

## 1 Aufgabenstellung

- Adjustierung des Prismenspektrographen, der Linsen und der Prismen im Spektrographen, um ein Spektrum für alle Lichtquellen abzubilden.
- Aufnahme des Spektrums von den Verschiedenen Lichtquellen und der Probe.
- Aufnahme der Spektren der gleichen Lichtquellen mit dem Gitterspektrographen.
- Erzeugung der Dispersionskurve des Prismenspektrographen.
- Bestimmung des Auflösevermögen der beiden Spektrographen.
- Bestimmung der Wellenlänge der Jod-Absorptionsbandkanten.
- Bestimmung der Dissoziationsenergie des Jodmoleküls.

## 2 Grundlagen

### 2.1 Molekulare Spektroskopie

test

Wenn elektromagnetische Strahlung mit einem Molekül in Wechselwirkung tritt, kann sie absorbiert, emittiert oder gestreut werden, je nach der Energie der Strahlung und den Energieniveaus des Moleküls. Die Energie der Strahlung wird durch ihre Wellenlänge oder Frequenz bestimmt, und die Energieniveaus des Moleküls werden durch seine elektronische Struktur, seine Vibrations- und Rotationsmoden bestimmt.

Die Absorption oder Emission elektromagnetischer Strahlung durch ein Molekül kann als Änderung der Intensität der Strahlung beim Durchgang durch die Probe beobachtet werden. Diese Änderung kann mit einem Spektrometer gemessen und zur Untersuchung der elektronischen Struktur und Bindung des Moleküls sowie seiner Schwingungs- und Rotationsenergieniveaus verwendet werden.

## 2.2 Prismenspektrograph

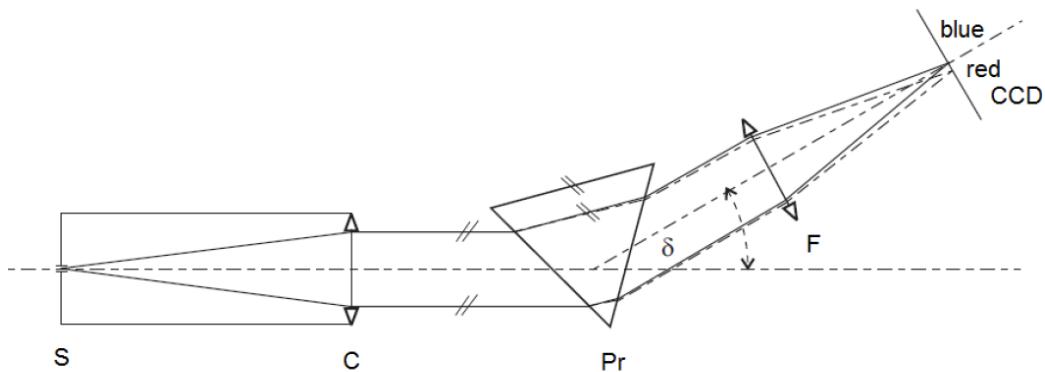
Ein Prismenspektrograph ist ein Spektrograph, der ein Prisma verwendet, um die elektromagnetische Strahlung in ihre einzelnen Wellenlängen oder Frequenzen zu zerlegen. Er besteht aus einer Lichtquelle, einem Prisma und einem Detektor und wird verwendet, um das elektromagnetische Spektrum einer Probe grafisch darzustellen.

Die Lichtquelle eines Prismenspektrographen ist in der Regel eine Lampe oder ein Laser, mit dem die Probe beleuchtet wird. Die Probe wird in den Strahlengang gestellt, und ein Teil der elektromagnetischen Strahlung wird von der Probe absorbiert, emittiert oder gestreut.

Das Prisma ist ein transparentes optisches Element, das aus einem brechenden Material wie Glas oder Quarz besteht. Es wird in den Weg der elektromagnetischen Strahlung gestellt und dient dazu, die Strahlung in ihre einzelnen Wellenlängen oder Frequenzen zu zerlegen. Das Prisma funktioniert, indem es die Strahlung je nach Wellenlänge oder Frequenz in unterschiedlichen Winkeln biegt. Dies führt dazu, dass sich die verschiedenen Wellenlängen oder Frequenzen der Strahlung nach dem Snelliusschen Brechungsgesetz, Gleichung 1, ausbreiten und ein Spektrum bilden.

$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{n_1(\lambda)}{n_2(\lambda)} \quad (1)$$

$\theta_1$  und  $\theta_2$  bezeichnet dabei den Eintritts bzw. Austrittswinkel und  $n_1$  und  $n_2$  die entsprechenden Brechzahlen. Der Strahlengang durch ein Prisma ist in folgender Abbildung 2.1 skizziert.



**Abbildung 2.1:** Skizziert Strahlengang durch ein Prisma, entnommen aus [unterlagen]

S ... Spalt  
C ... Kondensor-Linse  
Pr ... Prisma  
F ... Fokusier-Linse  
CCD ... Schirm  
 $\delta$  ... Brechungswinkel

Der Detektor eines Prismenspektrographen ist in der Regel eine Kamera oder ein Aufzeichnungsgerät, mit dem die Intensität der elektromagnetischen Strahlung über einen Bereich von Wellenlängen oder Frequenzen gemessen wird. Der Detektor erzeugt ein Diagramm der Intensität der Strahlung in Abhängigkeit von der Wellenlänge oder Frequenz, das als Spektrum bezeichnet wird.

## 2.3 Gitterspektrograf

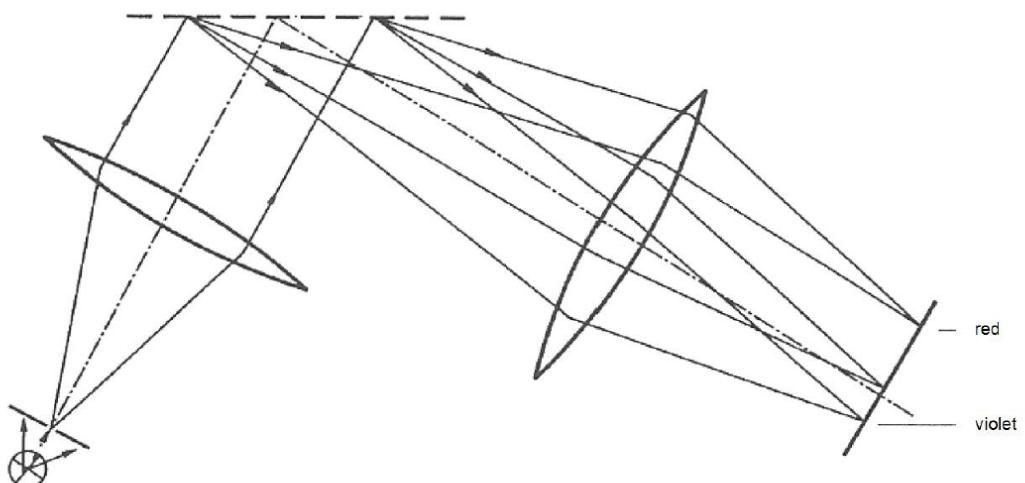
Ein Echelle-Spektrograf ist ein Spektrograf, der ein Beugungsgitter, das so genannte Echelle-Gitter, verwendet, um die elektromagnetische Strahlung in ihre einzelnen Wellenlängen oder Frequenzen zu zerlegen. Er besteht aus einer Lichtquelle, einem Echelle-Gitter, einem Kollimator, einer Kamera oder einem Aufzeichnungsgerät und einer Brennebene.

Bei der Lichtquelle eines Echelle-Spektrographen handelt es sich in der Regel um eine Lampe oder einen Laser, mit dem die Probe beleuchtet wird. Die Probe wird in den Strahlengang gestellt, und ein Teil der elektromagnetischen Strahlung wird von der Probe absorbiert, emittiert oder gestreut.

Das Echelle-Gitter ist ein Beugungsgitter, das aus einem transparenten optischen Element wie Glas oder Quarz besteht und mit einem periodischen Muster aus parallelen Linien versehen ist. Es wird im Strahlengang der elektromagnetischen Strahlung angebracht und dient dazu, die Strahlung in ihre einzelnen Wellenlängen oder Frequenzen zu zerlegen. Das Echelle-Gitter funktioniert, indem es die Strahlung je nach ihrer Wellenlänge oder Frequenz in unterschiedlichen Winkeln, nach Gleichung 2, beugt. Dies führt dazu, dass sich die verschiedenen Wellenlängen oder Frequenzen der Strahlung ausbreiten und ein Spektrum bilden.

$$d \sin \theta = n\lambda \quad (2)$$

$d$  bezeichnet dabei den Gitterabstand,  $\theta$  den Beugungswinkel,  $n$  die entsprechende Ordnung und  $\lambda$  die Wellenlänge. Eine schematische Skizze des Strahlengangs ist in Abbildung 2.2 sichtbar.



**Abbildung 2.2:** Skizzierter Strahlengang durch ein Gitter, entnommen aus [unterlagen]

Der Kollimator ist eine Linse oder ein Spiegel, der dazu dient, die elektromagnetische Strahlung zu kollimieren, d.h. zu einem parallelen Strahl zu bündeln. Der Kollimator befindet sich hinter dem Echelle-Gitter und sorgt dafür, dass die Strahlung auf die Kamera oder den Rekorder fokussiert wird.

