bilddaten als auch die beim Auslesen am Mikroskop eingestellten Kontrollparameter gemeinsam in einer LSM-Datei abgespeichert. Eine detaillierte Beschreibung des LSM-Fileformats ist in [12] gegeben.

## 2.5. Normierung der Countzahl $N$

Bei den im LSM-Dateiformat abgespeicherten Bilddaten ist in jedem Pixel als Wert die *Countzahl*  $N$ , also die gesamte Anzahl an detektierten Photonen, abgespeichert. Da  $N$  jedoch von den in Abschnitt 2.4.2 beschriebenen Parametern  $\tau$ ,  $R$  und  $p$  beeinflusst wird und diese oft variiert werden, um die Strukturen in verschiedenen Proben sichtbar zu machen, kann man die Bilddaten verschiedener Proben nicht immer vergleichen.

Um dies zu beheben, wurde in unserer Gruppe bereits eine Reihe von Experimenten zum Einfluss der verschiedenen Parameter durchgeführt. In [16] werden unter anderem einige dieser Experimente beschrieben und die daraus resultierenden Größen, die beim Vergleich verschiedener Proben zu verwenden sind, hergeleitet. Diese werden im Folgenden eingeführt, wobei angenommen wird, dass die übrigen Kontrollparameter außer  $\tau$ ,  $R$  und  $p$  konstant gehalten werden, was für die im Rahmen dieser Arbeit untersuchten Proben zutrifft.

Da sich eine lineare Abhängigkeit der *Countzahl*  $N$  von  $\tau$  und  $R$  gezeigt hat, wird zunächst die *Countrate*  $\eta$  definiert:

$$\eta = \frac{N}{\tau R}$$

Da  $\tau$  meist in  $\mu\text{s}$  angegeben ist, wird  $\eta$  im Rahmen dieser Arbeit in MHz angegeben. In der *Countrate* ist jedoch noch der *Dunkelstrom*  $\eta_{DC}$  des Mikroskopes enthalten. Dieser stellt einen positiven Offset zur *Countrate* dar und kann durch Messung der *Countrate* bei ausgeschaltetem Laser aber ansonsten gleichen äußeren Bedingungen wie bei normalen Messungen (z.B. Raumbeleuchtung) bestimmt werden. Für die hier untersuchten Proben ist  $\eta_{DC} = 0,0035$  MHz eine gute Näherung. Um den Dunkelstrom zu berücksichtigen, wird die *Netto-Countrate*  $\eta_{met}$  eingeführt:

$$\eta_{met} = \eta - \eta_{DC}$$

Diese kann zum Vergleich von Bildern, die mit gleicher *laser power*  $p$  aufgenommen wurden, herangezogen werden.

Es hat sich gezeigt, dass  $\eta_{met}$  linear von  $p$  abhängt. Um also die Einflüsse von  $p$  zu beseitigen, führt man die *angepasste Netto-Countrate*  $\eta_{met}^{adj}$  ein, für die gilt:

$$\eta_{met}^{adj} = \frac{\eta_{met}}{p}$$

Diese ist nun unabhängig von  $\tau$ ,  $R$  und  $p$  und kann damit zum Vergleich unterschiedlicher Proben benutzt werden.

Bei diesen Betrachtungen ist zu beachten, dass sie nur das Mikroskop betreffen. Jedoch zeigt sich in [16], dass der Einfluss der Farbzentren bei Variation dieser Parameter bei den hier untersuchten Proben vernachlässigt werden kann.

## 2.6. Spektroskopiedatensatz

Die im Rahmen dieser Arbeit entwickelten Werkzeuge werden beispielhaft an einem großen Spektroskopiedatensatz angewendet. Dieser Datensatz wurde bereits von anderen Mitgliedern unserer Gruppe in verschiedenen Experimenten aufgenommen. Er besteht aus einer Reihe mit dem LSM 710 ausgelesener FNTDs, die am MPIK und am HIT mit verschiedenen Ionen von  $^1H$  bis  $^{131}Xe$  mit unterschiedlichen Energien zwischen 3 und 430 MeV/u bestrahlt wurden. Dabei wurden die FNTD's ausschließlich orthogonal bestrahlt. Ausgelesen wurden meist 3D-Bildstapel mit bis zu einigen hundert Bildern in der Tiefe. Durch die orthogonale Bestrahlung stellt jedes Bild einen Querschnitt der Ionenbahnen durch den Detektor dar.

Da die verschiedenen Proben durch ihre Ionenart und Energie eindeutig bestimmt sind, wird im Folgenden in dieser Arbeit eine Abkürzung für die einzelnen Proben der Form (*Ionenart, Energie pro u*) eingeführt.

Zum Beispiel wird damit die Probe, die mit 1H-Ionen der Energie 12 MeV/u bestrahlt wurde, mit (1H, 12 MeV) abgekürzt.

Außerdem ist wichtig, dass in diesem Datensatz die Kontrollparameter  $\tau$ ,  $R$  und  $p$  des LSM 710 nicht einheitlich sind. Da diese wie in Abschnitt 2.4 erwähnt Einfluss auf die gesamte Anzahl an detektierten Photonen haben, wird später zur vergleichbaren Auswertung eine Normierung mit diesen Parametern nötig sein.

Eine genaue Liste mit den Daten des Datensatzes ist in Tabelle A.1 im Anhang zu finden. In der „Name“-Spalte ist der Name des LSM-Files zu der Probe eingetragen. Da sich bei einigen Bilderstapeln gezeigt hat, dass die ersten und letzten Bilder des Stapels für die Untersuchung in dieser Arbeit unbrauchbar sind, da entweder kein Signal zu sehen ist oder der Detektor die Sättigungsgrenze erreicht hat, ist in der Liste zusätzlich das erste und letzte Bild des brauchbaren Bereiches der Bilderstapel eingetragen. Die Auswahl dieses Bereiches wurde von Hand durchgeführt.

## 2.7. Software

In diesem Abschnitt wird die bereits vorhandene Software, auf deren Grundlage die im Rahmen dieser Arbeit bereitgestellten Werkzeuge aufbauen, beschrieben.

### 2.7.1. Java

Java ist eine 1995 von Sun Microsystems entwickelte objektorientierte Programmiersprache. Sie zeichnet sich durch ihre hohe Plattformunabhängigkeit aus und ist kostenlos verfügbar, weshalb sie in vielen Open-Source-Projekten verwendet wird.

Eine Einführung in die Java-Programmierung findet sich in [8] und ein umfassendes Nachschlagewerk für fortgeschrittene Programmierer in [9].

Im Folgenden werden einige notwendige Begriffe der Java-Programmierung eingeführt:

- In objektorientierten Programmiersprachen bezeichnet eine *Klasse* eine Datenstruktur, die Funktionen und Variablen enthalten kann. Sie kann als Bauplan für eine „neue Art“ von Variable dienen, die dann *Objekt* vom Typ der Klasse genannt wird.
- Da in Java jedes Programm in Form einer Klasse geschrieben wird, wird im Folgenden *Java-Klasse* synonym für Java-Programm verwendet.
- Funktionen einer Klasse werden *Methoden* genannt.
- *Standarddatentypen* bezeichnen in Java ganze Zahlen, boolesche Ausdrücke, einzelne Zeichen (character) und Fließkommazahlen.
- Ein *Feld (Array)* ist eine Datenstruktur, die aus einer indizierten Reihe eines einheitlichen Datentyps besteht.

### 2.7.2. ImageJ

ImageJ ist ein in Java geschriebenes und damit plattformunabhängiges Bildbearbeitungsprogramm, das von Wayne Rasband entwickelt wurde.

Es ist ein Open-Source-Projekt, was bedeutet, dass der Quellcode des Programms von jedem eingesehen und in eigene Projekte eingebunden werden kann.

Durch das einfache Einbinden von Funktionen in ImageJ in Form von Plugins, hat sich eine große Benutzergemeinde entwickelt, die neue Plugins für ImageJ entwickelt und somit den Funktionsumfang des ursprünglichen Programms erweitert. Dadurch ist ImageJ inzwischen ein Standardprogramm zur Bildverarbeitung in der Mikroskopie geworden.

[4] Die im Rahmen dieser Arbeit verwendete Version ist *ij146m* vom 23.04.2012.

### 2.7.3. LSM Reader

Der LSM Reader ist ein Plugin für ImageJ, um Bilder des LSM-Formates, wie sie vom Zeiss LSM 710 ausgegeben werden, zu lesen. Der Quellcode des LSM Readers ist frei zugänglich. Dies ermöglichte seine Einbindung in die eigenen zu entwickelnden Werkzeuge. Die in dieser Arbeit verwendete Version ist 4.0g vom 04.04.2011. [3]

### 2.7.4. ETH Mosaic Package

Die MOSAIC Gruppe der ETH-Zürich hat eine Vielzahl von eigen entwickelten Bildverarbeitungs-routinen in dem MOSAIC-Plugin für ImageJ zusammengefasst.

Das gesamte Softwarepaket ist Open Source und wird in einem frei zugänglichen Repository verwaltet. Damit lässt es sich für eigene Projekte verwenden. Im Rahmen dieser Arbeit wurde die Revision 4059 des Projektes verwendet. [6]

Im Folgenden werden die beiden Plugins beschrieben, die aus diesem Paket verwendet wurden.

#### Mosaic Background Subtractor

Das Plugin Background Subtractor ist entwickelt worden, um aus Bildern mit nicht konstanten Hintergründen, wie sie z.B. in der Lichtmikroskopie auftreten, diese abzuziehen und in einem eigenen Bild zu speichern.

In Abbildung 2.4 ist ein Beispiel für die Anwendung des Background Subtractors gezeigt.

Um den Hintergrund eines Bildes erfolgreich zu extrahieren ist jedoch vorausgesetzt, dass die dargestellten Objekte (Vordergrund) des Bildes im Vergleich zur Hintergrundstruktur klein sind. In der Diplomarbeit von F. Lauer [13] hat sich der Background Subtractor für FNTD's im Vergleich zu den anderen dort getesteten Methoden als die Beste herausgestellt, um die *Grobskalige Variation* (siehe Abschnitt 2.2.2) zu extrahieren.

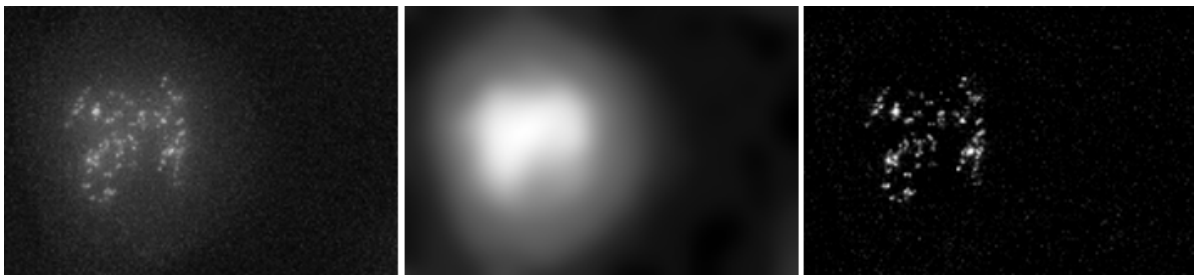


Abbildung 2.4.: Links: Originales Bild A, Mitte: Das erzeugte Hintergrundbild B, Rechts: Vom Hintergrund korrigiertes Bild (A-B); Nachdruck aus [11]

#### Algorithmus

Der Algorithmus, auf dem der Background Subtractor basiert, benutzt ein lokales Histogramm-Verfahren:

In einer Schleife rastert ein *sliding window* der Seitenlänge  $L$  mit der Schrittweite  $L/2$  pro Iteration durch das zu bearbeitende Bild. Die Seitenlänge sollte dabei so gewählt werden, dass die Vordergrundobjekte komplett im sliding window enthalten sind.

In jedem Iterationsschritt wird ein lokales Histogramm der im sliding window befindlichen Bildpunkte (Pixel) erstellt. D.h. es wird ein Diagramm erzeugt, indem die Häufigkeiten der in dem Bild auftretenden Pixelwerte über die Pixelwerte selbst aufgetragen sind. Das Maximum dieses Histogramms, also der Pixelwert, der am häufigsten

im sliding window vorkommt, wird in dem zu erzeugenden Hintergrundbild als Wert des Pixels in der Mitte des sliding windows gesetzt.

An den Rändern des Bildes, wird dieses gespiegelt um das lokale Histogramm erzeugen zu können.

Im nächsten Schritt werden die Pixel, die damit nicht bestimmt wurden, durch bilineare Interpolation der vier am nächsten liegenden bereits bestimmten Pixel, berechnet.

Anschließend wird das so bestimmte Hintergrundbild mit einer Gaußfunktion mit dem Radius  $2 \cdot L$  gefaltet um Rauschen, verursacht durch zu kleine  $L$  oder zu hohe Konzentrationen an Objekten, herauszufiltern. [11]

Das Verfahren wird in Abbildung 2.5 veranschaulicht.

Beim Abziehen des Hintergrundbildes wird jeder Pixelwert einzeln vom entsprechenden Pixel des Originalbildes abgezogen. Falls dabei negative Werte auftreten werden diese Pixelwerte automatisch auf 0 gesetzt.



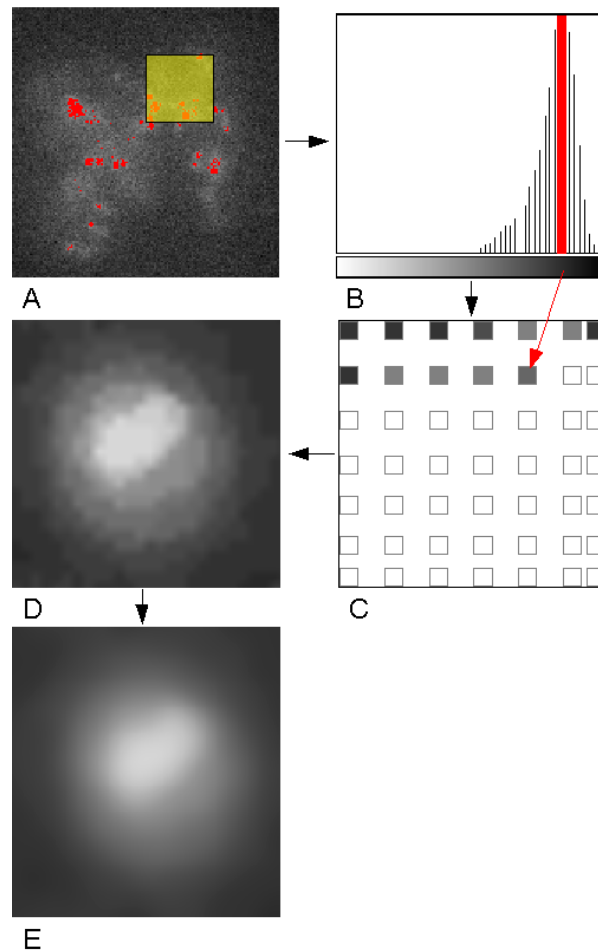


Abbildung 2.5.: *A: Originalbild mit sliding window. B: Histogramm der Pixelwerte innerhalb eines sliding windows. C: Hintergrund im ersten Schritt: Die Maxima der Histogramme werden in die Mitte des sliding window gesetzt. D: Komplettierung des Hintergrundbildes durch bilineare Interpolation der in C bestimmten Pixel. E: Geglättetes Hintergrundbild nach Faltung mit einer Gaußfunktion ; Nachdruck aus [11]*

### Mosaic Particle Tracker

Das Particle Tracker Plugin ist entwickelt worden, um in 3D-Bilderstapeln, wie sie z.B. auch vom LSM 710 erzeugt werden können, kleine Strukturen (verglichen zur Größe des aufgenommen Bildes), die sich vom Hintergrund des Bildes durch erhöhte Pixelwerte abheben und die im Folgenden *Teilchen* genannt werden, zu detektieren und durch Verbinden der detektierten Teilchen mit den zugehörigen aus anderen Bildern des Stapels eventuell vorliegende Teilchenbahnen zu rekonstruieren.

Verwendet wird in dieser Arbeit nur die Detektion der Teilchen, um diese herauszuschneiden und so ausschließlich die Teile eines Bildes, die nur Hintergrund enthalten, zu erhalten.

## Algorithmus

Eine genaue Beschreibung der Funktionsweise und der Fähigkeiten des zugrundeliegenden Algorithmus ist in [17] gegeben.

Für die zu entwickelnden Werkzeuge wird nur der Teil des Algorithmus gebraucht, indem die Teilchen detektiert d.h. ihre Positionen bestimmt werden.

Dies läuft in vier Schritten ab:

1. Bildwiederherstellung
2. Abschätzung der Teilchenposition
3. Verfeinerung der Teilchenposition
4. Unterdrückung falscher Teilchen

In Schritt 1 werden Fehler in den Bildern korrigiert, die bei der Aufnahme, z.B. mit einer CCD-Kamera, entstehen können.

In Schritt 2 wird anschließend durch Suche nach lokalen Maxima die Teilchenposition abgeschätzt. Ein Pixel wird als angenäherte Teilchenposition betrachtet, wenn kein anderes Pixel innerhalb eines Radius  $w$  heller ist (d.h. einen höhere Pixelwert hat) und gleichzeitig der Wert des Pixels oberhalb der Perzentile  $P(100-r)$  der Pixelwerte des Gesamtbildes ist.

Daraufhin wird in Schritt 3 aus den in Schritt 2 bestimmten Positionen durch Berücksichtigung der Helligkeiten aller Pixel innerhalb des Radius  $w$  eine genauere Position des geometrischen Zentrums des Teilchens bestimmt.

Zum Schluss werden in Schritt 4 fälschlich bestimmte Objekte, die die Bedingungen in Schritt 2 und 3 erfüllt haben, aussortiert.

Dabei wird angenommen, dass die meisten der detektierten Teilchen ähnliche Eigenschaften in Breite und Helligkeit aufweisen.

Um ein Ausschlusskriterium für Teilchen zu bekommen, die stark von den anderen Teilchen abweichen, wird eine mathematische Größe definiert, die für jedes detektierte Teilchen einen charakteristischen Wert für seine Ähnlichkeit in Größe und Helligkeit von allen anderen liefert. Dieser Wert wird um so größer je ähnlicher es den anderen Teilchen in diesen Eigenschaften ist.

Ist dieser Wert für ein Teilchen unter einer Schwelle  $T_s$ , so wird es ausgeschlossen.

Folglich hat der Benutzer drei Parameter zu wählen, die der Particle Tracker benötigt:

1. Den *radius*  $w$  in Pixeln, mit dem bestimmt wird, welcher Bereich um die detektierten Teilchen berücksichtigt werden soll.
2. Die *percentile*  $r$  in %, mit der man steuern kann, wie hoch der Wert eines Pixels relativ zu den anderen sein muss, damit es potentiell als Teilchenposition bestimmt werden kann.

3. Den *cutoff*  $T_s$ , mit dem man eine Schwelle setzt, ab der detektierte Teilchen aufgrund von abweichenden Eigenschaften im Vergleich zur Mehrheit der anderen detektierten Teilchen ausgeschlossen werden sollen.

### 2.7.5. Eclipse

Eclipse ist eine ursprünglich für Java entwickelte integrierte Entwicklungsumgebung, die durch viele Erweiterungen zu einem umfassenden Programmierwerkzeug auch für andere Sprachen wie C oder C++ gewachsen ist. Die im Rahmen dieser Arbeit verwendete Version ist *Eclipse SDK Version: 3.7.2*.

Eclipse wurde in dieser Arbeit vor allem deshalb verwendet, weil es das Importieren von in Java geschriebenen Open-Source-Projekten und ihr anschließendes Verknüpfen ermöglicht.

So konnten das ImageJ-, MOSAIC- und LSM Reader-Projekt mit einem eigenen Projekt zusammengeführt und somit in diesem auf die benötigten Ressourcen zur Bereitstellung der eigenen Werkzeuge zugegriffen werden.

### 2.7.6. R

R ist eine Programmiersprache mit Skriptfähigkeit und gleichzeitig Umgebung für statistische Berechnungen und Grafiken.

R steht als freie Software zur Verfügung und ermöglicht die Erweiterung der Sprache durch Erzeugen eigener Pakete. Dadurch steht auch eine große Menge bereits von anderen Nutzern entwickelten Pakete zur Verfügung, die man leicht durch Einbinden in eigene Projekte nutzen kann. [7]

#### rJava

rJava ist ein Paket für R, das als Schnittstelle zwischen R und Java dient. Es ermöglicht das Erzeugen von Objekten von Java-Klassen in R, auf deren Methoden zuzugreifen und zurückgelieferte Standarddatentypen und Datenfelder weiterzuverarbeiten.

Die hier verwendete Version von rJava ist *0.9-3*.

#### R-Paket - „FNTD“

FNTD ist ein Paket für R, das momentan in unserer Gruppe entwickelt wird, um Routinen zur Verfügung zu stellen, die das Bearbeiten der vom Zeiss LSM 710 erzeugten Bilder der FNTDs in R ermöglichen.

Beispiele hierfür sind die Bereitstellungen von mathematischen Operationen, statistischen Größen wie Mittelwert und Standardabweichung sowie von einheitlichen Plot-Routinen für diese Bilder.

Dazu wurde eine Datenstruktur *FNTDimage* implementiert, die diese Bilder in R repräsentiert. Die aktuellste hier verwendete Version ist *0.4.10*. In dieses Paket wurden auch die im Rahmen dieser Arbeit entwickelten Werkzeuge integriert.

## 3. Bereitstellung der Werkzeuge

### 3.1. Vorgehen

In diesem Abschnitt wird das Vorgehen beschrieben, mit dem aus den in Abschnitt 2.7 eingeführten Ressourcen die Werkzeuge bereitgestellt wurden.

#### 3.1.1. Reduzierung zu Java-Klassen

Die Grundlagen der Werkzeuge stellen die ImageJ-Plugins *LSM Reader*, *MOSAIC Background Subtractor* und *MOSAIC Particle Tracker* dar.

Zu Beginn der Arbeit waren diese Plugins derart an die Bildbearbeitungssoftware ImageJ gekoppelt, dass sie nur das Bearbeiten und den Zugriff auf komplette Bilddaten ermöglichen und auch die einzelnen Bestandteile der Plugins nur im Rahmen von ImageJ ausführbar waren. Das Paket *rJava* ermöglicht es jedoch nur, in R Objekte vom Typ normaler Java-Klassen zu erzeugen und von deren Methoden nur Standarddatentypen und Felder als Rückgabewerte zu bearbeiten.

Um eigene Klassen zu erstellen, die unabhängig von ImageJ arbeiten und dennoch die Ressourcen der anderen Projekte nutzen können, wurde zunächst mit Eclipse ein eigenes Projekt erstellt, das an die Projekte der Plugins und an ImageJ gekoppelt war.

Im nächsten Schritt wurden Teile der ImageJ-Plugins in diese Klassen übernommen und so abgeändert, dass sie anstatt mit Bildern mit Datenfeldern arbeiten. Dies wird im Nachfolgenden für die drei ImageJ-Plugins detaillierter beschrieben.

**LSM Reader** Die Klasse des LSM Readers, die den eigentlichen Lesevorgang der LSM-Datei durchführt, ist *Reader*. Sie besitzt die Methode *open*, die mit Eingabe eines Dateipfades zu einer Datei vom Typ LSM diese öffnen kann. Dabei werden die Bild- oder Bilderstapeldaten in einen extra dafür vorgesehenen Datentyp *ImagePlus* gespeichert. In Java sieht dieser Schritt wie folgt aus:

Speichern der Bilddaten eines LSM-Files

```
ImagePlus imp = new Reader().open(Path, true);
```

Dabei wird mit *new* ein Objekt der Klasse *Reader* erzeugt und mit „.“ auf die *Reader*-Methode *open* zugegriffen. Diese bekommt mit *Path* den Pfad zu einer LSM-Datei und mit dem boolean *true* wird der *Verbose*-Modus eingeschaltet, der für eine genauere Dokumentation bei Fehlern sorgt.

Die so geöffneten Bilddaten werden in die Variable *imp* vom Typ *ImagePlus* geladen.

Als nächstes müssen aus dieser Variable alle nötigen Informationen herausgelesen werden, um später in R den Bilderstapel zu rekonstruieren. Dabei können aber nur Standarddatentypen oder Felder in R übernommen werden. Deshalb wurde eine eigene Klasse *FNTD\_LSMFileReader* erstellt, die mehrere Methoden enthält, die jeweils einzelne Informationen aus *imp* herauslesen.

Die erste Methode ist *getFileData*. Ruft man sie ohne weitere Parameter auf, liest sie aus *imp* ein Feld aus, das die Pixelwerte aller Pixel des Bilderstapels enthält. Mit Angabe zweier ganzzahliger Parameter kann man die Bilder des Bilderstapels, die als Feld ausgelesen werden sollen, eingrenzen.

Um aus dem Feld, das *getFileData* ausgibt, auch einen Bilderstapel zu rekonstruieren, benötigt man zusätzlich die Information, wie breit und wie lang ein Bild in Pixeln ist und wie viele Bilder den gesamten Stapel bilden. Diese werden mit *getSize* in Form eines ganzzahligen Feldes mit drei Einträgen ausgelesen.

Damit sind alle Informationen, um das Bild in R zu rekonstruieren beisammen.

Jedoch sollten auch die Kontrollparameter, die in Abschnitt 2.4.2 genannt wurden, in R übernommen werden, um zum Beispiel eventuelle Abhängigkeiten von diesen bequem untersuchen zu können.

Diese sind ebenfalls im LSM-Format abgespeichert, können aber nicht mit dem *LSM Reader* gelesen werden. Deshalb musste hier „von Hand“ vorgegangen werden. Dies bewerkstelligt die Methode *getMetaData*. Im Folgenden ist ein Java-Codeausschnitt zu sehen, in dem das Vorgehen hierbei beispielhaft gezeigt wird. Dieser sucht nach einer hexadezimalen Bytefolge (*4C490004*), die als Identifikator für LSM-Dateien dient und daher immer vorkommen muss, und speichert deren Position in der Datei, falls er diese findet.

#### Beispiel zum Finden von Metainformatinen im LSM-File

```

1 RandomAccessFile file ;
2 file = new RandomAccessFile(new File(directory , fileName) , "r" );
3
4 int position = 0;
5 boolean running = true;
6 for (int i=position; running; i+=1) {
7     file.seek(i);
8     if (file.readInt() == 0x4C490004) {
9         position = i;
10        running = false;
11    }
12    if (i > 1000000) {
13        IJ.showMessage("File ist kein gültiges LSM!");
14        running = false;
15        break;
16    }
17 }

```

Hierbei wird in Zeile 2 ein Objekt *file* vom Typ *RandomAccessFile* erzeugt, wobei als Parameter ein Objekt vom Typ *File* und ein Zeichen, das die Zugriffsrechte regelt, benötigt wird. Das „r“ steht hierbei für *read only*. Das *File*-Objekt ist die Entsprechung einer realen Datei in Java. Um es zu erzeugen, benötigt man als Parameter den Pfad und den Namen der Datei.

Das *RandomAccessFile* stellt Methoden bereit, mit denen man auf die Daten des *File*-Objektes zugreifen kann. Mit *file.seek(i)* geht man an das *i*-te Byte der Datei und mit *file.readInt()* kann man von dem Byte der Datei, an dem man sich gerade befindet, den nächsten Integerwert (in Java 4 Byte lang) lesen.

