

BACHELORTHESIS
Leon Balke

Automatische und zeitaufgelöste Zählung von Insekten mit Bilderkennung zur Verhaltensanalyse von Stechmücken bei unterschiedlichen Lichtquellen

FAKULTÄT TECHNIK UND INFORMATIK
Department Informations- und Elektrotechnik

Faculty of Computer Science and Engineering
Department of Information and Electrical Engineering

Leon Balke

Automatische und zeitaufgelöste Zählung von Insekten mit Bilderkennung zur Verhaltensanalyse von Stechmücken bei unterschiedlichen Lichtquellen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Elektro- und Informationstechnik*
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Michael Erhard
Zweitgutachter: Dr. Claudius Noack

Eingereicht am: 6. Dezember 2020

Leon Balke

Thema der Arbeit

Automatische und zeitaufgelöste Zählung von Insekten mit Bilderkennung zur Verhaltensanalyse von Stechmücken bei unterschiedlichen Lichtquellen

Stichworte

Bildverarbeitung, automatische Zählung, Lichttechnik

Kurzzusammenfassung

Dieser Arbeit behandelt Testversuche und Erweiterungen eines Prototypen, welcher mit Hilfe von Bilderkennungssoftware eine automatische Zählung von Insekten ermöglicht. Die Tests werden an Fliegen und Mücken mit verschiedenen Leuchten als Lockmittel durchgeführt. Nach der Ergebnisanalyse kann bei den meisten Leuchtmittel ein Lockverhalten erkannt werden.

Leon Balke

Title of Thesis

Automatic and time-resolved counting of insects with image recognition for behavioral analysis of mosquitoes under different light sources

Keywords

Image processing, automatic counting, light technology

Abstract

This thesis deals with tests and extensions of a prototype, which enables an automatic counting of insects with the help of image recognition software. The tests are carried out on flies and mosquitoes with different lights as attractants. After the analysis of the results a lure behavior of most lights can be detected.

Inhaltsverzeichnis

Abbildungsverzeichnis	vi
Tabellenverzeichnis	viii
1 Einleitung	1
1.1 Motivation	1
1.2 Ziel der Arbeit	2
1.3 Aufbau der Arbeit	2
2 Grundlagen	3
2.1 Anwendungsbereiche Bildverarbeitung	3
2.1.1 Forschung	3
2.1.2 Industrie	4
2.1.3 Medizin	6
2.1.4 Beobachtung	7
2.2 Programmierung	10
2.2.1 Programme	10
2.2.2 Bildvorverarbeitung	10
2.2.3 Merkmalerkennung	11
2.3 Insekten	12
2.3.1 Allgemein	12
2.3.2 Mücken	12
2.3.3 Fliegen	14
2.4 Auswirkungen des Lichtes auf Insekten	15
3 Versuchsaufbau	17
3.1 Testumgebung und Entwurf	17
3.2 Software	18
3.2.1 Programmumgebung	18

3.2.2	Programmbeschreibung	19
3.3	Auswahl und Beschaffung von Insekten	21
4	Leuchtmittel	22
4.1	LED Lampen	23
4.2	Energiesparlampe FL	25
5	Experimente zur Verhaltensanalyse	27
5.1	Testverfahren	29
5.1.1	Grundlagentests	29
5.1.2	Vergleichstests	29
5.2	Tests allgemein und Grundlagentests mit Mücken	30
5.3	Grundlagentest mit Fliegen	35
5.3.1	Grundlagentests Leuchtmittel LED 6500K	36
5.3.2	Grundlagentests Leuchtmittel LED 3000K	39
5.3.3	Grundlagentests Leuchtmittel LED 2000K	41
5.3.4	Grundlagentests Leuchtmittel Energiesparlampe 6500K	43
5.3.5	Grundlagentests Leuchtmittel Energiesparlampe 2700K	45
5.3.6	Zusammenfassung	47
5.4	Vergleichstests	48
5.4.1	Vergleichstests mit Fliegen LED 6500k und FL 6500K	48
5.4.2	Vergleichstests FL 2700K und FL 6500K	51
5.4.3	Vergleichstests LED 2000K und LED 6500K	54
5.4.4	Zusammenfassung	57
6	Fazit	59
7	Ausblick	61
	Literaturverzeichnis	62
A	Anhang: Programmcode	64
B	Anhang: Bilder des Prototyps	72
C	Anhang: Messungen Leuchtmittel	74
	Selbstständigkeitserklärung	80

Abbildungsverzeichnis

2.1	Anwendungsbeispiel industrielle Bildverarbeitung [8]	4
2.2	Medizinische Bildverarbeitung, erkennen von Hirntumoren [2]	6
2.3	UAV LUNA beim Abheben [7]	8
2.4	Änderungsdetektion	9
2.5	Indikation von beweglichen Objekten(MTI)	9
2.6	„Merkmale der wichtigsten Mückengattungen (nach Marshall 1938)“[10] . .	13
2.7	Schematische Darstellungen der Stubenfliege (Musca domestica) [10] . . .	14
2.8	Wahrnehmungsbereiche von Menschen und Nachtfaltern in Gegenüberstel- lung einer warm-weißen-LED [5]	16
3.1	Frontansicht Konstruktion	18
3.2	Benutzeroberfläche	19
3.3	Beispielaufnahme der Kantenerkennung	20
4.1	Spektrum der Müller licht Retro LED 3000K Lampe [11]	23
4.2	Spektrum der Kopos-led BULB A60 LED 3000K Lampe [C Anhang]	23
4.3	Spektrum der Philips LED Cool daylight 6500K Lampe [C Anhang]	24
4.4	Spektrum der Philips warm white FL 2700K Lampe [C Anhang]	25
4.5	Spektrum der Philips cool daylight FL 6500K Lampe [C Anhang]	25
5.1	Zeitliche Auswertung der Positionen von Objekten in verschiedenen Be- reichen	28
5.2	Verteilungsstatistik der Aufenthaltsdauer von Objekten in verschiedenen Bereichen	28
5.3	Testbildaufnahme von fünf Mücken im Testbereich	30
5.4	Testbildaufnahme von einem Handy, von fünf Mücken im Testbereich . . .	32
5.5	Auswertung von Mücken bei einer LED 6500K	33
5.6	Verteilungsstatistik von 14h Video bei LED 3000K	34
5.7	Testbildaufnahme von acht Fliegen im Testbereich	35

5.8	Position der Fliegen während der Auswertung von der LED 6500K	36
5.9	Zeitliche Auswertung von acht Fliegen mit LED 6500K links	37
5.10	Zeitliche Auswertung von acht Fliegen mit LED 6500K rechts	38
5.11	Zeitliche Auswertung von acht Fliegen mit LED 3000K rechts	39
5.12	Zeitliche Auswertung von acht Fliegen mit LED 3000K links	40
5.13	Zeitliche Auswertung von acht Fliegen mit LED 2000K links	41
5.14	Zeitliche Auswertung von acht Fliegen mit LED 2000K rechts	42
5.15	Zeitliche Auswertung von acht Fliegen mit FL 6500K links	43
5.16	Zeitliche Auswertung von acht Fliegen mit FL 6500K rechts	44
5.17	Verteilungsstatistik von acht Fliegen mit FL 6500K rechts	45
5.18	Zeitliche Auswertung von acht Fliegen mit FL 2700K links	46
5.19	Zeitliche Auswertung von acht Fliegen mit FL 2700K rechts	47
5.20	Zeitliche Auswertung von acht Fliegen mit links LED 6500K, rechts FL6500K	49
5.21	Zeitliche Auswertung von acht Fliegen mit rechts LED 6500K, links FL 6500K	50
5.22	3. Auswertungsverfahren rechts LED 6500K, links FL 6500K	51
5.23	Zeitliche Auswertung von acht Fliegen mit rechts FL 2700K, links FL 6500K	52
5.24	3. Auswertungsverfahren mit rechts FL 2700K, links FL 6500K	53
5.25	Zeitliche Auswertung von acht Fliegen mit links FL 2700K, rechts FL 6500K	54
5.26	Verteilungsstatistik von acht Fliegen mit links FL 2700K, rechts FL 6500K	55
5.27	Zeitliche Auswertung von acht Fliegen mit links LED 2000K, rechts LED 6500K	56
5.28	Zeitliche Auswertung von acht Fliegen mit rechts LED 2000K, links LED 6500K	57
5.29	Aufnahme beim Vergleich zwischen LED 2000 K und LED 6500K	58

Tabellenverzeichnis

3.1	Materialien für die Konstruktion	18
4.1	Lampenauswahl	22
4.2	Bestrahlungsstärke von LED 3000K	24
4.3	Bestrahlungsstärke von LED 6500K	24
4.4	Bestrahlungsstärke von Energiesparlampe 2700K	26
4.5	Bestrahlungsstärke von Energiesparlampe 6500K	26
5.1	Ergebnisse der Grundlagentests	48
5.2	Ergebnisse der Vergleichstests	58

1 Einleitung

Basierend auf der These von Herr Dr. Noack „Stechmücken werden von klassischen Leuchtmitteln eher angezogen als von modernen LED-Leuchten (2018)“, wurde zur Beleuchtung oder Widerlegung ein Prototyp mit Bilderkennung zur Beobachtung von Insekten (Stechmücken) entworfen. Darauf aufbauend wird nun die Funktion des Prototyps durch das Analysieren des Verhaltens von Insekten bei unterschiedlichen Leuchtmitteln untersucht und dabei genauer auf die Leuchtmittelauswahl eingegangen [3]. Besonders bei diesem Versuch ist, dass nur ausgewählte Insektenarten untersucht werden können und nicht wie in anderen Versuchen willkürliche Insekten getestet werden.

