

# Table of Contents

Einleitung.....	2
Übertragen der SEL auf Einzelbilder.....	3
Übersicht über den Aufbau der Daten und die Dokumentation des GSI Experimentes.....	3
Übersicht über den Aufbau der Daten und die Dokumentation des LUNTE Experimentes.....	4
Vorbereitungen.....	4
Extrahieren der SEL-Koordinaten aus den *.txt Files.....	6
Einzeichnen der SEL auf den Bildern.....	7
Ansätze zum Zusammenfügen zu einem Panorama.....	8
Hugin.....	8
Alternativer Stitching-Ansatz mit Python Script.....	8
Python – Ansatz über Homographie-Matrizen.....	9
Manuelles Zusammenfügen.....	9
Berechnung des Wirkungsquerschnitts.....	9
GSI – Daten.....	9
LUNTE – Daten.....	10
Auswertung.....	12
Plausibilitätsprüfung Einzelbilder.....	12
Auswertung der Wirkungsquerschnitte.....	12
Zusammenfassung.....	12

# Einleitung

# Übertragen der SEL auf Einzelbilder

## Übersicht über den Aufbau der Daten und die Dokumentation des GSI Experimentes

Der Datensatz besteht aus Bildern und Textdateien, die in mehrere Scanbereiche aufgeteilt sind.

### 1. Mikroskopbilder:

Die Bilder sind nach dem Schema "Jenamicro18\_fs<Spalte><Zeile>.tif" (z.B. Jenamicro18\_fs12.tif) benannt. Sie bestehen aus einer Serie von Mikroskopbildern, die die zu den jeweiligen Scanbereichen korrespondieren.

### 2. Textdateien:

Die Textdateien enthalten Koordinaten, die mit Ionen beschossen wurden. Das Format beinhaltet:

- a. Die ersten 6 Zeilen bilden die Kopfzeile (Metadaten zum Experiment).
- b. Jede Zeile repräsentiert die spezifischen Koordinaten, die beschossen werden, mit 5 Werten:
  1. x-Koordinate in Bezug auf den entsprechenden Scanbereich basierend auf einem 12-Bit-ADC (abweichend von den Bildkoordinaten).
  2. y-Koordinate in Bezug auf den entsprechenden Scanbereich basierend auf einem 12-Bit-ADC (abweichend von den Bildkoordinaten).
  3. LatchUp (0 - kein SEL, 4 - SEL).
  4. Möglicher ADC-Eingang (in diesem Fall nicht zutreffend).
  5. Der Channeltron-Wert (entspricht der Anzahl der emittierten Elektronen).

### 3. Datenorganisation:

Die Daten sind auf 25 Scanbereiche verteilt, die jeweils ein Bild und eine \*.txt-Datei enthalten.

### 4. Dokumentation:

Für die Bilder mit 5-facher Vergrößerung sind folgende Dimensionsinformationen bereitgestellt:

- x: 597px = 900µm
- y: 461px = 700µm

Dimensionsinformationen zur Ionenablenkung:

- $x = 9 \cdot 33,7\mu\text{m} = 303,3\mu\text{m}$

- $y = 9 \cdot 35,7 \mu\text{m} = 321,3 \mu\text{m}$

Diese Übersicht soll ein besseres Verständnis der Datenstruktur für die Mikroskopbilder und Textdateien vermitteln, die nach den oben aufgeführten Spezifikationen organisiert sind und so eine effiziente Nutzung und Analyse der Daten ermöglichen.

## Übersicht über den Aufbau der Daten und die Dokumentation des LUNTE Experimentes

Der Datensatz besteht aus Bildern und csv-Dateien, die jeweils einen von drei, sich überlappende, Vertikalen Streifen darstellen. Diese Daten sind, im Gegensatz zu den GSI Daten, schon aufeinander abgestimmt.