Um auf diese Weise die Kontrollparameter auszulesen, benötigt man jedoch die Informationen, wie sie im LSM-Format gespeichert sind. Diese wurden aus der Beschreibung des LSM-Fileformates in [12] entnommen.

In dieser sind für alle Parameter charakteristische Bytefolgen (wie die im obigen Codebeispiel) beschrieben, mit deren Hilfe man sich durch die Datei zu den gesuchten Parametern navigieren kann.

**MOSAIC Background Subtractor** Beim *MOSAIC Background Subtractor* sind die notwendigen Funktionen, um den Hintergrund zu erhalten, in der Klasse *BackgroundSubtractor2\_* implementiert.

Da in dieser Klasse einige benötigten Variablen nicht von außerhalb zugreifbar waren, wurde aus ihr eine Klasse *OwnBackgroundsubtractor* erstellt, die identisch mit dem Original ist außer, dass die benötigten Variablen öffentlich verfügbar gemacht wurden.

Anschließend wurde eine eigene Klasse *FNTD\_Background* durch Vererbung von *OwnBackgroundsubtractor* erzeugt, damit in dieser auf die Methoden und Variablen von *OwnBackgroundsubtractor* zugegriffen werden kann.

In *FNTD\_Background* konnte damit die Methode *getBackgroundPixel* erstellt werden, die alles nötige macht, um den Hintergrund von R aus zu erzeugen.

Sie erstellt zunächst aus einem Feld mit Pixelwerten und den Informationen über die Länge, Breite und Anzahl der Bilder ein *ImagePlus*-Objekt. Da gerade diese Informationen von Methoden aus *FNTD\_LSMFileReader* an R geliefert werden, können sie ohne weiteres von R aus an die Methode übergeben werden.

Im zweiten Schritt wird die Methode *SubtractBackground* von *OwnBackgroundsubtractor* aufgerufen, welche das erstellte *ImagePlus*-Objekt und die Seitenlänge des sliding window L als Parameter benötigt. Sie führt den Algorithmus des Background Subtractors aus und speichert das Hintergrundbild intern ab.

Durch die Vererbung kann auf dieses zugegriffen werden und analog wie beim *LSM Reader* die Pixelwerte als Feld zurückgegeben werden. (Da sich Länge, Breite und Anzahl der Bilder nicht geändert haben, reicht diese Information, um in R das Hintergrundbild rekonstruieren zu können.)

**MOSAIC Particle Tracker** Der *MOSAIC Particle Tracker* ist das komplizierteste Plugin der hier behandelten. Hier werden beim Aufruf des Plugins Methoden vieler Klassen genutzt. Deshalb war der Ansatz, so viel wie möglich vom originalen Code zu über-

nehmen, um mögliche Fehlerquellen durch nicht bedachte Abhängigkeiten von anderen Klassen klein zu halten.

Die Klasse des Particle Trackers, die auf oberste Ebene aufgerufen wird, ist *ParticleTracker3DModular\_*. Deshalb wurde diese Klasse in eine eigene Klasse *FNTD\_ParticleDetector* übernommen, in der die Methoden zum Aufruf in R erstellt wurden. Diese sind:

- *getParticlePositions*
- *getParticlePositionsSilent*

Die erste Methode soll einen Dialog öffnen, in dem das zu bearbeitende Bild gezeigt wird und der Nutzer die Particle Tracker Parameter aussuchen, an dem Bild testen und zum Schluss die gewünschten detektierten Positionen abspeichern kann. Diese werden anschließend von der Methode zurückgegeben.

In der zweiten sollen beim Aufruf die Particle Tracker Parameter direkt übergeben und die damit bestimmten Positionen automatisch zurückgegeben werden.

Diese Methoden erzeugen analog zu *getBackgroundPixel* zu Beginn ein *ImagePlus*-Objekt. Dann durchlaufen sie die Methoden von *FNTD\_ParticleDetector*, die bei der ganz normalen Benutzung in ImageJ ebenfalls aufgerufen werden. Diese wurden jedoch so abgekürzt, dass sie, sobald die Teilchenpositionen detektiert wurden, diese zurückliefern und den normalen Ablauf abbrechen.

Die so erstellten Klassen wurden zum Schluss in Eclipse gemeinsam mit den benötigten Klassen aus dem ImageJ-, MOSAIC- und LSM Reader-Projekt zu einem Paket, *FNTDJavaMethods.jar* zusammengeschnürt, welches in R verwendet werden kann.

### 3.1.2. Einbindung in R

Im Folgenden wird anhand eines Codebeispielles gezeigt, wie man in R das erstellte Java-Paket einbinden, um die Methoden der darin beinhalteten Klassen zu benutzen:

Beispiel zum Aufrufen der Java Methoden in R

```

1 library(rJava)
2 .jinit()
3
4 .jaddClassPath("C:\\Users\\FNTDJavaMethods.jar")
5 Reader <- .jnew("fntd/lsm/FNTD_LSMFileReader")
6
7 directory <- "C:\\Users\\Sg35041_c_spect.stack.lsm"
8
9 pixelarray <- .jcall(Reader, "[T", "getFileData", directory)
10 metadaten <- .jcall(Reader, "[S", "getMetaData", directory)
11 sizes <- .jcall(Reader, "[I", "getSize", directory)

```



In Zeile 1 wird das Paket rJava geladen, um dessen Funktionen verfügbar zu machen. In Zeile 2 wird mit dem rJava-Befehl `.jinit()` die virtuelle Maschine von Java für R initialisiert. Diese dient als Schnittstelle zum PC und Betriebssystem und führt den Java-Code aus.

In Zeile 4 wird mittels des rJava-Befehls `.jaddClassPath` der Pfad zu dem erstellten Java-Paket, `FNTDJavaMethods.jar`, in den *Klassenpfad* des Betriebssystems geladen. Dort würd beim Ausführen des Java-Codes nach den benötigten Klassen gesucht.

Nun kann in Zeile 5 ein Java-Objekt in R vom Typ der Klasse `FNTD_LSMFileReader` erzeugt werden. Der dafür zuständige rJava-Befehl `.jnew` benötigt dafür den Pfad zu der Klasse innerhalb des Paketes.

In dieser Klasse wurden, wie in Abschnitt 3.1.1 beschrieben, die Methoden implementiert, um Informationen aus einem LSM-File zu lesen. Als Beispiel wird in Zeile 7 der Pfad zu dem LSM-File `Sg35041_c_spect.stack.lsm` in eine Variable geladen.

In den Zeilen 9 bis 11 werden nun die Methoden aufgerufen, um die Bilddaten aus dem LSM-Files zu lesen, und diese anschließend in Variablen gespeichert. Der dafür verwendete rJava-Befehl `.jcall` benötigt als erstes Argument ein Java-Objekt, dessen Methoden aufgerufen werden sollen.

Als zweites Argument wird ein Zeichen benötigt, das den Rückgabotyp der Methode identifiziert. „T“ steht hierbei für *short* (Ganzzahltyp der Größe 2 Byte), „S“ für *String* (Zeichenkette) und „I“ für *integer* (Ganzzahltyp der Größe 4 Byte). Ein „[“ vor dem Zeichen bedeutet, dass ein Feld vom nachfolgenden Rückgabotyp ausgegeben wird.

Das dritte Argument benennt die Methode, die aufgerufen werden soll. Die Funktionsweise der Methoden in diesem Beispiel wurde in Abschnitt 3.1.1 beschrieben.

Alle weiteren Argumente stehen für die Parameter, die die aufgerufenen Java-Methoden benötigen. In diesem Fall ist das für alle drei Methoden nur der Pfad zum Beispielfeld. Insgesamt wurden in diesem Beispiel also alle Informationen in R zur Verfügung gestellt, um den Bilderstapel aus dem LSM-File zu rekonstruieren.

Das Vorgehen bei den Methoden zur Hintergrunderzeugung und der Teilchendetektion ist analog.

### 3.1.3. Bereitgestellte Werkzeuge

Die so in R eingebundenen Java-Methoden wurden als nächstes in handlichere R-Funktionen eingebaut und diese in das FNTD-Paket integriert.

Dadurch wurde es um die folgenden drei Funktionen erweitert.

- `read.lsm.image (filename, first.frame = NULL, last.frame = NULL)` : Diese Funktion benötigt einen String `filename` als Eingabeparameter, der den Dateipfad zu einem LSM-Bild enthält, und gibt dieses in R als `FNTDimage` aus. (Wahlweise können mit zwei Ganzzahlwerten `first.frame` und `last.frame` im Falle eines Bilderstapels nur Teilstapel eingelesen werden.)
- `get.background (img, length.parameter = 50)` : Diese Funktion benötigt ein Objekt `img` vom Typ eines `FNTDimages` und einen Ganzzahlwert `length.parameter` als

Seitenlänge des sliding window  $L$ , der bei fehlender Eingabe standardmäßig 50 gesetzt wird.

Sie liefert ein `FNTDimage`, das das von `img` erhaltene Hintergrundbild enthält, zurück.

- `find.tracks (img, radius = NULL, cutoff = NULL, percentile = NULL)`: Mit dieser Funktion erhält man bei Eingabe eines `FNTDimage`s `img` die Positionen der vom Algorithmus detektierten Teilchen.

Bei Eingabe der Particle Tracker Parameter `radius`, `cutoff` und `percentile` geschieht dies automatisch. Ansonsten öffnet sich ein Dialog samt Bild, an dem man durch Ausprobieren die Parameter bestimmen kann.

## 3.2. Überprüfung

Diese erste Reihe von Tests dient zur Überprüfung, ob die erstellten Werkzeuge die gleichen Ergebnisse liefern wie die zugrundeliegenden originalen Werkzeuge und damit korrekt funktionieren. Darüber hinaus werden qualitative Untersuchungen des Einflusses der mathematischen Parameter dieser Funktionen geführt.

### 3.2.1. Einlesen von LSM-Files

Die zum Einlesen von LSM-Files erstellte Funktion `read.lsm.image` wird auf die Bilddaten des Datensatzes angewandt. Dabei werden aber nur einzelne Bilder der Bilderstapel der verschiedenen Proben eingelesen.

In Abbildung 3.1 wird links ein Beispiel eines in R erhaltenen Bildes gezeigt. Dieses zeigt das Bild eines FNTDs der Probe (1H, 3 MeV/u). Es wurde ein mittleres Bild des Datenstapels, in diesem Fall das 14. Bild ausgewählt. Um zu überprüfen, ob `read.lsm.image` richtig funktioniert, ist zum Vergleich in Abbildung 3.1 rechts das entsprechende Bild mit dem `LSM Reader`- Plugin von ImageJ gezeigt.

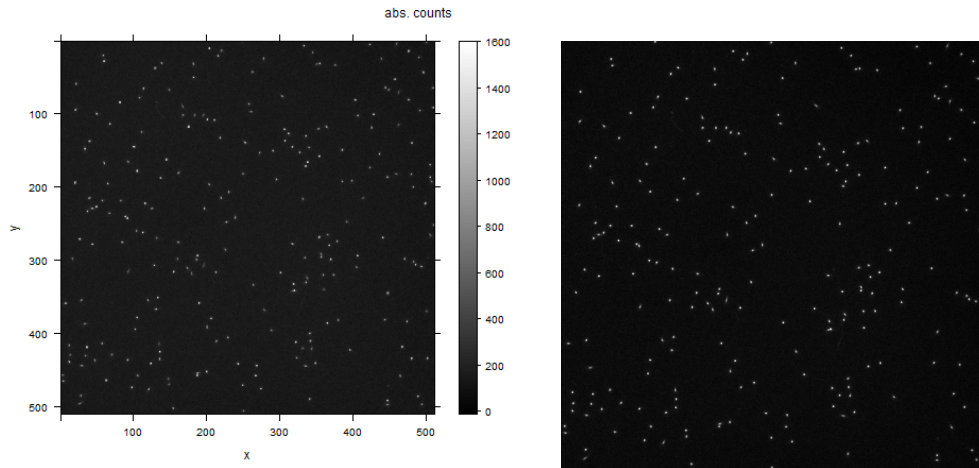


Abbildung 3.1.: Vergleich zwischen dem Ergebnis von `read.lsm.image` (links) und dem originalen Bild aus ImageJ, das gedreht und gespiegelt wurde (rechts) für  $1H, 3\text{ MeV/u}$ , Bild 14

Die x- und y-Achse des von `read.lsm.image` erhaltenen Bildes enthält die Position in Pixel in der entsprechenden Richtung und jeder Pixel des Bildes enthält als Zahlenwert die Countzahl an dieser Stelle. Die Kodierung der Countzahlen in Graustufen wird am Balken auf der rechten Seite gezeigt.

Die hellen sternartigen Punkte entsprechen hohen Countzahlen und damit hohen Energiedepositionen von Teilchen im Detektor. Da die Ionen das Bild, wie in Abschnitt 2.6 erwähnt, orthogonal passiert haben, ist dieses Muster als Querschnitt der Ionenbahnen (*Tracks*) durch den Detektor zu verstehen.

Beim Vergleich mit dem ImageJ-Bild fällt auf, dass dieses zunächst um  $90^\circ$  nach links gedreht und anschließend um die horizontale Achse gespiegelt werden muss, damit beide Bilder das gleiche Muster zeigen. Vermutlich unterscheiden sich die Darstellungsweisen der Bilder in R und ImageJ. Ein Vergleich der Pixelwerte der zugrundeliegenden Datenobjekte zeigt jedoch, dass diese identisch sind und die Funktion damit korrekt funktioniert. In den Bildern sind leichte Unterschiede in der Helligkeit zu sehen, die aber durch unterschiedliche Graustufenkodierungen der Countzahlen verursacht wird und damit nur beim Betrachten eine Rolle spielt.

Um einen Überblick über die erhaltenen Bildern aus den unterschiedlichen Proben zu bekommen, zeigt Abbildung A.1 im Anhang ein Bild aus jeder Probe des Datensatzes. Dabei wurde jeweils ein mittleres Bild des Bilderstapels gewählt. Da im Folgenden öfters nur ein mittleres Bild betrachtet wird, wurde in Tabelle A.1 für jede Probe in der Spalte „Mittl. Bild“ eingetragen welches Bild dabei verwendet wird.

Bei diesen Bildern muss man jedoch berücksichtigen, dass sie nicht mit einheitlichen Kontrollparametern aufgenommen wurden, weshalb man sie nicht ohne weiteres vergleichen kann.

Deshalb wurden in Tabelle A.2 alle Metadaten und weitere charakteristische Größen wie die mittlere Countzahl etc. für diese Bilder zusammengestellt.

Ein Vergleich dieser Bilder mit den entsprechenden aus ImageJ hat für den ganzen Da-

tensatz bestätigt, dass *read.lsm.image* korrekt funktioniert.

### 3.2.2. Automatische Erzeugung von Hintergründen

Die Funktion *get.background* wird probeweise auf Bilder angewandt, die mit *read.lsm.image* eingelesen wurden. Dabei wird die Seitenlänge des sliding windows  $L$  von 2 bis 200 variiert um einen qualitativen Eindruck des Einflusses von  $L$  zu gewinnen.

Ein mit *get.background* erzeugtes Hintergrundbild ist in Abbildung 3.2 links dargestellt. Dieses Bild stellt den Hintergrund von Abbildung 3.1 mit der Seitenlänge des sliding window  $L = 20$  dar. Zum Vergleich ist diesem das entsprechende Resultat des *MOSAIC Background Subtractors* gegenübergestellt.

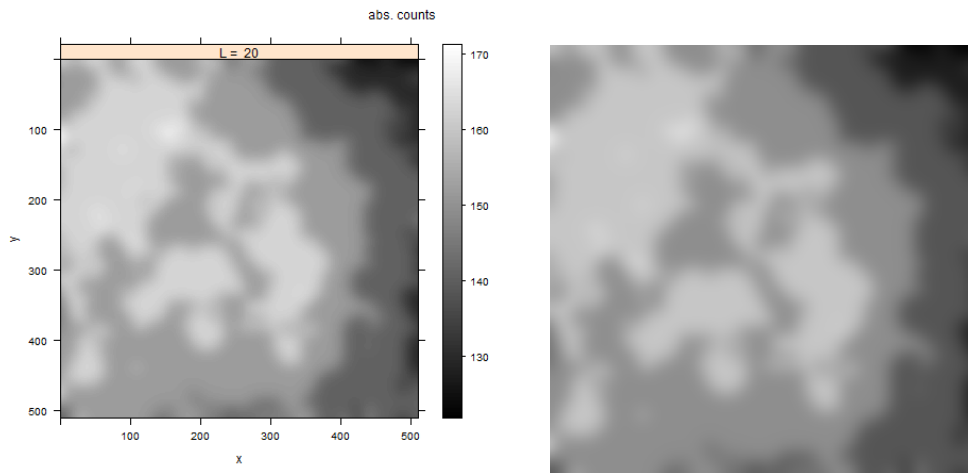


Abbildung 3.2.: Vergleich zwischen dem Hintergrund von *get.background* (links) und dem des *Background Subtractors* (rechts) für  $1H, 3\text{ MeV/u}$ , Bild 14

Bei den mit *get.background* erhaltenen Hintergrundbild kann am Balken auf der rechten Seite erkennen, dass die Countzahlen in diesem Bild in einem viel geringeren Bereich variieren, als im Ausgangsbild in Abbildung 3.1. Außerdem sieht man, dass die *Tracks* verschwunden sind. Dies stimmt mit den Erwartungen an ein Hintergrundbild überein. Dennoch sieht man immernoch gröbere Strukturen im Bild. Darunter fallen besonders hellere Flecken in der Mitte des Bildes auf und eine allgemeine Tendenz des Hintergrundes zu den Rändern hin dunkler zu werden.

Der Vergleich mit dem Resultat des *Background Subtractors* zeigt wieder, dass die Bilder identisch sind. Die leichten Unterschiede in der Helligkeit entstehen wieder durch die unterschiedlichen Graustufenkodierungen der Countzahlen. In Abbildung A.-1 im Anhang wurden für alle Proben des Datensatzes jeweils der Hintergrund des mittleren Bildes mit  $L=10$  zusammengestellt. Ein Vergleich mit den entsprechenden Hintergrundbildern des *Background Subtractors* bestätigt, dass die Funktion korrekt funktioniert. Außerdem zeigt dieser Überblick, dass die Hintergrundbilder starke Unterschiede von Größe und Variation für die verschiedenen Proben aufweisen.

Um einen Überblick des Verhaltens der erzeugten Hintergrundbilder für verschiedene L-Werte zu bekommen, zeigt Abbildung 3.3 eine Reihe von Hintergrundbildern von Abbildung 3.1 für verschiedene L-Werte.

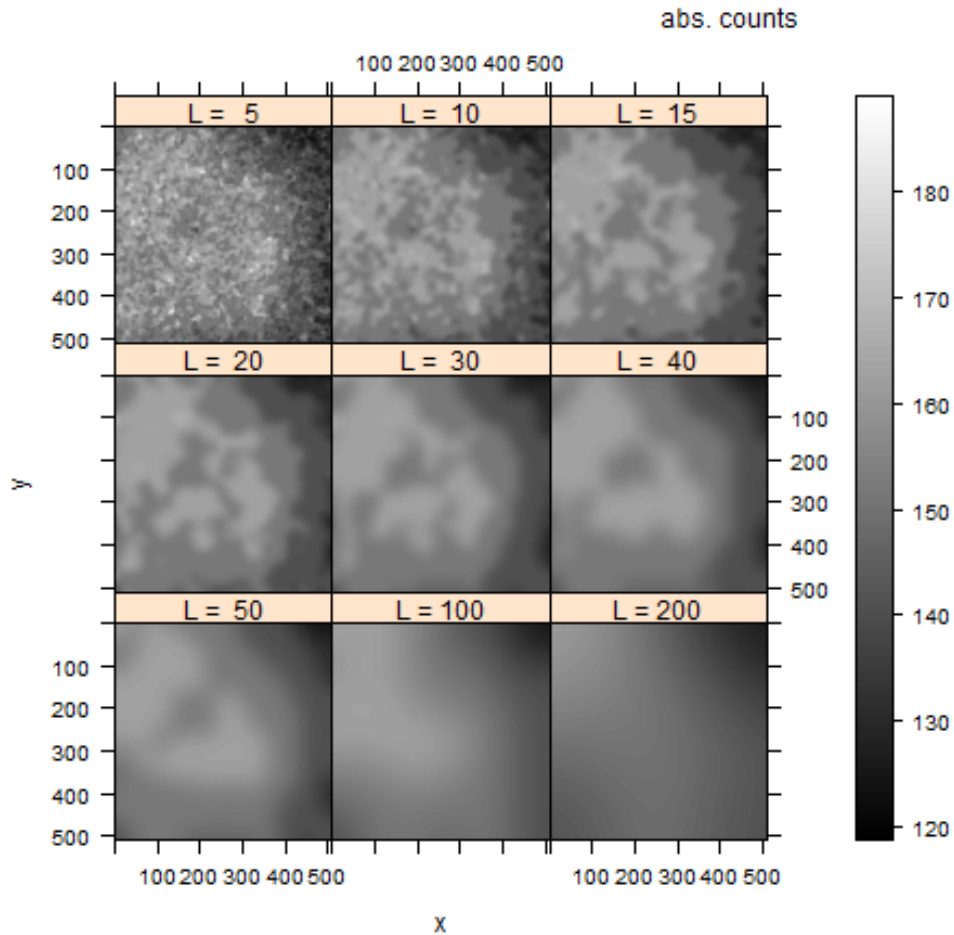


Abbildung 3.3.: Hintergründe für verschiedene L-Werte von  $1H$ ,  $3\text{ MeV/u}$ , Bild 14

Man kann erkennen, dass die bereits in Abbildung 3.1 gefundenen Strukturen mit zunehmenden L verschwinden und mit kleinerem L feiner werden. Außerdem bleibt die Tendenz zum Abnehmen des Hintergrundes zum Rand hin für alle L bestehen.

Wegen der Funktionsweise des zugrundeliegenden Algorithmus würde man bei Anwendung von *get.background* an einem Bild mit  $L=1$  erwarten, wieder das gleiche Bild zu erhalten. Die Funktion benötigt allerdings  $L \geq 2$  als Parameter. Daher wird in Abbildung 3.4 das Verhalten der Hintergründe für kleine  $L \geq 2$  dargestellt.

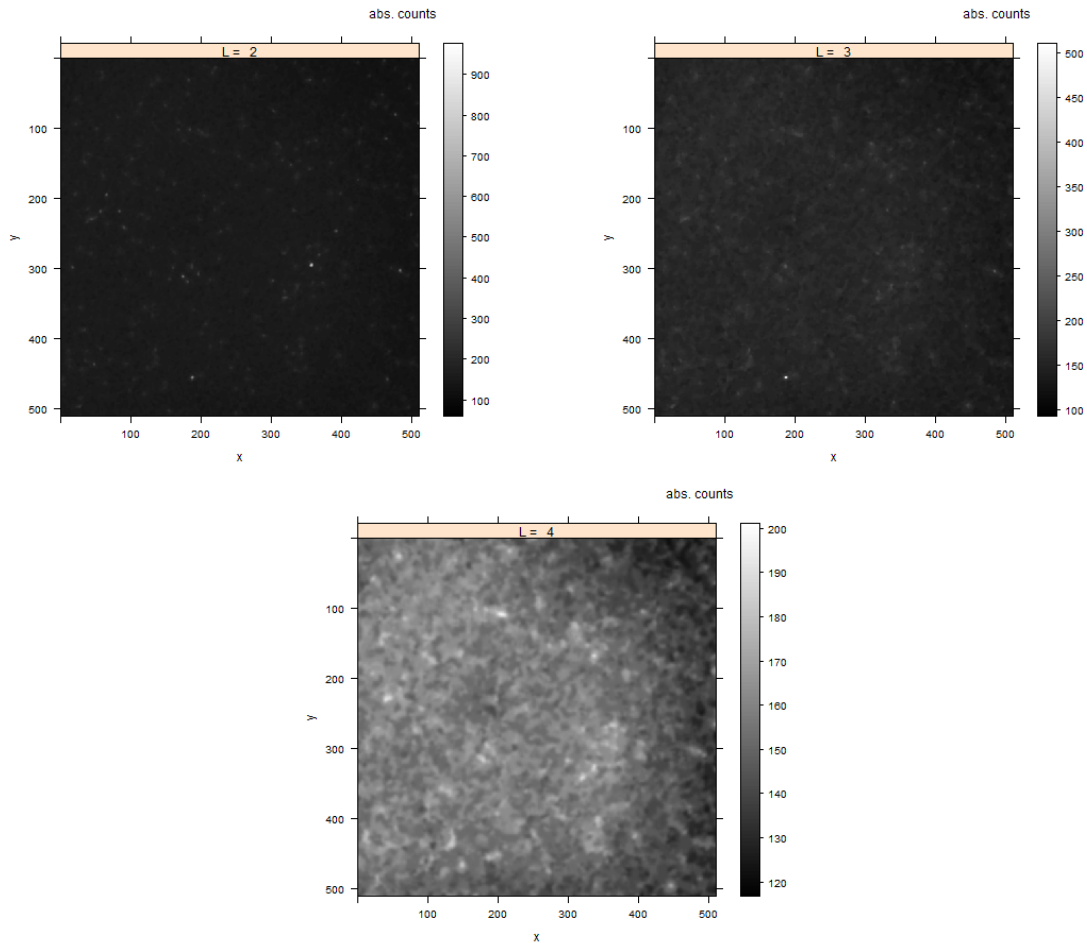


Abbildung 3.4.: Hintergründe für kleine  $L$ -Werte von  $1H$ ,  $3 \text{ MeV/u}$ , Bild 14

Diese Bilder zeigen, dass für kleine Werte von  $L$  eindeutig Reste der sternartigen Energiedepositionen im Hintergrundbild übrig bleiben.

Folglich kann man diese Bilder nicht wirklich als Hintergrundbilder betrachten.

Da das Verschwinden der Reste kontinuierlich ist, stellt sich die Frage ab welchem  $L$ -Wert die erzeugten Hintergrundbilder tatsächlich als wahr betrachtet werden können. Diese Frage wird in Kapitel 4 genauer untersucht.

Außerdem kann man in diesen Bildern bei  $L = 2$  und  $L = 3$  die Reste der kleinskaligen Variation als Rauschen im ganzen Bild erkennen. Bei  $L = 4$  ist dieses bereits in eine größere Struktur übergegangen. Folglich verschwindet ungefähr ab diesem  $L$  die kleinskalige Variation im automatischen Hintergrund.

### 3.2.3. Teilchenerkennung

In diesem Abschnitt wird *find.tracks* getestet und durch Variation der Parameter qualitativ deren Einfluss untersucht.

Ein Beispiel für durch *find.tracks* detektierte *Tracks* ist in Abbildung 3.5 links gegeben. Zum Vergleich ist rechts das entsprechende Ergebnis des Particle Trackers dargestellt. Bei beiden Bildern wurden die Standardparameter des Particle Trackers verwendet. Das zugehörige originale Bild war wieder Abbildung 3.1.