Mit der Kamera oder dem Rekorder wird die Intensität der elektromagnetischen Strahlung über einen Bereich von Wellenlängen oder Frequenzen gemessen. Es wird

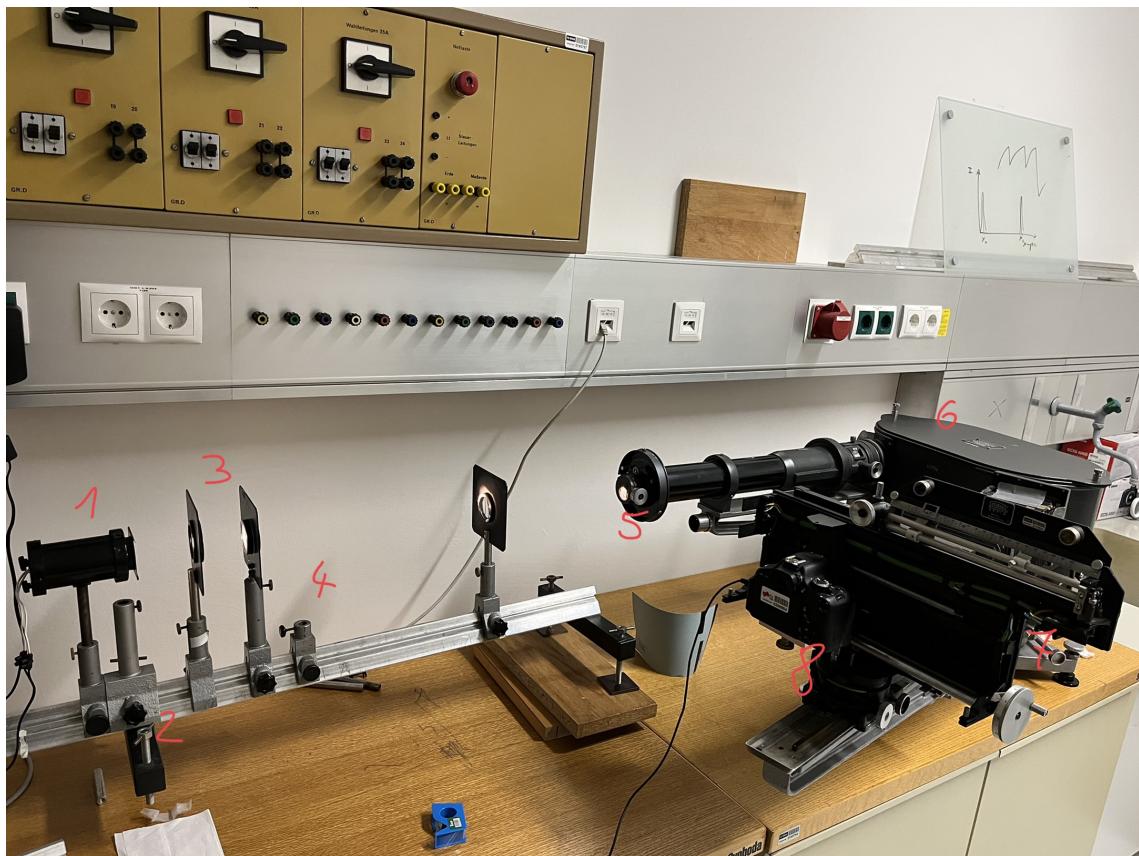
ein Diagramm der Strahlungsintensität in Abhängigkeit von der Wellenlänge oder Frequenz erstellt, das als Spektrum bezeichnet wird.

Die Brennebene ist die Ebene, in der das Spektrum gebildet wird, und sie befindet sich in der Regel im Brennpunkt des Kollimators. Die Kamera oder der Rekorder wird in der Fokusebene platziert, um die Intensität der elektromagnetischen Strahlung zu messen.

## 3 Versuchsaufbau

### 3.1 Prismenspektrograph

Der Versuchsaufbau ist in folgender Abbildung 3.1 ersichtlich. Dabei wird die Position der Lichtquelle mit einem leeren Linsenhalter auf der optischen Bank markiert, um später beim Wechsel der Lichtquelle die selbe Position zu ermöglichen. Bei den Linsen wird dabei versucht, den vorliegenden Lichtstrahl, so gut wie möglich, auf den Eingangsspalt des Spektrographen zu fokussieren, um eine möglichst hohe Intensität zu gewährleisten. Auch die Position, auf die die Iodzelle gestellt wird, wird mit einem leeren Linsenhalter markiert.



**Abbildung 3.1:** Versuchsaufbau des Prismenspektrographen

- 1 ... Lichtquelle
- 2 ... Markierung für die Lichtquelle
- 3 ... Verwendete Sammellinsen (Brennweiten: v.l.n.r. 250, 50, 120)
- 4 ... Markierung für die Iodquelle
- 5 ... Eintrittsspalt in den Spektrographen
- 6 ... Prismen in Spektrographen
- 7 ... Halterung für Kamera
- 8 ... Befestigte Kamera

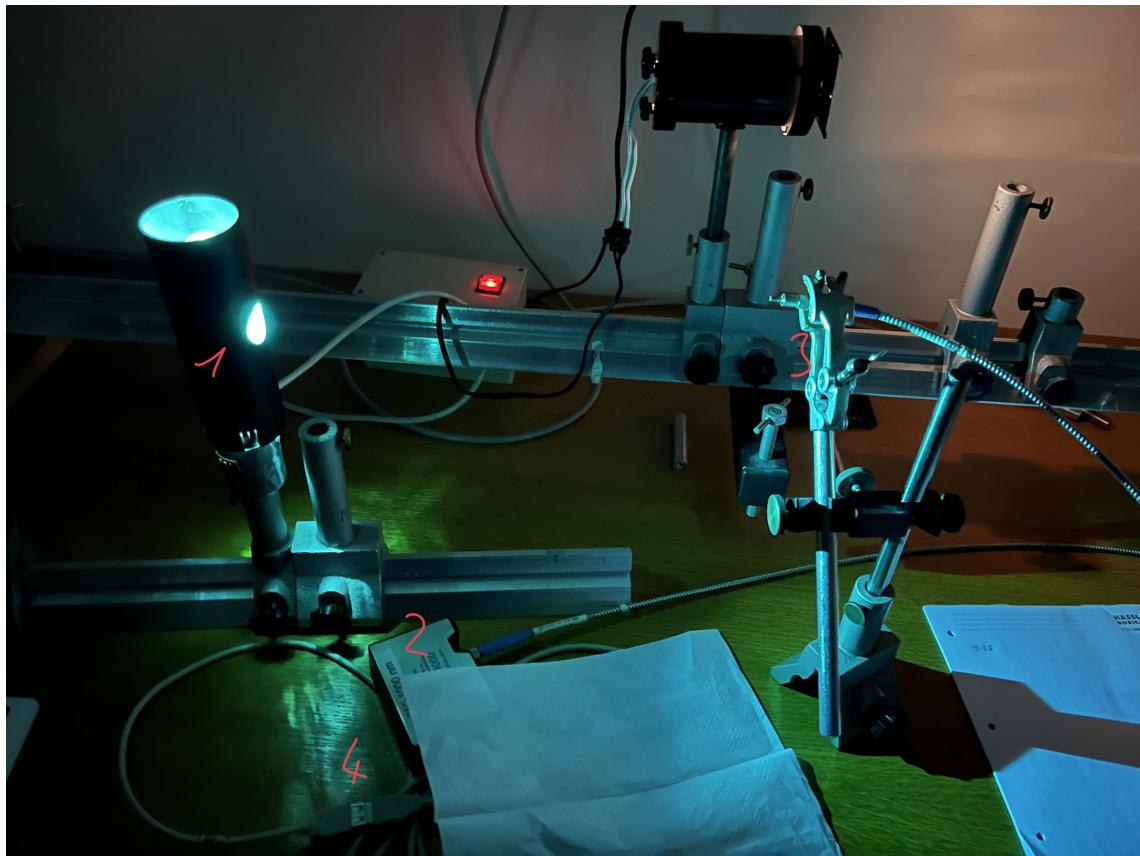
Im inneren des Spektrographen müssen nun die Prismen durch vorsichtiges Verdrehen richtig eingestellt werden, sodass der eintreffende Lichtstrahl durch die 3 Prismen zur Ausgangsöffnung abgelenkt wird und dort das komplette Spektrum, innerhalb des Schirms sichtbar ist. Um dies gut beurteilen zu können, wird ein Streifen Klebeband über die entsprechende Öffnung geklebt, sodass das gebrochene Licht gut sichtbar wird. Die entsprechenden Positionen der Prismen sind in Abbildung 3.2 sichtbar.



**Abbildung 3.2:** Anordnung der Prismen im Spektrographen

### 3.2 Gitterspektograph

Der Aufbau des Gitterspektrographen ist in Abbildung 3.3 sichtbar. Dabei wird der Sensor des Gitterspektrographen mithilfe einer passenden Halterung direkt vor die Lichtquelle gestellt.



**Abbildung 3.3:** Versuchsaufbau des Gitterspektrographen

- 1 ... Lichtquelle
- 2 ... Gitterspektograph
- 3 ... Sensor der Spektrographen
- 4 ... Schnittstelle für Computer

## 4 Geräteliste

Für den Versuch werden die in Tabelle 4.1 aufgelisteten Geräte verwendet.

**Tabelle 4.1:** Verwendete Geräte

Gerätetyp	Hersteller	Typ	Inventar-Nr	Anmerkung
Prismaspektograph	Steinheil		0153624	
Gitterspektograph	Ocean Optics	USB4000	0101417	350-1000 nm
Kamera	Canon	EOS 600D	0206884	ohne Linse
Lichtleiter	Ocean Optics	OP1000-2-UV-BX		
Hg-Dampflampe		Hg-Cd	F21	
Halogenlampe	Leybold		F4	
Iodprobe	Spindler & Hoyer		V / 154d	
Sammellinsen		f = 250, 50, 120		3 x
Optische Bank				höhenverstellbar
Linsenhalter				

## 5 Durchführung und Messergebnisse

### 5.1 Prismenspektrograph

Nachdem die Linsen und Prismen wie bereits in Abschnitt 3 angeführt, aufgebaut wurden, wird das, von der Kamera erfasste, Spektrum am Computer mit der Software SpectraSuite betrachtet. Es ist dafür zu sorgen, dass die einzelnen Linien der Lichtquelle scharf erscheinen und eine möglichst gute Nutzung der gesamten Breite des zur Verfügung stehenden Schirms vorliegt. Dies wird durch Verstellen und anschließender Feinjustierung der Neigung und des Abstands der Kamera zum Spektographen erreicht. Um das abgebildete Spektrum sichtbar zu machen, wird ein Streifen Klebeband an der entsprechenden Stelle des Spektrographen angebracht. Um dafür zu sorgen, dass kein Störlicht auf die Kamera gelangt, wird das gesamte Raumlicht abgedunkelt und ein Karton seitlich an die Kamera gehalten, um kein Licht seitlich von der Lichtquelle zu bekommen.