1.1 Motivation

Insbesondere in Städten sorgt die Außenbeleuchtung zum Aufhellen der Nacht. Dafür sorgen zum Beispiel Beleuchtungen für Gebäude, der Straße oder von Reklamen. Aktuell wird immer mehr in die Umrüstung der Straßenbeleuchtung von den Gemeinden und Städten investiert. Dabei handelt es sich um 9 Millionen Straßenlampen [5]. Diese können einen großen Einfluss auf die umgebende Umwelt haben. In Deutschland leben rund 33.000 Insektenarten, welche eine sehr wichtige Schlüsselposition in der Nahrungskette vieler anderer Tierarten darstellen. In Ökosystemen sorgen die Insekten für die Bestäubung der Pflanzen, welche für die Landwirtschaft notwendig sind. Laut [5] schätzen Experten, dass allein in Deutschland pro Jahr rund 150 Billionen (150.000.000.000.000) Insekten an Straßenlaternen sterben. Darunter befinden sich häufig bedrohte Arten. Gründe für die Vielzahl an toten Insekten sind Verbrennungen durch die heißen Lampen sowie Erschöpfung, nach dem sie im Lichtkegel umher geflogen sind. Zudem werden sie zur leichten Beute für ihre Jäger, wenn sie orientierungslos im Licht umherfliegen. Aus den Gründen fehlen Insekten zur Fortpflanzung. Durch das Massensterben der Insekten wird

die Blütenbestäubung stark reduziert. Die im Lichtschein gefangenen Motten hatten Pollen von 28 Pflanzenarten transportiert, wie eine Studie aus England raus festgestellt hat [5].

1.2 Ziel der Arbeit

In dieser Arbeit geht darum, die Funktionalität eines Prototyps sicherzustellen, welcher dafür ausgelegt ist, mithilfe einer Bilderkennungssoftware die Positionen der Testobjekte in festen Testarealen zu zählen und nachträglich auszuwerten. Dazu wird eine ausgewählte Anzahl von bestimmten Testobjekten, in diesem Fall Stechmücken und Fliegen in dem Versuchsaufbau analysiert. Als Lockmittel werden in diesem Studienaufbau verschiedene Leuchtmittel verwendet. Besondere Unterschiede der Lampen sind die Lichtfarben und die Arten des Lichtes. Differenziert wird hier zwischen LED-Lampen und Energiesparlampen.

1.3 Aufbau der Arbeit

Zu Beginn der Arbeit werden grundlegende Anwendungsbereiche der Bildverarbeitung beschrieben. Dabei liegt der Fokus auf den Bereichen Industrie, Medizin und Beobachtung von Objekten. Zudem werden noch Verfahren beschreiben, die für eine Bildverarbeitung verwendet werden. Damit ein Grundwissen über die Testobjekte verfügbar ist, wird auch dies in dem Kapitel beleuchtet. Zunächst wird das Wahrnehmungsspektrum von Nachtfaltern mit dem Spektrum der Leuchtmittel verglichen und analysiert. Der anschließend dargestellte Versuchsaufbau bezieht sich auf die Testumgebung und die verwendete Software. Des Weiteren wird die Auswahl und Beschaffung der Insekten aufgeführt. In dem nächsten Kapitel sind die verschiedenen Leuchtmittel und deren Spektren aufgeführt und es werden Vermutungen des Lockverhaltens aufgestellt. Abschließend findet eine Verhaltensanalyse statt. Dafür werden Aufnahmen der Testobjekte erstellt und diese mit Hilfe des Bildverarbeitungsprogrammes ausgewertet. Die Auswertungen werden nacheinander dargestellt und analysiert.

2 Grundlagen

2.1 Anwendungsbereiche Bildverarbeitung

Der Begriff Bildverarbeitung hat in den letzten Jahren immer mehr an Bedeutung gewonnen. Es ist eine Technik, mit der es möglich ist: Objekte, Formen, Farben und Muster aus Bildern zu identifizieren. Hierbei werden die Bilder als digitale Signale betrachtet und zu Beginn aufbereitet, um später die zu erkennenden Objekte zu extrahieren. Mit den erlangten Informationen werden die erkannten Objekte analysiert oder weiterverarbeitet.

Die Anwendung der Bildverarbeitung ist sehr weit gefächert. Speziell in den Bereichen Forschung, Industrie, Medizin und zur Beobachtung ist die Bildverarbeitung vertreten. In dem folgenden Abschnitt werden die unterschiedlichen Bereiche grundlegend beschrieben.

2.1.1 Forschung

In dieser Bachelorarbeit findet die Bildverarbeitung ihre Anwendung bei Forschungszwecken und Messwerverfassung. Mit dessen Hilfe ist es leicht die Messwerte aufzunehmen, digital zu verarbeiten und darzustellen. Es gibt noch einige weitere Forschungsgebiete, in denen Bildverarbeitung Anwendung findet. Beispiele hierfür zu nennen sind andere Zählvorgänge die schnell, präzise und über einen längeren Zeitraum stattfinden sollen oder genaue und schnelle Messungen von Bohrungen oder Abständen. Wichtig ist die Bildverarbeitung auch für Roboter oder KI-Systeme (künstliche Intelligenz), dadurch habe sie die Möglichkeit zu „sehen“, also Positionen, Farben und Formen zu erkennen. Mit der Fähigkeit „sehen“ zu können, ist es möglich auf unterschiedliche Situationen zu reagieren, wenn diese fest einprogrammiert werden. In dem Fall wären es ein schwaches KI-Systeme, weil kein eigenständiges Lernen erfolgen kann.

2.1.2 Industrie

Die Fähigkeit einer Maschine, das „sehen“ zu ermöglichen, ist auch bei Fertigungsprozessen und in der Industrie essenziell, denn Fertigungsprozesse werden immer effizienter und die Qualität der Produkte steigt weiter an. Um die komplexen Aufgaben und wiederkehrende Ergebnisse zuverlässig zu erreichen ist daher die Bildverarbeitung ausschlaggebend. Dabei spielt auch das Lernen, um Gegenstände zu erkennen eine zentrale Rolle.

Die Bildverarbeitungssysteme der Industrie bestehen aus digitalen Sensoren in Kameras mit speziellen Optiken zur Bilderfassung. Dort werden solche Systeme eingesetzt, um Produktionsketten während der Fertigung automatisch, genau und über einen langen Zeitraum zu überwachen. Häufig wird es auch im Industriebereich für Messungen und Zeichenerkennung verwendet wie in der Abbildung 2.1 veranschaulicht wurde [12].

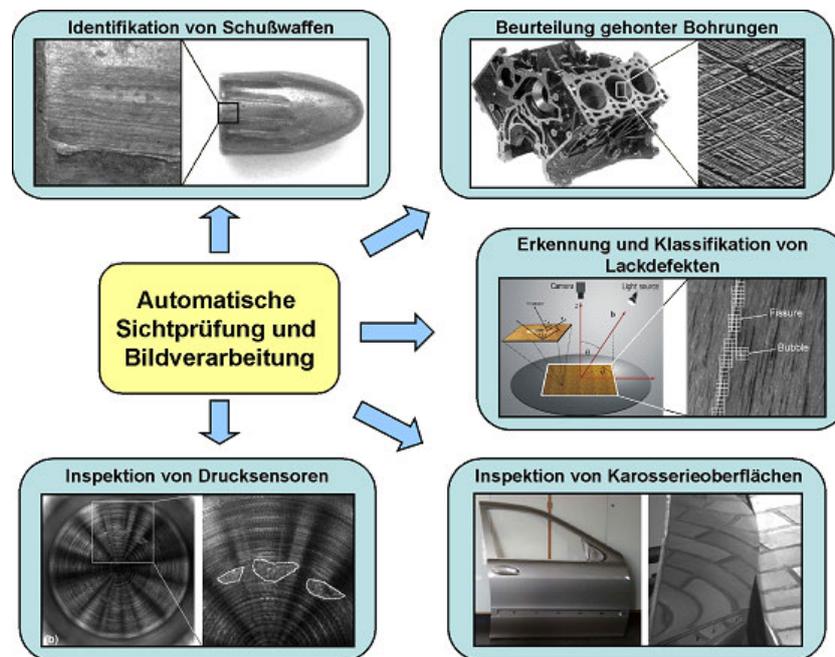


Abbildung 2.1: Anwendungsbeispiel industrielle Bildverarbeitung [8]

Lernen ist für die Bildverarbeitung relevant, damit ist es möglich verschiedene Gegenstände zu erkennen und einzuordnen. Die KI-Kamera-Software Google Lens kann zum Beispiel durch die Verwendung von Smartphone Kameras inzwischen mehr als eine Milliarde Gegenstände erkennen und zuordnen. Solch ein Lernprozess geschieht mittels Deep

Learning. Meist basierend auf „Convolutional Neural Network“ (CNN) was übersetzt „faltendes neuronales Netzwerk“ bedeutet [1].