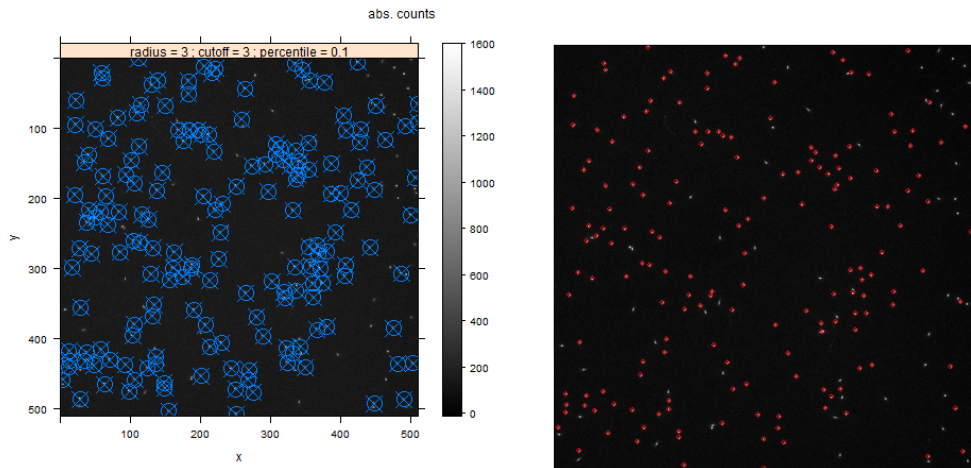


Abbildung 3.5.: Vergleich der detektierten Teilchen von *find.tracks* (links) und Particle Tracker (rechts) bei  $1H$ ,  $3\text{ MeV/u}$ , Bild 14

Die blauen Kreise im Bild links markieren die von *find.tracks* detektierten *Tracks*. Im rechten Bild sind es entsprechend die roten Kreise. Ein genauer Vergleich der beiden Bilder zeigt, dass exakt die gleichen *Tracks* markiert wurden. Das bestätigt, dass *find.tracks* korrekt funktioniert. Man kann erkennen, dass einige *Tracks* nicht detektiert wurden. Die Sensibilität der Trackererkennung wird durch die Particle Tracker Parameter im oberen Balken des linken Bildes bestimmt.

In Abbildung 3.6 wird die Abhängigkeit der Trackererkennung von den Particle Tracker Parameter illustriert. Dargestellt sind drei Beispiele von mit *find.tracks* detektierten Teilchen von Abbildung 3.1. Das Bild links oben zeigt ein Beispiel für zu sensible, das rechts oben für zu grobe und das in der Mitte unten für optimale Parameter.

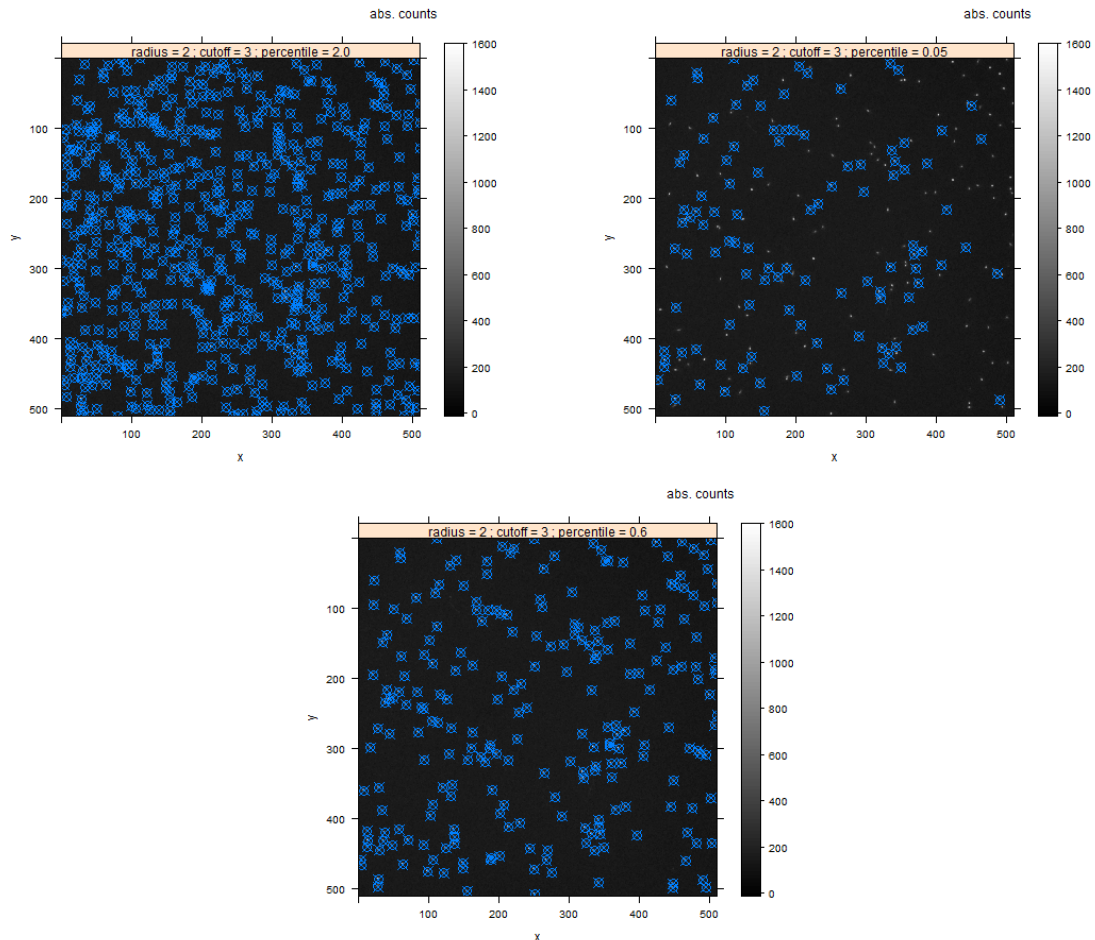


Abbildung 3.6.: *Einfluss der Particle Tracker Parameter für 1H, 3 MeV/u, Bild 14. Links oben sind zu sensible, rechts oben zu grobe und in der Mitte unten optimale Parameter dargestellt.*

Aus Abbildung 3.6 kann man auch schließen, dass die Wahl optimaler Parameter nicht eindeutig ist.

Sie hängt davon ab, welche Abweichungen der Countzahl man als Track ansieht. Damit gibt die Bestimmung der optimalen Parameter dem Experimentator Spielraum.

Da die optimalen Parameter von Probe zu Probe unterschiedlich sein können, wie in Abbildung 3.7 gezeigt, in dem die optimalen Parameter für (1H, 3 MeV) auf (12C, 4 MeV) angewandt wurden und dennoch Tracks undetektiert bleiben, müssen diese für jede Probe einzeln gefunden werden.

Dies wurde durch Variation verschiedener Parameter an dem mittleren Bild jeder Probe durchgeführt. Dabei wurde der Dialog-Modus von *find.tracks* bei Aufruf ohne Parameter verwendet. Die Ergebnisse wurden in Tabelle A.1 eingetragen.



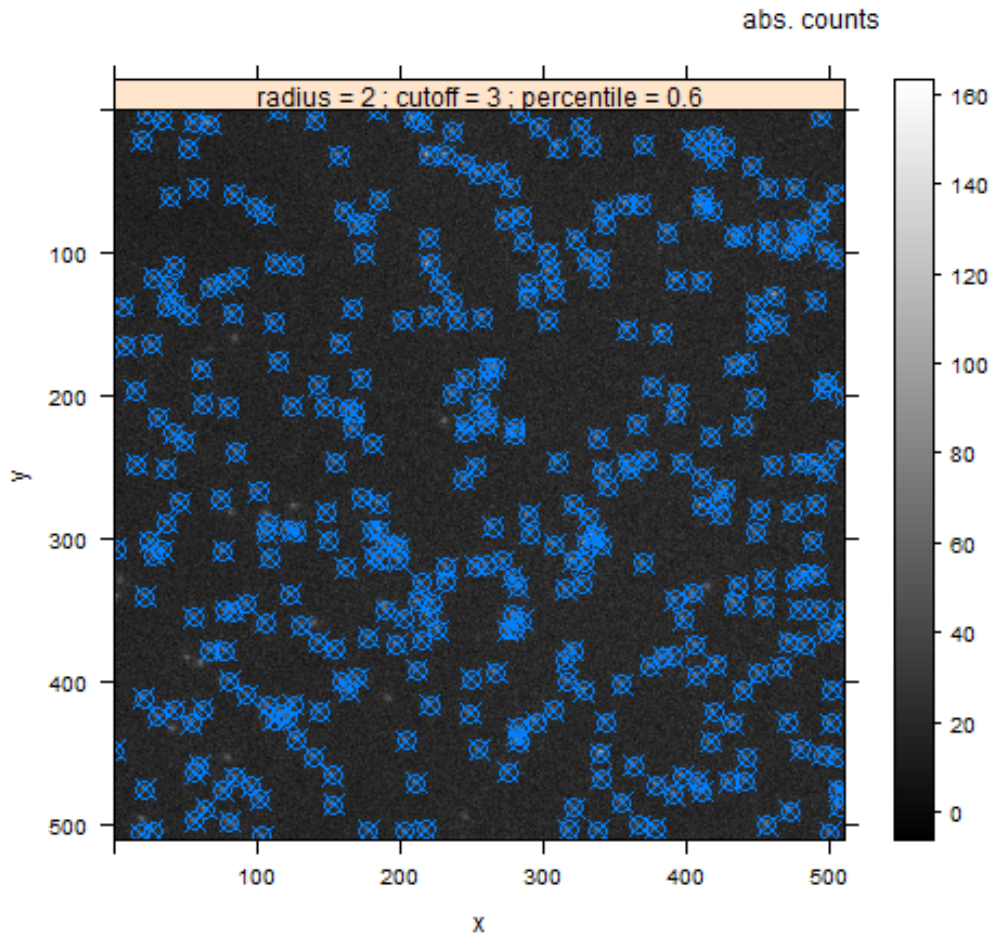


Abbildung 3.7.: Abweichung der optimalen Parameter bei ( $^{12}\text{C}$ , 4 MeV), Bild 6



## 4. Anwendung

In diesem Kapitel soll die erstellte Funktion *get.background* zur automatisierten Bestimmung der Hintergründe an den Spektroskopiedatensatz angewendet und optimale Werte für den L-Parameter der Funktion bestimmt werden.

Um die Qualität der damit bestimmten Hintergründe untersuchen zu können, wird in Abschnitt 4.1 zunächst ein Verfahren vorgestellt, in dem die Hintergründe durch Vergleich mit einem Referenzhintergrund überprüft werden können. Dieses Verfahren hängt jedoch von L und einem weiteren hierfür notwendigen Parameter ab.

In Abschnitt 4.2 wird deshalb untersucht, inwieweit es möglich ist einen optimalen Satz dieser Parameter zu finden und gleichzeitig ein Vorschlag für diesen gemacht. Dazu wird der Einfluss dieser Parameter auf die Ergebnisse des Prüfverfahrens untersucht.

Zum Schluss werden in Abschnitt 4.3 die Ergebnisse von *get.background* mit den vorher bestimmten optimalen Parametern untersucht, um damit Rückschlüsse auf die Qualität dieses Werkzeuges zu ziehen.

### 4.1. Ansatz zur Auswertung des Hintergrundes

Die Tests in Abschnitt 3.2 haben gezeigt, dass die Funktionsweise von *get.background* mit dem ursprünglichen Background Subtractor übereinstimmt. Um abzuschätzen wie gut die mit dem lokalen Histogrammverfahren des Background Subtractors *automatisch* erzeugten Hintergründe mit den wirklichen Hintergründen der ursprünglichen Bilder übereinstimmen, wird ein Referenzhintergrund benötigt.

Die Idee, um diesen Referenzhintergrund zu erhalten, ist mit der *find.tracks*-Funktion die Positionen der Energiedepositionen in einem Bild zu markieren und alle Pixel innerhalb eines Ausschneideradiuses  $R_{cut}$  um diese auszuschneiden. Aufgrund der eng begrenzten Energiedepositionen erhält man somit ein Bild, das nur die Bereiche enthält, in denen bei der Bestrahlung keine Energie deponiert wurde und die damit dem Hintergrund entsprechen.

Um dieses *manuell* erzeugte Hintergrundbild mit dem vorher erzeugten *automatischen* zu vergleichen, werden sie durcheinander geteilt, wobei das manuelle Hintergrundbild im Zähler steht. Das Teilen von Bildern ist dabei pixelweise zu verstehen.

Dies wird in Abbildung 4.1 dargestellt. Das entstandene Bild wird im Folgenden *Prüfbild* genannt.

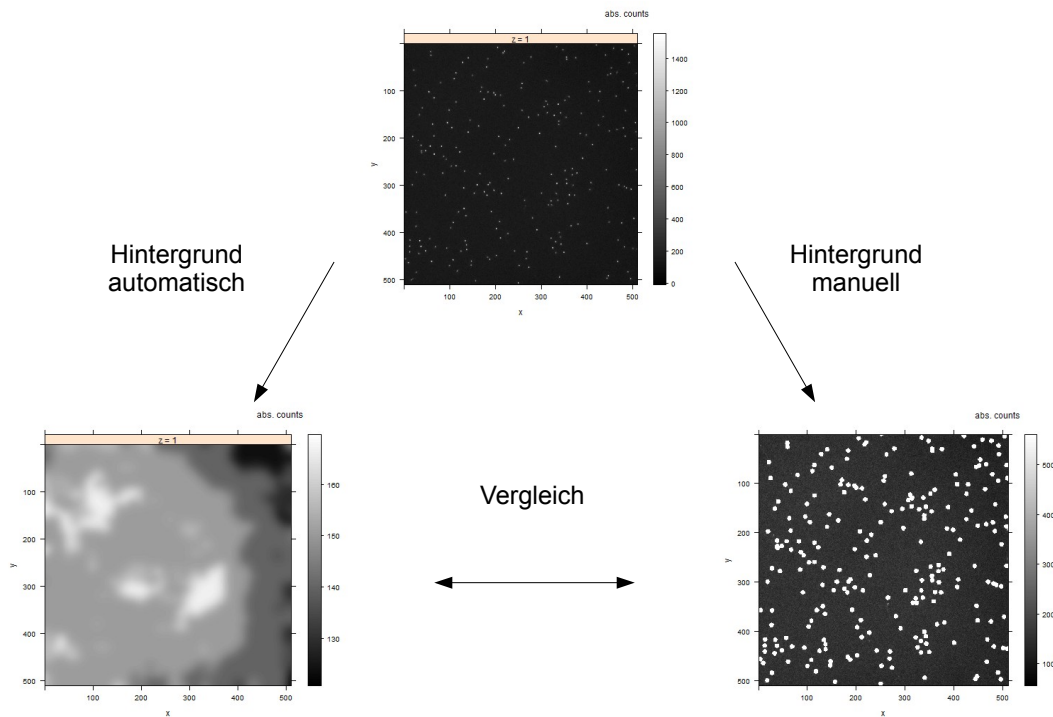


Abbildung 4.1.: Schema zur Auswertung der Hintergründe.

## Prüfwerte

Um quantitative Größen zu bekommen, die beschreiben wie gut der automatisch erzeugte Hintergrund mit dem manuellen übereinstimmt, werden im Folgenden zwei Prüfwerte definiert.

Seien  $\mu(x)$  der Mittelwert einer Größe  $x$ ,  $H_{man}$  das manuelle Hintergrundbild und  $H_{auto}$  das automatische Hintergrundbild. Dann sind die beiden Prüfwerte wie folgt definiert:

1. Mittelwert des Prüfbildes  $\mu_p$ :

$$\mu_p = \mu \left( \frac{H_{man}}{H_{auto}} \right)$$

2. Verhältnis der Mittelwerte  $f_p$ :

$$f_p = \frac{\mu(H_{man})}{\mu(H_{auto})}$$

Idealerweise wären diese Werte beide 1. Je weiter sie von 1 abweichen, desto stärker unterscheiden sich die beiden Hintergrundbilder.

Da  $H_{man}/H_{auto}$  pixelweise zu verstehen ist, ergibt sich bei  $\mu_p$  das Problem, dass dieser nicht mehr definiert ist, sobald mindestens ein Pixel von  $H_{auto}$  den Wert 0 besitzt. Problematisch bei diesem Ansatz ist zudem, dass räumlich begrenzte starke Abweichungen durch die Mittelung im ganzen Bild abgeschwächt oder durch andere Abweichungen in entgegengesetzter Richtung ausgeglichen werden können und dadurch übersehen werden. In solchen Fällen müsste dieses Verfahren auf Teilbilder des ursprünglichen Bildes angewandt werden.

## 4.2. Bestimmung des optimalen Parameters L

In diesem Abschnitt wird der Einfluss der Parameter, die für den in Abschnitt 4.1 beschriebenen Ansatz zur Auswertung der Hintergrundbilder notwendig sind, untersucht. Das Ziel hiervon ist es, zu klären welcher Parameter L optimal für eine Untersuchung der Qualität der erzeugten Hintergründe ist und diesen für die Proben des Spektroskopiedatensatzes zu bestimmen.

### 4.2.1. Einfluss des Ausschneideradius

Wie in Abbildung A.1 gesehen, besitzen die Energiedepositionen variierende aber stets räumlich sehr begrenzte Größen. Um den in Abschnitt 4.1 beschriebenen manuellen Hintergrund korrekt zu bestimmen, muss der Ausschneideradius  $R_{cut}$  so groß gewählt werden, dass der komplette Bereich der Energiedepositionen herausgeschnitten wird. Um zu bestimmen, wann dies der Fall ist, wird für den gesamten Datensatz von dem mittleren Bild der manuelle Hintergrund gebildet, wobei  $R_{cut}$  variiert wird. Von den so erzeugten Bildern werden anschließend Mittelwert, Maximum und Standardabweichung aller Pixelwerte untersucht.

Da die Bereiche der Energiedepositionen stark erhöhte Countzahlen im Vergleich zu den restlichen Teilen der Bilder darstellen, müsste man an allen drei Größen erkennen können, ob die Energiedepositionen noch im Bild vorhanden sind oder bereits komplett hereausgeschnitten wurden.

Der Mittelwert müsste beim Herausschneiden der Energiedepositionen stark abnehmen und anschließend für große  $R_{cut}$ -Werte ungefähr konstant bleiben. Beim Maximum würde man das gleich Verhalten erwarten, da wenn alle Energiedepositionen im Bild komplett herausgeschnitten werden, das Maximum auf die Größenordnung der Hintergrund-Countzahlen fällt. Die Standardabweichung müsste sich ebenfalls gleich verhalten, da die Energiedepositionen große Abweichungen von den mehrheitlich vorkommenden Hintergrundwerten darstellen.

Abbildungen 4.2 bis 4.4 zeigen die Ergebnisse dieser Untersuchungen für verschiedene  $R_{cut}$ .

Bei den Mittelwerten kann man erkennen, dass diese bei den meisten Proben mit zunehmenden  $R_{cut}$  abnehmen.

Dieser Abfall ist zu Beginn meist stark und wird anschließend schwächer. Insgesamt be-

trägt er nur einige Counts. Bei den Bildern mit hohen Countzahlen ist der Abfall damit relativ gering, bei denen mit kleinen kann er dagegen fast 50% betragen wie bei (131Xe, 9 MeV).

Der Grund für den starken Abfall zu Beginn liegt darin, dass bei kleinen  $R_{cut}$  die Tracks noch nicht komplett herausgeschnitten werden und deren Reste den Hintergrund im Mittel immer noch erhöhen. Sobald die Tracks komplett ausgeschnitten werden, gibt es keine hohen Energiedepositionen mehr, die bei zunehmenden  $R_{cut}$  verschwinden. Theoretisch würde man jedoch erwarten, dass die Mittelwerte sich nach dem starken Abfall für höhere  $R_{cut}$  nicht mehr ändern, weil irgendwann nur noch der Hintergrund des Bildes übrig sein müsste. Tatsächlich ist dies nie der Fall. In vielen Proben sieht man bei hohen  $R_{cut}$  zwar einen sehr schwachen Abfall. Jedoch nimmt in den Proben (1H, 6 MeV), (1H, 12 MeV), (1H, 50 MeV), (1H, 220 MeV) der Abfall sogar zu.

Die Proben (1H, 21 MeV), (12C, 430 MeV), (32S, 3 MeV) zeigen bei hohen  $R_{cut}$  wieder einen Anstieg. In Abbildung 4.5 werden für diese drei möglichen Verläufe jeweils zwei Beispiele bei  $R_{cut} = 30$  gezeigt. Hier kann man mit bloßem Auge keine klaren Unterschiede zwischen diesen Proben erkennen. Jedoch sieht man, dass bei einigen Proben durch das Herausschneiden der Tracks mit großem Radius kaum noch etwas vom eigentlichen Bild übrig bleibt. Dies könnte ein Grund für die verschiedenen Verläufe sein, da die Mittelwerte nur über einen kleinen Teil des Bildes gebildet werden und damit mögliche Außreißer einen deutlich stärkeren Einfluss haben.

Daher kann man aus diesen Bildern folgern, dass die  $R_{cut}$ -Werte nicht zu groß gewählt werden dürfen um noch sinnvolle Statistik mit den Bildern betreiben zu können. Da man aufgrund dieses Verhaltens aus den meisten Bildern keinen klaren Punkt erkennen kann, ab wann der steile Abfall der Mittelwerte aufhört und die Tracks damit komplett herausgeschnitten werden, ist dieser Plot ungeeignet um optimale  $R_{cut}$ -Werte zu bestimmen.

Bei den Maxima zeigt sich am Anfang meist ein starker Abfall und anschließend ein nahezu konstantes Verhalten.

An dem Übergang zum konstanten Verhalten sieht man gut, für welche  $R_{cut}$  die Tracks komplett herausgeschnitten werden. Theoretisch ist dieser Plot also gut geeignet um gute Werte für  $R_{cut}$  für die verschiedenen Proben abzuschätzen.

An der Probe (32S, 3 MeV) sieht man jedoch ein Problem dabei. Hier ist das Maximum konstant. Das liegt daran, dass von *find.tracks* nicht alle Tracks detektiert wurden und ein undetektierter Track für alle  $R_{cut}$  das Maximum darstellt.

Bei der Probe (12C, 4 MeV) ist das gleiche für ein Track mit einer niedrigeren Countzahl von ca. 60 passiert, wie man an dem Plateau erkennt. Dieses wurde jedoch bei größeren  $R_{cut}$  von einem detektierten Nachbartrack aus ausgeschnitten.

Damit eignet sich dieser Plot nur für den Fall, dass alle Tracks vom Particle Tracker bestimmt werden, für die Bestimmung optimaler  $R_{cut}$ -Werte. Dies kann jedoch nicht immer gewährleistet werden.

Bei den Standardabweichungen zeigt sich in den meisten Fällen ein ähnlicher Verlauf wie bei den Maxima.

Das liegt daran, dass durch das Ausschneiden der Tracks weniger Pixelwerte vorkommen, die stark von dem Hintergrund und damit von den meisten anderen Pixelwerten abweichen.

Das die Standardabweichung dennoch nie gegen 0 geht, liegt an den kleinskaligen Hintergrundschwankungen.

In manchen Proben steigen die Standardabweichung für größere  $R_{cut}$  wieder an. Das kann vernachlässigt werden, da wie bereits in Abbildung 4.5 festgestellt wurde, bei diesen Werten der größte Teil des Bildes ausgeschnitten wird und dadurch nichtdetektierte Tracks einen stärkeren Einfluss auf die Standardabweichung haben.

Damit eignet sich dieser Plot gut für die Bestimmung der  $R_{cut}$ -Werte für die verschiedenen Proben. Um zu berücksichtigen, dass die Tracks komplett herausgeschnitten werden und gleichzeitig so wenig wie möglich vom manuellen Hintergrund verloren geht, wird der  $R_{cut}$ -Wert als der „Optimale“ angesehen, bei dem der Übergang des starken Abfalls der Standardabweichungen in den konstanten Bereich stattfindet.

Die Ergebnisse hierfür wurden in Tabelle A.1 in der „ $R_{cut}$ “-Spalte eingetragen.

Die nachfolgenden Untersuchungen werden stets mit diesen  $R_{cut}$ -Werten geführt.