Weil nicht das gesamte Spektrum mit einem Foto aufgezeichnet werden kann wird zunächst bestimmt, wie groß der erfasste Bereich ist. Dazu wird zunächst die Kamera mithilfe der entsprechenden Kurbel so lange bewegt, bis der rote Peak gerade am rechten Rand des Bildschirms verschwindet. Nun wird die Kamera, unter Bestimmung der entsprechenden Distanz, solange weiterbewegt, bis sich diese rote Linie am linken Rand des Bildschirms befindet. Dadurch wird die Distanz des aufgezeichneten Spektrums bestimmt. Nun wird auf diese Art das gesamte Spektrum aufgezeichnet,



**Abbildung 5.1:** Das Software Interface für die Kontrolle der Kamera

welches schließlich Bild für Bild zusammengesetzt werden kann. Dabei ist zu beachten, dass das Spektrum immer von beiden Lichtquellen und der Iodprobe festgehalten wird, bevor die Kameraposition verschoben wird. Weil bei den verschiedenen Farbeindrücken unterschiedliche Helligkeiten vorliegen, muss auch immer der Kontrast neu eingestellt werden. Ein Bild eines Teils des Spektrums in entsprechenden Computerprogramm ist symbolisch in folgender Abbildung 5.1 sichtbar.

Um den Versuch reproduzieren zu können, sollte exakt der gleiche Spektrograph verwendet werden. Alle abgelesenen und eingestellten Werte sind in folgender Tabelle 5.1 ersichtlich. Weil die Messwerte nicht immer klar von der entsprechenden Skala ablesbar waren wurde dies bei den Unsicherheiten berücksichtigt und in der Tabelle in der entsprechenden Spalte vermerkt.

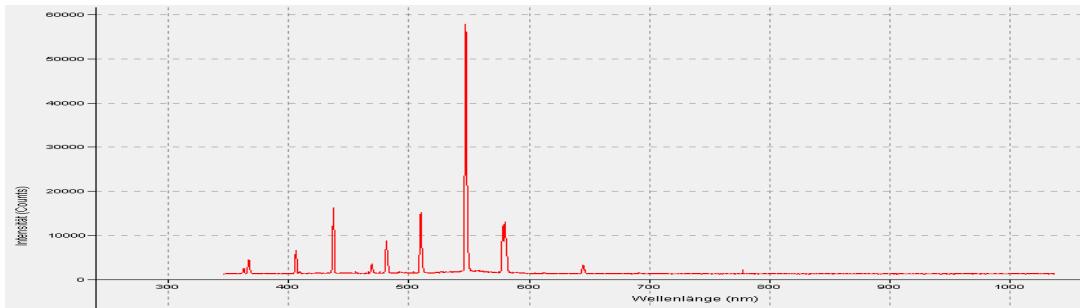
**Tabelle 5.1:** Abgelesene Einstellungen

- $f_i$  ... Brennweite der i-ten Sammellinse in mm
- $f_k$  ... Brennweite des Kollimators
- $SP$  ... Spaltbreite des Eingangsspalt
- $x$  ... Entfernung der Kamera
- $N_{Pi}$  ... Neigung des i-ten Prismas (nicht in °)
- $N_{PT}$  ... Neigung des Tisches in °
- $N_{PK}$  ... Neigung der Kamera in °

Bezeichnung	Messwert	Anmerkung zur Unsicherheit
$f_1$	250 mm	Implizit gegeben
$f_2$	50 mm	Implizit gegeben
$f_3$	120 mm	Implizit gegeben
$f_k$	$(141,6 \pm 0,1) \mu\text{m}$	Skala
$SP$	$(80 \pm 1) \mu\text{m}$	Mikrometerschraube
$x$	$(43,2 \pm 0,1) \text{ mm}$	Skala
$N_{P1}$	$(65 \pm 2)$	Geschätzt mit Entfernungsmessung
$N_{P2}$	$(10,5 \pm 0,5)$	Abgelesen auf entsprechender Skala
$N_{P3}$	$(6,5 \pm 0,2)$	Abgelesen auf entsprechender Skala
$N_{PT}$	$(10,88 \pm 0,02)^\circ$	Skala
$N_{PK}$	$(20 \pm 5)^\circ$	Augenmaß

## 5.2 Gitterspektograph

Um das Spektrum mithilfe des Gitterspektrographen zu analysieren, wird der Aufbau aus Abschnitt 3 in den Strahlengang der entsprechenden Probe gestellt. Der Computer wird über die entsprechende Verbindung angeschlossen und das Auswertungsprogramm gestartet. Bei der genauen Position des Sensors wird dabei darauf geachtet, dass die verzeichneten Peaks möglichst groß werden, diese aber nicht saturieren. Ein Ausschnitt dieser Einstellungen im Programm ist dabei symbolisch in Abbildung 5.2 sichtbar.



**Abbildung 5.2:** Aufnahme des Referenzspektrums mittels des Gitterspektrographen und Software

## 6 Auswertung

Um zu sehen wie sich die Unsicherheit der Messungen bis in die Ergebnisse fortpflanzt, ist erweiterte Gauss-Methode verwendet worden. Die Grundlagen dieser Methode stammen von den Powerpointfolien von GUM [[wolfgang\\_kessel\\_isobipm-gum\\_2004](#)]. Für die Auswertung ist die Programmiersprache Python im speziellen die Pakete `labtool-ex2`, `pandas`, `sympy`, `lmfit` zur Hilfe genommen worden. `lmfit` wurde für das Fitten hergenommen, `sympy` wurde für symbolische Manipulation hergenommen und die restlichen Pakete für leichteres handhaben der Daten. Dies wurde aber alles durch `labtool-ex2` abstrahiert.

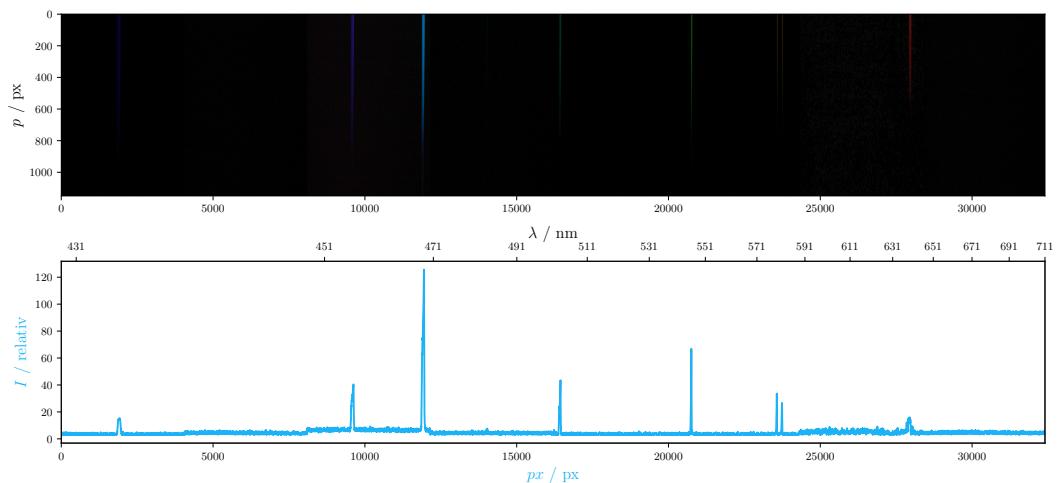
### 6.1 Prismenspektrograph

Zunächst galt es die aufgenommenen Bilder in ein gesamtes Spektrum zusammenzufügen. Dazu wurde bei Abschnitt 5 sichergestellt, dass die Aufnahmen einen Überlapp besitzen welcher groß genug ist, damit etwaige Ausrichtungsfehler bzw nichtbelichteten Anteile des Bildes weggescchnitten werden können. Um herauszufinden wie groß Überlapp und somit der Spielraum für Korrektur finden zu können wurde mittels GIMP, eine Gelben Spektrallinie im zweiten und dritten Bild deren Position gemessen und im Python-Skript eingefüllert. Die Differenz ist genau die Erhaltene Breite eins dieser Bilder. Diese Bildbreite ist jedoch dynamisch von ganz links nach ganz rechts verschiebbar. Durch Zusammenfügen ist herausgefunden worden, dass ganz links die besten Resultate, mit den niedrigsten Sprüngen, bei den Übergängen erzeugt. Da die Bilder auf Empfehlung sehr kurz Belichtet wurden, musste die Intensität mittels einer Gamma-Korrektur geboostet werden. Damit die Peaks im Quecksilberspektrum noch besser ersichtlich, wurde eine CLAHE Algorithm auf die Bilddaten angewendet, wodurch der Noise reduziert wurde aber die Peaks erhalten

bleiben. Schlussendlich wurden diese Bilder nun zusammengesetzt. Dadurch sind folgende Spektren:

### 6.1.1 Spektrogramme des Prisma

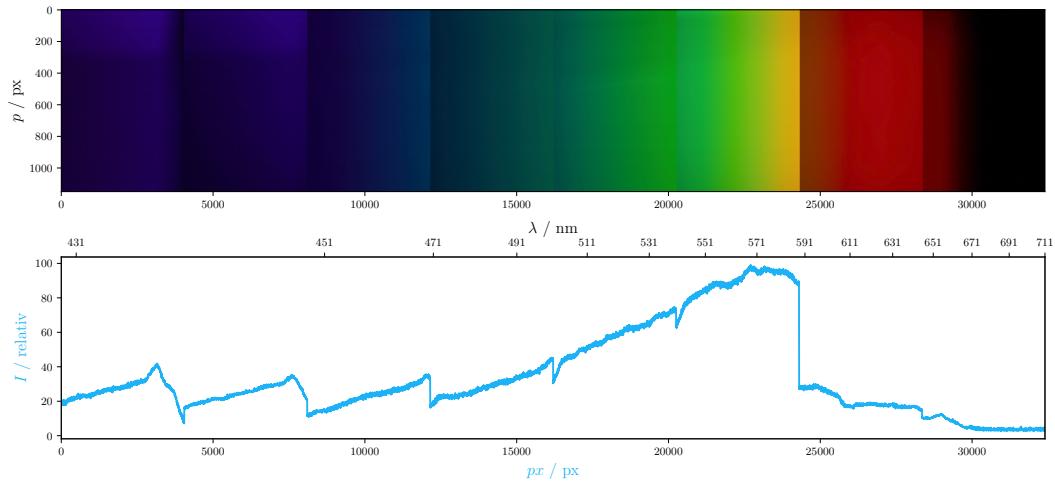
Zunächst wird das gemessene Kalibrierungsspektrum einer Quecksilber-Cadmium Lampe dargestellt, siehe Abbildung 6.1.