### Vorbereitungen

Um die bereitgestellten \*.dat Dateien in Textdateien umzuwandeln, wird das Perl-Skript "extract\_dat2txt.pl" verwendet, das von GSI bereitgestellt wurde. Die Benutzung des Skripts erfolgt in einem ordnungsgemäß definierten Verzeichnis, um die ordnungsgemäße Funktion des Programms zu gewährleisten.

Die Ordnerstruktur ist wie folgt:

1. Hauptordner, in dem sich die Pythonskripte befinden:

- fun1.py
- LatchUpMap\_v0\_9.py

2. Unterordner innerhalb des Hauptordners:

a. "pics" für die Bilder:

- Mikroskopbilder (Beispielsweise von Elektronikbauteilen oder Halbleitern)
- Latchup Unterordner: Enthält generierte Bilder nach der Ausführung von LatchUpMap\_v0\_9.py.

b. "docs" für die umgewandelten Textdateien:

- Hier werden die Textdateien gespeichert, die aus den ursprünglichen \*.dat Dateien mithilfe von "extract\_dat2txt.pl" generiert wurden.

Um die volle Funktionsfähigkeit des Programms zu garantieren, ist es wichtig, diese Ordnerstruktur strikt einzuhalten. Durch die richtige Organisation können die verschiedenen Skripte und Dateien effizient verarbeitet und genutzt werden, um den gewünschten Workflow zu ermöglichen.

## Extrahieren der SEL-Koordinaten aus den \*.txt Files

Um die SEL auf die Mikroskopbilder zu projizieren müssen die Koordinaten eingelesen und sortiert werden. Dazu werden folgende Schritte ausgeführt:

### 1. Erstellung der Listen mit relevanten Dokumenten:

Es werden zwei separate Listen für die in den Verzeichnissen "docs" und "pics" gefundenen relevanten Dokumente zu erstellen. Relevante Dokumente haben die Dateiendungen ".dat.txt" oder ".tif". Die Dateien im Unterordner "Latchups" sind davon ausgenommen und werden bei diesem Schritt nicht berücksichtigt.

### 2. Ablauf für jedes relevante Dokument in "docs":

Für jedes in Schritt 1 identifizierte relevante Dokument im Verzeichnis "docs" wird folgender Ablauf abgearbeitet:

*a) Zeilenweise einlesen:* Das Dokument wird zeilenweise eingelesen. Jede Zeile wird anschließend einzeln verarbeitet.

*b) Header löschen:* Der Header, also die ersten 6 Zeilen mit Metadaten, wird in diesem Schritt entfernt.

*c) Zeile in 5 Elemente aufteilen:* Die verbleibenden Zeilen des Dokuments werden nun entsprechend dem Aufbau der Daten in fünf separate Elemente aufgeteilt. Damit wird eine Struktur geschaffen, die es ermöglicht, die Informationen später leichter weiterzuverarbeiten.

*d) Prüfung und Speicherung von Latchups:* In diesem Schritt wird überprüft, ob ein Single Event Latchup (SEL) für die jeweilige Zeile vorliegt. Wenn dies der Fall ist, wird die Zeile in der Liste "LatchUps" gespeichert. Diese Liste dient als Sammlung der Latchup-Events, die in den verschiedenen Dokumenten gefunden wurden.

Zusammengefasst besteht der Prozess also aus der Erstellung zweier Listen mit relevanten Dokumenten und der Durchführung bestimmter Operationen wie dem Einlesen, Löschen von Headern und Aufteilung von Zeilen für jedes relevante Dokument im Verzeichnis "docs". Ziel ist es, Latchup-Events extrahieren und in einer separaten Liste zur weiterverarbeitung zu speichern.