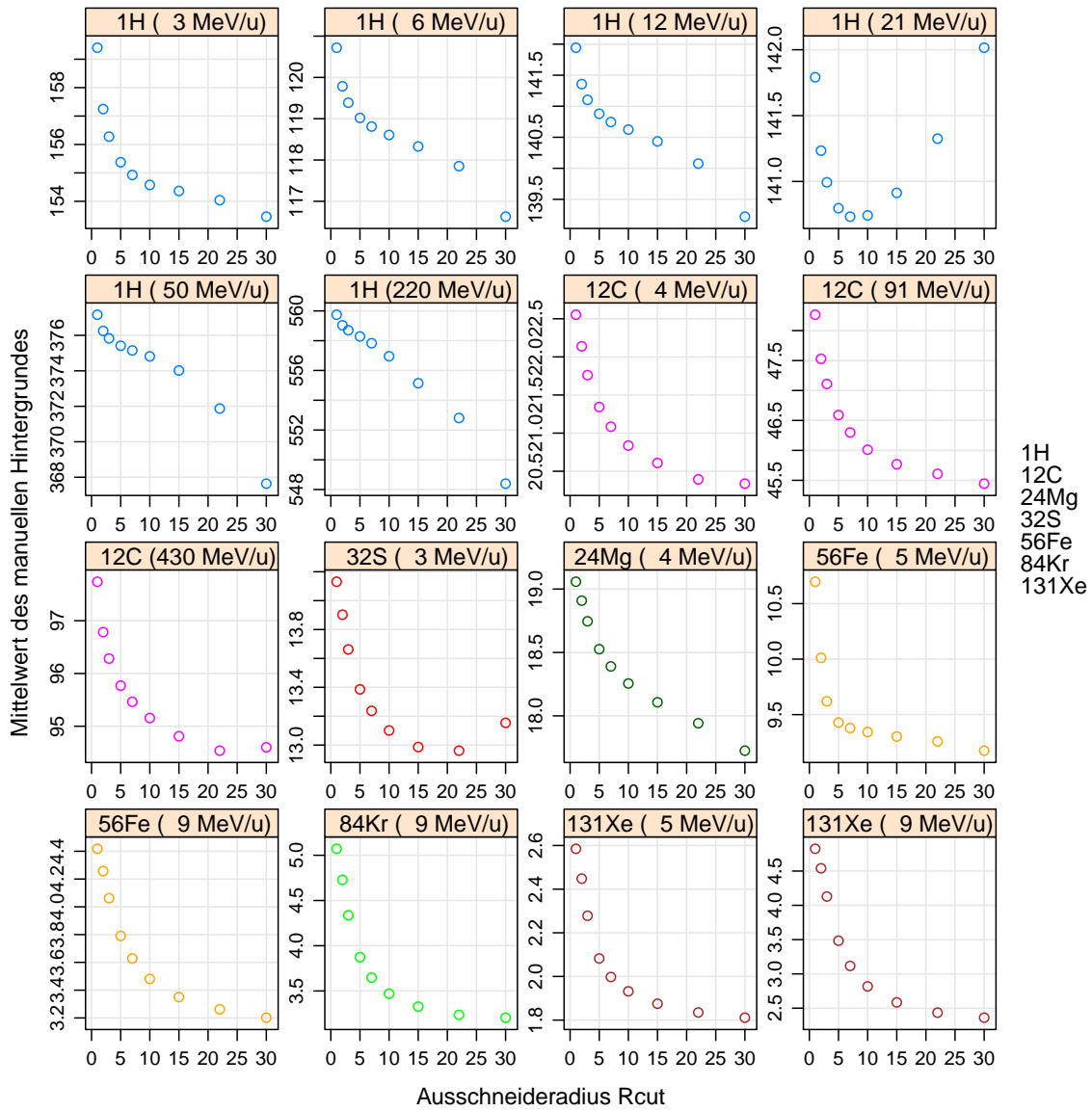


Abbildung 4.2.: Mittelwerte der manuellen Hintergründe über  $R_{cut}$



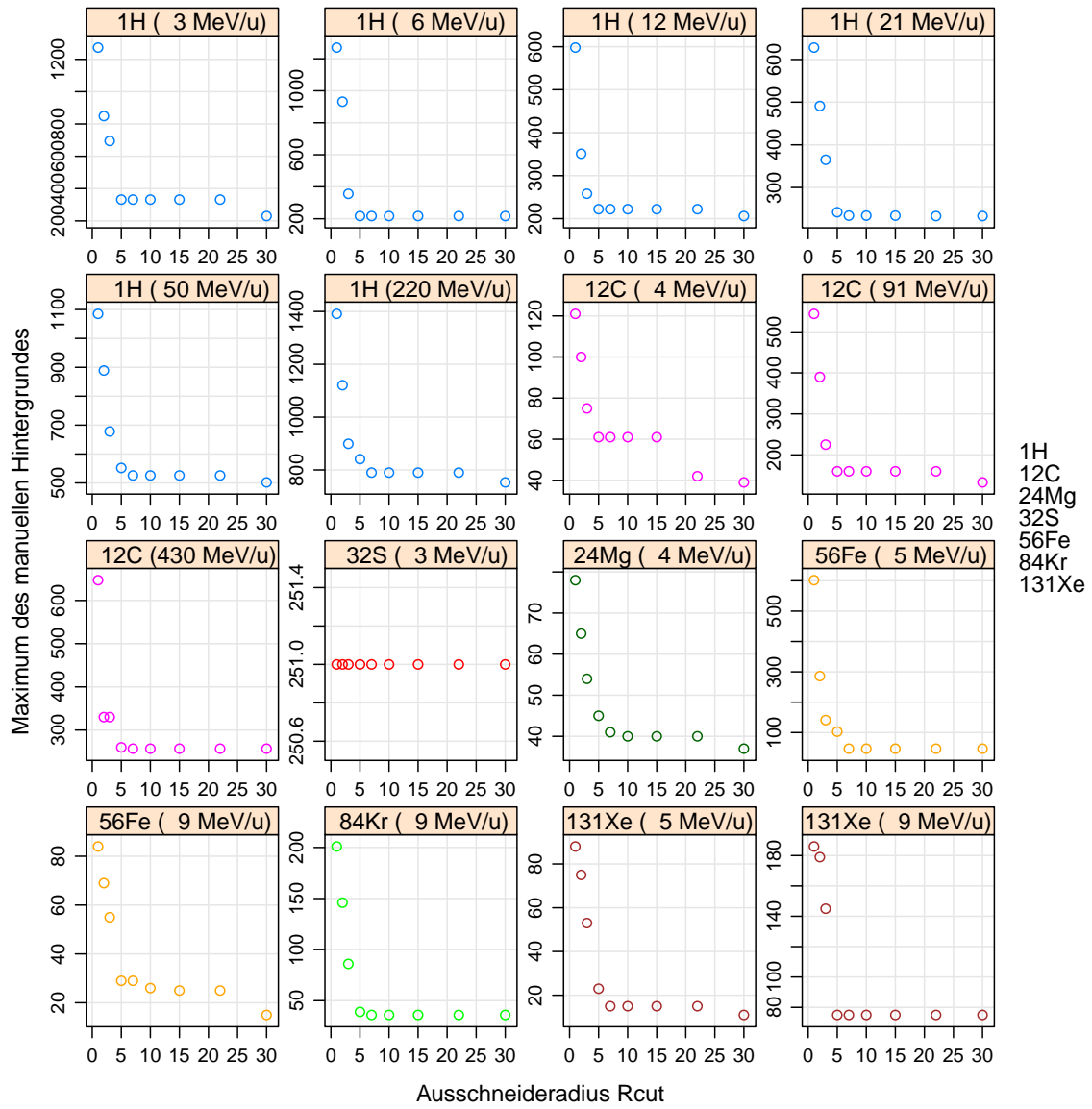


Abbildung 4.3.: Maxima der manuellen Hintergründe über  $R_{cut}$

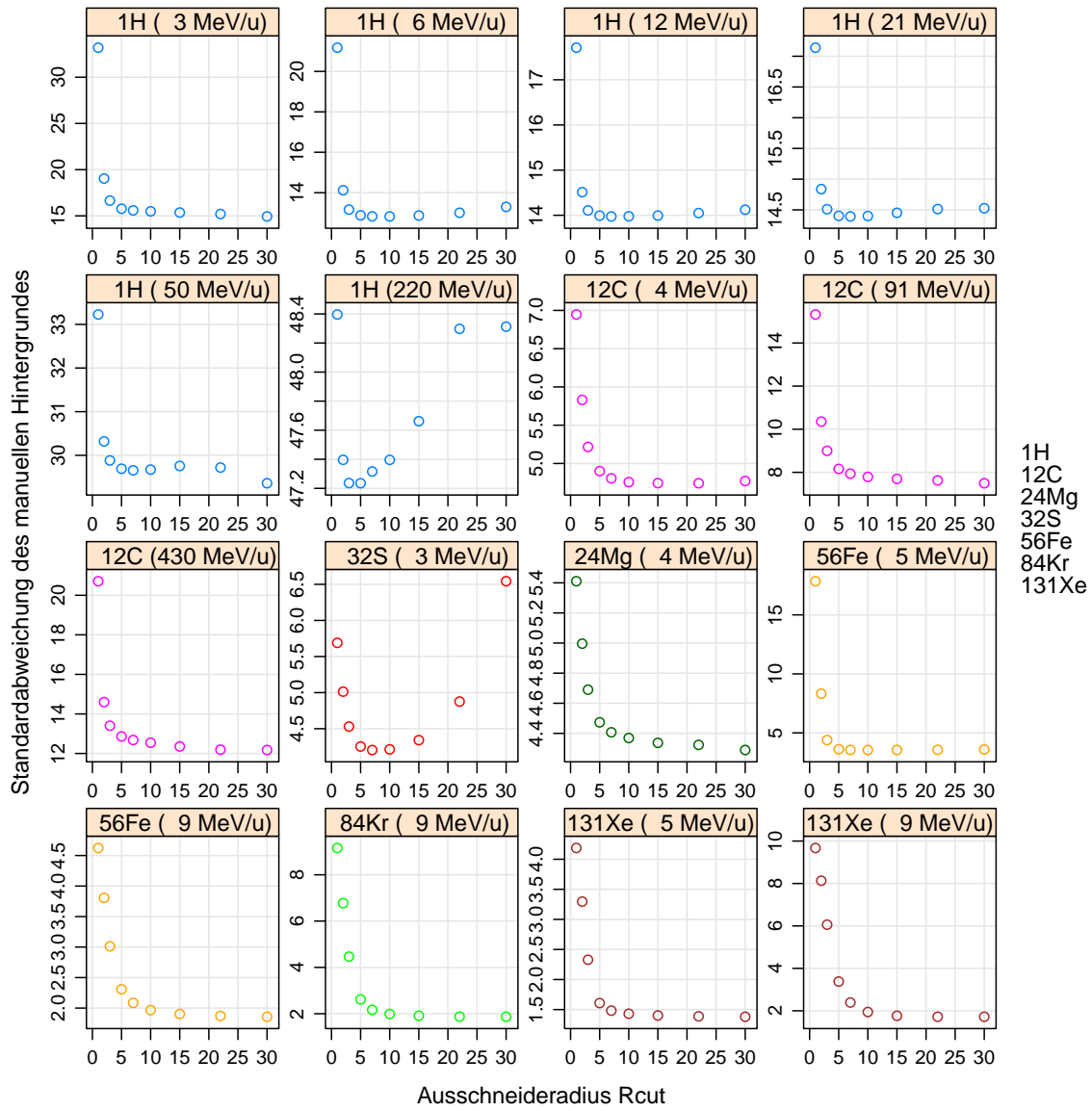


Abbildung 4.4.: Standardabweichung der manuellen Hintergründe über  $R_{cut}$

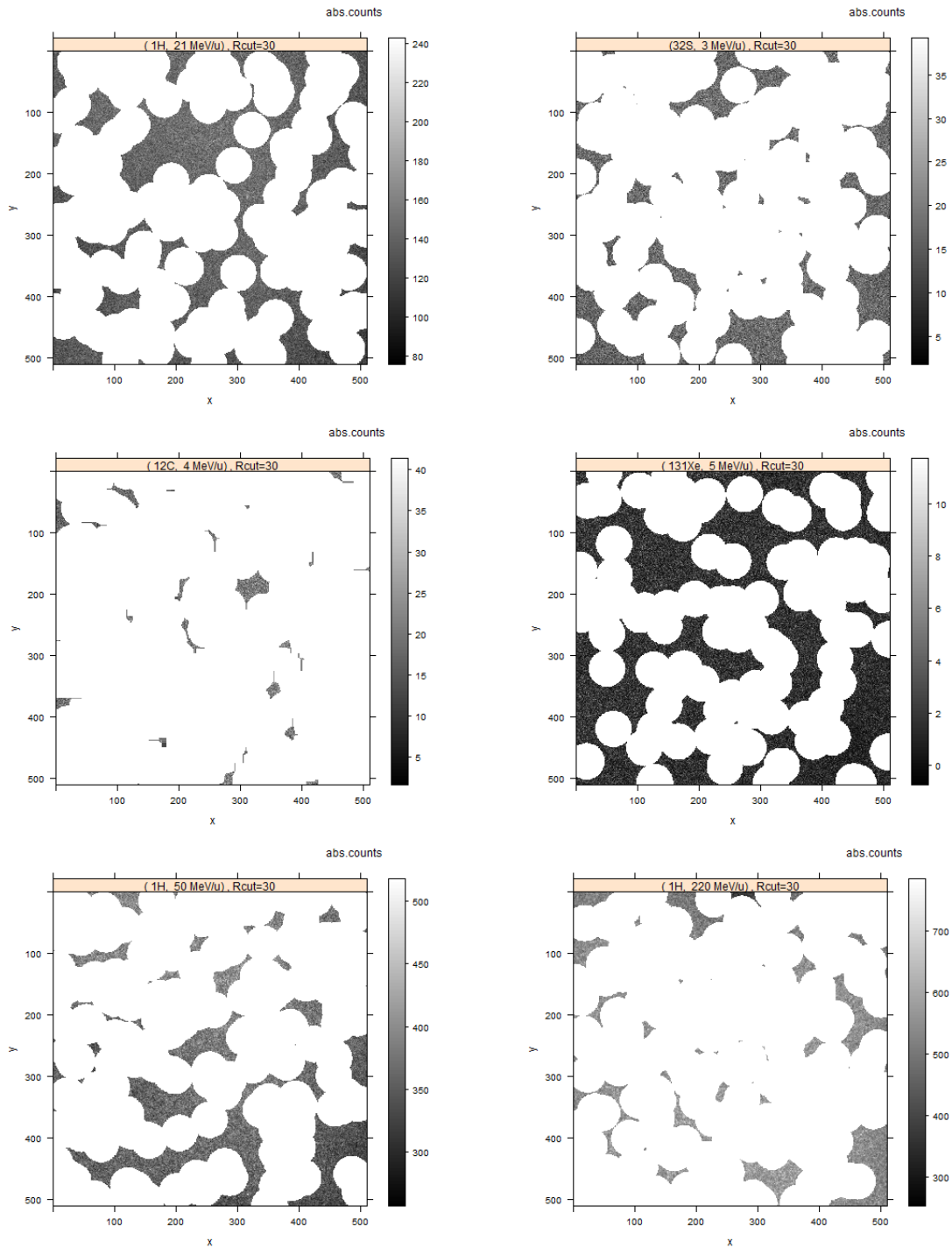


Abbildung 4.5.: Beispiele für Proben mit unterschiedlichen Verläufen der Mittelwerte bei hohen  $R_{cut}$ -Werten. Dargestellt sind jeweils die manuellen Hintergrundbilder mit  $R_{cut} = 30$ . Oben: Ansteigende Mittelwerte, Mitte: Schwach abfallende Mittelwerte, Unten: Stark abfallende Mittelwerte

### 4.2.2. Abhängigkeit von L

In Abschnitt 3.2.2 ist bereits festgestellt worden, dass die Tracks in den automatischen Hintergrundbildern mit zunehmenden L verschwinden. Dieser Prozess findet jedoch kontinuierlich statt, sodass mit bloßem Auge nicht zu bestimmen ist, ab welchem L die Tracks verschwinden. In diesem Abschnitt wird daher die Abhängigkeit der automatischen Hintergrundbilder von L quantitativ untersucht.

Darüber hinaus sollen die L-Werte bestimmt werden, die optimale Resultate bei dem in Abschnitt 4.1 vorgestellten Ansatz liefern. D.h. es muss ein Kompromiss gefunden werden zwischen guten Prüfwerte, weil bei diesen der automatische Hintergrund am ehesten dem manuellen und damit theoretischen Hintergrund entspricht, und dem Verschwinden der Tracks, weil diese per Definition nicht im Hintergrund vorkommen dürfen.

Gute Prüfwerte erwartet man bei kleinen L-Werten, da hier der automatische Hintergrund dem originalen Bild und damit dem manuellen Hintergrundbild am ähnlichsten sind. Die Tracks verschwinden jedoch erst bei größeren L-Werten komplett.

Um dies zu bewerkstelligen, werden zunächst die Prüfwerte für eine Reihe unbestrahlter Proben untersucht. Diese bestehen theoretisch nur aus Hintergrund. Damit kann man an den Prüfwerten den Einfluss des Background Subtractor Algorithmus auf den Hintergrund untersuchen ohne eventuell auftretende Effekte durch die Bestrahlung berücksichtigen zu müssen.

Anschließend wird wieder der Spektroskopiedatensatz untersucht, um aus den Ergebnissen und dem Vorwissen der Untersuchungen der unbestrahlten Proben optimale Parameter für L zu bestimmen.

Da sich in Abschnitt 4.2.1 gezeigt hat, dass die Mittelwerte der Bilder sehr unterschiedliche Countzahlen aufweisen und auch Tabelle A.2 zeigt, dass die Kontrollparameter für die verschiedenen Proben variieren, werden in diesem Abschnitt die in Abschnitt 2.5 eingeführte angepasste Nettocountrate  $\eta_{net}^{adj}$  betrachtet.

#### Unbestrahlte Referenzproben

Es wurden 16 unbestrahlte FNTDs (rm1000\_BG bis rm1015\_BG) untersucht.

Diese mussten zuerst am LSM 710 ausgelesen werden. Dabei wurde pro FNTD nur ein einzelnes Bild im Zentrum des FNTDs ausgelesen und die Kontrollparameter dabei konstant gehalten.

Den manuellen Hintergrund der so erhaltenen Bilder stellen diese selbst dar, da sie nicht bestrahlt wurden.

Untersucht werden die beiden Prüfwerte für jede Probe.

In Abbildungen 4.6 und 4.7 sind die Prüfwerte in Abhängigkeit von L für alle 16 unbestrahlten FNTDs dargestellt.

Man kann erkennen, dass beide Prüfwerte stets über 1 sind. Daraus folgt, dass der automatische Hintergrund im Mittel kleiner ist als das originale Bild.

Die Prüfwerte zeigen in allen Fällen eine steigende Tendenz mit Zunahme von L. Insgesamt sind jedoch die Abweichungen vom Idealwert 1 nur gering. In Tabelle 4.1 werden

einige Werte für  $\mu_p$  und  $f_p$  zusammengefasst:

	Minimum	Maximum	Mittelwert (L=2)	Mittelwert (L=10)	Mittelwert (L=50)
$\mu_p$	1,002365	1,014449	1,004105	1.005634	1,006437
$f_p$	1,002235	1,014543	1,003845	1.005533	1,006462

Tabelle 4.1.: Übersicht der Prüfwerte für unbetsrahlte Proben. Aufgelistet sind der insgesamt kleinste und größte Prüfwert und die Mittelwerte der Prüfwerte für  $L=2$ ,  $L=10$  und  $L=50$ .

Die besseren Werte bei kleinen  $L$  sind dadurch zu erklären, dass der Algorithmus von *get.background* hier Ergebnisse produziert, die dem originalen Bild am meisten ähneln, wie schon in Abschnitt 3.2.2 festgestellt wurde.

Auffällig ist weiter, dass die Prüfwerte der Probe `rm1003_BG` deutlich stärker ansteigen als bei den anderen Proben, wodurch auch die hohen Maximalwerte der Prüfwerte entstanden sind.

Ein Blick auf diese Probe in Abbildung 4.8 zeigt, dass hier ein großer Fleck, also eine starke Abweichung von transformierten Farbzentren des Detektors, zu sehen ist.

Wie ebenfalls in Abbildung 4.8 dargestellt, wird vor allem bei großen  $L$  dieser Fleck im automatischen Hintergrund unterdrückt und da der manuelle Hintergrund in diesem Fall dem originalen Bild entspricht, lässt sich somit die starke Abweichung zwischen automatischen und manuellen Hintergrund und damit die schlechten Prüfwerte erklären. Darüber hinaus sieht man in Abbildung 4.8 einige kleinere Fluktuationen der Countrate. Diese sind in jeder Probe zu finden. Solche Fluktuationen würden bei kleinen  $L$  von *get.background* unterdrückt werden und damit eine schwache Abweichung von manuellen und automatischen Hintergrund erzeugen. Dies könnte damit eine mögliche Erklärung für den stärkeren Anstieg der Prüfwerte bei kleinen  $L$  sein, vor allem da dieser absolut auch nur sehr klein ist und in jeder Probe auftritt.

Da man solche kleinen und größeren Fluktuationen in einer Probe momentan nicht ausschließen kann, müssen diese mit berücksichtigt werden.

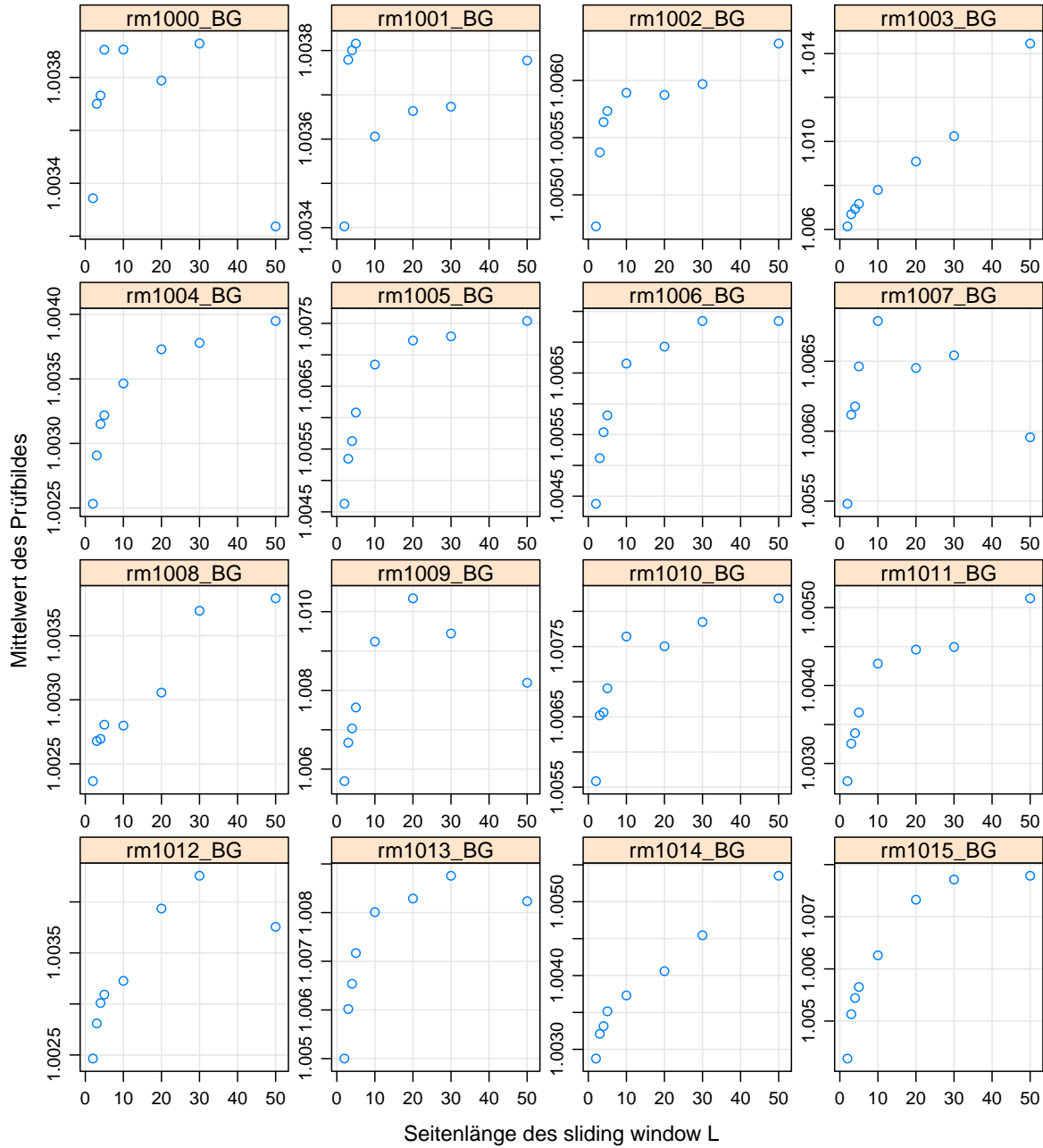


Abbildung 4.6.:  $\mu_p$  in Abhängigkeit von  $L$  für unbestrahlte Proben

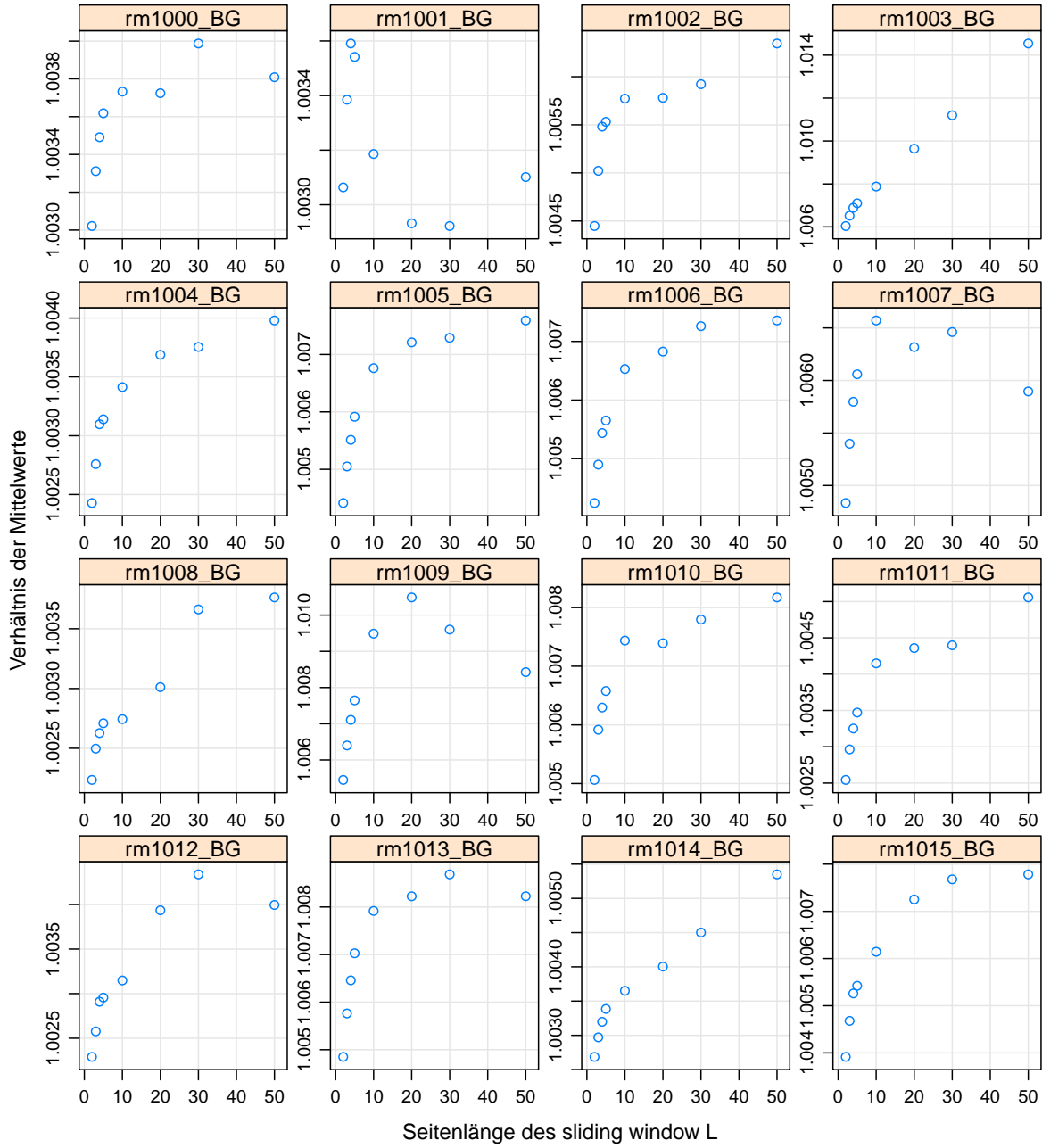


Abbildung 4.7.:  $f_p$  in Abhängigkeit von  $L$  für unbestrahlte Proben

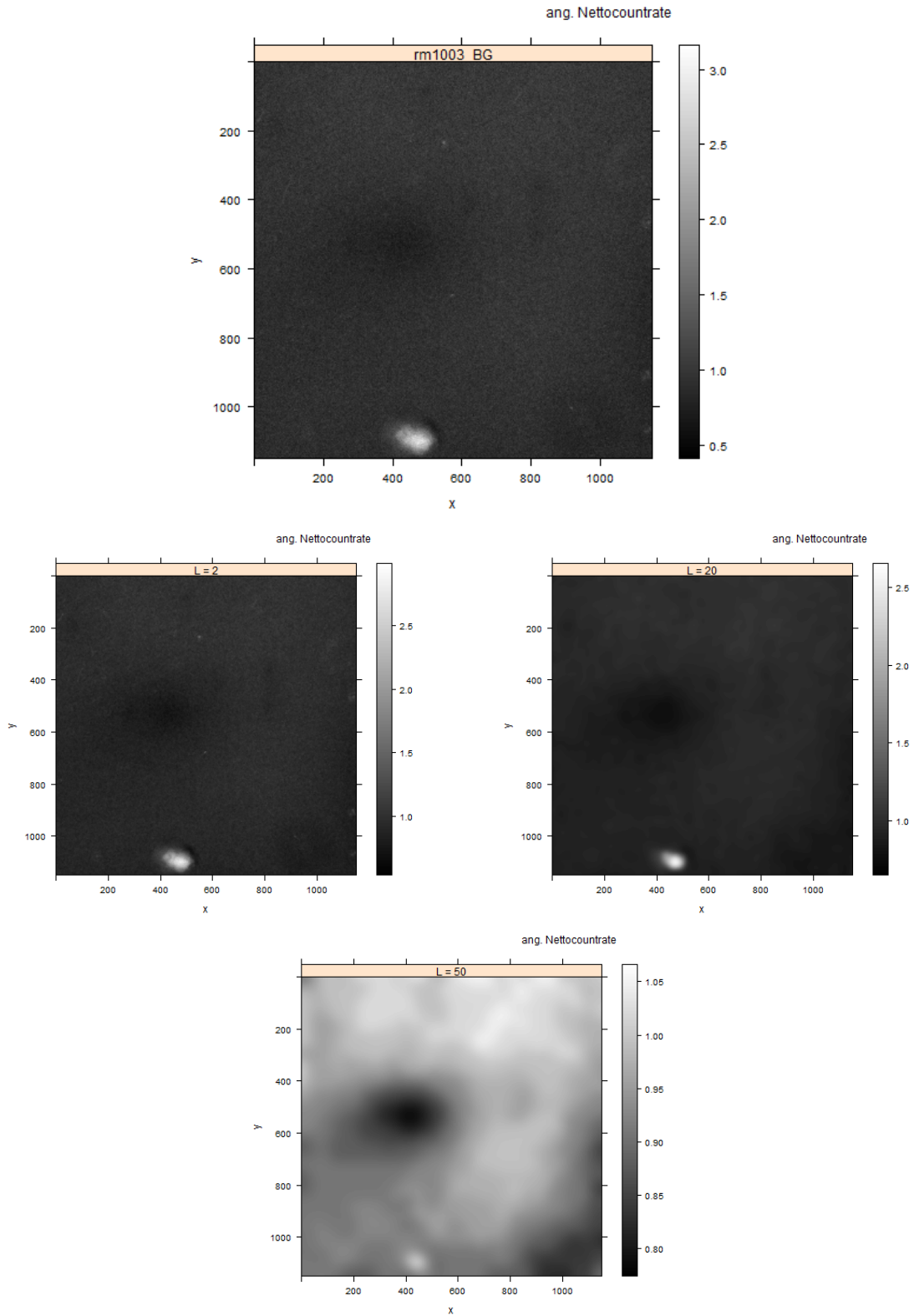


Abbildung 4.8.: Probe *rm1003\_BG* mit entsprechenden Hintergrundbilder für  $L = 2$ ,  $20$ ,  $50$ . In der Mitte unten ist ein großer Fleck zu sehen, der für die schlechten Prüfwerte dieser Probe verantwortlich ist. Außerdem sieht man einige kleine Felcken im Bild.