**Abbildung 6.1:** Diese Graphik beinhaltet das Farb und die Intensitätsspektrum einer Hg-Cd-Lampe und stellt das Kalibrierungsspektrum für folgende Berechnungen mit dem Prismenspektrographen dar.

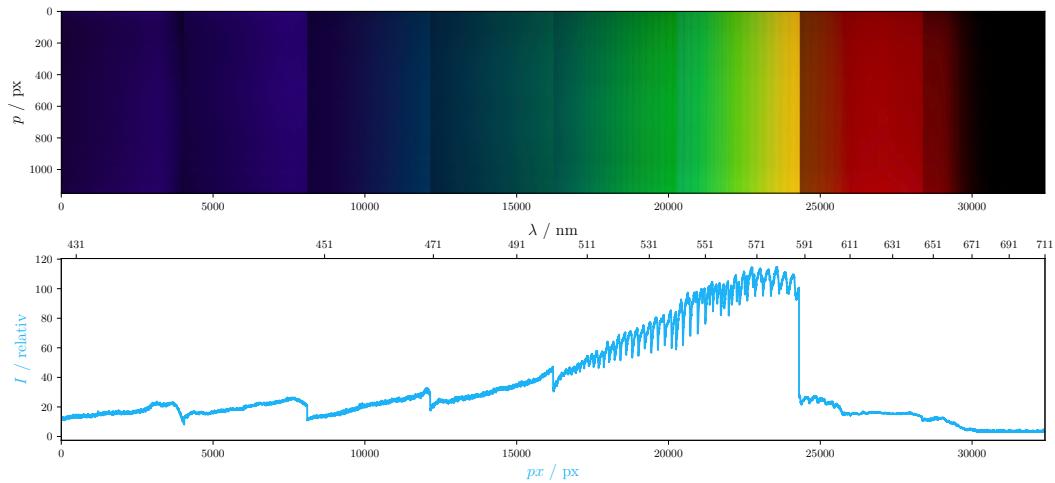
Dieses Spektrum wird verwendet um die Dispersionrelation des Prismas zu finden und einen Isomorphismus zwischen Pixel und Nanometer erstellen. Unter Verwendung dieses Isomorphismus wurde den Pixeln in diesen Graphiken auch eine Wellenlänge zugeordnet. Mehr dazu in Unterunterabschnitt 6.1.2.

Nun wird das Halogenspektrum dargestellt, siehe Abbildung 6.2.



**Abbildung 6.2:** Diese Graphik beinhaltet das Farb und die Intensitätsspektrum einer Halogenlampe und stellt das Referenzspektrum für das Iodspektrum dar.

Zuletzt nun das Spektrum von Iod beleuchtet mit der Halogenlampe, siehe Abbildung 6.3.

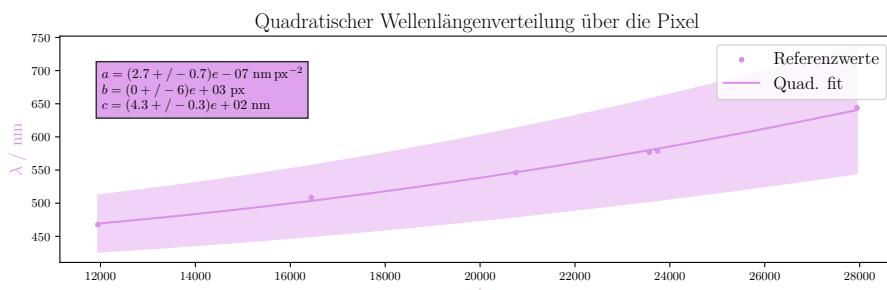


**Abbildung 6.3:** Diese Graphik beinhaltet das Farb und die Intensitätsspektrum einer Halogenlampe mit Iodmolekülen im Abbildungspfad

### 6.1.2 Dispersionskurve des Prismenspektrographen

Durch Identifikation der Peaks im Referenzspektrum und der Zuordnung mit den bekannten Wellenlängen dieser Peaks, aus der Angabe [], ist es möglich die Dispersionskurve mittels eines quadratischen Fit (in Scheitelpunktform  $\lambda = a(x - b)^2 + c$ ) zu beschreiben. Da jedoch nur gewisse Wellenlängen der Peaks bekannt sind und der Cyan Peak bei unserer Messung nicht aufscheint, sind folgende Wellenlängen und deren zugeordneten Peaks fürs Fitten genommen worden.

1. Rot 643,9 nm
2. Gelb 1 579,0 nm
3. Gelb 2 576,9 nm
4. Grün 546,0 nm
5. Türkis 508,5 nm
6. Blau 467,8 nm

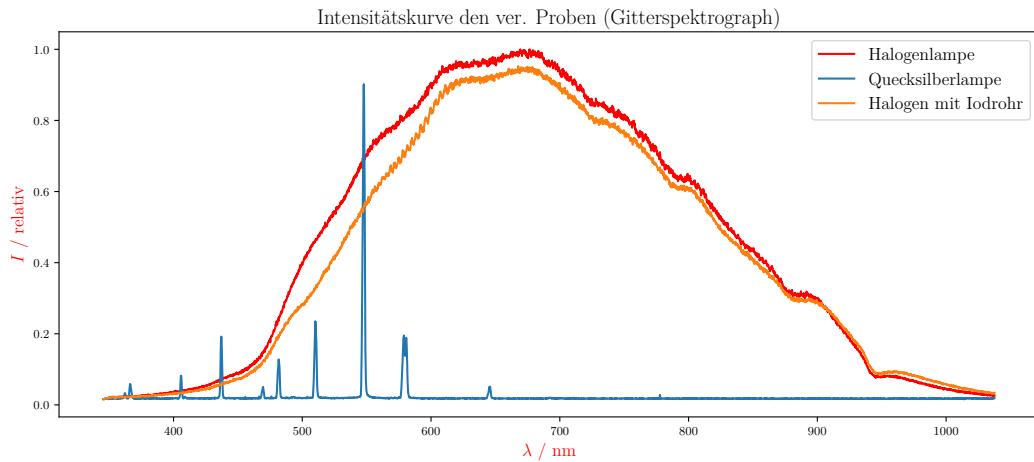


**Abbildung 6.4:** Diese Graphik beschreibt das Dispersion der verschiedenen in Pixeln und stellt die Dispersionrelation zwischen Pixel zu Wellenlängen dar.

Diese gefittet Funktion wird nun, wann auch immer beim Prismenspektrometer von Pixel zu Nanometer und dessen Inverse wenn Nanometer zu Pixel umgerechnet werden muss.

## 6.2 Gitterspektrometer

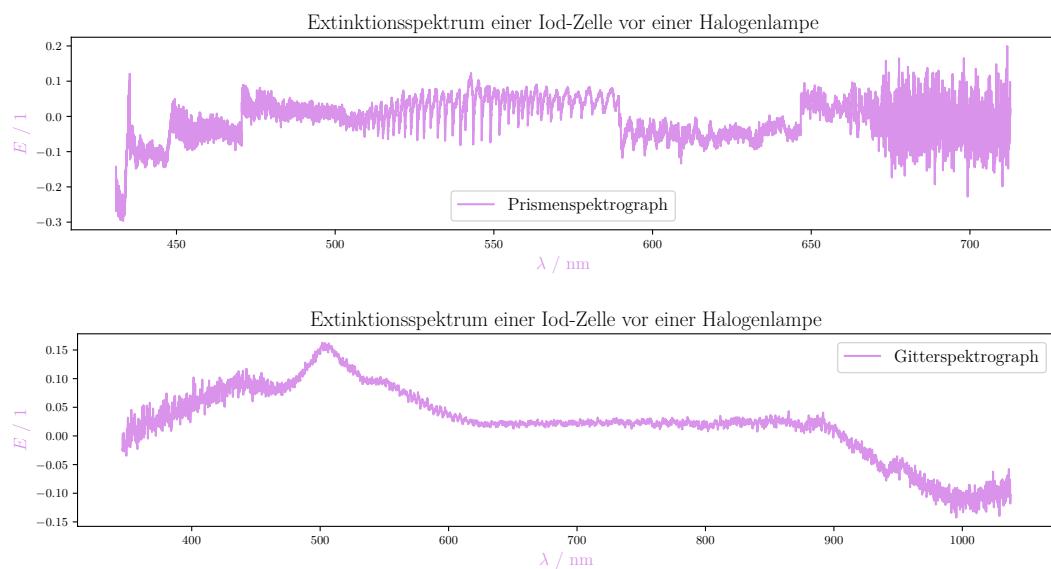
Hier werden nun die Intensität-Spektren der verschiedenen Belichtungen des Gitterspektrometer dargestellt.



**Abbildung 6.5:** Diese Graphik beinhaltet die gemessenen Spektren für die Hg-Cd Lampe, der Halogenlampe und der Halogenlampe mit dem Iod vorgeschoben.