Deep Learning ist eine spezielle Methode, um Informationen zu verarbeiten. Dazu werden eine ausreichend große Menge an Datenpunkten von einem Programm abgebildet, um komplizierte Zusammenhänge zu schaffen. Es werden benutzerdefinierte Modelle verwendet, welche über einen längeren Zeitraum, aus tief hintereinander aufgebauten Schichten bestehen. Die Identifikation von passenden Merkmalen erfolgt durch wiederholende Programmabläufe und wird häufig als Trainingsprozess bezeichnet. Der Prozess wird auch Lernen genannt, wodurch auch der Name entstanden ist [6].

Der Trainingsprozess ist ein automatischer Ablauf, indem spezifizierete Eigenschaften von Objekten eingelernt werden. Anschließend ist die Zuordnung von den neu erfassten Bild-daten mit einer sehr hohen Genauigkeit möglich. Häufig findet eine Anwendung zur Objekterkennung und Klassifizierung statt. Damit solche Technologien direkt von Firmen verwendet werden können, werden die Programme vor Auslieferung mit mehreren Millionen ausgesuchten Bildern aus dem Industriebereich trainiert. Somit müssen nur noch verhältnismäßig wenige eigene Bilder in den Lernprozess eingebunden werden, um die Funktionalität zu gewährleisten.

Solche Deep-Learning verfahren kommen dann zum Einsatz, wenn eine herkömmliche Bildverarbeitung auf Grund von komplexen und zeitaufwändigen Prüfverfahren oder Herstellungsprozessen nicht mehr ausreicht. Zudem können auch Leihen neue Bilder in das Programm trainieren oder einbinden und es müssen keine ausgebildeten Programmierer sein [1].

Durch industrielle Bildverarbeitung ist es möglich komplizierte und teilweise unterschiedliche Abläufe mit einer hohen Geschwindigkeit zu verwirklichen. Durch intelligente Vermessungen kann viel Material gespart werden. Wie zum Beispiel in der Holzindustrie. Dort ist die Vermessung von Baumstämmen essenziell, um herauszufinden wo sich die Merkmale wie: die Position des Kerns, die Jahresringe oder die Beschaffenheit des Holzes liegen. Dies wurde bis her von Menschen durchgeführt. Bei neuen Fabriken geschieht das durch eine KI-System, wodurch die Vermessungen schneller und genauer verlaufen [1].

2.1.3 Medizin

Ein weiterer wichtiger Aufgabenbereich, in dem Bilderverarbeitung von großem Nutzen ist, ist die Medizintechnik. Auch hier wird das Deep Learning angewandt, um beispielsweise Bilder aus der Neurologie oder aus Röntgenaufnahmen auswerten zu können.

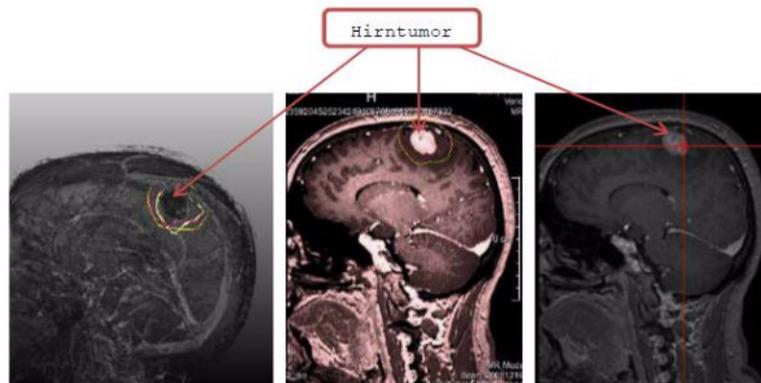


Abbildung 2.2: Medizinische Bildverarbeitung, erkennen von Hirntumoren [2]

Das Thema wurde in Fachartikel „Künstliche Intelligenz in der Medizin – Holzweg oder Heilversprechen?“ von dem Dr. Daniel Sonntag ausführlich untersucht. Des Weiteren geht er davon aus, dass sich innerhalb der nächsten sechs Jahre die KI-Forschung in dem Bereich etablieren werden wird. Wichtige Voraussetzung dafür ist, dass sich die Qualität der Ergebnisse von den KI-Systemen mit denen von Fachärzten vergleichbar sind.

Beispielhafte Anwendungen einer solchen Technik ist, in erster Linie eine schnelle und kostengünstige Zweitmeinung für den behandelnden Arzt. Dafür würde das System medizinische Texte, Bilder und Aufnahmen der Patienten intelligent Vorverarbeiten. Mit dem Ziel, dass Auffälligkeiten bei Befunden dem Arzt sichtbar werden. Eine weitere wichtige Anwendung wäre in dem nicht klinischen Bereich möglich. Dort könnten Angehörige von pflegebedürftigen Menschen durch ein KI-System entlastet werden. In dem durch sensomotorische Intelligenz einfache Haushaltsaufgaben abgenommen und durch Interpretation der Sprache und Videos eine Kommunikation stattfinden kann. Zuletzt hat jeder Patient die Möglichkeit sich schon zuhause, bevor er zum Arzt geht, eine Diagnose beziehungsweise Meinung oder Therapie per Knopfdruck einzuholen [13].

Damit die Anwendung eines KI-Systems zweifelsfrei ist, müssen drei Probleme berücksichtigt werden:

- eine dauerhafte und sichere Vorhersage
- alle Krankheitsfälle für die Entscheidungsunterstützung
- Nachvollziehbarkeit

Die ersten beiden Schwierigkeiten bestehen auf Grund fehlender und qualitativ guten Trainingsdaten für die KI-Systeme. Dafür wurde im Jahr 2016 das Programm Medizininformationen initiiert, um in Deutschland Datenzentren zu erstellen. Diese Datenzentren sollen dafür verwendet werden, erfasste Daten zu verarbeiten und Programmen mit Hilfe Deep Learning zu Trainieren. Die Datenerfassung gestaltet sich jedoch häufig schwierig, denn dafür werden eine Vielzahl an Bilddaten und Patientendaten benötigt. Da jedoch viele Kliniken keine elektrische Datenerfassung der Patienten haben, steht die Patientenakte häufig nicht digital zur Verfügung und somit ist es erschwert an die nötigen Daten zu gelangen.

Das Thema Nachvollziehbarkeit bezieht sich nicht nur auf die Analyse der Diagnosen. Sondern auch auf die daraus geschlossenen Entscheidungen wie der Patient behandelt oder therapiert werden sollte. Denn menschliche Intelligenz ist den KI-Systemen in bestimmten Punkten besonders überlegen, zum Beispiel den Kontextbezug oder persönliche Werte zu berücksichtigen [13].

Die Bildverarbeitung ist in dem Bereich Medizin und medizinische KI-Systeme schon jetzt ein wichtiger Bestandteil und es wird in naher Zukunft eine immer wichtigere Rolle spielen. Es sind noch viele Fragen zu beantworten und es müssen groß Datenmengen ausgewertet werden, bevor die KI-Systeme in den Praxisgebrauch Anwendung findet [4].

2.1.4 Beobachtung

Desweiteren gibt es bei Drohnen einen weiteren Anwendungsbereich der Bildverarbeitung. Besonders im Bereich Militär werden Drohnen als Spionagewerkzeuge verwendet. Mit Hilfe solcher Drohnen lassen sich Bild- und Videoaufnahmen von Bereichen aufnehmen und diese Auswerten. In dem Artikel „ABUL - Automatisierte Bildauswertung für Unbemannte Luftfahrzeuge“ von dem Fraunhofer IOSB von Diol.-Inform. Heinze werden

detailliert die Aufgaben und Möglichkeiten solcher Drohnen beschrieben [7].

Die Vorteile eines unbemannten Flugfahrzeugs sind im Wesentlichen, dass kein Mensch in Gefahr gerät und diese von einem sicheren Standort gesteuert werden können. Diese werden mit Videotechnik ausgestattet und dienen für Erkundung und Aufklärungszwecke. Sie werden auch in Such- und Rettungsmissionen eingesetzt. Die Bodenstation erhält die von den Drohnen aufgenommenen Videodaten auf verschiedenen Übertragungswegen. Dort wird der angezeigte Videostream von menschlichen Bedienern mithilfe der neusten Technologie ausgewertet. Um die Personen während der Auswertung zu unterstützen, wurde das ABUL System entwickelt. ABUL bietet optimierte Echtzeitfunktionen für die Online-Erkundung und taktische Erkundung sowie wertvolle Funktionen für Offline-Erkundungsmissionen. Zur Aufklärung und Überwachung im Sicherheitsbereich werden zunehmend unbemannte Systeme eingesetzt, die hauptsächlich mit Videosensoren ausgestattet sind. Bei der Bundeswehr kommt schon seit Jahren die Drohne Luna zum Einsatz. Sie hat eine universelle Integrationsplattform implementiert, die einen automatischen Bildauswertungsprozess integriert [7].