### Bestrahlte Proben (Teilchenabhängig)

In diesem Abschnitt wird wieder der gesamte Spektroskopiedatensatz der bestrahlten Proben untersucht. Von jeder Probe wird ein Bild aus der Mitte des Bilderstapels genommen und die  $R_{cut}$ -Werte, wie sie in Abschnitt 4.2.1 ermittelt wurden, gewählt (siehe Tabelle A.1). Um den Einfluss von  $L$  auf die Abweichungen des automatischen vom manuellen Hintergrund zu bestimmen, werden die Prüfwerte bei Variation von  $L$  untersucht.

Um gleichzeitig quantitativ bestimmen zu können, ab welchem  $L$  die Tracks in den Hintergrundbildern verschwinden, wird die Differenz von Maximum und Mittelwert des automatischen Hintergrundes untersucht. Denn, da die Tracks gerade positive Abweichungen von Countraten von der mittleren Countrate in einem Bild darstellen, ist zu erwarten, dass die Differenz von Maximum und Mittelwert der Countraten stark abnimmt, wenn mit zunehmenden  $L$  die Tracks verschwinden. Anschließend müsste sie nahezu konstant bleiben.

An diesem Übergang müsste man den  $L$ -Wert bestimmen können, bei dem gerade die Tracks verschwinden.

In Abbildungen 4.9 bis 4.11 sind für alle Proben die beiden Prüfwerte und die Differenz zwischen Maximum und Mittelwert der automatischen Hintergründe für verschiedene  $L$  dargestellt.

Bei Betrachtung der Prüfwerte zeigt sich im Falle von  $\mu_p$  ein komplizierteres Bild als im unbestrahlten Fall, wohingegen  $f_p$  sich ähnlich verhält.

In Tabelle 4.2 werden einige Werte für  $\mu_p$  und  $f_p$  zusammengefasst:

	Minimum	Maximum	Mittelwert (L=2)	Mittelwert (L=10)	Mittelwert (L=50)
$\mu_p$	1,008988	1.750872	1.079388	1.134496	1.160402
$f_p$	0.7365584	1.715919	0.9863928	1.108316	1.143901

Tabelle 4.2.: Übersicht der Prüfwerte für bestrahlten Proben. Aufgelistet sind der insgesamt kleinste und größte Prüfwert und die Mittelwerte der Prüfwerte für  $L=2$ ,  $L=10$  und  $L=50$ .

Die Tabelle zeigt, dass beide Prüfwerte über einen deutlich größeren Bereich variieren und auch stärkere Abweichungen von 1 zeigen, als bei den unbestrahlten Proben in Tabelle 4.1. An den Mittelwerten für  $L=2$ , 10 und 50 sieht man jedoch, dass die Tendenz der Prüfwerte bei größeren  $L$ -Werten anzusteigen gleich ist wie bei den unbestrahlten Proben.

Der  $\mu_p$ -Prüfwert in Abbildung 4.9 zeigt kein einheitliches Verhalten für alle Proben. In der Hälfte der Proben steigen die  $\mu_p$ -Werte, wie im unbestrahlten Fall, mit zunehmenden  $L$  an. Die Proben (1H, 12 MeV), (12C, 4 MeV), (56Fe, 5 MeV) und (84Kr, 9 MeV) hingegen zeigen eine abfallende Tendenz für zunehmende  $L$ -Werte. In den Proben (1H, 6 MeV), (12C, 91 MeV), (24Mg, 4 MeV) und (56Fe, 9 MeV) lässt sich keine eindeutige

Tendenz erkennen. An diesen Proben sieht man, dass die Proben, die eine gleiche Tendenz zeigen, völlig unterschiedlich von der Teilchenart und Energie sein können. Insgesamt zeigt sich jedoch, dass die  $\mu_p$ -Werte für leichte Teilchen bessere Werte zeigen und die Schwankungen in Abhängigkeit von L hier sehr gering sind. Erst für die  $^{131}\text{Xe}$ -Proben steigen sie sehr deutlich an. Außerdem erkennt man, dass die Proben mit steigender Tendenz in der Regel sehr deutlich ansteigen, während die Proben mit fallender Tendenz im Vergleich dazu nur gering abfallen.

Der  $f_p$ -Prüfwert in Abbildung 4.10 zeigt für alle Proben ein ähnliches Verhalten. Er steigt zuerst stark an und geht anschließend in ein Plateau über.

Dieser starke Anstieg zu Beginn erinnert an die Verläufe bei den unbestrahlten Proben. Hier erstreckt er sich aber über einen deutlich größeren Bereich.

In einigen Proben ist der  $f_p$ -Wert für niedrige L sogar kleiner als 1. Hier scheint also der automatisch erzeugte Hintergrund größer als der manuelle zu sein.

Dieses Verhalten zeigt sich vor allem bei schwereren Teilchen.

Dies zeigt ein Problem des  $f_p$ -Prüfwertes. Im manuellen Hintergrundbild wird der Mittelwert nur aus den Pixeln gebildet, die nicht ausgeschnitten wurden, also nur aus dem Hintergrund.

Beim automatischen Hintergrund wird hingegen der Mittelwert über alle Pixel gebildet. Da wie schon in Abschnitt 3.2.2 gesehen bei kleinen L die Tracks nicht verschwinden, erhöhen diese den automatischen Hintergrund. Dieser Effekt ist umso stärker, je mehr sich die Tracks vom Hintergrund abheben.

Wie in Abschnitt 3.2.1 gesehen, ist dies gerade bei den schweren Teilchen der Fall.

(Bei dem  $\mu_p$ -Prüfwert tritt dies nicht auf, da der Mittelwert des Prüfbildes über die Bereiche gebildet wird, in denen die Pixel des manuellen Hintergrundbildes nicht ausgeschnitten wurden.) Dies zeigt, dass der  $f_p$ -Prüfwert durch das Vorhandensein von Resten von Tracks im automatischen Hintergrundbild abgefälscht wird und damit nicht mehr richtig den Unterschied zwischen manuellen und automatischen Hintergrund charakterisiert. Er sollte also nur in Bereichen herangezogen werden, in denen die Tracks im automatischen Hintergrund bereits verschwunden sind.

In Abschnitt 3.2.2 konnte man qualitativ sehen, dass die Tracks in den automatisch erzeugten Hintergrundbildern kontinuierlich verschwinden. Dort kam auch die Frage auf, ab welchem L man die Tracks als verschwunden und die automatischen Hintergrundbilder damit wirklich als Hintergründe bezeichnen kann.

Zu diesem Zweck wurde in Abbildung 4.11 die Differenz der Maxima und Mittelwerte der automatischen Hintergrundbilder gezeigt.

In allen Proben zeigt sich hier ein starker Abfall der Differenz für kleine L und anschließend ein nahezu konstantes Verhalten.

Das liegt daran, dass bei kleinen L-Werten die Tracks, die für ein erhöhtes Maximum sorgen, verschwinden und für alle Proben ab  $L=10$  nur noch Hintergrund übrigbleibt, der keine starken Abweichungen vom Mittelwert mehr zeigt.

Damit kann man, vorausgesetzt die verschiedenen Bilder einer Probe verhalten sich ähnlich, aus diesem Plot über die Stellen, an denen der steile Abstieg der Differenz endet,

einen L-Wert für jede Probe bestimmen, ab dem man die Teilchenreste vernachlässigen kann.

Wenn man nur diesen Aspekt betrachtet, könnte man für die weitere Auswertung auch ein größeres L wählen, um absolut sicher zu sein, dass keine Teilchenreste mehr übrig sind.

Beim  $\mu_p$ -Prüfwert hat sich jedoch die Tendenz gezeigt, dass er für größere L-Werte schlechter wird. Beim  $f_p$ -Prüfwert kann man aus dem Resultat aus Abbildung 4.11, dass die Tracks im automatischen Hintergrund für alle Proben ab  $L=10$  zu vernachlässigen sind, schließen, dass er für  $L \geq 10$  zuverlässige Ergebnisse liefert. Dieser zeigt hier eine etwas deutlichere Verschlechterung für große L-Werte als der  $\mu_p$ -Wert.

Außerdem hat sich auch im Fall der unbestrahlten Proben gezeigt, dass die Prüfwerte mit größeren L schlechter werden.

Deshalb bietet sich an für die optimalen L-Werte für die weitere Auswertung gerade die Grenzwerte, an denen die Tracks verschwinden, zu wählen. Denn an diesen sind die Bedingungen, dass die Hintergrundbilder durch das Verschwinden der Tracks als tatsächliche Hintergrundbilder betrachtet werden können und die Prüfwerte möglichst gut sind, am optimalsten erfüllt.

Um jedoch sicher zu sein, dass ein einheitliches L für alle Bilder einer Probe sinnvoll ist, wird im nächsten Abschnitt das Verhalten innerhalb eines Bilderstapels einer Probe für verschiedene L beispielhaft untersucht.

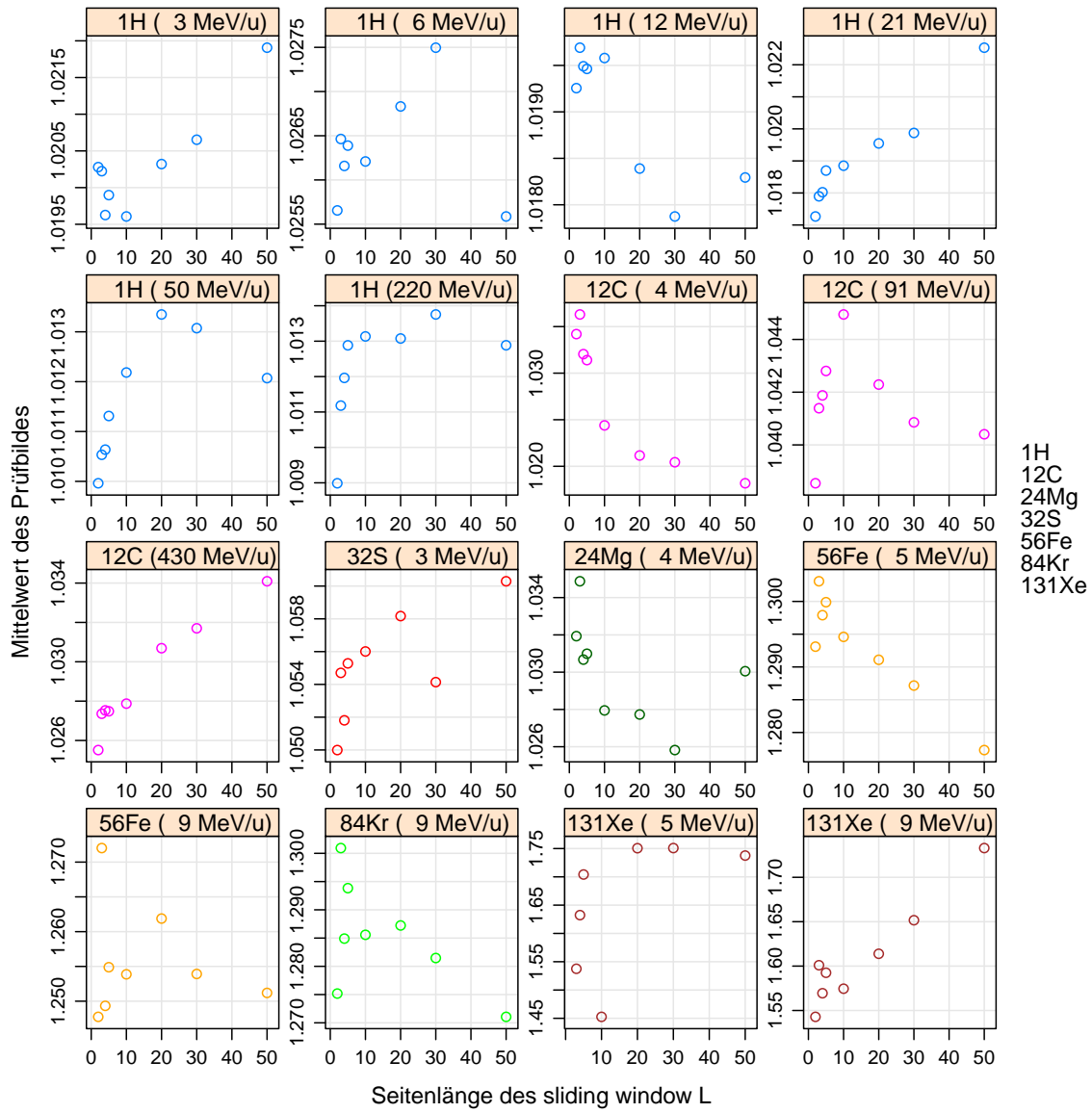


Abbildung 4.9.:  $\mu_p$  in Abhängigkeit von  $L$  für bestrahlte Proben

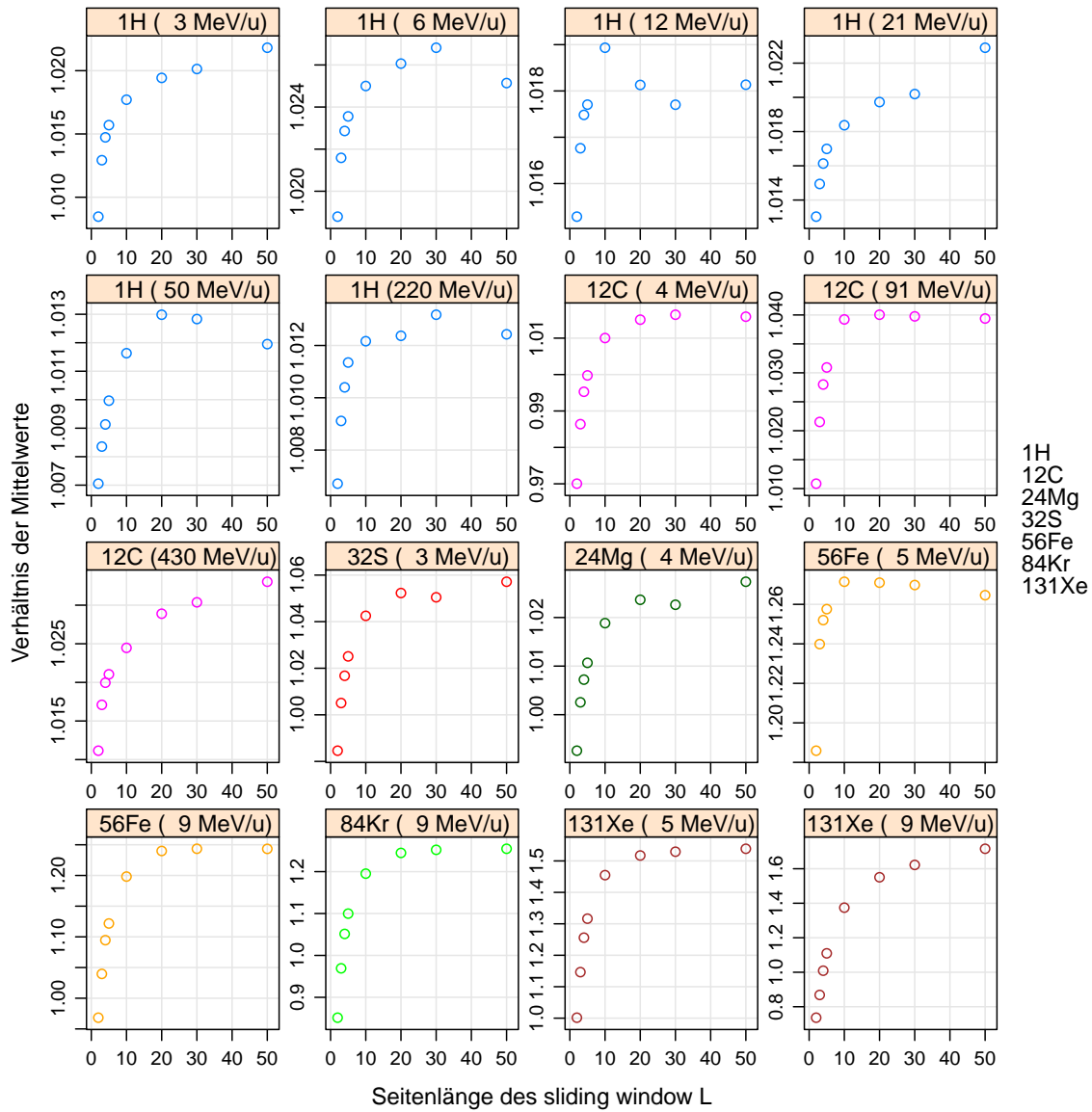


Abbildung 4.10.:  $f_p$  in Abhängigkeit von  $L$  für bestrahlte Proben

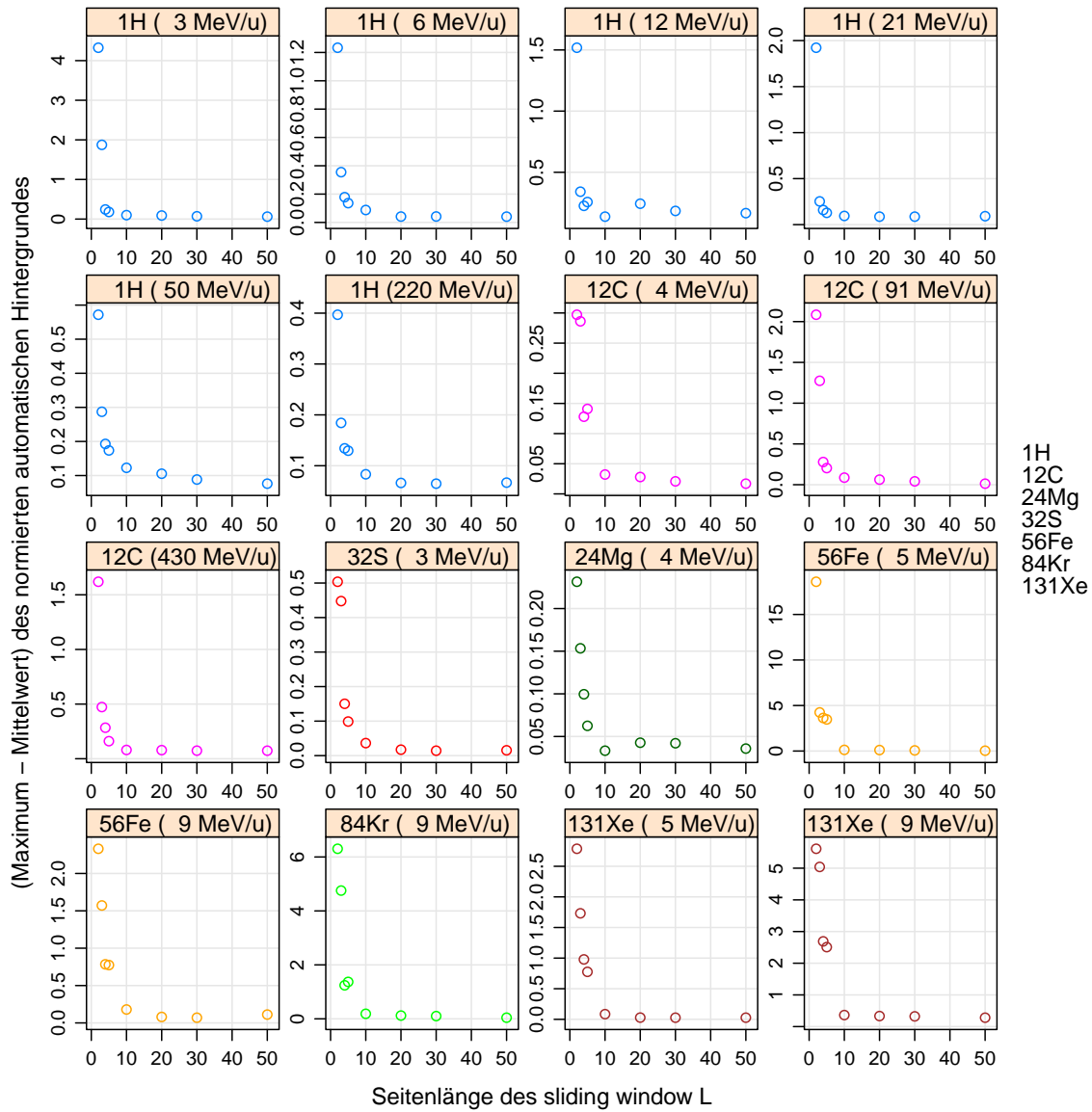


Abbildung 4.11.: Differenz zwischen Maximum und Mittelwert der angepassten Counten der automatischen Hintergründe in MHz über  $L$

**Bestrahlte Proben (Tiefenabhängig)**

Um zu überprüfen, ob das Verhalten der Hintergründe bezüglich des Verschwindens der Tracks innerhalb eines Bilderstapels konsistent ist, wird beispielhaft an den Proben ( $^{56}\text{Fe}$ , 4 MeV) und ( $^{131}\text{Xe}$ , 9 MeV) die Differenz zwischen Maximum und Mittelwert in Abhängigkeit von  $L$  für alle Bilder der beiden Bilderstapel d.h. für die Tiefenprofile untersucht.

Abbildungen 4.12 und 4.13 zeigen die Ergebnisse hiervon. Es zeigt sich, dass in beiden Proben der Verlauf für alle Bilder sehr ähnlich ist und auch mit den Verläufen in Abbildung 4.11 übereinstimmen.

Vor allem die Stelle, an der der steile Abfall in einen konstanten Bereich übergeht, kann in beiden Fällen für den gesamten Bilderstapel einheitlich bestimmt werden. In diesen beiden Beispielen ist dies bei  $L=10$  der Fall.

Damit bekräftigt dies die Annahme, dass man anhand des Verlaufes eines einzelnen Bildes einer Probe einen guten Wert für  $L$  bestimmen kann, ab dem gerade die Tracks verschwinden.

Dies wurde anhand von Abbildung 4.11 gemacht und die Ergebnisse in der Spalte „ $L$ “ in Tabelle A.1 eingetragen.