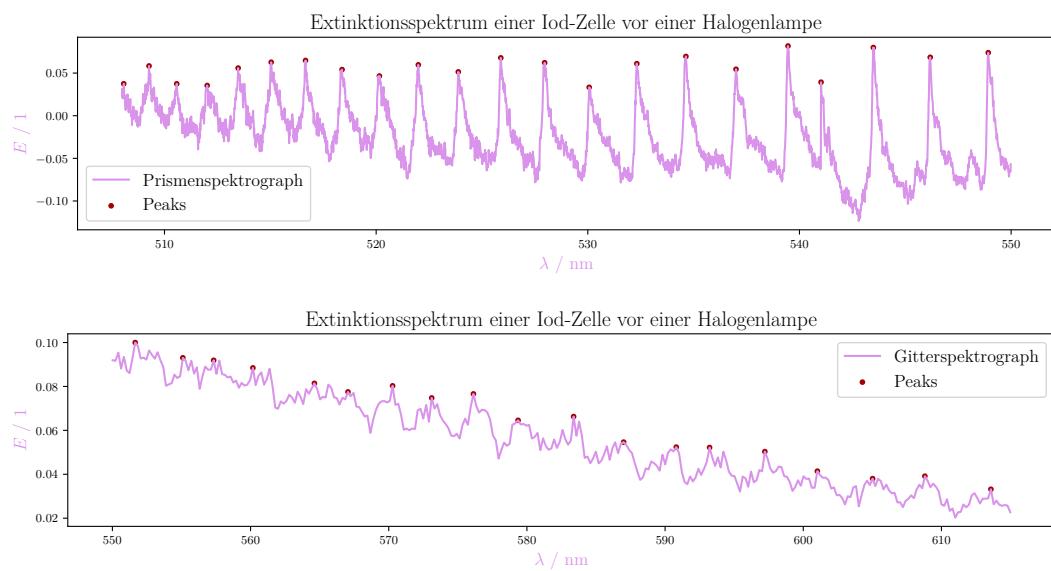
### 6.2.1 Wellenlänge der Jod-Absorptionsbandkanten

Um die Valleys im Absorptionsband von Iod besser finden zu können wird die Extinktion gebildet indem das Halogenspektrum als Referenzspektrum dient.



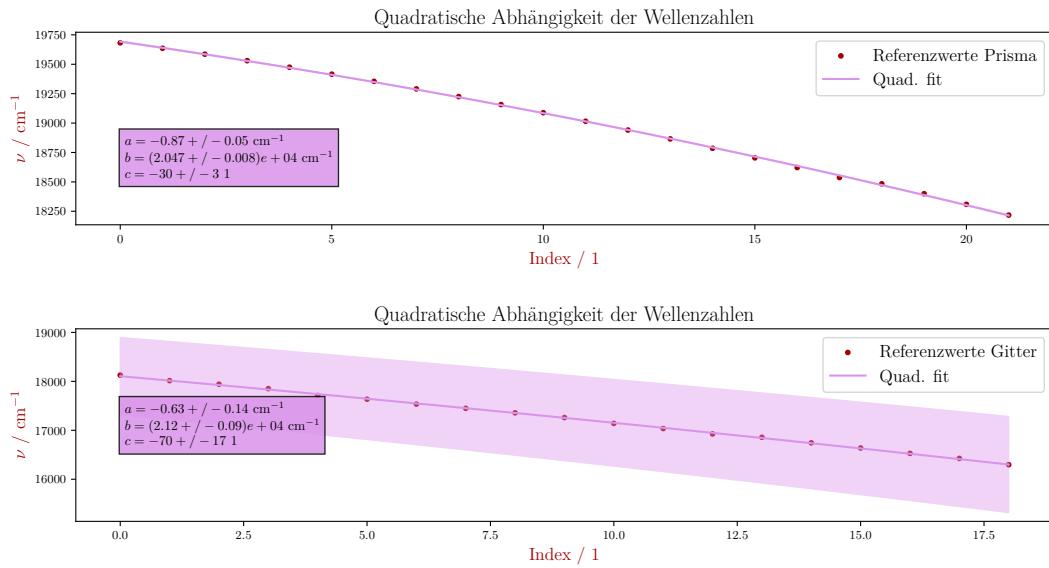
**Abbildung 6.6:** Diese Graphik beinhaltet die Extinktionskurven des Prismenspektrographen (oben) und des Gitterspektrographen (unten) und wurde mittels der Abbildung 6.5 Abbildung 6.2 und Abbildung 6.3 erstellt.

Zudem wird die Extinktion negiert damit die Valleys zu markante Peaks werden, welche leicht mittels Vorhanden Peak-Finding-Algorithmen gefunden werden kann. Dabei wurde sich auf markante Bereich der Extinktionsspektren fokussiert, wo die Absorption klar ersichtlich sind. Damit ist die einfache Findung und Bearbeitung der Peaks gewährleistet.



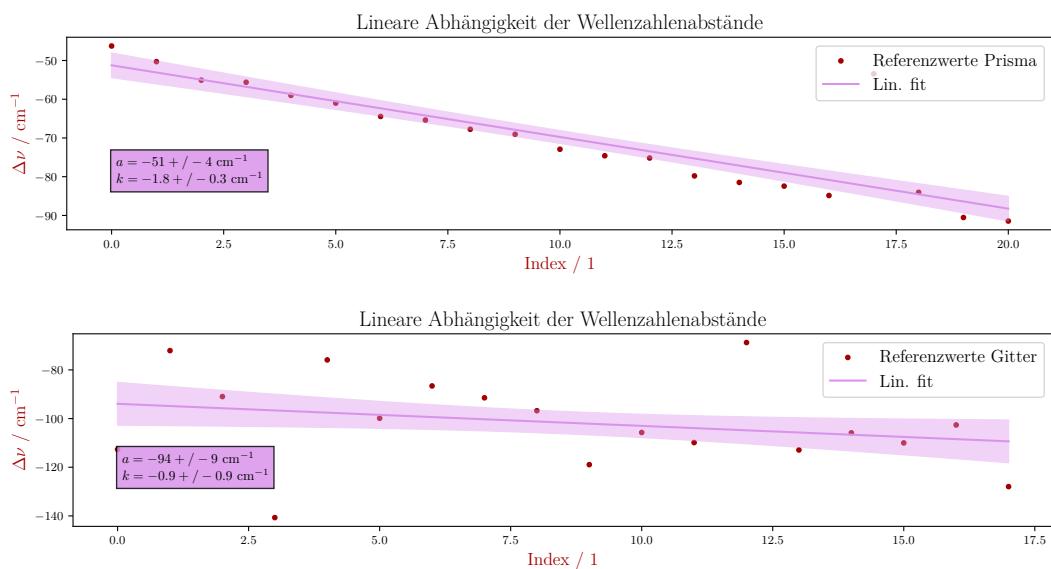
**Abbildung 6.7:** Diese Graphik beinhaltet Ausschnitte aus der Extinktionskurven des Prismenspektrographen (oben) und des Gitterspektrographen (unten), welche markante Peaks aufweisen.

Die Wellenlängen der Peaks lassen nochmals, Aufgrund des Deslandre'sche Gesetzes, mittels eines quadratischen Fits (in Scheitelpunktf orm  $\nu = a(x-b)^2 + c$ ) beschreiben und finden dadurch einen Isomorphismus zwischen den Indexen und deren Wellenzahlen  $\nu$ .



**Abbildung 6.8:** Diese Graphik modelliert, dass Verhalten der Bandlückenverteilung eines Gitterspektrographen und eines Prismenspektrographen nach dem ersten Deslandre'schen Gesetz. Jedoch wurde das quadratische Verhalten mittels der Scheitelpunktform gefittet.

Die Differenzen der Wellenzahlen können auf Grund der quadratischen Natur der  $\nu$  als lineare Funktion modelliert und gefittet werden.



**Abbildung 6.9:** Diese Graphik beinhaltet einen linearen Fit der Differenzen der zweier benachbarten Schwingungsniveaus aus Abbildung 6.8

Diese Funktion modelliert die Steigung des Isomorphismus und erlaubt uns den Extreme Wert für die Wellenzahl zu finden.

### 6.2.2 Dissoziationsenergie des Jodmoleküls

Aufgrund der größeren Unsicherheit des linearen Fits beim Gitterspektrometer, in Abbildung 6.9, wird nur die Auswertung für den Prismenspektrometer gemacht.

Indem in die invertierte lineare Funktion vom Fit 0 eingesetzt wird kann der Index gefunden werden für den die Wellenzahl im Isomorphismus maximiert wird (siehe Vorzeichen der quadratischen Funktion).

Die maximierte Wellenzahl ist:

$$\nu = (2,047 \pm 0,009) \cdot 10^4 \text{ cm}^{-1} \quad (3)$$

Jedoch kann auch Aufgrund der gewählten quadratischen Scheitelpunktform die Scheitelpunktwellenzahl (maximiert) auch direkt am Fitparameter  $b = (2,047 \pm 0,008) \cdot 10^4 \text{ cm}^{-1}$ , welches Aufgrund des niedrigeren Auswertungsschritten eine niedrigere Unsicherheit hat. Mittels der Strahlungsenergieformel ist die Konvergenzenergie zu berechnen:

$$E_s = hc\nu = (4,066 \pm 0,017) \cdot 10^{-19} \text{ J} = (2,538 \pm 0,011) \text{ eV} \quad (4)$$

Zieht man schlussendlich die Anregungsenergie  $E_A = (0,970 \pm 0,005) \text{ eV}$  (Unsicherheit implizit angenommen) von der Konvergenzenergie ab erhält man die Dissoziationsenergie:

$$E_{\text{Diss}} = E_s - E_A = (1,568 \pm 0,012) \text{ eV} \quad (5)$$

## 6.3 Auflösevermögen

# 7 Diskussion

## 7.1 Prismenspektrograph

Allgemein ist die Benutzung des Prismenspektrographen sehr aufwändig, weil die Justierung der Linsen und Prismen sehr viel Zeit in Anspruch nimmt. Auch ist die Aufnahme des Spektrums, welches sich aus mehreren Einzelbildern zusammensetzt, klar zeitintensiver.