**(Verarbeitetes und unverarb. Dokument in Anhang)**

## Einzeichnen der SEL auf den Bildern

Nachdem die SEL aus den Textdateien extrahiert wurden, können sie nun auf die Mikroskopbilder projiziert werden. Dazu werden folgende Schritte ausgeführt:

1. *Finden des korrespondierenden Bildes:* Basierend auf dem Namen des \*.dat.txt Dokuments wird das passende Bild ausgewählt.
2. *Öffnen des Bildes mit OpenCV:* Nachdem das korrespondierende Bild gefunden wurde, wird es mit der OpenCV-Bibliothek geöffnet. OpenCV ermöglicht die Berechnung der Dimensionen des Bildes und konvertiert es in ein Numpy-Array zur weiteren Verarbeitung.
3. *Berechnung der SEL-Koordinaten:* Für jeden SEL im Datensatz werden die genauen Koordinaten im Bild berechnet. Dies bezieht sich auf den Scanbereich des Bildes und liefert die Positionsdaten für die anstehenden Transformationen. Dazu werden die folgenden Formeln verwendet:

$$P_x = \frac{d_x}{ADC} x_{ADC}$$

$$P_y = \frac{d_y}{ADC} y_{ADC}$$

P – Position im Scanbereich in Pixeln  
d – Ablenkung  
 $x_{ADC}/y_{ADC}$  – Position im Scanbereich  
bezogen auf ADC  
ADC – 12bit (4096)

4. *Umrechnung der Positionen in Bildkoordinaten:* Die zuvor berechneten Positionen werden nun in Bildkoordinaten umgerechnet. Dazu werden folgende Schritte durchgeführt:
  1. Spiegelung an der Y-Achse: Die Koordinaten werden so angepasst, dass sie symmetrisch bezüglich der Y-Achse angeordnet sind.
  2. Drehung um 90° gegen den Uhrzeigersinn: Der Koordinatenvektor wird um 90° gegen den Uhrzeigersinn gedreht.
  3. Koordinatentransformation: Transformation von Scan-Bereich-Koordinaten in Bildkoordinaten

$$x_{Bild} = P_{x, neu} + \frac{höhe}{2} - \frac{d_y}{2} \quad y_{Bild} = P_{y, neu} + \frac{breite}{2} - \frac{d_x}{2}$$

5. *Ändern der Pixel im Numpy-Array:* Innerhalb des Bildes (Numpy-Array) werden die Pixelwerte für SEL auf [0, 0, 255] (rot) gesetzt. Optional können auch die Pixelwerte für den Rahmen auf [255, 0, 0] (blau) gesetzt werden um die Analyse zu vereinfachen.
6. *Speichern des modifizierten Bildes:* Schließlich wird das veränderte Bild mithilfe von OpenCV im Ordner "Latchups" gespeichert. Dadurch ist das Bild für die spätere Analyse und Verwendung verfügbar.

**(Einzelbild einfügen)**

# Ansätze zum Zusammenfügen zu einem Panorama

Die neuen Bilder sollen miteinander kombiniert werden, um den Vergleich mit den Ergebnissen von LUNTE zu erleichtern. Außerdem soll geprüft werden, ob der gesamte Mikrocontroller wirklich gescannt wurde oder ob einige Bereiche ausgelassen wurden.

## Hugin

Hugin ist ein FOSS (Free Open Source Software) Stitching Programm zur Erstellung von Panoramas.

Die neu generierten Bilder werden in drei Schritten zusammengefügt:

1. *Einlesen der Bilder:* Die Bilder können per drag and drop in Hugin eingelesen werden.
2. *Korrespondierende Punkte finden:* Korrespondierende Punkte werden zuerst automatisch gefunden und können anschließend manuell bearbeitet werden. Punkte können gelöscht oder hinzugefügt werden. Es wird eine Projektdatei angelegt.
3. *Stitchen:* Die Bilder werden automatisiert zusammengefügt. Das resultierende Panorama wird im Angegebenen Pfad gespeichert.