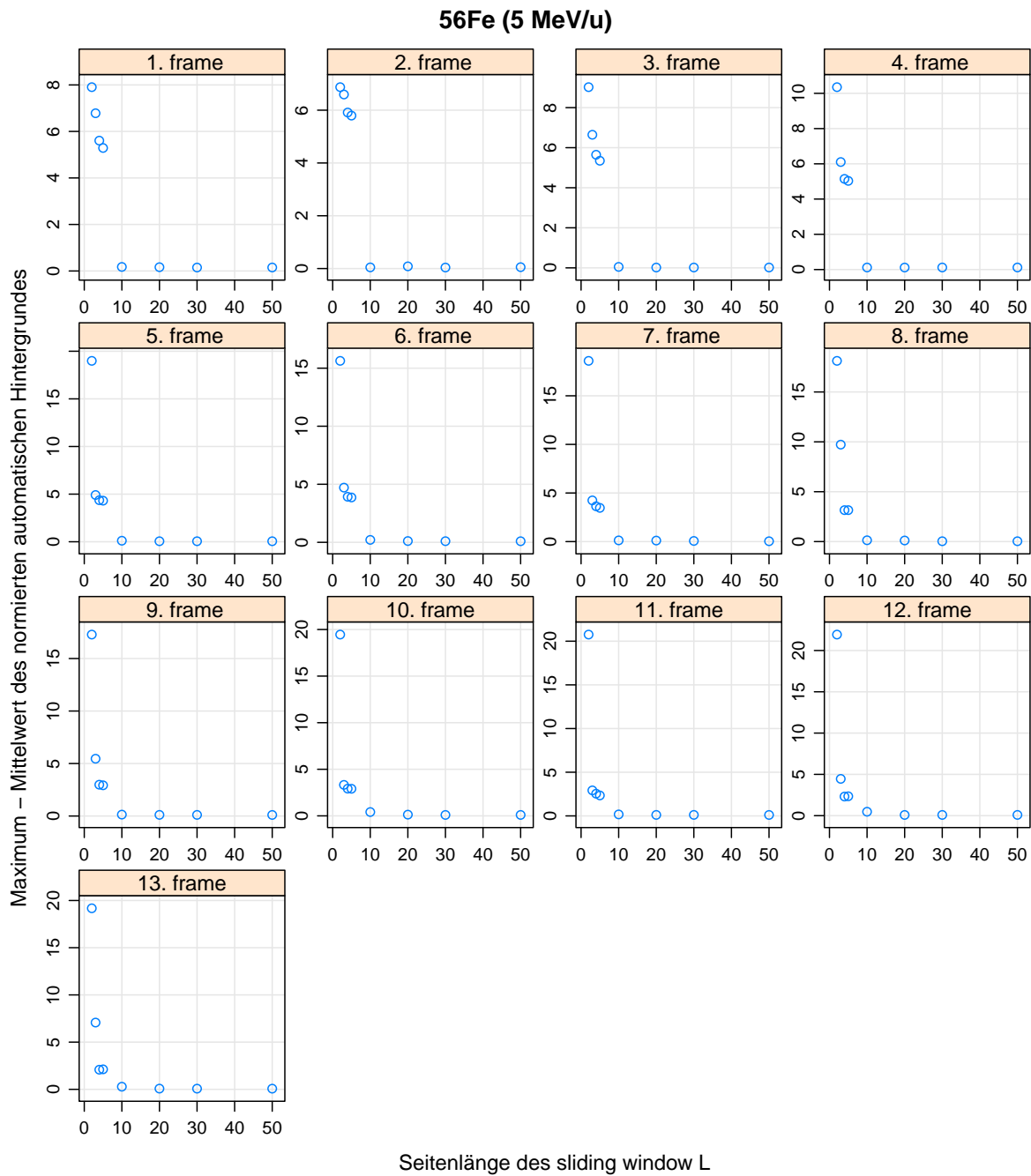


Abbildung 4.12.: *Differenz zwischen Maximum und Mittelwert der angepassten Count-raten der automatischen Hintergründe in MHz über L für ( $^{56}\text{Fe}$ , 5 MeV)*



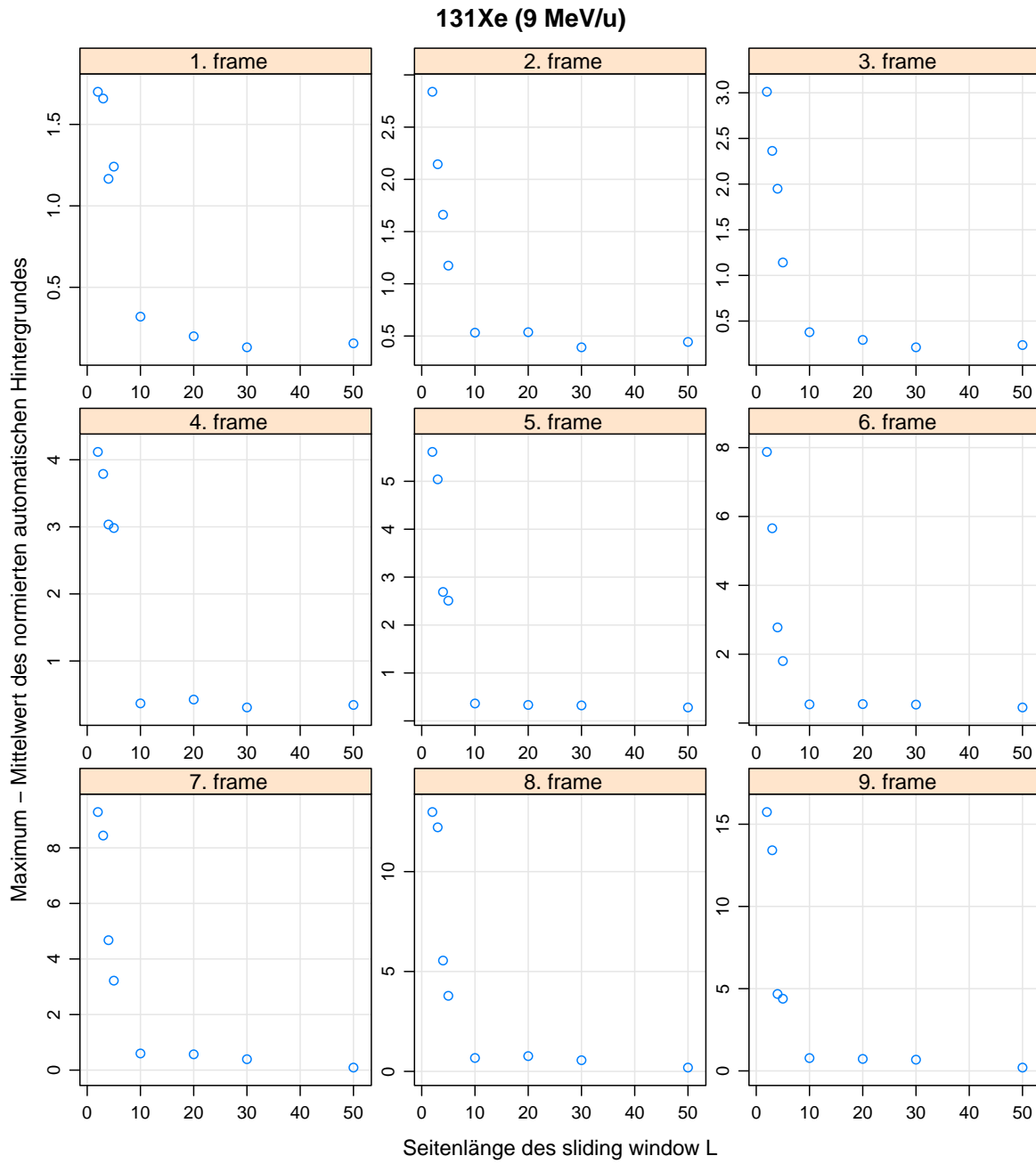


Abbildung 4.13.: Differenz zwischen Maximum und Mittelwert der angepassten Countraten der automatischen Hintergründe in MHz über  $L$  für ( $^{131}\text{Xe}$ , 9 MeV)

### 4.3. Teilchenabhängige Qualität des Hintergrundes

Da im vorherigen Abschnitt eine Auswahl für die optimalen Parameter getroffen wurde, kann nun der Ansatz zur Überprüfung des automatischen Hintergrundbildes mit diesen Werten durchgeführt werden. Es wird im Folgenden also für ein mittleres Bild aus jeder Probe des Spektroskopiedatensatzes mit den „optimalen“ Werten für die Particle Tracker Parameter, der Länge des sliding window  $L$  und dem Ausschneideradius  $R_{cut}$  (siehe Tabelle A.1) die beiden Prüfwerte untersucht.

In Tabelle 4.3 wurden die Ergebnisse für die beiden Prüfwerte für die verschiedenen Proben zusammengetragen.

Probe	$\mu_p$	$f_p$
( 1H, 3 MeV/u)	1.019607	1.017708
( 1H, 6 MeV/u)	1.026209	1.025001
( 1H, 12 MeV/u)	1.019580	1.018934
( 1H, 21 MeV/u)	1.018851	1.018372
( 1H, 50 MeV/u)	1.012184	1.011633
( 1H, 220 MeV/u)	1.013134	1.012164
( 12C, 4 MeV/u)	1.024391	1.010035
(12C, 91 MeV/u)	1.044947	1.039230
(12C, 430 MeV/u)	1.027874	1.024469
(24Mg, 4 MeV/u)	1.027949	1.018844
(32S, 3 MeV/u)	1.056009	1.042521
(56Fe, 5 MeV/u)	1.294592	1.271443
(56Fe, 9 MeV/u)	1.253868	1.198224
(84Kr, 9 MeV/u)	1.285577	1.195226
(131Xe, 5 MeV/u)	1.452611	1.454824
(131Xe, 9 MeV/u)	1.574556	1.374432

Tabelle 4.3.: Übersicht der Prüfwerte für bestrahlte Proben bei optimalen Parametern. Betrachtet wurde dabei das mittlere Bild aus jeder Probe.

Die Tabelle zeigt eine klare Tendenz der Prüfwerte für schwerere Teilchen schlechter zu werden.

Aber im Vergleich zu Abbildungen 4.6 und 4.7 zeigt sich, dass die Werte dennoch deutlich höher sind als die entsprechenden Werte mit  $L=10$  bei den unbestrahlten Proben. Jedoch sieht man, dass die *get.background* Funktion für leichte Teilchen relativ gute Ergebnisse liefern kann. Bei mit Wasserstoff bestrahlten Proben ist die Abweichung des automatischen Hintergrundes für alle Energien weniger als 3%.

Bei schwereren Teilchen ab Eisen sind die Abweichungen jedoch zwischen (19 - 57)%. Folglich stößt hier entweder die *get.background*-Funktion oder der verwendete Ansatz

zur Überprüfung der Funktion an seine Grenzen.

Um einen anschaulicherem Blick auf die Hintergrundbilder zu bekommen, werden diese in Abbildung 4.14 ihrem originalen Bild gegenübergestellt.

Es ist für Protonstrahlung jeweils ein Beispiel für niedrige (  $^1\text{H}$ , 3 MeV/u) und hohe Energien (  $^1\text{H}$ , 220 MeV/u) dargestellt und ein Beispiel für schwere Teilchen ( $^{131}\text{Xe}$ , 9 MeV/u).

Bei ( $^{131}\text{Xe}$ , 9 MeV/u) kann man erkennen, dass die Bereiche, in denen Teilchen durchgekommen sind, im Hintergrund immer noch verschwommen sichtbar sind.

Das originale Bild dieser Probe zeichnet sich jedoch durch eine sehr hohe Differenz zwischen Hintergrund- und Teilchensignal aus, wie man an der Skala zur Farbkodierung sehen kann.

Relativ zu dieser Differenz beträgt die maximale Differenz im Hintergrundbild nur etwa 5% von dieser.

Damit ist dieses Bild dennoch im Einklang mit Abbildung 4.11, aus der geschlossen wurde, dass für diese Probe ab  $L=10$  die Differenz zwischen maximalen und mittleren Signal des Hintergrundbildes so gering ist, dass dieses tatsächlich als Hintergrundbild angesehen werden kann.

An den beiden Protonbildern kann man beobachten, dass sich bei hohen Energien ein homogenerer Hintergrund ergibt, während man bei dem Beispiel mit 3 MeV/u noch zum Teil im Hintergrund Regionen erkennen kann, in denen im Original viele Teilchen zu sehen sind.

Der Vergleich der Bilder von (  $^1\text{H}$ , 3 MeV/u) und ( $^{131}\text{Xe}$ , 9 MeV/u) zeigt zudem, dass, der Hintergrund von ( $^{131}\text{Xe}$ , 9 MeV/u) Bereiche aufweist, die deutlich dunkler sind. In diesen Bereichen muss demnach auch im Originalbild eine sehr niedrige Countrate gemessen worden sein.

Dies lässt die Idee aufkommen, dass die schlechten Prüfwerte mit niedrigen Countraten korrelieren. Erklären könnte man dies durch den Ansatz, dass die kleinskaligen Fluktuationen des Hintergrundes ähnlich wie ein symmetrisches Rauschen um einen mittleren Wert schwanken. Ist dieser mittlere Wert sehr nah bei Null, sorgt dies für ein asymmetrisches Verhalten, da Werte über den mittleren Wert schwanken können aber nicht weit unter ihm, weil sie nicht negativ werden können.

Da im automatischen Hintergrund solche kleinskaligen Fluktuationen verschwinden, würden dann vor allem Fluktuationen in die positive Richtung verschwinden und der automatische Hintergrund damit im Mittel dunkler werden als das originale Bild, was zu hohen Prüfwerten führt. Um dies zu untersuchen sind in Abbildung 4.15 die beiden Prüfwerte  $\mu_p$  und  $f_p$  über den Mittelwert des manuellen Hintergrundbildes aufgetragen. Man sieht tatsächlich, dass die Proben mit hohen Prüfwerten stets im Mittel niedrige Countraten aufweisen.

Jedoch gibt es auch Proben, die noch niedrigere Mittelwerte aufweisen und gleichzeitig deutlich bessere Prüfwerte zeigen.

Demnach scheint diese Idee keine Erklärung zu liefern für die hohen Prüfwerte.

Vielmehr ist in Abbildung 4.15 anhand der Farbkodierung der Zusammenhang zwischen schweren Teilchen und hohen Prüfwerte nochmals verdeutlicht.

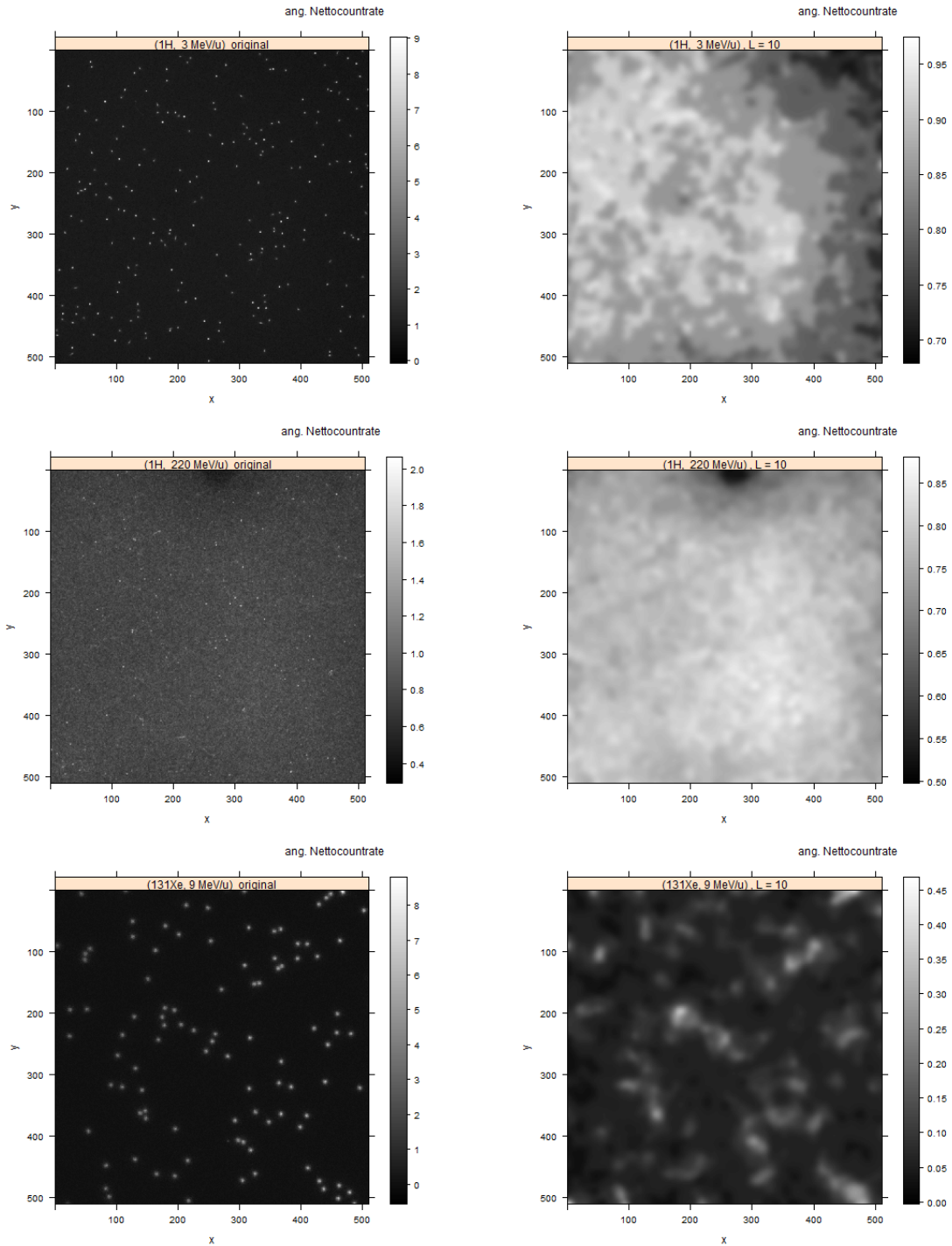


Abbildung 4.14.: Vergleich einiger Bilder mit ihren optimalen Hintergrundbildern

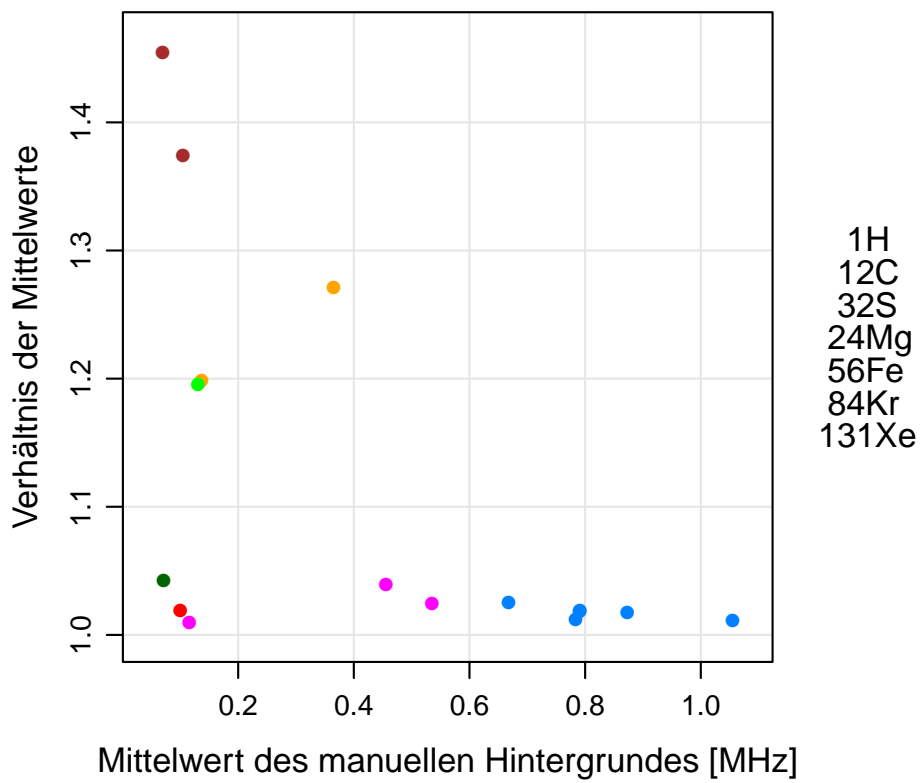
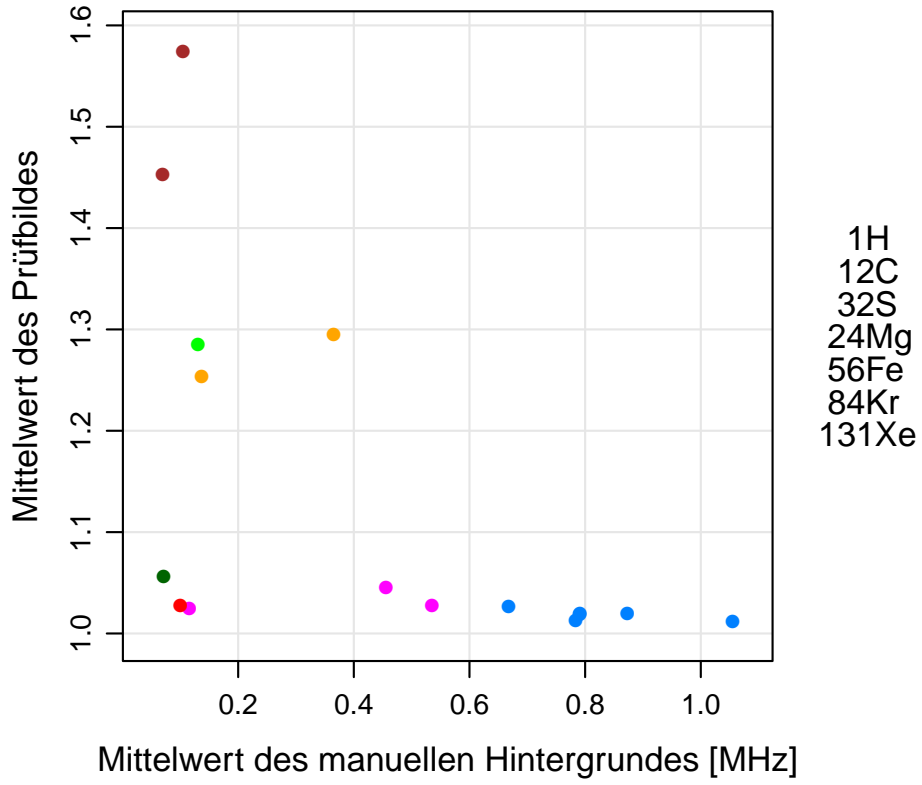


Abbildung 4.15.:  $\mu_p$  und  $f_p$  aufgetragen über die Mittelwerte der angepassten Countraten der manuellen Hintergrundbilder.



# 5. Zusammenfassung und Diskussion

In dieser Arbeit wurden Softwarewerkzeuge zur automatisierten Bestimmung des Hintergrüdes von FNTDs, die am Zeiss LSM710 ausgelesen wurden, bereitgestellt, überprüft und an einem Datensatz angewendet.

## 5.1. Werkzeuge

Die in dieser Arbeit bereitgestellten Werkzeuge basieren auf ImageJ-Plugins, die zur automatisierten Funktionsweise angepasst und in ein R-Paket integriert wurden, welches zum Bearbeiten von FNTD-Bilddaten in R dient.

Das Hauptwerkzeug ist die Funktion *get.background*, die aus einer Reihe gegebener FNTD-Bilder automatisiert deren Hintergründe bestimmen kann.

Um mit FNTD-Bildern in R arbeiten zu können, wurde zudem die Funktion *read.lsm.image* bereitgestellt, welche aus den vom LSM710 erzeugten Bilddaten ein entsprechendes Bildobjekt in R mit allen relevanten Metadaten erzeugt.

Als weiteres Hilfswerkzeug wurde die Funktion *find.tracks* bereitgestellt, die aus einem gegebenen FNTD-Bild die Positionen von lokalen Schwankungen der Countzahlen (*Tracks*) bestimmt.

Überprüfungen aller drei Werkzeuge, in denen ihre Funktionsweisen mit den der ursprünglichen ImageJ-Plugins verglichen wurden, ergaben, dass sie die gleichen Resultate liefern und damit korrekt in das R-Paket integriert wurden.

## 5.2. Bestimmung des optimalen Parameters $L$

Die *get.background*-Funktion wurde an einen Datensatz von FNTDs, die mit verschiedenen Teilchen und Energien bestrahlt wurden, angewendet. Um die Resultate hiervon zu überprüfen, wurde in Abschnitt 4.1 ein Ansatz hierfür vorgestellt, bei dem für jede Probe zwei Prüfwerte  $\mu_p$  und  $f_p$  erstellt werden. Dieser Ansatz hängt neben dem hauptsächlich zu bestimmenden Parameter  $L$  auch vom  $R_{cut}$ -Parameter ab. Für beide mussten die optimalen Einstellungen gefunden werden.

### 5.2.1. Einfluss von $R_{cut}$

Es wurde Mittelwert, Maximum und Standardabweichung aller Pixelwerte der manuellen Hintergründe in Abhängigkeit von  $R_{cut}$  untersucht, um optimale Werte für  $R_{cut}$  zu erhalten.

Die Ergebnisse (siehe Abschnitt 4.2.1) haben gezeigt, dass Mittelwert und Maximum ungeeignet für die Bestimmung optimaler  $R_{cut}$ -Werte sind. Die Standardabweichung hat sich als bester Ansatz hierfür herausgestellt und wurde im Rahmen dieser Arbeit verwendet. Die daraus resultierenden empfohlenen  $R_{cut}$ -Werte wurden in Tabelle A.1 zusammengefasst.

### 5.2.2. Unbestrahlte Referenzproben

Bei der Untersuchung des Einflusses von  $L$  auf einen Satz nicht bestrahlter Proben haben sich sehr gute Werte für die Prüfwerte  $\mu_p$  und  $f_p$  ergeben (Abschnitt 4.2.2). Daraus kann man schließen, dass der Background Subtractor, wenn er auf einen reinen Hintergrund angewandt wird, diesen unter Erhaltung der Countraten und damit korrekt bestimmt. Dennoch konnte man sehen, dass der Background Subtractor Algorithmus einen kleinen Einfluss auf die Prüfwerte hat, sodass diese minimal schlechter werden, wenn man  $L$  größer wählt. Das zeigt, dass der  $L$ -Parameter bei der Bestimmung von Hintergründen klein gewählt werden sollte, wobei er gleichzeitig aber auch groß genug sein muss, dass die Strukturen des Vordergrundes verschwinden.

Da die Prüfwerte in diesem Abschnitt trotz der guten Werte dennoch stets über 1 waren, kann an dem Verschwinden der kleinskaligen Variation in den automatischen Hintergrundbildern liegen. Denn bisher ist noch nicht klar, ob das lokale Histogramm-Verfahren (Abschnitt 2.7.4) von *get.background* tatsächlich den mittleren Wert der kleinskaligen Variation als Maximum des lokalen Histogramms bestimmt. Die hier erhaltenen Ergebnisse könnten dadurch erklärt werden, dass tendenziell Werte unter dem mittleren Wert der kleinskaligen Variation als Maxima bestimmt werden, wodurch die automatisch erzeugten Hintergrundbilder im Mittel kleinere Countzahlen aufweisen und die Prüfwerte damit größer werden.

Wie man an den dennoch guten Prüfwerten sehen kann, muss dieser Effekt jedoch sehr klein sein. Einen Aufschluss darüber könnten Untersuchungen geben, in denen man die lokalen Histogramme von Bereichen, die nur kleinskaliges Rauschen enthalten, betrachtet und mit den Ergebnissen des Background Subtractors vergleicht.

### 5.2.3. Bestrahlte Proben (Teilchenabhängig)

Bei der Untersuchung des Einflusses von  $L$  auf den Datensatz bestrahlter Proben sollten optimale Werte für diesen Parameter bestimmt werden. Dabei musste berücksichtigt werden, dass in den erzeugten Hintergrundbildern die Tracks nicht mehr vorkommen dürfen und gleichzeitig die Prüfwerte  $\mu_p$  und  $f_p$  möglichst gut sind.

Es hat sich gezeigt, dass zur Bestimmung des Verschwindens der Tracks die Betrachtung der Differenz zwischen Maximum und Mittelwert der Countraten im automatischen Hintergrundbild in Abhängigkeit von  $L$  gut geeignet ist (siehe Abbildung 4.11). Diese ergab für alle Proben die Bedingung  $L \geq 10$ .

Um die Prüfwerte zu begutachten, wurden diese selbst in Abhängigkeit von  $L$  untersucht. Dies ergab jedoch kein klares Bild (Abbildung 4.9). Im größten Teil der Proben zeigte sich eine Verschlechterung der Prüfwerte mit zunehmendem  $L$ . Einige Proben zeigten



jedoch auch keine Abhängigkeit oder eine Verbesserung mit  $L$ . Da aber die Verschlechterungen öfters vorkamen, meist stärker ausgeprägt waren und im Fall der unbestrahlten Proben ebenfalls aufgetreten sind, wurde aus den Resultaten die Bedingung geschlossen, dass der  $L$ -Wert für alle Proben möglichst klein gewählt werden sollte. Diese Bedingung ist jedoch mehr als Empfehlung zu betrachten, da die Ergebnisse hierfür nicht eindeutig sind.

Diese beiden Bedingungen zusammen ergeben für alle Proben  $L = 10$  als Vorschlag für einen optimalen Wert.

#### 5.2.4. Bestrahlte Proben (Tiefenabhängig)

Die Wahl des optimalen  $L$ -Parameters bestand auf der Annahme, dass die Bedingung  $L \geq 10$  für alle Bilder einer Probe d.h. für das Tiefenprofil der Probe gilt. Diese Annahme wurde beispielhaft an den Bilderstapeln zweier Proben überprüft. Die Ergebnisse haben bestätigt, dass die Tracks in jedem Bild des Datensatzes bei dem gleichen  $L$ -Wert verschwinden und damit die Annahme gerechtfertigt ist (siehe Abbildungen 4.12 und 4.13).