Beim Betrachten des Spektrums aus Abbildung 6.2 wird klar ersichtlich, dass die Linien durch diverse Abbildungsfehler eine leichte Krümmung aufweisen. Besonders stark ist dieser Effekt bei den niedrigeren Wellenlängen, was sich durch die höhere Energie erklären lässt. Für die Auswertung wurde dadurch nur ein schmaler Bereich aus der Mitte genommen, um die Werte nicht durch diese Krümmung zu verfälschen. Die Schärfe der Kamera wurde auf den gelben Doppellinie bei 579,0 nm und 576,9 nm fokussiert, da diese besonders signifikant für die Lichtquelle ist.

Aufgrund der kurzen Belichtungszeit konnte der Cyan Peak nicht war genommen werden und eine Gamma-Korrektur musste durchgeführt werden, damit die Peaks markanter werden. Daher wird empfohlen die Belichtungszeit nicht zu kurz zu wählen.

## 7.2 Gitterspektrograph

Der Vorteil des Gitterspektrographen liegt vor allem in der einfachen Handhabung. So muss dieser nur angeschlossen werden und liefert sofort einen Intensitätsverlauf. Jedoch wird kein Spektrum erzeugt. Auch ist darauf zu achten, dass die erzeugt Grafik nicht übersättigt ist, was zu einem Fehler führen würde.

Im Bezug auf die Genauigkeit des Gitterspektrographen sei angemerkt, dass dessen letzte Kalibrierung im Jahr 2008 stattgefunden hat, was darauf schließen lässt, dass eine erneute Kalibrierung die Genauigkeit verbessern könnte.

### 7.3 Auflösevermögen

Die erhaltenen Werte der Auflösevermögen sind in folgender ?? gegenübergestellt.

Ein Vergleich dieser Werte zeigt klar, dass der Prismenspektrograph eine deutlich höhere Auflösung hat. Dies wird auch bei der qualitativen Betrachtung der entsprechenden Bilder sichtbar.

### 7.4 Dissoziationsenergie des Jodmoleküls

Der erhaltene Wert für die Dissoziationsenergie ist in folgenden Liste dem Literaturwert gegenübergestellt.

1.  $E_{\text{Diss}} = (1,568 \pm 0,012) \text{ eV}$
2.  $E_{\text{Diss}_{\text{lit}}} = 1,57 \text{ eV}$

Der Literaturwert ist also in beiden Fehlerintervallen enthalten. Der erhaltene Wert des Prismenspektrographen liegt jedoch deutlich näher am Literaturwert. Auch ist dessen Unsicherheit kleiner.

## 8 Zusammenfassung

Zunächst lässt sich feststellen, dass die Handhabung des Gitterspektrographen deutlich einfacher und vor allem schneller geht. Die erhaltenen Ergebnisse des Prismenspektrographen sind jedoch genauer am Literaturwert und haben auch eine geringere Unsicherheit. Weiters liegt auch eine bessere Auflösung vor. Wodurch sich der Zeitaufwand insgesamt lohnt. Es sei jedoch auch angemerkt dass sich die Untersuchung von Lichtquellen durch den Prismenspektrographen nur für Lichtquellen eignet, die leicht zum Spektrographen gebracht werden können, da dieser aufgrund seiner Größe und des Justierungsaufwands nur sehr schwer bewegt werden kann.

## 8.1 Spektren

Im Rahmen dieses Praktikums wurden die Spektren einer Hg-Lampe, einer Halogenlampe und einer Jodprobe, sowohl mit einem Prismenspektrographen als auch mit einem Gitterspektrographen bestimmt. Die erhaltenen Spektren sind in den Abbildung 6.1 Abbildung 6.2 Abbildung 6.3 und Abbildung 6.5 ersichtlich.

## 8.2 Auflösevermögen

Die erhaltenen Werte für die Auflösevermögen sind in folgender ?? ersichtlich.

## 8.3 Wellenlänge der Jod-Absorptionsbandkanten

Die Wellenlänge der Jod-Absorptionsbandkanten ist das reziproke der gefundenen Wellenzahl  $\nu$  und beträgt:

$$\lambda = \frac{1}{\nu} = (498 \pm 2) \text{ nm}$$

## 8.4 Dissoziationsenergie des Jodmoleküls

Der erhaltene Werte für die Dissoziationsenergie ist  $E_{\text{Diss}} = (1,568 \pm 0,012) \text{ eV}$  und der Literaturwert ist  $E_{\text{Disslit}} = 1,57 \text{ eV}$

## 9 Appendix

### 9.1 Code

```
1 # pyright: reportUnboundVariable=false
2 # pyright: reportUndefinedVariable=false
3
4 import numpy as np
5 import pickle
6 from skimage.measure import profile_line
7 from scipy.signal import find_peaks
8 import pandas as pd
9 import cv2
10 import matplotlib.pyplot as plt
11 from numpy.typing import NDArray
12 from labtool_ex2 import Project
13
14
15 # (0,0)----- (x,0)
16 # |-----|
17 # |-----|
18 # |-----|
19 # |-----|
20 # |-----|
21 # (0,y)----- (x,y)
22 # reference pictures Hg_01 and Hg_02: 5184 x 3456
23 # pixel position at the right end of the red line with closest HEX code
# possible
24 # y val of pixel 1350
25 # 558 left val right val 4710
26 pic_width = 5184
27 pic_height = 3456
28 # Using a picture editing tool find non overlapping span using any of
29 # the Mercury spectral lines which appears in two pictures
30 loc1 = 4610
31 loc2 = 560
32 cropped_picture_span = loc1 - loc2
33 links_lastig = 1 # a value between 0 and 1
34 x_start = (pic_width - cropped_picture_span) * (1 - links_lastig)
35 x_end = (pic_width - cropped_picture_span) * (1 - links_lastig) +
cropped_picture_span
36 y_start = 1350
37 y_end = 2500
38 panowidth = 8 * cropped_picture_span
39 panoheight = y_end - y_start
40
41 DATAPATH = "./data/"
42 PICTURE_DIR = f"{DATAPATH}edited/"
```

```
44
45 def getIntensity(greyscale_img, profile_line_height) -> np.ndarray:
46     intensity = profile_line(
47         greyscale_img,
48         src=(profile_line_height, 0),
49         dst=(profile_line_height, panowidth),
50         linewidth=50,
51     )
52     return intensity
53
54
55 # (x_start,y_start)----D-----(x_end,y_start) <<<
56 # |-----D-----| <<<
57 # |-----D-----| <<<
58 # |-----| <<<
59 # |-----| <<<
60 # |-----| <<<
61 # (x_start,y_end)-----|(x_end,y_end)
62 scanline_height = panoheight // 7
63
64
65 def analysis(P: Project, peaks: NDArray, pan: NDArray, name: str):
66     intensity = getIntensity(peaks, scanline_height)
67     print(intensity.shape)
68
69     # pickle intensities
70     with open(f"{DATAPATH}/{name}pickle", "wb") as file:
71         pickle.dump(intensity, file)
72
73     # plot the whole thing
74     # fig, ax = plt.subplots(nrows=2, sharex=True)
75     pxToNm, nmToPx = genBasisMap(P)
76
77     P.figure.clear()
78     ax_picture: plt.Axes = P.figure.add_subplot(2, 1, 1)
79     ax_int: plt.Axes = P.figure.add_subplot(2, 1, 2)
80
81     P.vload()
82
83     extra_ax = ax_int.twiny()
84
85     lticks = np.arange(int(pxToNm(0)) + 1, int(pxToNm(len(intensity))), 20)
86     pxTicks = nmToPx(lticks)
87     lxTicks = pxToNm(pxTicks)
88
89     ax_picture.imshow(pan, aspect="auto")
90     ax_picture.set_ylabel("$p$ / px")
91     ax_picture.set_xlim(0, len(intensity))
92     P.data = pd.DataFrame(None)
93
```

```
94     P.data["I"] = intensity
95     P.data["px"] = np.arange(len(intensity))
96
97     P.plot(
98         ax_int,
99         px,
100        I,
101        label="Intensität",
102        style="#1cb2f5",
103    )
104
105    ax_int.set_xlim(0, len(intensity))
106    extra_ax.set_xlim(ax_int.get_xlim())
107    extra_ax.set_xticks(pxTicks)
108    extra_ax.set_xticklabels(map(str, lticks))
109    extra_ax.set_xlabel(r"$\lambda$ / nm")
110    # ax[1].grid()
111    P.figure.tight_layout(pad=1)
112    P.figure.subplots_adjust(top=0.90)
113
114    P.savefig(f"{name}plot.pdf")
115
116    print(f"{name}[:-1] done")
117
118
119 # KNOWN_SPECTRAL_LINES = [467.8, 479.9, 508.5, 546, 576.9, 579, 643.9]
120 # # nm Blue, Cyan, Turquoise, Green, Yellow1, Yellow2, Red
121 KNOWN_SPECTRAL_LINES = [467.8, 508.5, 546, 576.9, 579, 643.9]
122 # nm Blue, Cyan, Turquoise, Green, Yellow1, Yellow2, Red
123 OUTPUTPATH = "./output/"
124
125
126 def genBasisMap(P: Project) -> tuple[callable, callable]:
127     P.figure.clear()
128     P.data = pd.DataFrame(None)
129     ax = P.figure.add_subplot()
130     with open("./data/Hg_pickle", "rb") as file:
131         intensity = pickle.load(file)
132
133     distance = 130 # Minimum distance before another peak is searched
134     height = 11 # height of lowest peak
135     peaks, *_ = find_peaks(intensity, distance=distance, height=height)
136     P.vload()
137     violet, violet2, *peaks = peaks
138     P.data["px"] = peaks
139     P.data["l"] = KNOWN_SPECTRAL_LINES
140     print(P.data)
141     P.plot_data(
142         axes=ax,
143         x=px,
144         y=l,
```

```
145         label="Referenzwerte",
146         style="#da93ea",
147     )
148     l = a * (px - b) ** 2 + c
149
150     params = P.plot_fit(
151         axes=ax,
152         x=px,
153         y=l,
154         eqn=l,
155         style=r"#da93ea",
156         label="Quad.",
157         offset=[0, 80],
158         use_all_known=False,
159         guess={"a": 3.5e-7, "b": -4e-3, "c": 470},
160         bounds=[
161             {"name": "a", "min": 0, "max": 1},
162             {"name": "b", "min": -1, "max": 1},
163             {"name": "c", "min": 0, "max": 700},
164         ],
165         add_fit_params=True,
166         granularity=10000,
167         # gof=True,
168         # scale_covar=True,
169     )
170     print(peaks)
171     p = params
172
173     ax.set_title(f"Quadratischer Wellenlängenverteilung über die Pixel")
174
175     P.ax_legend_all(loc=1)
176     ax = P.savefig(f"mappingPxToWaveLength.pdf")
177
177     def pxToNm(x):
178         return p["a"].value * (x - p["b"].value) ** 2 + p["c"].value
179
180     def nmToPx(l):
181         return np.sqrt((l - p["c"].value) / p["a"].value) + p["b"].value
182
183     # lambda_violet = wavelength(violet)
184     # print(f"Wavelength of violet line: {lambda_violet}")
185     # print(f"Spectral resolution: {lambda_violet/0.4}")
186     return (pxToNm, nmToPx)
187
188
189 def imgPrep() -> None:
190     base_names = ["Hg_", "Halo_", "I_"]
191
192     gm = {
193         "I": r"I",
```

```
194         "l": r"\lambda",
195         "px": r"px",
196         "a": r"a",
197         "b": r"b",
198         "c": r"c",
199     }
200     gv = {
201         "I": r"relativ",
202         "a": r"\si{\nano\meter\per\px\squared}",
203         "b": r"\si{\px}",
204         "c": r"\si{\nano\meter}",
205         "l": r"\si{\nano\meter}",
206         "px": r"\si{\px}",
207     }
208
209     pd.set_option("display.max_columns", None)
210     plt.rcParams["axes.axisbelow"] = True
211     P = Project("Spektrometer", global_variables=gv, global_mapping=gm,
212     font=13)
213     P.output_dir = "./"
214     P.figure.set_size_inches((12, 6))
215
216     for name in base_names:
217
218         pics = list()
219         peakspics = list()
220
221         for picture_name in [f"{name}{i}.JPG" for i in range(8)]:
222
223             I = cv2.imread(f"{PICTURE_DIR}{picture_name}")
224             imm = I[y_start:y_end, x_start:x_end]
225
226             # Gamma correction for low exposure
227             gamma = 0.6
228             lookUpTable = np.empty((1, 256), np.uint8)
229             for i in range(256):
230                 lookUpTable[0, i] = np.clip(pow(i / 255.0, gamma) *
231                 255.0, 0, 255)
232
233             gamma_pic = cv2.LUT(imm, lookUpTable).copy()
234             pics.append(gamma_pic)
235             # color_converted = cv2.cvtColor(gamma_pic, cv2.
236             COLOR_BGR2RGB)
237             # pil_image = Image.fromarray(color_converted)
238             # YUV = cv2.cvtColor(I, cv2.COLOR_BGR2YUV)
239             # YUV[:, :, 0] = cv2.equalizeHist(YUV[:, :, 0])
240             # # convert the YUV image back to RGB format
241             # img_output = cv2.cvtColor(YUV, cv2.COLOR_YUV2BGR)
242             # cv2.imwrite(f"./cleared/{picture_name}", img_output)
243             RGB = cv2.cvtColor(imm, cv2.COLOR_BGR2RGB) # convert to
244
245             RGB
```

```

241     R, G, B = cv2.split(RGB)
242
243     # Create a CLAHE object: The image is divided into small
244     # block 8x8 which they are equalized as usual.
245     clahe = cv2.createCLAHE(clipLimit=2.5, tileSize=(8, 8))
246     # Applying this method to each channel of the color image
247     output_2R = clahe.apply(R)
248     output_2G = clahe.apply(G)
249     output_2B = clahe.apply(B)
250
251     # merge each channel back to one
252     img_output = cv2.merge((output_2R, output_2G, output_2B))
253     eq = cv2.cvtColor(img_output, cv2.COLOR_BGR2GRAY)
254     peakspics.append(eq)
255
256     cv_panorama = cv2.flip(np.hstack(pics), 1)
257     pan = cv2.cvtColor(cv_panorama, cv2.COLOR_BGR2RGB)
258     peaks = cv2.flip(np.hstack(peakspics), 1)
259     cv2.imwrite(f"./cleared/{name}comb.jpg", cv_panorama)
260     cv2.imwrite(f"./cleared/{name}comb_eq.jpg", cv_panorama)
261     analysis(P, peaks, pan, name)
262
263 if __name__ == "__main__":
264     imgPrep()