Abbildung 2.3: UAV LUNA beim Abheben [7]

Neben der einfachen Visualisierung von Videostreams, der Bildoptimierung und der unregelmäßigen Korrektur von Infrarotsensormaterialien umfasst ABUL auch effiziente Funktionen für die Videoverarbeitung und -auswertung. Sowie intelligente Funktionen für die Verarbeitung von Videodaten und die Videoausgabe. Beispielhaft zu nennen sind hier:

- Änderungsdetektion
- Indikation von beweglichen Objekten(MTI)

- Videodatenbank zum Suchen nach Videos z. B. geografischen Regionen oder anderen Attributen wie Zeit- und Bodenauflösung



Abbildung 2.4: Änderungsdetektion



Abbildung 2.5: Indikation von beweglichen Objekten (MTI)

Wie in der Abbildung 2.4 zu erkennen ist, sind dort einige Fahrzeuge und Personen rot markiert und andere nicht. Diese haben sich im Vergleich der letzten Aufnahme verändert, daher werden die als Änderung aufgefasst und hier rot markiert. Weitere Anwendungen findet diese Auswertung auch im Bereich der Schadensbewertung nach Naturkatastrophen, für die Verkehrsüberwachung oder für die Dokumentation der Stadtentwicklung [9].

In der Abbildung 2.5 sind bewegende Fahrzeuge in grün markiert. Hierfür werden Videoaufnahmen ausgewertet. Das Verfahren ähnelt dem der Änderungsdetektion, nur dass hier Bilder aus einem Video miteinander verglichen werden, um somit eine Veränderung/Bewegung festzustellen.

2.2 Programmierung

In dem folgenden Abschnitt wird auf die Software und Eigenschaften der Bildverarbeitung eingegangen. Die Entwicklung der Software des Prototyps wurde auf einem Raspberry Pi vorgenommen. Auf dem Raspberry Pi wurde das Betriebssystem Raspbian installiert, welches auf Debian und Linux basiert. Es ist mit allen gängigen Betriebssystemen kompatibel und wird daher sehr häufig verwendet [3].

2.2.1 Programme

Die Programmierung wurde mit der Programmiersprache Python hergestellt. Sie ist eine der häufigsten verwendeten Programmiersprachen auf Raspbian. Eine wichtige Eigenschaft ist hierbei, dass Funktionsblöcke durch Einrücken und nicht wie in anderen Programmiersprachen durch geschweifte Klammern abgetrennt werden [3].

Zur Bildverarbeitung ist die Programmbibliothek OpenCV notwendig. Dort sind zahlreiche Algorithmen diesbezüglich implementiert. Die Software steht seit dem Jahr 2000 frei zur Verfügung. Wichtige Vorteile an der Bibliothek sind, dass das Wissen in einer großen Community verbreitet wird und auch viele Programmcodes zur Verfügung stehen [3].

2.2.2 Bildvorverarbeitung

Es gibt verschiedene Filter für die Bildvorverarbeitung. Beschrieben werden hier nur die, die auch im späteren Verlauf Anwendung findet. Nähere Beschreibungen und Funktionen sind in [3] oder in den dort verwiesenen Quellen zu finden.

Erstmals muss ein Glättungsfilter eingesetzt werden, um das Hintergrundrauschen zu minimieren und die Kanten zu Schärfen. Dies geschieht mit einem bilateralen Filter oder auch mit einem Gaußfilter. Um zum Beispiel Hintergrundrauschen zu minimieren in den kleinere Strukturen verschwommen werden, ist der Gaußfilter gut zu verwenden. Jedoch muss dabei beachtet werden, dass das Filter richtig dimensioniert ist, um keine Kanten und Objekte zu beeinflussen. Das bilaterale Filter basiert auf den Gaußfilter, mit dem Unterschied, dass er nicht nur den Räumlichen Abstand, sondern auch die Intensität der Pixel bewertet. Pixelunterscheide mit hoher Intensität haben in dem Filter eine größere Priorität, als welche mit weniger Intensität. Dadurch lassen sich Kanten besser hervorheben und werden nicht verwischt, solange auch hier auf die Dimensionierung des Filters

geachtet wird.

Das Filter ist bereits in der OpenCV Bibliothek enthalten und kann mit dem Befehl „cv2.bilateralFilter()“ aufgerufen werden [3].

2.2.3 Merkmalerkennung

Um Merkmale mit Hilfe von Algorithmen in der Bilderkennung festzustellen, werden besondere Punkte wie Kanten oder Ecken gesucht. Diese sind durch die Bildvorverarbeitung qualitativ besser aufgelöst und dargestellt. Solche Merkmale werden häufig durch Helligkeitsunterschieden zwischen zwei Pixeln festgestellt. Dazu wird der Canny-Algorithmus verwendet, welcher sich in der OpenCV Bibliothek befindet. Der Algorithmus kann über den Befehl „cv2.Canny()“ aufgerufen werden und als Ergebnis entsteht ein Bild welches nur die Objektkanten enthält. Somit ist eine guter Grundbaustein zur Kantenerkennung sichergestellt [3].

Die Anwendung des Canny-Algorithmus besteht aus mehreren Verarbeitungsschritten. Hierzu wurde eine kurze Zusammenfassung dargestellt, um die Anwendung aufzuzeigen. Nähere Informationen sind in [3] und dessen Quellen enthalten:

Der erste Verarbeitungsschritt, ist die Anwendung des Sobel Operators. Mit diesem Operator werden die Gradienten und deren Richtung festgelegt. Da hier breite Kanten entstehen, der Canny-Algorithmus jedoch Kanten von nur einem Pixel breit abbilden soll, wird hier das non-maximum-suppression angewendet. Dies bedeutet, dass nur die lokalen Maxima an der Kante hervorgehoben und nicht unterdrückt werden.

Die Intensitätsunterschiede der Pixel an den Kanten, werden mit Hilfe des Thresholdingverfahren festgelegt. Es entstehen zwei Schwellenwerte, T1 und T2, welche der Funktion übergeben werden. Das Verhältnis von T1:T2 entspricht in etwa 1:2. Nach abtasten eines höheren Wertes als T2 in einem Bild, wird an dieser Stelle ausgehend entlang der Kante getestet, ob diese Pixel größer als T1 sind. Anschließend erhält man eine Objektkante mit der Breite von nur einem Pixel, dessen Intensität größer als T1 ist.

Nun müssen die Kanten noch zu Objekten zusammengefasst werden. Dies geschieht mit der in der OpenCV-Bibliothek enthaltenden Funktion „cv2.findcontours()“. Der Funktion wird der Parameter „CV_RETR_EXTERNAL“ übergeben. Dieser fasst nur äußere Kan-

ten von Strukturen zusammen und speichert diese in einer Liste. Aus dieser Liste lassen sich Informationen, wie zum Beispiel Koordinaten und Größe des Objektes entnehmen und anschließend auswerten oder bearbeiten.

2.3 Insekten

In diesem Kapitel wird auf die allgemeinen Eigenschaften der Insekten eingegangen. Besonders wichtig ist hierbei die Betrachtung der Stechmücke und Fliegen. Da hier das Verhalten der Insekten im weiteren Verlauf der Arbeit näher betrachtet wird.

2.3.1 Allgemein

Die Tiergruppe der Insekten ist die größte der Welt. Sie zeichnen sich durch die Einkerbung zwischen Brust und Kopf sowie Brust und Hinterkörper aus, wodurch auch der Name entstanden ist. Der Begriff kommt von dem lateinischen Wort *insectus* „eingekerbt“. Je nach Art befinden sich unterschiedliche Mundwerkzeuge am Kopf. Es ist von der jeweiligen Nahrung abhängig ob das Werkzeug zum Sägen, Lecken oder Saugen entwickelt wurde. Solange von der bevorzugten Nahrung genug vorhanden ist, sind sie ein großer Vorteil für die Insekten. Sollte dies jedoch nicht möglich sein (durch Witterungsbedingungen) treten Komplikationen auf.

Die Entwicklung vollziehender Insekten als Metamorphose. Das beschreibt die Umwandlung von einer Larvenform in eine Erwachseneform. Von der Umwandlung gibt es verschiedene Varianten, zum einen welche die die Entwicklung in einer Stufe durchlaufen und andere die mindestens eine Larvenform benötigen, sich anschließend verpuppen und aus denen danach die erwachsenen Insekten schlüpfen. Außerdem gibt es noch einen gleitenden Entwicklungsverlauf, über verschiedene Larvenstadien [10].

2.3.2 Mücken

Stechmücken, aus der Familie Culicidae in Englisch Moskitos, fliegen im Vergleich zu Fliegen sehr grazil und fallen durch ihre langen, Fadenförmigen Antennen auf. In Deutschland gibt es ungefähr 40 und in Europa über 100 verschiedene Mückenarten, welche von Frühjahr bis Herbst aktiv sind [10].

Männliche Mücken unterscheiden sich von den weiblichen durch eine auffallend buschige Antenne. Durch die buschige Antenne werden die Vibrationen der Weibchen aufgenommen, um diese zu lokalisieren. Die Weibchen werden dadurch meistens in der Luft von den Männchen begattet. Zur Produktion der Larven benötigen die Weibchen Proteine. Diese bekommen sie durch das Blut von Menschen oder Tieren. Angelockt werden sie durch Hautausdünstungen und das darin enthaltene CO_2 . Ansonsten ernähren sich männliche und weibliche Mücken von Blütennektar. Die Larven werden meistens in stillen Süßwasser wie Seen, Teichen, Regentonnen oder ähnliches abgelegt [10].