### 5.3. Teilchenabhängige Qualität des Hintergrundes

Die Untersuchung der Prüfwerte der Proben der bestrahlten FNTDs mit den vorher bestimmten optimalen Parametern zeigte, dass diese schlechter sind, als die vergleichbaren Prüfwerte der unbestrahlten FNTDs (Abschnitt 4.3). Dabei werden die Werte immer schlechter je schwerer die Teilchen sind, mit denen die FNTDs bestrahlt wurden. Die mit  $^1H$ ,  $^{12}C$  und  $^{24}Mg$  bestrahlten FNTDs zeigten dabei akzeptable Abweichungen vom Idealwert der Prüfwerte von weniger als 5%. Für die Proben, die mit schwereren Teilchen bestrahlt wurden, betragen die Abweichungen jedoch bis zu 56% (siehe Tabelle 4.3). Damit scheint die *get.background*-Funktion nur für leichte Proben bis einschließlich  $^{24}Mg$  geeignet zur korrekten Bestimmung des Hintergrundes zu sein.

Da die Prüfwerte der nichtbestrahlten Proben sehr gut waren und bis zu den mit schweren Teilchen bestrahlten Proben sukzessive schlechter wurden, muss dies an Effekten liegen, die mit schwereren Teilchen zunehmen.

Sehr niedrige Countzahlen in unbestrahlten Bereichen, die in vielen der mit schweren Teilchen bestrahlten Proben vorkommen, konnten als Ursache ausgeschlossen werden, da diese auch in einigen mit leichten Teilchen bestrahlten Proben mit guten Prüfwerten vorkamen.

Damit bleibt eine Erklärung für dieses Verhalten bisher noch offen.

Insgesamt bleibt festzuhalten, dass die *get.background*-Funktion mit optimalen  $L$ -Werten den Hintergrund von Proben zufriedenstellend bestimmen kann, die nicht oder mit leichten Teilchen bis einschließlich  $^{24}Mg$  bestrahlt wurden.

Die optimalen  $L$ -Werte konnten dabei gut nach unten abgeschätzt aber mit den hier an-

gewendeten Verfahren nicht absolut sicher bestimmen werden, sondern nur eine Empfehlung hierfür gegeben werden. Bei Proben, die mit schwereren Teilchen bestrahlt wurden, scheidet die Funktion aus bisher noch nicht geklärten Gründen.

# Anhang

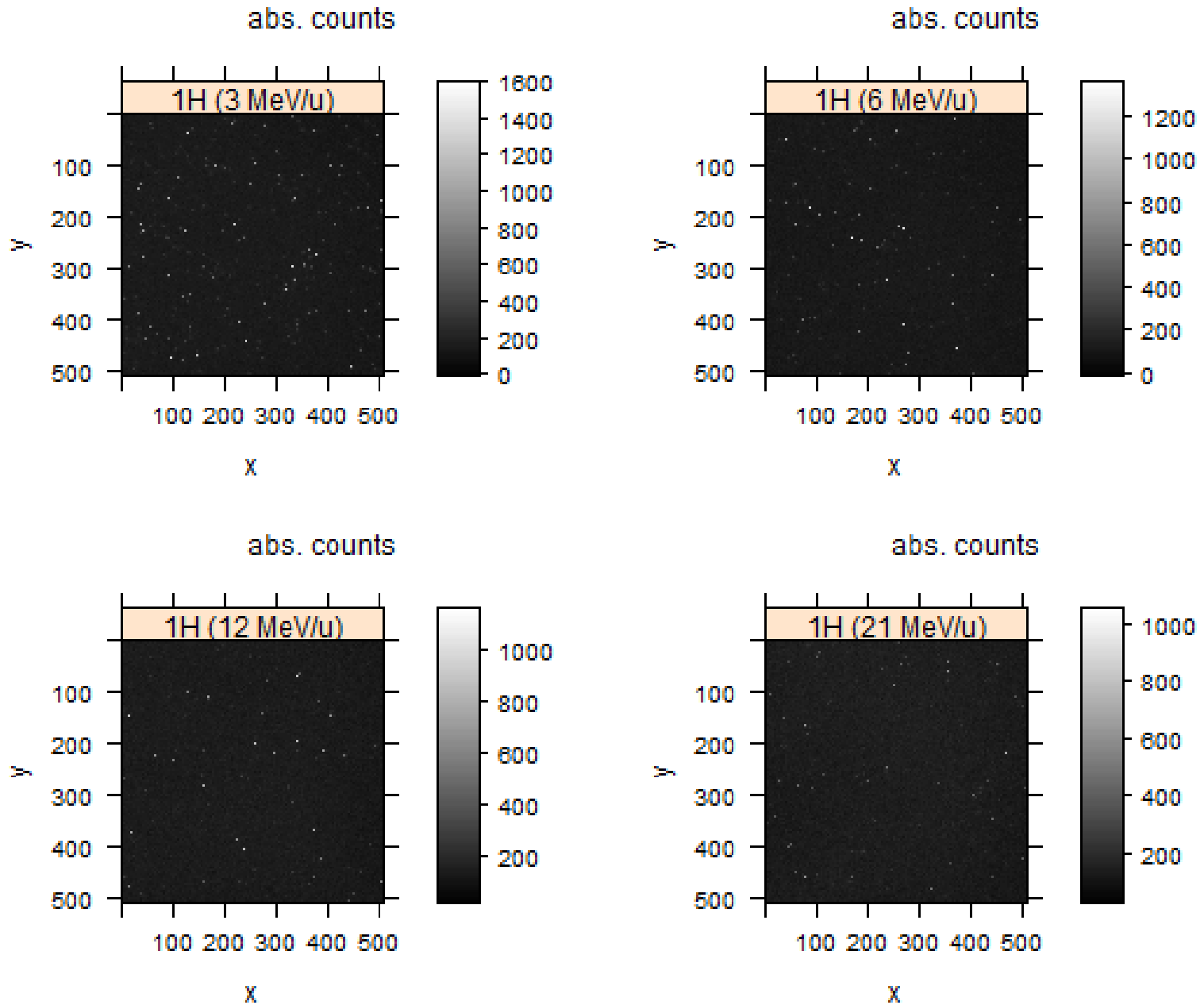


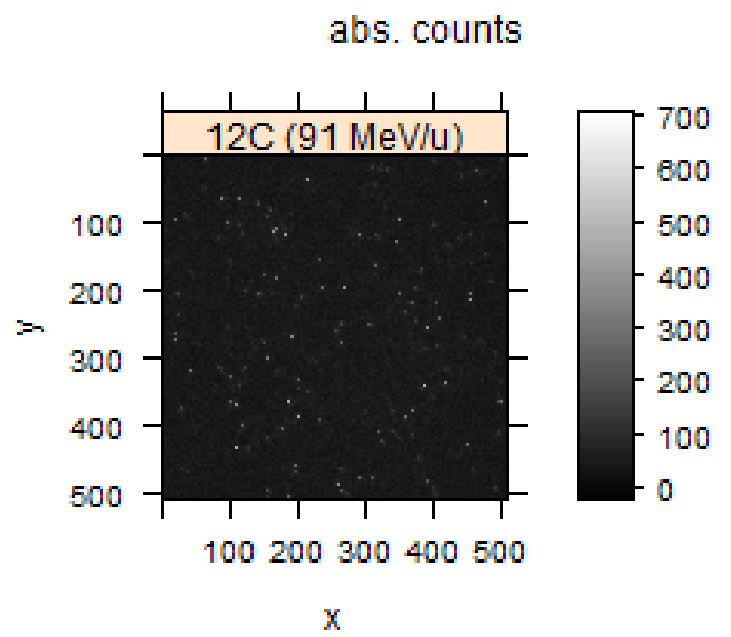
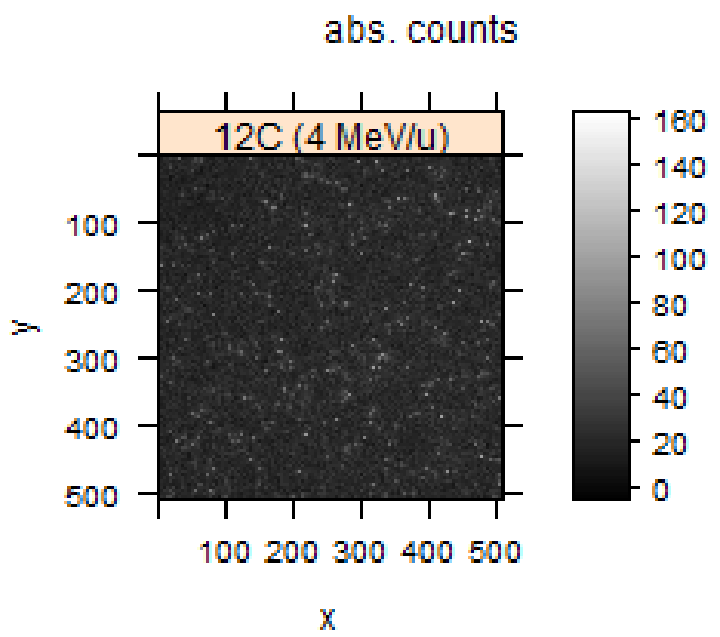
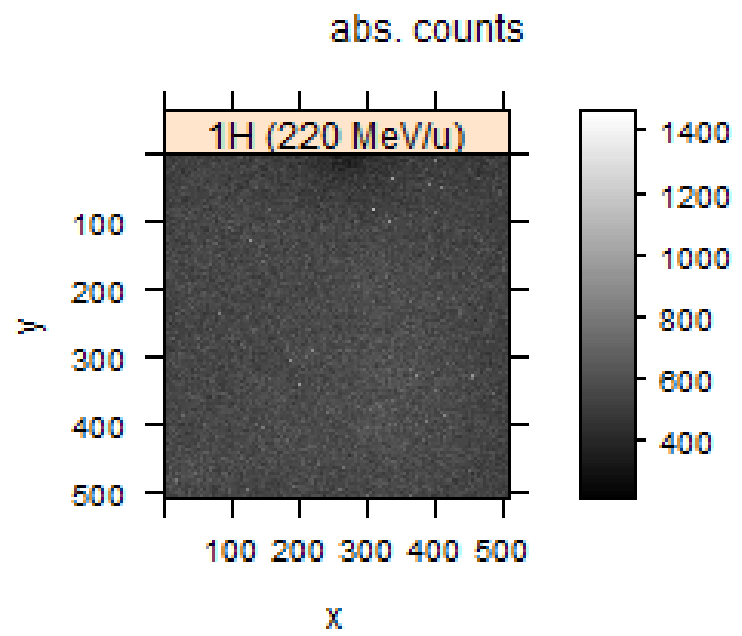
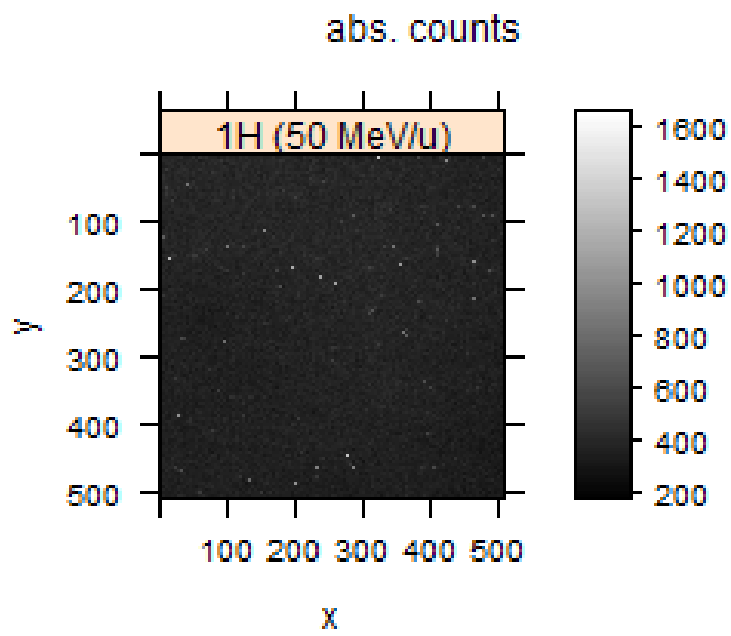
# A. Spektroskopiedatensatz

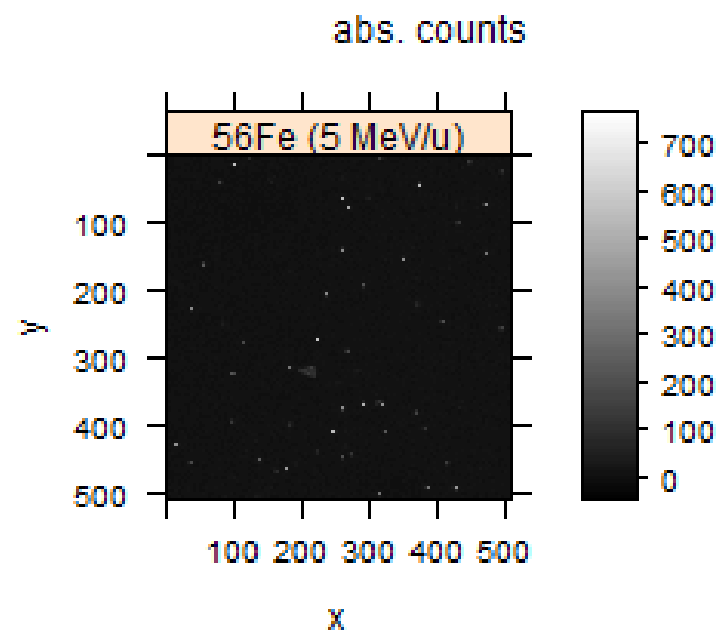
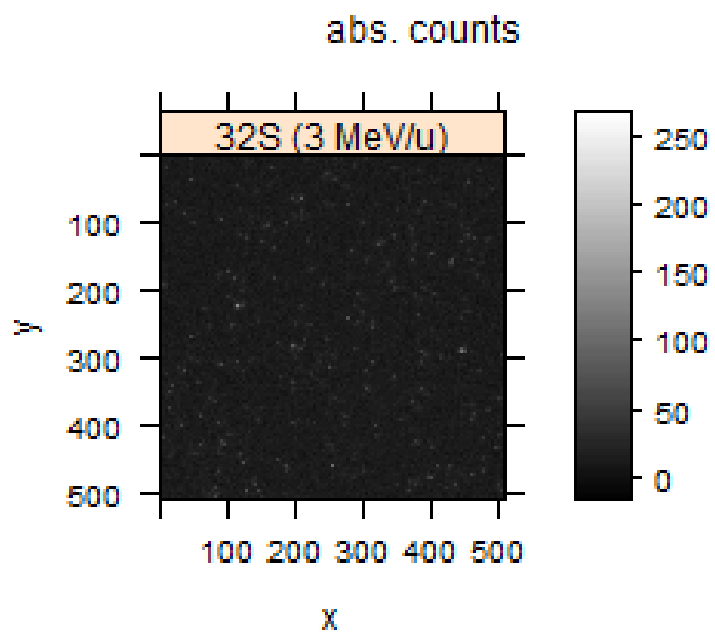
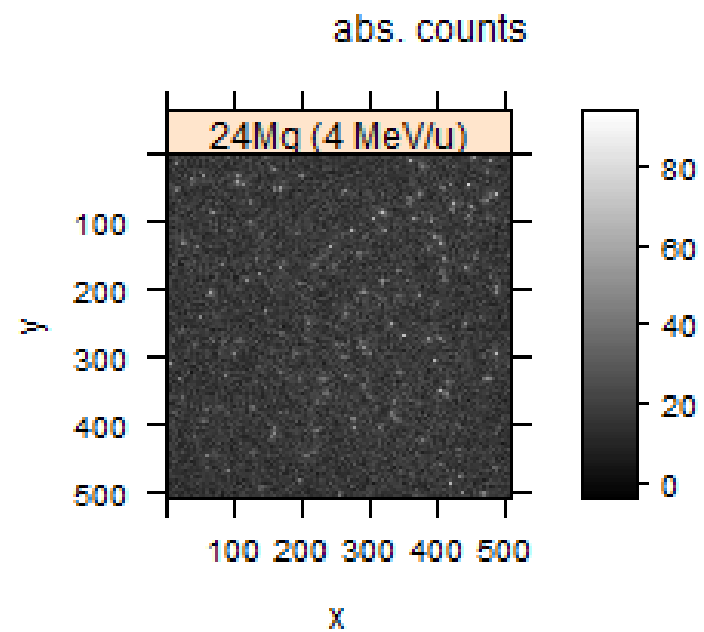
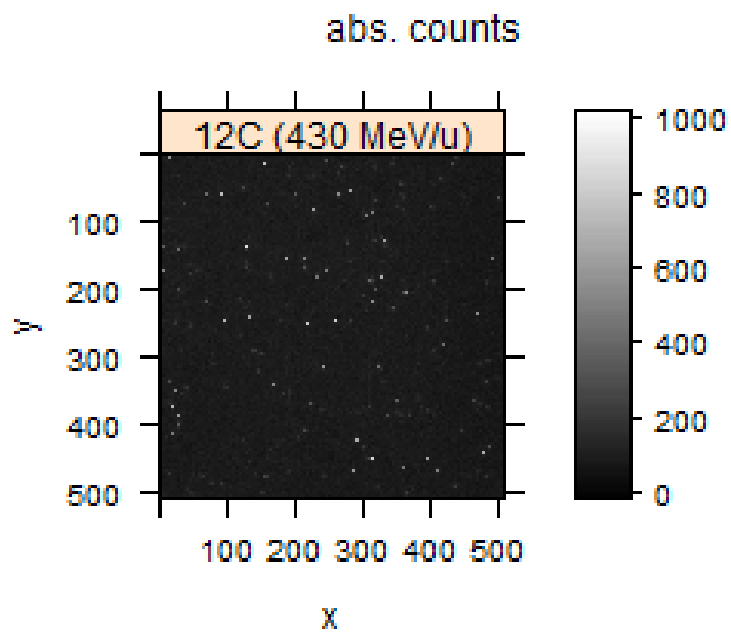
Name	Teilchen	$E$ [MeV/u]	Erstes Bild	Letztes Bild	Mittl. Bild	$\omega$	$T_s$	$r$	$R_{cut}$	$L$
Sg35041_c_spect.stack	1H	3	1	27	14	2	3	0,6	5	10
Sg35043_d_spect.stack	1H	6	1	30	15	2	3	0,6	5	10
Sg35045_a_spect.stack	1H	12	1	30	15	2	3	0,6	5	10
Sg35046_c_spect.stack	1H	21	1	30	15	2	3	0,6	5	10
jmo2011_a_spect.stack	1H	50	1	30	15	2	3	0,6	5	10
jmo2009_spect.nostack	1H	220	1	1	1	2	3	0,4	5	10
jmo3901_d_spectroscopy_512er	12C	4	4	7	6	2	3	1,2	7	10
jmo2007_stack_final	12C	91	60	89	75	2	3	0,2	7	10
jmo2005_a_spect.stack	12C	430	1	30	15	2	3	0,5	7	10
jmo3910_a_spectroscopy	24Mg	4	5	9	7	2	3	0,75	7	10
jmo3907_f_spectroscopy	32S	3	8	12	10	2	3	0,8	7	10
rh16_a_spect.stack	56Fe	5	10	22	16	2	3	0,6	7	10
rh1_a_spect.stack	56Fe	9	5	12	8	2	3	0,6	7	10
rh2_a_spect.stack	84Kr	9	4	12	8	2	3	0,6	10	10
rh11_a_spect.stack	131Xe	5	3	10	6	2	3	0,7	10	10
rh5_a_spect.stack	131Xe	9	6	14	10	2	3	0,5	10	10

Tabelle A.1.: Spektroskopiedatensatz mit Name, Energie  $E$  und Teilchenart der Probe, den Bildernummern des verwendbaren Bereiches, mittlerem Bild, optimalen Particle Tracker Parametern Radius  $\omega$  in Pixel, Cutoff  $T_s$  und Percentile  $r$  in % und den Parametern für manuellen und automatischen Hintergrund  $R_{cut}$  und  $L$  in Pixel.