```

**Listing 1:** For Merging the recorded images and determination of the dispersion relation. The resolution power of the prism was also determined.

```

1 # pyright: reportUnboundVariable=false
2 # pyright: reportUndefinedVariable=false
3
4 from typing import Callable
5 import numpy as np
6 import matplotlib.pyplot as plt
7 import pickle
8 import pandas as pd
9 from scipy.constants import speed_of_light, h, elementary_charge
10 from scipy.signal import find_peaks
11 from imageprep import genBasisMap
12 from numpy.typing import NDArray
13 from labtool_ex2 import Project
14 from uncertainties import ufloat
15
16 DATAPATH = "./data/"
17 OUTPUTPATH = "./output/"
18 PICTURE_DIR = f"{DATAPATH}edited/"
19
20 pxToNm = callable
21 nmToPx = callable
22

```

```
23
24 def readIn(path, skiprows=0):
25     lines = []
26     with open(path, "r", encoding="ISO-8859-1") as f:
27         lines = f.readlines()
28     if skiprows:
29         lines = lines[skiprows:]
30
31     L, I = [], []
32     for str_ in lines:
33         try:
34             str_ = str_.replace(", ", ".")
35             l, i = str_.split("\t")
36             L.append(float(l))
37             I.append(float(i))
38         except:
39             pass
40     return np.asarray(L), np.asarray(I)
41
42
43 def extinction(val: NDArray, ref: NDArray) -> NDArray:
44     return np.log10(val / ref)
45
46
47 def plot_extinction(P: Project) -> None:
48     P.figure.clear()
49     P.data = pd.DataFrame(None)
50     ax = P.figure.add_subplot()
51
52     # Extinktion prism
53     with open(f"{DATAPATH}Halo_pickle", "rb") as file:
54         I_ref = pickle.load(file) # Halogen
55     with open(f"{DATAPATH}I_pickle", "rb") as file:
56         I_Iod = pickle.load(file)
57     Ext = extinction(I_Iod, I_ref)
58     P.data["E"] = Ext
59     pixels = np.arange(len(Ext))
60     _l = pxToNm(pixels)
61     P.data["_l"] = _l
62     P.vload()
63
64     P.plot(
65         axes=ax,
66         x=_l,
67         y=E,
68         label="Prismenspektrograph",
69         style="#da93ea",
70     )
71
72     ax.set_title("Extinktionsspektrum einer Iod-Zelle vor einer  
Halogenlampe")
```

```
73 P.ax_legend_all(loc=0)
74 P.figure.tight_layout(pad=1)
75 ax = P.savefig(f"prism_extinction.pdf", clear=True)
76
77 low = int(nmToPx(508))
78 high = int(nmToPx(550))
79 _l = pxToNm(np.arange(len(Ext)))
80 l_band = _l[low:high]
81 E_band = -Ext[low:high]
82 peaks, *_ = find_peaks(E_band, distance=130, prominence=0.01)
83 l_peaks, E_peaks = l_band[peaks], E_band[peaks]
84 P.data = pd.DataFrame(None)
85 P.data["E"] = E_band
86 P.data["l"] = l_band
87 P.plot(
88     axes=ax,
89     x=l,
90     y=E,
91     label="Prismenspektrograph",
92     style="#da93ea",
93 )
94 P.data = pd.DataFrame(None)
95 P.data["E"] = E_peaks
96 P.data["l"] = l_peaks
97 P.plot_data(
98     axes=ax,
99     x=l,
100    y=E,
101   label="Peaks",
102   style="#a30000",
103 )
104 ax.set_title("Extinktionsspektrum einer Iod-Zelle vor einer
105 Halogenlampe")
106 P.ax_legend_all(loc=0)
107 P.figure.tight_layout(pad=1)
108 ax = P.savefig(f"prism_ausschnitt_extinction.pdf", clear=True)
109
110 waveNumber(P, l_peaks / 10e6, "Prisma")
111 P.figure.clear()
112 P.data = pd.DataFrame(None)
113 ax = P.figure.add_subplot()
114 _l, halogen = readIn(f"{PICTURE_DIR}halogen.txt", skiprows=17)
115 _, HG = readIn(f"{PICTURE_DIR}quecksilber.txt", skiprows=17)
116 _, Iod = readIn(f"{PICTURE_DIR}iod.txt", skiprows=17)
117
118 Ext = -extinction(Iod, halogen)
119 P.data = pd.DataFrame(None)
120 P.data["E"] = Ext
121 P.data["l"] = _l
122 P.plot()
```

```
123         axes=ax,
124         x=l,
125         y=E,
126         label="Gitterspektrograph",
127         style="#da93ea",
128     )
129     ax.set_title("Extinktionsspektrum einer Iod-Zelle vor einer
130     Halogenlampe")
131     P.ax_legend_all(loc=1)
132     P.figure.tight_layout(pad=1)
133     ax = P.savefig(f"gger_extinction.pdf", clear=True)
134
135     filter = np.logical_and((550 < _l), (_l < 615))
136     l_band = _l[filter]
137     E_band = Ext[filter]
138     peaks, *_ = find_peaks(E_band, distance=11, prominence=0.001)
139     l_peaks, E_peaks = l_band[peaks], E_band[peaks]
140     P.data = pd.DataFrame(None)
141     P.data["E"] = E_band
142     P.data["l"] = l_band
143     P.plot(
144         axes=ax,
145         x=l,
146         y=E,
147         label="Gitterspektrograph",
148         style="#da93ea",
149     )
150     P.data = pd.DataFrame(None)
151     P.data["E"] = E_peaks
152     P.data["l"] = l_peaks
153     P.plot_data(
154         axes=ax,
155         x=l,
156         y=E,
157         label="Peaks",
158         style="#a30000",
159     )
160     ax.set_title("Extinktionsspektrum einer Iod-Zelle vor einer
161     Halogenlampe")
162     P.ax_legend_all(loc=0)
163     P.figure.tight_layout(pad=1)
164     ax = P.savefig(f"gger_ausschnitt_extinction.pdf", clear=True)
165     waveNumber(P, l_peaks / 10e6, "Gitter")
166     P.figure.clear()
167     P.data = pd.DataFrame(None)
168     ax = P.figure.add_subplot()
169
170     max_val = max(halogen.max(), HG.max(), Iod.max())
171     P.data["l"] = _l
172     P.data["I"] = halogen / max_val
173     # ax.plot(_l, halogen/max_val, label="Halogen")
```

```
172     P.plot(
173         axes=ax,
174         x=l,
175         y=I,
176         label="Halogenlampe",
177     )
178
179     ax.plot(_l, HG / max_val, label="Quecksilberlampe")
180     ax.plot(_l, Iod / max_val, label=r"Halogen mit Iodrohr")
181     ax.set_title("Intensitätskurve den ver. Proben (Gitterspektrograph")
182     " )
183     P.ax_legend_all(loc=0)
184     P.figure.set_size_inches((11, 6))
185     ax = P.savefig(f"intensity_spektrum_gitter.pdf", clear=True)
186
187 def waveNumber(P: Project, peaks, name: str):
188     P.figure.clear()
189     P.data = pd.DataFrame(None)
190     ax = P.figure.add_subplot()
191     P.data["v"] = 1 / peaks
192     P.data["i"] = np.arange(len(P.data["v"]))
193     P.vload()
194
195     P.plot_data(
196         axes=ax,
197         x=i,
198         y=v,
199         label=f"Referenzwerte {name}",
200         style="#a30000",
201     )
202
203     v = d * (i - e) ** 2 + f
204
205     params = P.plot_fit(
206         axes=ax,
207         x=i,
208         y=v,
209         eqn=v,
210         style=r"#da93ea",
211         label="Quad.",
212         offset=[0, 10],
213         use_all_known=False,