In der Abbildung 2.6 ist die Entwicklung einer Mücke aufgeführt.

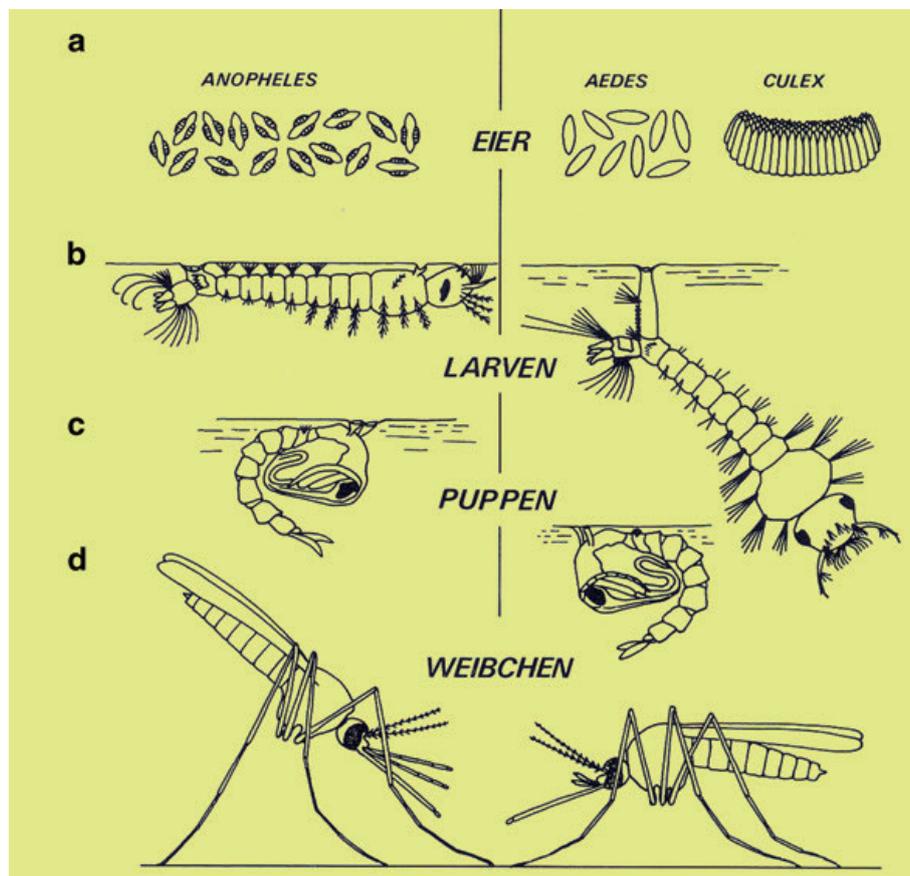


Abbildung 2.6: „Merkmale der wichtigsten Mückengattungen (nach Marshall 1938)“ [10]

- (a) Form der Eigelege
- (b) Larvenstellung im Wasser beim Luftholen

- (c) Puppenstellung im Wasser beim Luftholen
- (d) Ruhestellung der Weibchen bis 6mm groß

2.3.3 Fliegen

Die Stubenfliege (*Musca Domesica*) ist häufig in Mist, Fäkalien, Gülle oder Kompostanlagen aufzufinden. Sie treten das ganze Jahr, aber vermehrt im Sommer auf. Die Größe der Fliege liegt zwischen 6-7 mm und sie weisen eine dunkelgraue Farbe mit vier schwarzen Streifen auf. Ihre Beine sind schwarz-braun. Im Ruhezustand sitzt die Fliege meistens über Kopf an einer Wand, wobei sich die Flügelränder überschneiden. Zur Nahrungsaufnahme haben sie einen nach unten gerichteten Leckrüssel mit dem sie sich selbst zersetzende organische Substanzen, wie zum Beispiel Speisereste, aufnehmen [10].

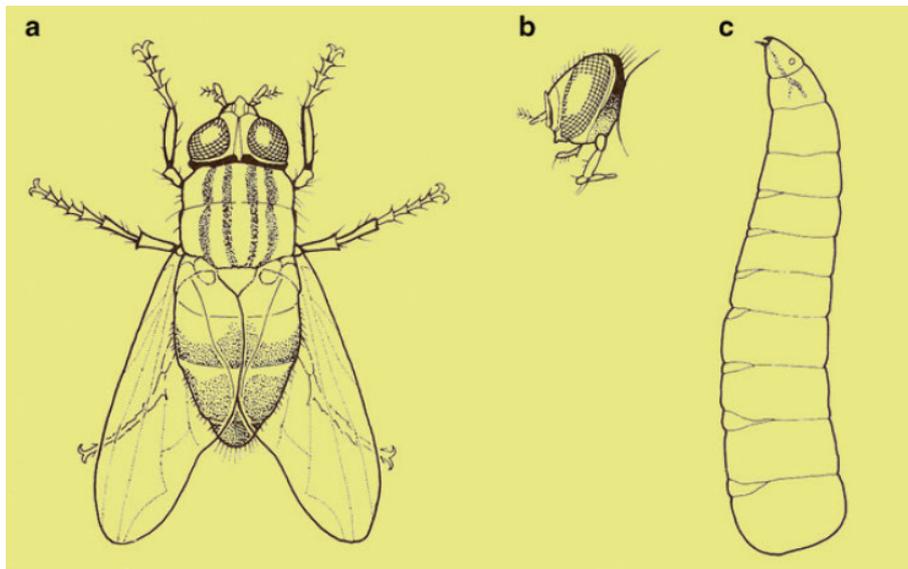


Abbildung 2.7: Schematische Darstellungen der Stubenfliege (*Musca domestica*) [10]

In Abbildung 2.7 ist folgendes dargestellt:

- (a) Adulte von Dorsal
- (b) Kopf von der Seite
- (c) Larve

Die Weibchen werden nur einmal befruchtet und legen danach ca. 4-8 Tage lang etwa 1000, $1 \times 0,26$ mm große Eier. Bevorzugte Orte hierfür sind, die Ausscheidungen von Tieren und Menschen, aber auch faulendes Material oder Wunden von Menschen und Tieren. Innerhalb eines Tages werden bis zu 120 Eier abgelegt und etwa 4-8 solcher Gelege erschafft eine Fliege innerhalb ihrer 70-tägigen Lebenszeit. Aus den Eiern schlüpfen mardenartige Larven. Sie sind 12mm lang, haben einen weißen beinlosen Körper und weisen Mundhaken auf. (Abb.2.7c) Das Larvenstadium dauert zwischen 4 und 10 Tagen und durchläuft dabei mehrere Häutungen, um am Ende zur braunen Tönnchenpuppe zu werden. So dauert es nun noch etwa 10 Tage bis die Adulten Fliege schlüpfen. Somit dauert eine Entwicklung je nach Temperatur ungefähr 8-50 Tage [10].

2.4 Auswirkungen des Lichtes auf Insekten

Die Wahrnehmung des Lichtes ist bei Insekten anders als bei Menschen. Dies ist notwendig, damit Insekten sich bei Dunkelheit von Mond- oder Sternenlicht orientieren können. Aus dem Grund wirken vor allem ultraviolettes, violettes, blaues und grünes Licht anziehend auf Insekten. Wie in der Abbildung 2.8 zu erkennen ist. In der Abbildung stehen die Nachtfalter stellvertretend für eine Vielfalt von nachtaktiven Insekten.

Es ist ein Maximum der Empfindlichkeit bei einer Wellenlänge von ungefähr 350 nm zu erkennen. Die Grenze von sichtbarem Licht zu UV-Licht liegt in etwa bei 400nm. Dies bedeutet, dass die Insekten Farben sehen, die von Menschen nicht wahrgenommen werden können. Ein weiteres Maximum liegt bei 520 nm, welches auch im Wahrnehmungsbereich von Menschen liegt und eine blau-grüne/ türkis artige Farbe aufweist. In der dargestellten Abbildung ist auch das Spektrum einer warm-weißen-LED dargestellt. Es ist deutlich zu erkennen, dass sich der größte Bereich nicht mit dem des Wahrnehmungsbereiches der Nachtfalter überschneidet. Lediglich im Bereich von 425nm bis ungefähr 550nm, was eine blaue bis grüne Farbe aufweist. Das Spektrum der LED überschneidet sich fast mit der Wahrnehmung der Menschen [5].