Abbildung A.1.: Überblick über die Bilder des Spektroskopiedatensatzes









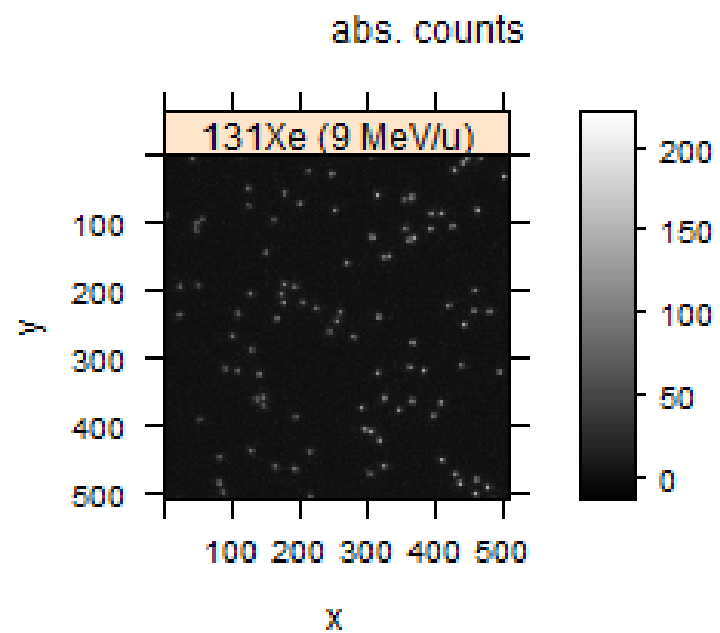
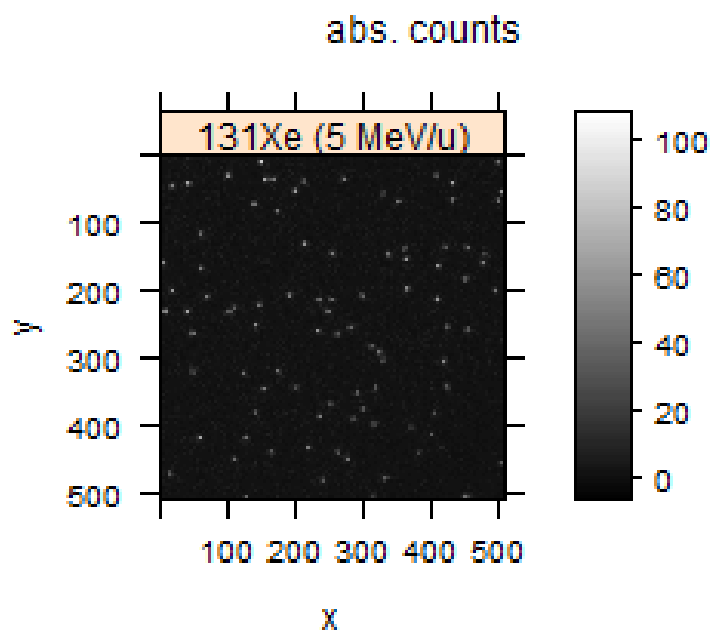
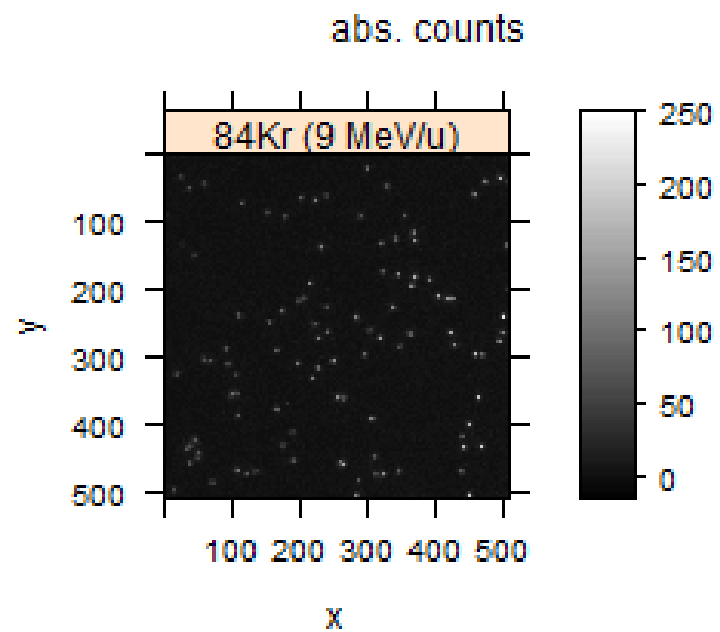
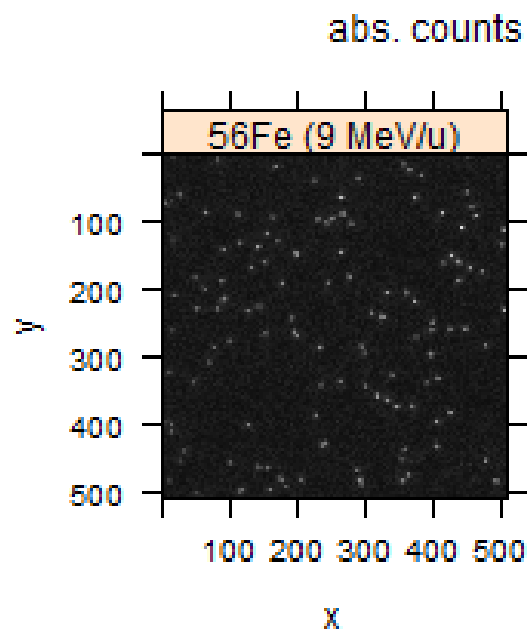
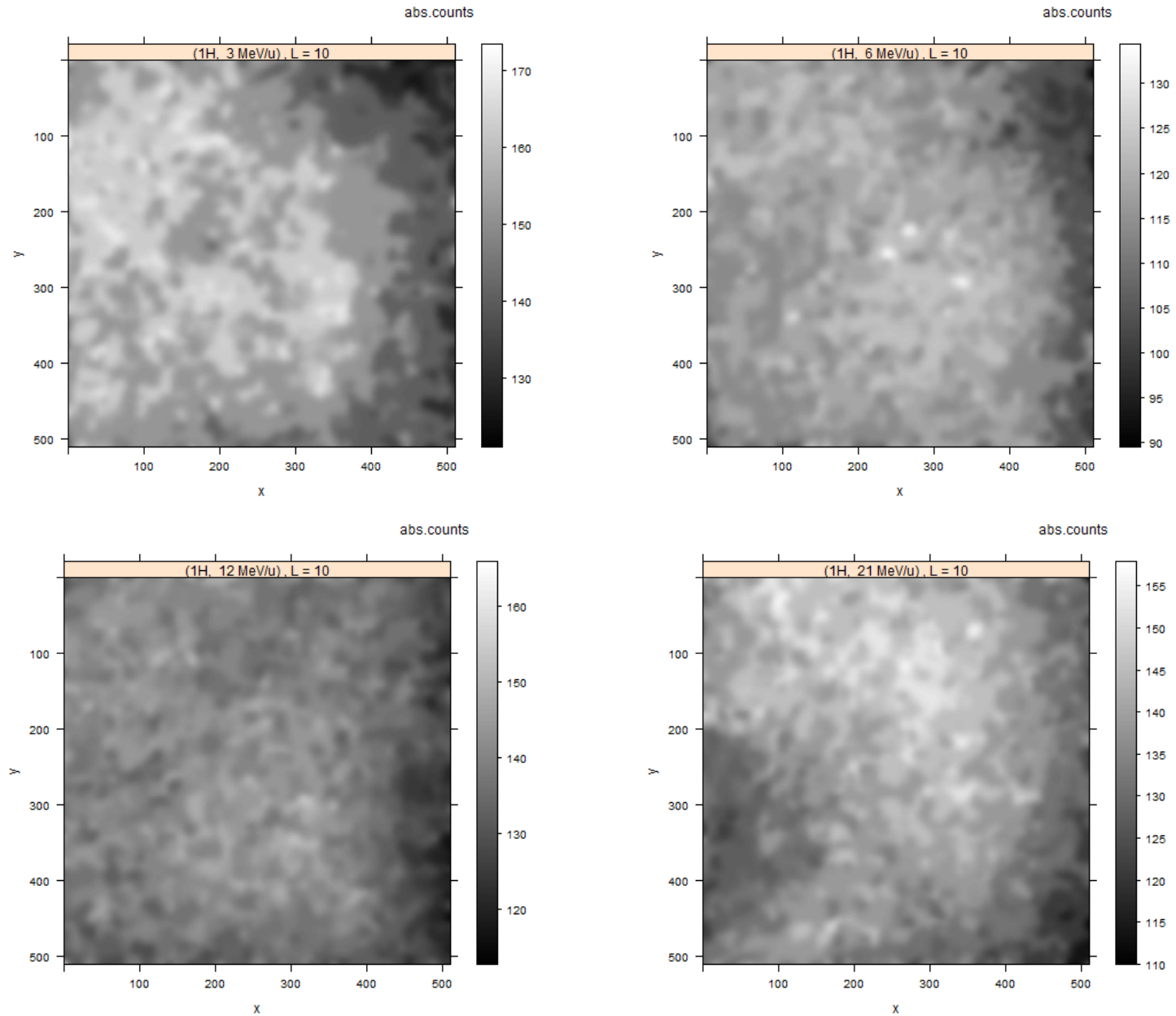
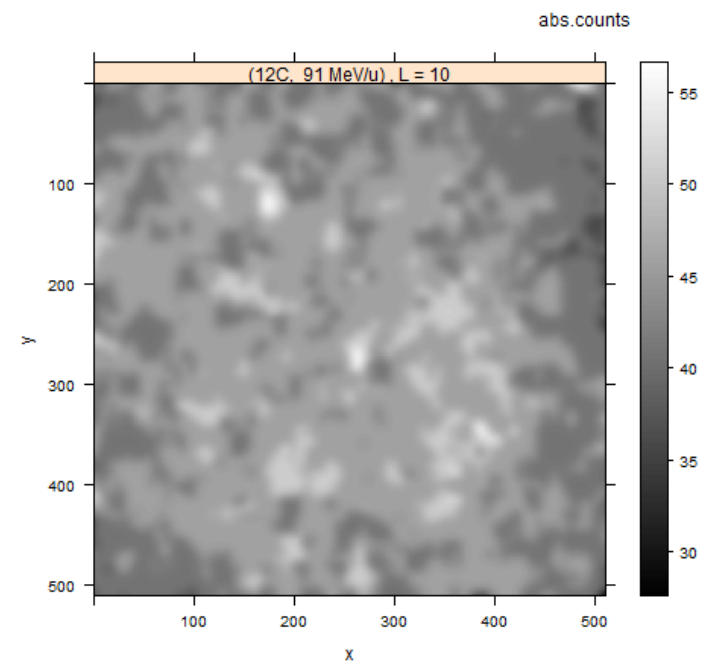
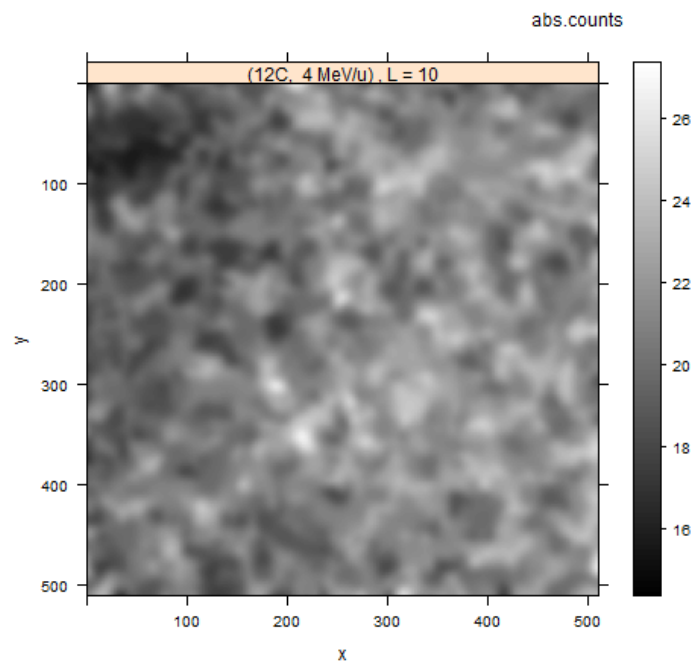
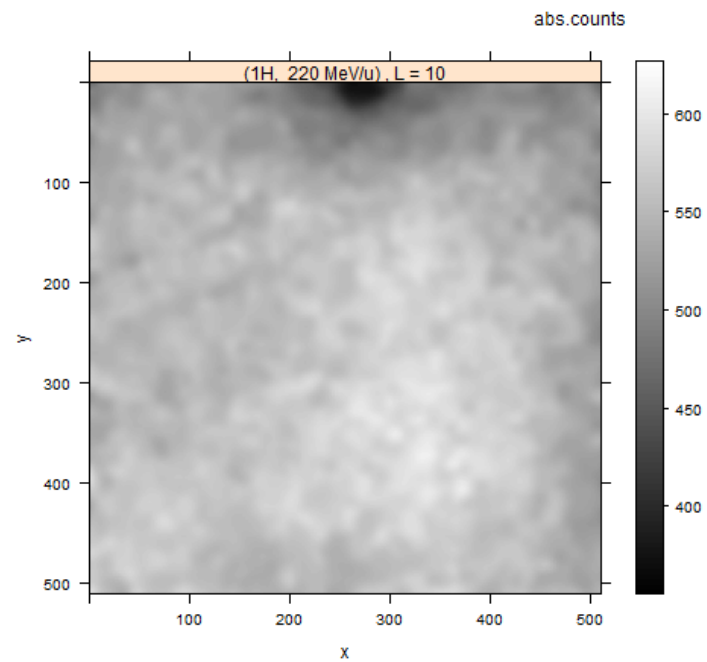
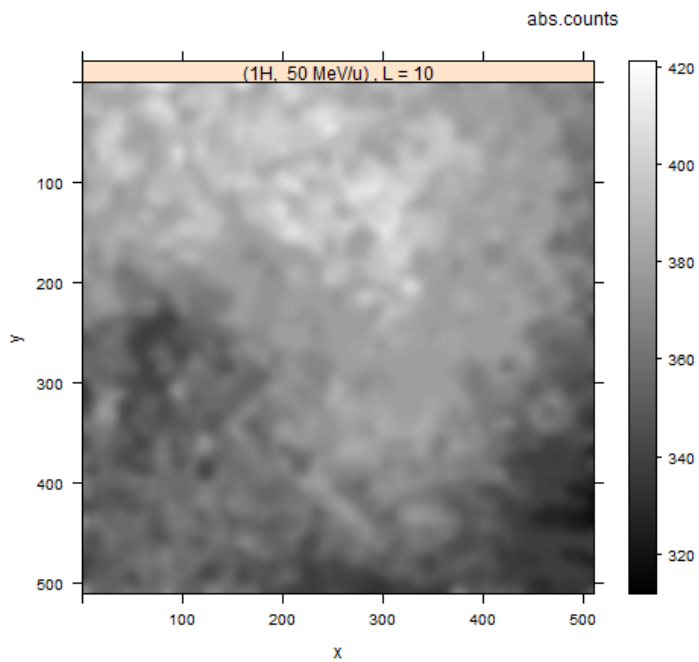
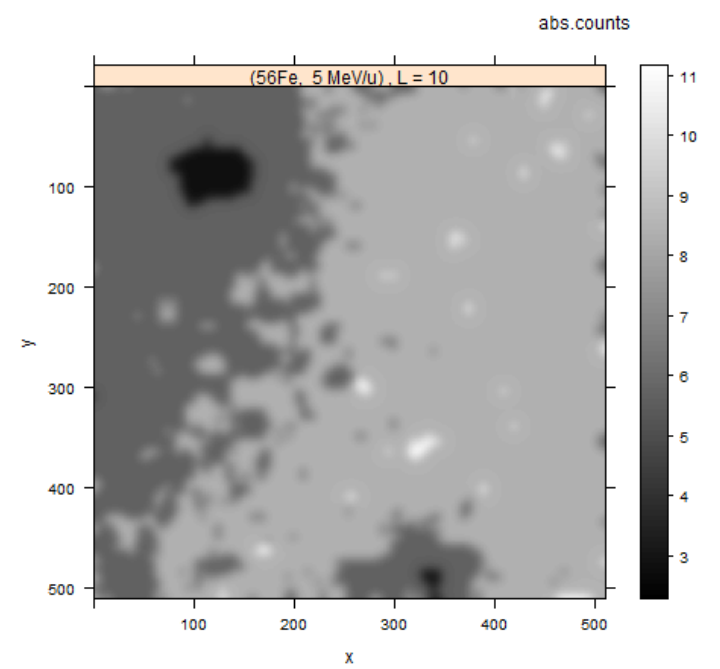
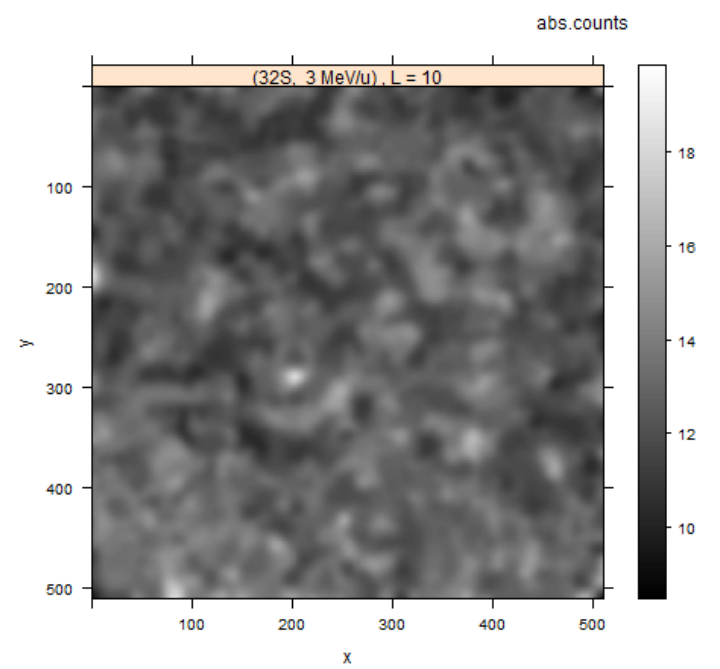
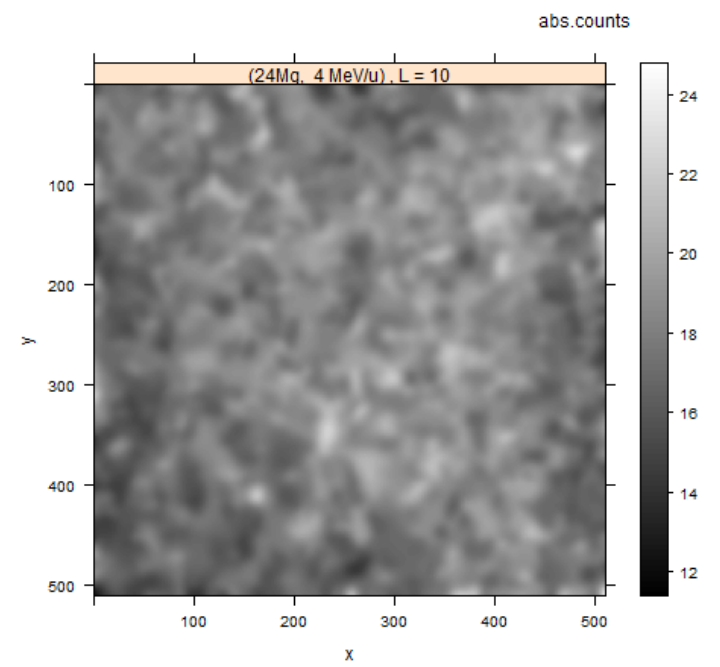
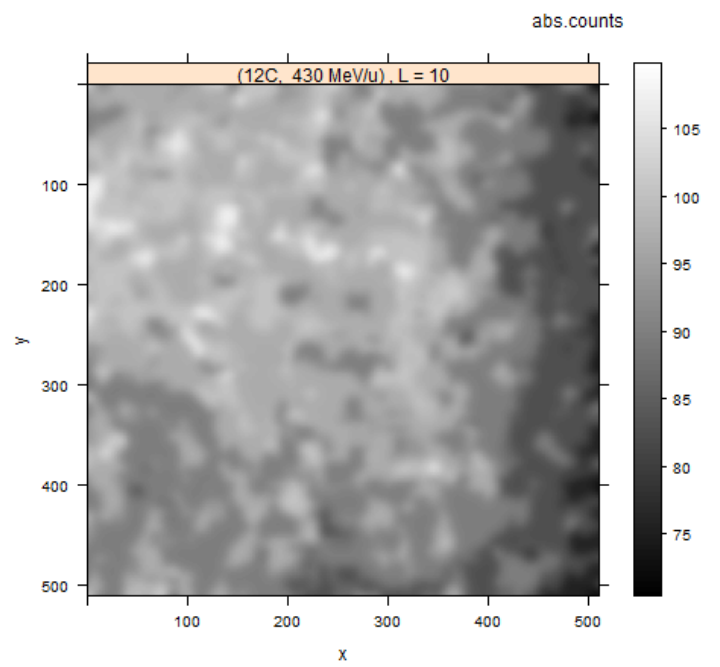
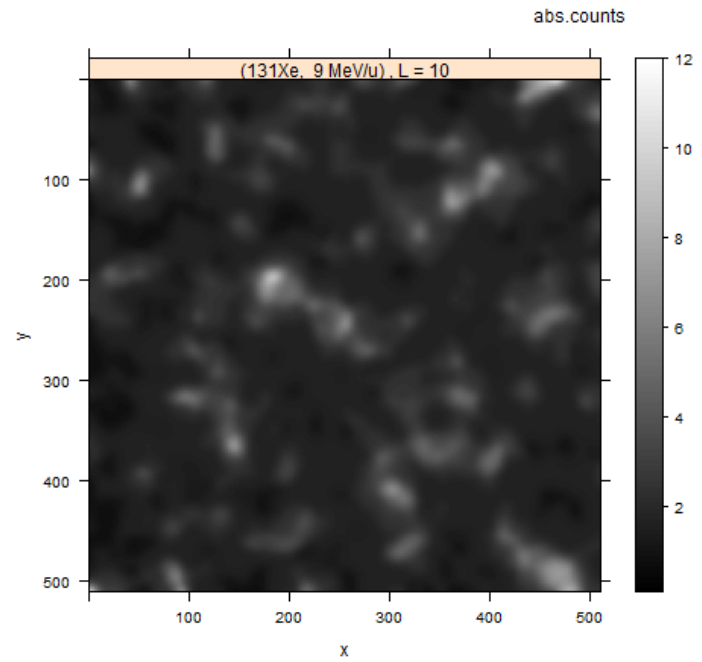
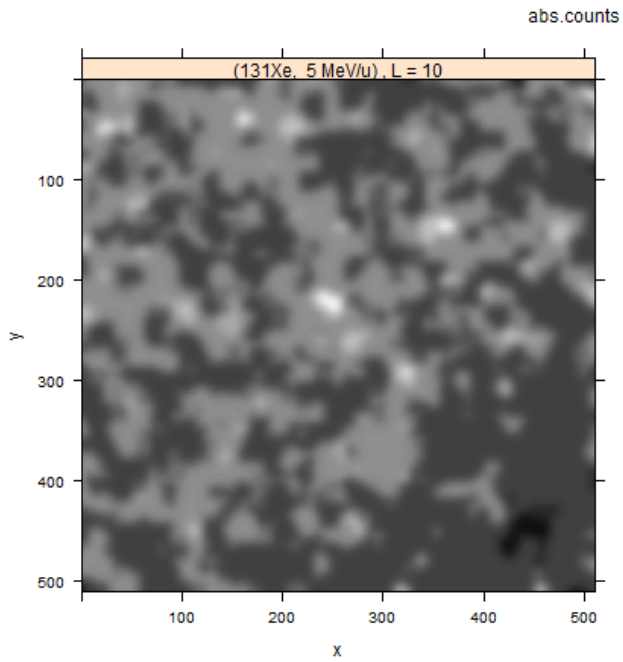
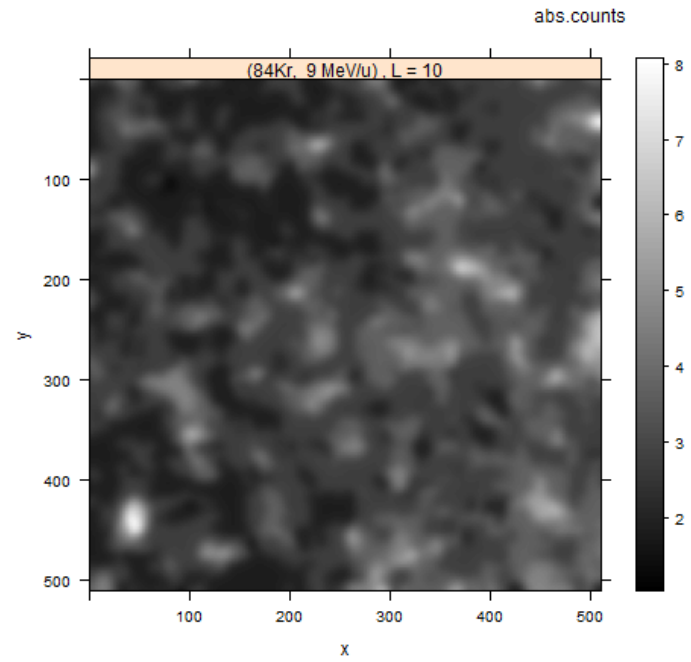
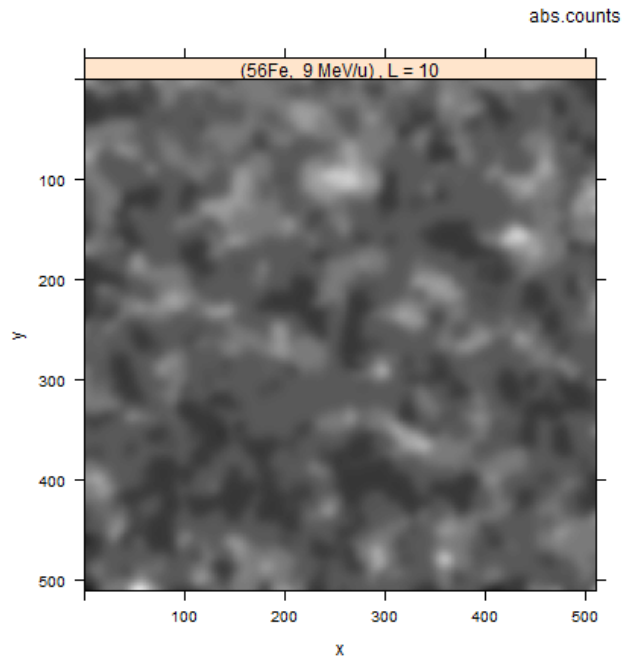


Abbildung A.-1.: Überblick über die Hintergrundbilder des Spektroskopiedatensatzes mit  $L=10$









Probe	$\tau$ [ $\mu\text{s}$ ]	$R$	$p$	$\bar{N}$	$\max(N)$	$\bar{\eta}_{net}$	$\max(\eta_{net})$	$\bar{\eta}_{net}^{adj}$	$\max(\eta_{net}^{adj})$
( 1H, 3 MeV/u)	177.31700	1	100	161.642437	1505	0.90810147	8.4841239	0.90810147	8.4841239
( 1H, 6 MeV/u)	177.31700	1	100	121.814251	1285	0.68348574	7.2434081	0.68348574	7.2434081
( 1H, 12 MeV/u)	177.31700	1	100	142.694183	1094	0.80124057	6.1662412	0.80124057	6.1662412
( 1H, 21 MeV/u)	177.31700	1	100	142.487434	996	0.80007458	5.6135587	0.80007458	5.6135587
( 1H, 50 MeV/u)	177.31700	2	100	378.222618	1569	1.06301539	4.4207797	1.06301539	4.4207797
( 1H, 220 MeV/u)	177.31700	4	100	560.497112	1391	0.78674729	1.9576769	0.78674729	1.9576769
(12C, 4 MeV/u)	177.31700	1	100	22.767113	153	0.12489780	0.8593614	0.12489780	0.8593614
(12C, 91 MeV/u)	50.42424	2	100	48.777885	664	0.48017495	6.5806346	0.48017495	6.5806346
(12C, 430 MeV/u)	177.31700	1	100	98.681442	957	0.55302556	5.3936136	0.55302556	5.3936136
(24Mg, 4 MeV/u)	177.31700	1	100	19.125740	88	0.10436185	0.4927863	0.10436185	0.4927863
(32S, 3 MeV/u)	177.31700	1	100	14.239296	251	0.07680418	1.4120439	0.07680418	1.4120439
(56Fe, 5 MeV/u)	50.42424	1	50	11.070007	718	0.21603740	14.2356827	0.43207481	28.4713654
(56Fe, 9 MeV/u)	50.42424	1	50	4.489990	91	0.08554428	1.8011875	0.17108855	3.6023750
(84Kr, 9 MeV/u)	50.42424	1	50	5.233280	235	0.10028500	4.6569567	0.20057001	9.3139135
(131Xe, 5 MeV/u)	50.42424	1	50	2.644524	102	0.04894548	2.0193365	0.09789096	4.0386731
(131Xe, 9 MeV/u)	50.42424	1	50	4.950142	208	0.09466988	4.1215000	0.18933976	8.2430000

Tabelle A.2.: Metadaten der mittleren Bilder des Datensatzes. Eingetragen ist für jede Probe die *dwell time*  $\tau$  in [ $\mu\text{s}$ ], die *number of rescans*  $R$ , die *laser power*  $p$  in %, die mittleren und maximalen Countzahlen  $\bar{N}$  und  $\max(N)$ , die mittleren und maximalen Nettocoutraten  $\bar{\eta}_{net}$  und  $\max(\eta_{net})$  in [MHz] und die mittlere und maximale angepasste Nettocoutrate  $\bar{\eta}_{net}^{adj}$  und  $\max(\eta_{net}^{adj})$  in [MHz].

# Literaturverzeichnis

- [1] Carl Zeiss Microscopy GmbH: LSM 710 - Die Kraft der Empfindlichkeit.  
[http://www.zeiss.de/sensitivity\\_d](http://www.zeiss.de/sensitivity_d); visited July 5th 2012.
- [2] Heidelberger Ionenstrahl-Therapiezentrum (HIT).  
<http://www.klinikum.uni-heidelberg.de/UEber-uns.124447.0.html>; visited July 5th 2012.
- [3] ImageJ Documentation Wiki: LSM Reader.  
[http://imagejdocu.tudor.lu/doku.php?id=plugin:inputoutput:lsm\\_reader:start](http://imagejdocu.tudor.lu/doku.php?id=plugin:inputoutput:lsm_reader:start); visited July 11th 2012.
- [4] ImageJ: Features.  
<http://rsbweb.nih.gov/ij/features.html>; visited July 11th 2012.
- [5] Max-Planck-Institut für Kernphysik: The heidelberg ion beam facilities.  
<http://www.mpi-hd.mpg.de/blaum/accelerators/index.de.html>; visited July 5th 2012.
- [6] MOSAIC Plugins for ImageJ.  
<http://www.mosaic.ethz.ch/Downloads/ParticleTracker>; visited July 11th 2012.
- [7] R: What is r?  
<http://www.r-project.org/>; visited July 11th 2012.
- [8] *Grundkurs Programmieren in Java*. Hanser, München ; Wien, 6., aktualisierte und erw. Aufl. edition, 2011.
- [9] *Java ist auch eine Insel*. Galileo Press, Bonn, 10., aktualisierte und überarb. Aufl. edition, 2012.
- [10] GM Akselrod, MS Akselrod, ER Benton, and N. Yasuda. A novel  $Al_2O_3$  fluorescent nuclear track detector for heavy charged particles and neutrons. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 247(2):295–306, 2006.
- [11] Janick Cardinale. Manual: Histogram-based background subtractor for ImageJ. Website, 2008. Available online at [http://www.mosaic.ethz.ch/Downloads/ParticleTracker/BGS\\_manual.pdf](http://www.mosaic.ethz.ch/Downloads/ParticleTracker/BGS_manual.pdf); visited on July 11th 2012.

- [12] Carl Zeiss MicroImaging GmbH, Advanced Imaging Microscopy. *Image File Format Description*, March 2006.  
LSM 510, LSM 5 LIVE, LSM 5 PASCAL, Release 4.0.
- [13] F.Lauer. Investigations of a fluorescent nuclear track detector for use in therapeutic ion beams. Diploma thesis, University of Heidelberg, 2011.
- [14] ICRU. *Fundamental Quantities and Units for Ionizing Radiation*. International Commission on Radioation Units and Measurements, 2011. Report 85.
- [15] H. Krieger. Grundlagen der Strahlungsphysik. *Vieweg+ Teubner Verlag*, 2009.
- [16] Greilich, S. , Osinga, J.-M. , Niklas, M. , Lauer, F.M. , Bestvater, F. , Bartz, J.A. , Akselrod, M.S. , Jäkel, O. Quantative read-out of  $Al_2O_3 : C, Mg$ -based fluorescent nuclear track detectors using a commercial confocal microscope. (*in Vorbereitung*).
- [17] I.F. Sbalzarini and P. Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of structural biology*, 151(2):182–195, 2005.



# Danksagung

Als erstes möchte ich mich bei Prof. Dr. Oliver Jäkel dafür bedanken, dass er mich in seine Forschungsgruppe aufgenommen und die Betreuung dieser Arbeit übernommen hat. Darüber hinaus danke ich Prof. Dr. Peter Bachert für seine Bereitschaft die zusätzliche Betreuung dieser Arbeit von Seiten der Fakultät für Physik und Astronomie zu übernehmen.

Ein ganz besonderer Dank geht an Dr. Steffen Greilich, der diese Arbeit mit seiner umfangreichen Unterstützung erst möglich gemacht hat. Die Gespräche mit ihm und seine Ratschläge waren sehr lehrreich für mich. Außerdem danke ich Martin Niklas, der mir ebenfalls mit vielen Erklärungen und Ratschlägen sehr geholfen hat. Weiter danke ich Julia-Maria Osinga, die mir bei der Auslese geholfen und mich ebenfalls mit Tipps unterstützt hat sowie Nora Hünemohr für die Unterstützung bei der Bestrahlung. Darüber hinaus danke ich allen Mitgliedern der Forschungsgruppe *Heavy Ion Therapy* für viele weitere hilfreiche Anregungen und Gespräche.

Schließlich danke ich meiner Familie für ihre permanente Unterstützung, die mir das Studium überhaupt erst ermöglicht hat, und Friederike Krüger für das Korrekturlesen dieser Arbeit.



# Erklärung

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

Heidelberg, den 03.08.2012