214         guess={"d": -1, "e": -52, "f": 20000},
215         bounds=[
216             {"name": "d", "min": -20, "max": 400},
217             {"name": "e", "min": -70, "max": 100},
218             {"name": "f", "min": 0, "max": 30000},
219         ],
220         add_fit_params=True,
221         granularity=10000,
```

```
222         # gof=True ,
223         # scale_covar=True ,
224     )
225
226     def quad(x):
227         return ufloat(params["d"].value, params["d"].stderr) * (
228             x - ufloat(params["e"].value, params["e"].stderr)
229         ) ** 2 + ufloat(params["f"].value, params["f"].stderr)
230
231     ax.set_title("Quadratische Abhangigkeit der Wellenzahlen")
232     P.ax_legend_all(loc=1)
233     P.figure.tight_layout(pad=1)
234     ax = P.savefig(f"waveNumberFit{name}.pdf", clear=True)
235
236     # Wavenumber prism difference
237     v1 = P.data["v"].values[:-1]
238     v2 = P.data["v"].values[1:]
239     P.data = pd.DataFrame(None)
240     P.data["Dv"] = v2 - v1
241     P.data["i"] = np.arange(len(P.data["Dv"]))
242     P.vload()
243     P.plot_data(
244         axes=ax,
245         x=i,
246         y=Dv,
247         label=f"Referenzwerte {name}",
248         style="#a30000",
249     )
250
251     Dv = k * i + d
252
253     pp = P.plot_fit(
254         axes=ax,
255         x=i,
256         y=Dv,
257         eqn=Dv,
258         style=r"#da93ea",
259         label="Lin.",
260         offset=[0, 10],
261         use_all_known=False,
262         guess={"k": -2, "d": -52},
263         bounds=[
264             {"name": "k", "min": -4, "max": 0},
265             {"name": "d", "min": -102, "max": 1},
266         ],
267         add_fit_params=True,
268         granularity=10000,
269         # gof=True ,
270         # scale_covar=True ,
271     )
272
```

```

273     def lin(x):
274         return ufloat(pp["k"].value, pp["k"].stderr) * x + ufloat(
275             pp["d"].value, pp["d"].stderr
276         )
277
278     def invlin(v):
279         return (v - ufloat(pp["d"].value, pp["d"].stderr)) / ufloat(
280             pp["k"].value, pp["k"].stderr
281         )
282
283     ax.set_title("Lineare Abhängigkeit der Wellenzahlenabstände")
284     P.ax_legend_all(loc=1)
285     P.figure.tight_layout(pad=1)
286     ax = P.savefig(f"waveNumberDeltasFit{name}.pdf", clear=True)
287
288     # Calculation
289     vindex = invlin(0)
290     v = quad(vindex) * 1e2 # To cm to m SI
291     print("-----")
292     print(f"{{vindex}}")
293     print(f"{{v}}")
294     wavelength = 1 / v
295     print(f"{{wavelength}}")
296
297     print("Scheitelenergie")
298     E = h * speed_of_light * v
299     E_ev = E / elementary_charge
300     E_diss = E_ev - ufloat(0.970, 0.005) # Anregungsenergie
301     print(f"{{E}}", f"{{E_ev}}", f"{{E_diss}}")
302     print("-----")
303
304
305 if __name__ == "__main__":
306
307     gm = {
308         "I": r"I",
309         "i": r"\text{Index}",
310         "l": r"\lambda",
311         "px": r"px",
312         "a": r"a",
313         "b": r"b",
314         "c": r"c",
315         "d": r"^{a}",
316         "f": r"^{b}",
317         "e": r"^{c}",
318         "k": r"^{k}",
319         "v": r"\nu",
320         "Dv": r"\Delta\nu",
321         "E": r"E",
322     }
323     gv = {

```

```

324     "I": r"relativ",
325     "i": r"1",
326     "a": r"\si{\nano\meter\per\px\squared}",
327     "b": r"\si{\px}",
328     "c": r"\si{\nano\meter}",
329     "d": r"\si{\per\centi\meter}",
330     "f": r"\si{\per\centi\meter}",
331     "k": r"\si{\per\centi\meter}",
332     "e": r"1",
333     "v": r"\si{\per\centi\meter}",
334     "Dv": r"\si{\per\centi\meter}",
335     "l": r"\si{\nano\meter}",
336     "px": r"\si{\px}",
337     "E": r"1",
338 }
339
340 pd.set_option("display.max_columns", None)
341 plt.rcParams["axes.axisbelow"] = True
342 P = Project("Spektrometer", global_variables=gv, global_mapping=gm,
343             font=13)
344 P.output_dir = "./"
345 P.figure.set_size_inches((11, 3))
346 pxToNm, nmToPx = genBasisMap(P)
347 plot_extinction(P)

```

**Listing 2:** The wavelength extraction of the band edges in the Iodine absorptions spectrum and calculation of the dissociation energy of iodine molecule in ground state.

## Abbildungsverzeichnis

2.1	Skizzierter Strahlengang durch ein Prisma . . . . .	2
2.2	Skizzierter Strahlengang durch ein Gitter . . . . .	4
3.1	Versuchsaufbau des Prismenspektrographen . . . . .	5
3.2	Anordnung der Prismen im Spektrographen . . . . .	6
3.3	Versuchsaufbau des Gitterspektrographen . . . . .	7
5.1	Das Software Interface für die Kontrolle der Kamera . . . . .	9
5.2	Aufnahme des Referenzspektrums mittels des Gitterspektrographs und Software . . . . .	11
6.1	Diese Graphik beinhaltet das Farb und die Intensitätsspektrum einer Hg-Cd-Lampe und stellt das Kalibrierungsspektrum für folgende Berechnungen mit dem Prismenspektrographen dar. . . . .	12
6.2	Diese Graphik beinhaltet das Farb und die Intensitätsspektrum einer Halogenlampe und stellt das Referenzspektrum für das Iodspektrum dar. . . . .	13

6.3	Diese Graphik beinhaltet das Farb und die Intensitätsspektrum einer Halogenlampe mit Iodmolekülen im Abbildungpfad . . . . .	13
6.4	Diese Graphik beschreibt das Dispersion der verschiedenen in Pixeln und stellt die Dispersionrelation zwischen Pixel zu Wellenlängen dar.	14
6.5	Diese Graphik beinhaltet die gemessen Spektren für die Hg-Cd Lampe, der Halogenlampe und der Halogenlampe mit dem Iod vorgeschoben. . . . .	15
6.6	Diese Graphik beinhaltet die Extinktionskurven des Prismenspektrographen (oben) und des Gitterspektrographens (unten) und wurde mittels der Abbildung 6.5 Abbildung 6.2 und Abbildung 6.3 erstellt.	16
6.7	Diese Graphik beinhaltet Ausschnitte aus der Extinktionskurven des Prismenspektrographen (oben) und des Gitterspektrographens (unten), welche markante Peaks aufweisen. . . . .	17
6.8	Diese Graphik modelliert, dass Verhalten der Bandlückenverteilung eines Gitterspektrographen und eines Prismenspektrographen nach dem ersten Deslandre'schen Gesetz. Jedoch wurde das quadratische Verhalten mittels der Scheitelpunktform gefittet. . . . .	18
6.9	Diese Graphik beinhaltet einen linearen Fit der Differenzen der zweier benachbarten Schwingungsniveaus aus Abbildung 6.8 . . . . .	19

## Tabellenverzeichnis

4.1	Verwendete Geräte . . . . .	8
5.1	Abgelesene Einstellungen . . . . .	10