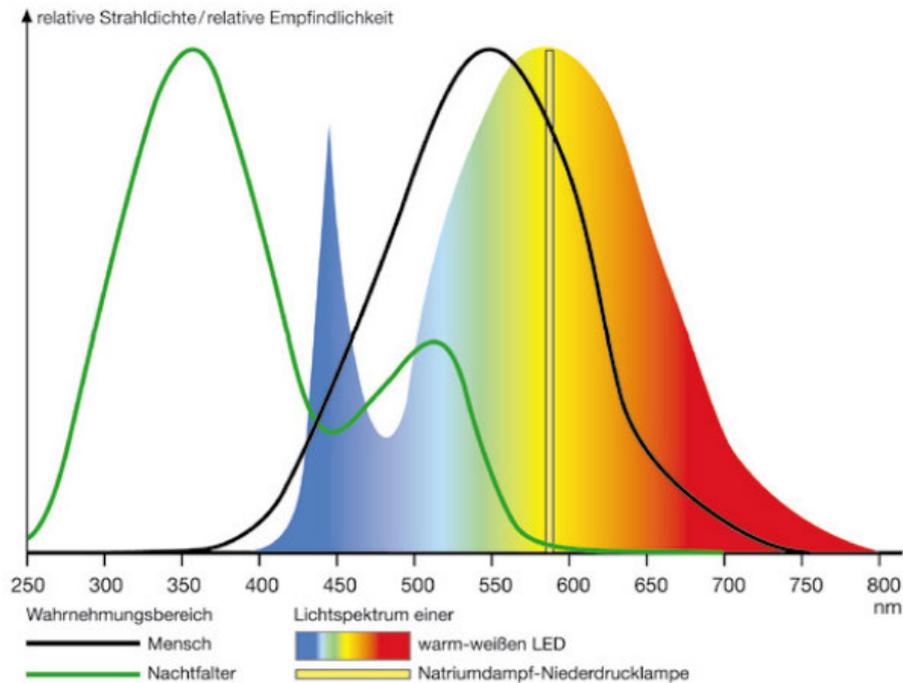


Abbildung 2.8: Wahrnehmungsbereiche von Menschen und Nachtfaltern in Gegenüberstellung einer warm-weißen-LED [5]

Vergleiche mit den hier verwendeten Leuchtmitteln werden in dem Kapitel 4 Leuchtmittel dargestellt.

Es ist erschwert, die Wirkung von künstlichem Licht auf einzelne Insektenarten abzuschätzen. Es gibt jedoch Hinweise darauf, dass die folgenden Arten besonders durch künstliches Licht bedroht sind:

- Arten, mit einer geringen Populationsdichte, die sich langsam vermehren und sich auf einen bestimmten Lebensraum spezialisiert haben.
- Arten, die selten in der Natur vorkommen oder welche die durch Menschen dezimiert wurden.
- Arten, die nur eine kurze Lebensdauer als Erwachsene haben, wodurch ihr Fortpflanzungszeitraum nur kurz ist. Dabei werden Insekten-Generationen bei der Fortpflanzung stark eingeschränkt, was zu massiven Populationseinbrüchen führen kann

[5]

3 Versuchsaufbau

Im ersten Kapitel wird der Versuchsaufbau inklusive Material dargestellt. Zudem wird die Implementierung des Programms und die Auswahl und Beschaffung der zu testenden Insekten beschrieben.

3.1 Testumgebung und Entwurf

Um eine geeignete Konstruktion zu erstellen, musste im ersten Schritt das Material bestimmt werden. Da in der Arbeit des Prototypen beschrieben wurde, dass die Konstruktion aus Kartons bestand, diese jedoch zum Ende der Versuche sehr gelitten haben und die Höhe nicht ausreichend war [3], fiel die Entscheidung auf eine Holzkonstruktion. Die Höhe der Kamera ist aus dem Betrachtungswinkel zu berechnen. Mit einem Betrachtungswinkel von $\alpha = 67^\circ$ und einer Gehäuselänge von $l = 51\text{cm}$, kann mit

$$\tan\left(\frac{\alpha}{2}\right) = \frac{l}{h} \Rightarrow h = \frac{l}{\tan\left(\frac{\alpha}{2}\right)} = 39,42\text{ cm} \quad (3.1)$$

die Höhe der Kamera bestimmt werden. Die Höhe wird somit auf 40 cm festgelegt. Die Konstruktion ist ähnlich wie im Prototyp [3] aufgebaut. Um die verwendeten Leuchtmittel unter den Testbereich zu platzieren ist das Gestell 30 cm hoch. Als untere Abgrenzung des Testbereiches und um die Lichtstrahlen zu brechen, damit es nicht zur Blendung der Kamera kommt, wurden zwei Lagen Diffusionsstoff befestigt. Zur oberen Abgrenzung wurde eine durchsichtige Folie, um die Sicht in den Testbereich nicht eingeschenkt. Die Außenwände des gesamten Aufbaus sind aus sechs mm dicken Sperrholzplatten und die Innenseiten sind mit weißem Lack bemalt, um Unreinheiten zu vermindern. Die Kamera und der Raspberry Pi befinden sich auf dem Deckel, welcher aus Pappe besteht. In der Tabelle 3.1 sind die Materialien für den Messaufbau aufgeführt und in der Abbildung 3.1 ist die fertige Konstruktion dargestellt.

Tabelle 3.1: Materialien für die Konstruktion

Anzahl	Beschreibung
4	Rahmenholz 24 x 44 x 3000 mm
2	Sperrholzplatten 1200 x 600 x 6 mm Buche
X	Schrauben 4,5 mm
X	Schrauben 3,5 mm
1	Weißer Lackfarbe
1	Diffusionsstoff
1	Durchsichtige Folie



Abbildung 3.1: Frontansicht Konstruktion

3.2 Software

3.2.1 Programmumgebung

Um das angefertigte Programm aus dem Prototypen in Betrieb nehmen zu können wird, ein Raspberry Pi und ein dazu passende Kamera benötigt

- Raspberry Pi 3 B, 4x 1,2 GHz, 1 GB RAM, WLAN, BT
- Raspberry Pi - Kamera, 8MP, v2.1, IMX219PQ

Der Programmcode ist im Anhang von [3] zu finden und wurden, von dort übernommen. Zuvor muss jedoch das Betriebssystem für den Raspberry Pi und das OpenCV installiert werden. Hier kam es zu einigen Schwierigkeiten, da bei einigen aufgeführten Installationen nicht alle Libaris eingebunden sind und die aktuelle Version 4.1.0 Änderungen im Code erfordert. Dadurch sind Syntaxfehler aufgetreten, die behoben werden mussten. Aus diesem Grund ist ein aktualisierter Code für die OpenCV Version 4.1.0 im Anhang A zu finden.

3.2.2 Programmbeschreibung

Nach dem Start der main.py öffnet sich eine grafische Benutzeroberfläche (GUI). (Abb. 3.2) Hier kann durch Anklicken ein gewünschtes Programm zur Analyse ausgewählt werden. Optional zu wählen ist der zwei Zonenmodus, wo das Bild in zwei Hälften geteilt und jede einzeln ausgewertet wird. Zu dem kann noch entschieden werden, ob ein zuvor aufgenommenes Bild oder Video ausgewertet werden soll, oder eine Liveansicht der Kamera. Je nach Auswahl werden Felder, die keine Verwendung dafür haben, rot hinterlegt, um optisch darzustellen, dass diese nicht verwendet werden können. Zum Schließen des Programms oder des Videos ist die „Q“ Taste zu drücken.

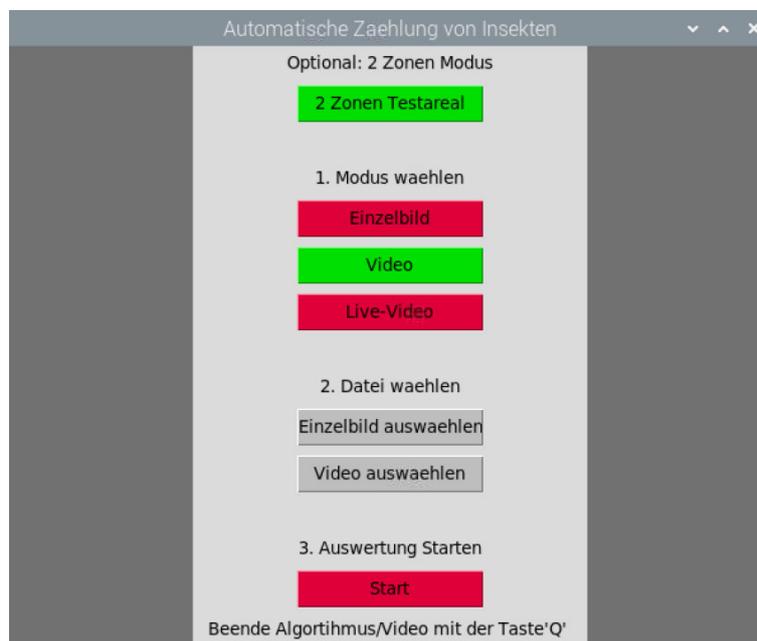


Abbildung 3.2: Benutzeroberfläche

Wie schon angedeutet, müssen Video- und Bildaufnahmen zuvor aufgezeichnet werden. Die dafür verwendeten Codes sind im Anhang A zu finden. Nach der Aufnahme kann die Datei über die GUI aufgerufen werden, welche sich daraufhin in einem Image öffnet. Zu sehen ist ein Bild wie in Abbildung 3.3. Solange dieses Image geöffnet ist, wird das Programm zur Objekterkennung durchlaufen.

Dafür wird der Frame in schwarz-weiß gewechselt, um nur Graustufen zu berücksichtigen. Dieser wird dann mit einem Bilateralfilter gefiltert, um wie in den Grundlagen beschrieben, kleinere Strukturen im Hintergrund zu minimieren.