### (mit hugin generiertes Panorama einfügen)

Bei der Verwendung von Hugin treten folgende Probleme auf:

1. Da die Scanbereiche jeweils nur einen Teil der Mikroskopaufnahme einnehmen gibt es erhebliche Überschneidungen. Da die SEL nur auf einem der Bilder eingezeichnet sind kommt es zum 'Verschlucken' der eingezeichneten SEL und das resultierende Panorama ist nicht zu gebrauchen.
2. Aus dem generierten Projektfile können keine relevanten Informationen zur Transformation der Bilder rekonstruiert werden. Dadurch ist es nicht möglich die SEL nach dem Erstellen des Panorama einzuzeichnen.

Dementsprechend ist Hugin für diese Anwendung nicht geeignet und muss als Methode verworfen werden.

## Alternativer Stitching-Ansatz mit Python Script

Für das Zusammenfügen der Mikroskopbilder wird anstatt Hugin ein Python-Script verwendet. Hierzu wird ein Stitching-Algorithmus aus dem Internet (**Quelle**) eingesetzt.

Allerdings gibt es nach wie vor Probleme, da die SEL trotzdem noch "verschluckt" werden, ähnlich wie bei der Verwendung von Hugin. Deshalb ist es notwendig, die Herangehensweise anzupassen.



## Python – Ansatz über Homographie-Matrizen

Python und OpenCV ermöglichen es mit Hilfe von Homographiematrizen

Koordinatentransformationen zwischen zwei, sich überlappenden, Bildern durchzuführen. Um zwischen den einzelnen Bildern und dem erstellten Panorama Homographiematrizen zu berechnen, müssen folgende Schritte durchgeführt werden:

1. Zuerst müssen korrespondierende Punkte in beiden Bildern gefunden und die "guten" Korrespondenzen gespeichert werden.
2. Im nächsten Schritt wird OpenCV verwendet, um die Homographiematrix zu erstellen.
3. Anschließend werden die SEL-Koordinaten für das Panorama berechnet.

Es gilt:  $x' = Hx$ , wobei  $H$  die Homographiematrix,  $x$  die Koordinaten im Originalbild (Einzelbild) und  $x'$  die transformierten Koordinaten im Panorama sind.

4. Schließlich werden die SELs im Panorama eingezeichnet.

(mit normales und 'extended' Bild einfügen)

Trotz der vielversprechenden Idee gibt es einige Probleme. Zum Beispiel sind die Scanbereiche an der Y-Achse gespiegelt und um 90° gegen den Uhrzeigersinn gedreht. Obwohl das Potential der Idee besteht, ist der Aufwand im Vergleich zum Nutzen zu hoch.

## Manuelles Zusammenfügen

Bei der Manuellen Verarbeitung der von Python erzeugten Bildern müssen diese zunächst auf den Scanbereich zugeschnitten werden. Anschließend müssen die Scanbereiche manuell so ausgerichtet werden, dass sie nahtlos zusammengefügt werden können. Dies erfordert möglicherweise einige Anpassungen und eine sorgfältige Handhabung, um das beste Ergebnis zu erzielen.

(evt. Probleme erwähnen)

(Ergebnis einfügen)

## Berechnung des Wirkungsquerschnitts

### GSI – Daten

Zur Berechnung des Wirkungsquerschnitts der GSI Daten wird ein vorhandenes Skript von Dr.-Ing. Hannes Zöllner verwendet. Um das Skript verwenden zu können muss das Panorama in eine csv Datei umgewandelt werden.

Dazu wird das Bild eingelesen und Pixelweise ausgewertet. Wenn das Pixel rot ist, wird an seiner Position die Impulsenergie der Goldionen in eine Liste eingetragen, ansonsten eine 0. Diese Liste wird anschließend als csv-Datei gespeichert.