Anschließend werden die Kanten mit dem Canny-Algorithmus ausgewertet. Dafür existiert eine neue Funktion. Mit dessen Hilfe die Maxima und Minima für das Canny-Filter bestimmt werden. Als Rückgabewert erhält man ein Bild, in dem nur Kanten dargestellt werden. Im nächsten Schritt werden diese Kanten zu Umrissen, wie in den Grundlagen beschrieben, zusammengefasst. Zuletzt werden die Konturen mit den eingestellten maximalen und minimalen Größen abgeglichen, gezählt und mit Vierecken umrandet. (Abb. 3.3)

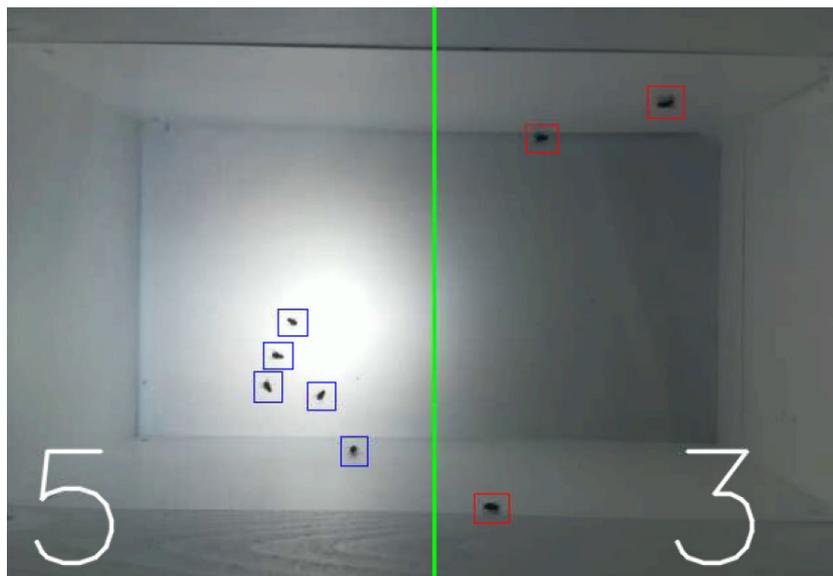


Abbildung 3.3: Beispielaufnahme der Kantenerkennung

Die zuvor implementierten Auswertungsdiagramme werden mit einem Balkendiagramm ergänzt. Diese werden in dem Kapitel 5. Verhaltensanalyse näher beschrieben.

3.3 Auswahl und Beschaffung von Insekten

Die Auswahl der Insekten ist auf die Stechmücke gefallen, da diese wie in den Grundlagen geschrieben, nachts aktiv sind. Zudem ist die Stechmücke leicht zu beschaffen, sofern Zugang zu einem Garten oder See besteht und Sommer beziehungsweise früher Herbst wird.

Nach mehreren missglückten Versuchen rote oder weiße Mückenlarven, aus einer Zoo-handlung, schlüpfen zu lassen. Wurde sich hierfür an das Aquarium-Center Löhning gewendet. Zur Beschaffung der Larven, wurde nur ein Eimer mit Wasser und einigen getrockneten Blättern befüllt und in einen Garten oder in die Nähe eines Sees gestellt. In das Wasser legen die Mücken dann ihre Eier ab, die sich nach einiger Zeit zu Larven entwickeln. Aus denen dann, wie in den Grundlagen beschrieben, die Mücken entstehen.

Sobald das Larvenstadium erreicht ist, wurden diese in ein kleineres Wasserbecken innerhalb einer Art Terrarium gefüllt. Aus diesem sie nach dem Schlüpfen nicht wegfliegen können.

Um die Mücken in den Testaufbau zu transportieren ist Geduld und Planung gefragt. Dafür wurde an die Außenseite der Eingangsklappe des Testbereiches ein Fliegennetz befestigt und über das Terrarium gezogen. Damit es nur einen Weg gibt in den Testaufbau zu gelangen.

Da die Tests mit den Mücken nicht aussagekräftig waren, haben die späteren Tests auch mit Fliegen stattgefunden. Diese sind einfacher und ganzjährig zu beschaffen. Hierfür gibt es verschieden Händler im Internet die Fliegen in dem braunen Tönnchenpuppen Stadium verkaufen. Diese schlüpfen dann nach etwa 10 Tagen wie schon in den Grundlagen beschrieben.

Fliegen werden nach einem kurzen Aufenthalt im Kühlschrank langsam und träge, wodurch sie einfach in den Testaufbau gelegt werden können.

4 Leuchtmittel

Um das Verhalten der Insekten auf verschiedenen Lichtfarben untersuchen zu können, werden für die folgenden Tests verschiedene Leuchtmittel mit unterschiedlichen Lichtfarben verwendet. Die verwendeten Leuchtmittel sind in der Tabelle 4.1 aufgelistet.

Tabelle 4.1: Lampenauswahl

	Lichttemperatur [K]	Beschreibung	Lichttemperatur gemessen [K]	Lichtstrom [lm]
LED	2000	Müller licht Retro LED	X	150
LED	3000	Kopos-LED BULB A60	3065	461
LED	6500	Philips LED cool daylight	6819	1337
FL	2700	Philips warm white	2787	466
FL	6500	Philips cool daylight	6547	186

Unter dem Grundlagenpunkt 2.4, wurde der Wahrnehmungsbereich von Nachtfaltern bereits aufgezeigt. Im nun folgenden Verlauf werden die Spektren der verwendeten Leuchtmittel dargestellt, um diese mit dem Wahrnehmungsbereich von Nachtfaltern zu vergleichen. Die durchgeführten Messungen von ESYLUX sind unter Anhang C zu finden. Zu der „Müller licht Retro LED“ existieren keine Messungen, da diese während des Transportes beschädigt wurde. Aus diesem Grund wurde das Spektrum eines alternativen Leuchtmittels aus dem Internet verwendet, um das Spektrum ungefähr betrachten zu können (Abb. 4.1).

4.1 LED Lampen

Nachfolgend werden die Spektren der verwendeten LED-Lampen dargestellt und ihre Besonderheiten beschrieben. Zum dem werden die Spektren mit dem Wahrnehmungsbereich des Nachtfaltern verglichen.

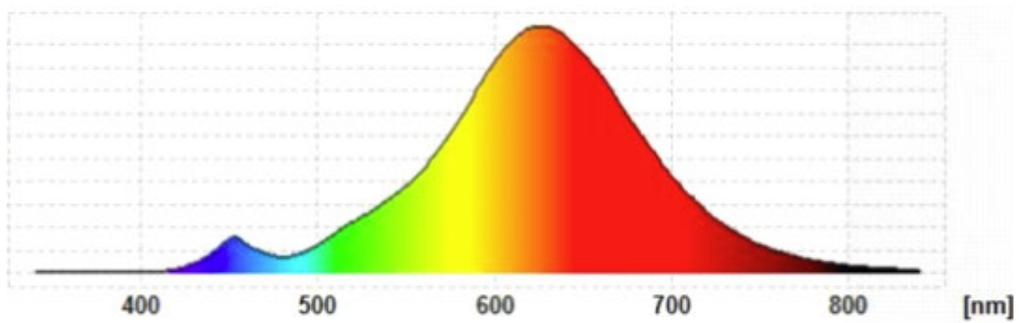


Abbildung 4.1: Spektrum der Müller licht Retro LED 3000K Lampe [11]

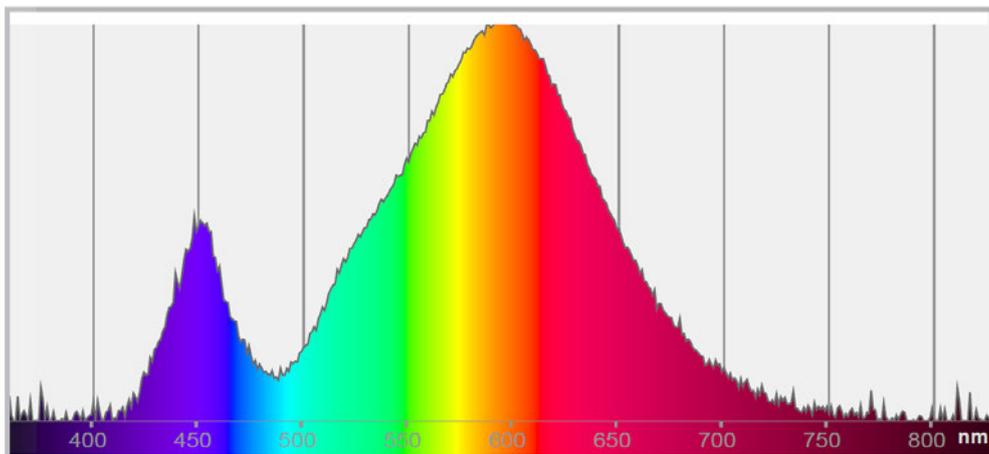


Abbildung 4.2: Spektrum der Kopus-led BULB A60 LED 3000K Lampe [C Anhang]