## LUNTE – Daten

Zur Berechnung des Wirkungsquerschnitts der LUNTE Daten wird ein vorhandenes Skript von Dr.-Ing. Hannes Zöllner verwendet. Um die beiden Datensätze vergleichbar zu machen müssen vier Schritte ausgeführt werden bevor das Skript ausgeführt werden kann:

1. Panoramas zusammen setzen
2. Größe ändern
3. Überlappende Pixel Löschen
4. Streifen zusammenfügen

### Panoramas zusammen setzen

Die Panoramas werden aus den Bildern genau wie die GSI-Bilder manuell zusammen gesetzt und so eng wie möglich auf die Größe des Chips zugeschnitten.

### Größe ändern

Um die Wirkungsquerschnitte vergleichen zu können, müssen beide Bilder und csv-Dateien die gleiche Größe haben. Es wurde entschieden die Größe der LUNTE Daten anzupassen.

Zum ändern der Größe müssen zuerst die Umrechnungsfaktoren entlang der x- und y-Achse berechnet werden. Dazu werden die Panoramas in einem Pythonscript geöffnet und die Größen ermittelt. Die Umrechnungsfaktoren werden mit folgender Formel berechnet:

$$U_x = \frac{Breite_{GSI}}{Breite_{LUNTE}} \quad U_y = \frac{Höhe_{GSI}}{Höhe_{LUNTE}}$$

Nach der Berechnung der Umrechnungsfaktoren können die csv- und Bilddateien geöffnet werden. Deren Dimensionen werden mit den Umrechnungsfaktoren multipliziert um die neue Größe der Streifen zu berechnen.

Zur tatsächlichen Änderung der Größe wurden in einem ersten Ansatz fertige Funktionen der Bibliotheken numpy und cv2 (openCV) verwendet. Für die csv Datei wurde np.resize verwendet. Es gibt allerdings Bedenken, dass durch das Verwenden dieser Funktion einzelne Pixel 'verschluckt' werden und eine alternative Lösung musste gefunden werden.

Dabei werden folgend Schritte ausgeführt:

1. SEL Koordinaten im originalen Array identifizieren.
2. Erstelle ein neues Array mit den neuen Dimensionen und fülle es mit Nullen.
3. Erstelle eine 2 Dimensionale Liste mit den selben Dimensionen und speichere die Werte aller Elemente des originalen Arrays, die auf das selbe Pixel des neuen Arrays gemappt werden.
4. Für jedes Element in der Liste wird der Minimalwert ausgewählt und in dem neuen Array gespeichert. Das Array wird darauf hin in einer neuen csv Datei im Unterordner '~/csv/resized' gespeichert.

Für die Bilder wird cv2.resize verwendet und das Ergebnis im Unterordner '~/pics/resized' gespeichert. (**Besser erklären welche Algorithmen zur interpolation verwendet werden**)

### Überlappende Pixel Löschen

Um herauszufinden, wie viele Pixel-Spalten sich überlappen werden die skalierten Bilder zu einem Panorama zusammengefügt und dann so zugeschnitten, dass nur die sich überlappenden Bereiche übrig bleiben. Diese können gezählt und anschließend die selbe Breite am rechten und linken Rand der csv-Datei löschen. Eine eventuelle Verschiebung entlang der y-Achse hat keinen Einfluss auf die Berechnung des Wirkungsquerschnitts.

### Csv Dateien zusammen fügen

Zum Zusammenfügen der csv-Dateien werden diese im ersten Schritt eingelesen, und die Dimensionen der einzelnen Arrays ermittelt. Die Dokumente werden Reihenweise zusammengefügt, wenn es zu Unterschieden in der Länge der Arrays kommt werden diese mit Nullen (0) aufgefüllt. Der so entstandene Array wird in eine csv-Datei geschrieben und im Unterordner 'resized' gespeichert.

# **Auswertung**

## **Plausibilitätsprüfung Einzelbilder**

- Vergleich mit Strukturen
- Vergleich mit LUNTE
- Hypothese zu Artefakten

## **Auswertung der Wirkungsquerschnitte**

## **Zusammenfassung**