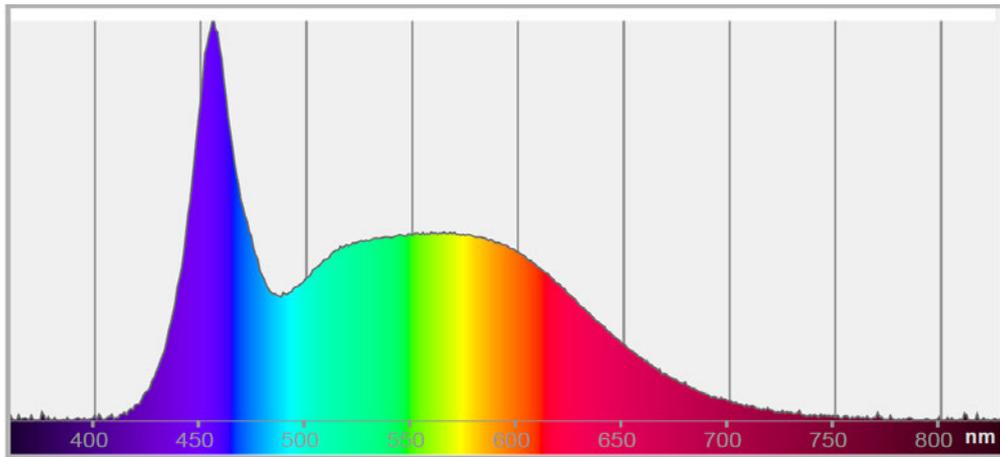


Abbildung 4.3: Spektrum der Philips LED Cool daylight 6500K Lampe [C Anhang]

Tabelle 4.2: Bestrahlungsstärke
LED 3000K

In nm	In $mW/m^2/nm$
400	0,45
450	4,60
500	1,72
550	6,09
600	9,33
650	4,52

von Tabelle 4.3: Bestrahlungsstärke
LED 6500K

In nm	In $mW/m^2/nm$
400	0,5
450	34,2
500	15,2
550	19,8
600	18
650	8,2

Auffällig ist bei diesen Leuchtmitteln das erste Maximum, welches sich bei allen etwa auf 450nm befindet. Der Wert bei der 6500K LED ist jedoch um ein Vielfaches höher, als bei den anderen LED-Lampen. 450nm liegt ungefähr auf derselben Wellenlänge, wie das Tal des Wahrnehmungsbereiches von Nachtfaltern. Daraus folgt, dass das Maximum der 6500k LED nicht komplett wahrgenommen werden kann. Ein weiterer zu beachtender Bereich ist, das Zweite Maximum des Wahrnehmungsbereiches bei etwa 520nm. Auch wenn die 3000K LED ungefähr $4mW/m^2/nm$ und die 6500K LED knapp $19mW/m^2/nm$ abgibt. Der entschiedene Bereich bei der 2000K LED liegt noch weiter im roten Spektrum. Dadurch und weil sie fast kein blaues Licht ausstrahlt, sollte dieses Leuchtmittel nahe zu keine Anziehung auf Insekten haben. Das Maximum der 3000K LED nimmt ein Nachtfalter kaum wahr, wodurch diese Lampe kein großes Lockverhalten aufweisen sollte. Die 6500K LED hingegen sollte durch die hohe Intensität ein größeres Lockverhalten aufweisen.

4.2 Energiesparlampe FL

Auch hier werden die Spektren der Energiesparlampen dargestellt, analysiert und mit dem Wahrnehmungsspektrum des Nachtfalters verglichen.

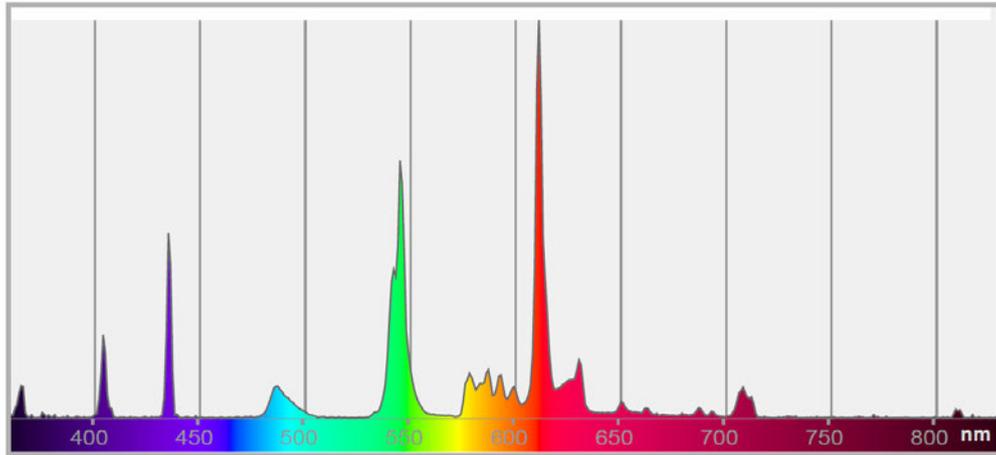


Abbildung 4.4: Spektrum der Philips warm white FL 2700K Lampe [C Anhang]

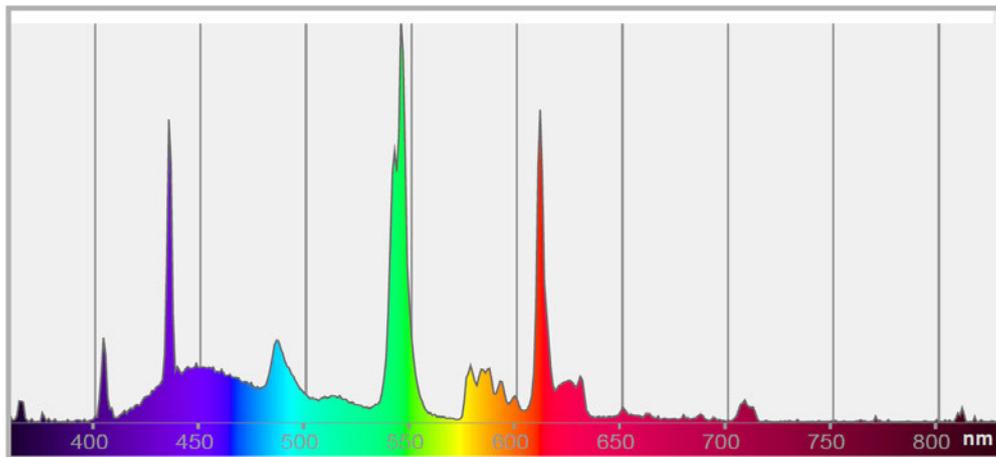


Abbildung 4.5: Spektrum der Philips cool daylight FL 6500K Lampe [C Anhang]

Die Messwerte aus den Messungen von ESYLUX sind leider nicht auf den Maxima der Bestrahlungsstärke, daher ist es schwer diese zu bestimmen. Jedoch sind die Werte bei beiden Leuchtmitteln auf der gleichen Wellenlänge. Mit Hilfe eines Dreisatzes wurden ungefähre Werte ermittelt, wie in den Tabellen 4.4 und 4.5 zu erkennen ist.

Tabelle 4.4: Bestrahlungsstärke von Energiesparlampe 2700K

In nm	In $mW/m^2/nm$
360	2,25
410	5,61
435	12,37
490	2,22
540	17,2
620	26

Tabelle 4.5: Bestrahlungsstärke von Energiesparlampe 6500K

In nm	In $mW/m^2/nm$
360	1,11
410	4,2
435	14,13
480	3,86
540	14,58
620	20,44

Auffällig ist, dass beide Energiesparlampen kurz nach 350nm ein kleinen Ausschlag haben, welcher genau in dem Wahrnehmungsbereich der Nachtfalter liegt. Jedoch ist der Bereich zwischen 400nm und 450nm entscheidender für das Lockverhalten. Denn in dem Bereich sind zwei Maxima und zusammen bilden diese ungefähr ein Drittel der gesamten Leuchtkraft. Zudem sind bei den ersten Maxima die Wellenlängen größer und somit mehr in dem starken Wahrnehmungsbereich der Insekten. Weiterhin sind die Maxima bei 540nm auch in dem Wahrnehmungsbereich.

Jedoch ist die Bestrahlungsstärke bei beiden Energiesparlampen ähnlich. Aus dem Grund kann keine Vermutung zum Lockverhalten aufgestellt werden.

Beim Vergleich von den Spektren der LED 6500K und der FL6500K ist auffällig, dass auch wenn die Maxima der FL 6500k weiter in dem Wahrnehmungsbereich eines Nachtfalters liegen, die Intensität der LED 6500 K weitaus höher ist. Das ist dem Unterschied, des Lichtstromes zuzuschreiben. Denn dadurch ist die Bestrahlungsstärke der Energiesparlampe nicht so intensiv wie die der LED-Lampe. Aus dem Grund wird die LED 6500K vermutlich ein stärkeres Lockverhalten aufweisen können.

5 Experimente zur Verhaltensanalyse

In diesem Abschnitt wird die Durchführung der Verhaltensanalyse beschrieben. Hierfür werden die zuvor aufgenommenen Videos ausgewertet, um die Ergebnisse im späteren Verlauf diskutiert. Als Ergebnis der Auswertung werden die folgenden zwei Darstellungen ausgegeben:

1. In der Darstellung (Abb. 5.1) sind drei Graphen in einem Diagrammen aufgeführt. In jedem Graphen werden die Anzahl der Insekten in den jeweiligen Bereichen (left-, right-, completearea) über die Zeit dargestellt. Hier kann die Bewegung der Insekten verfolgt und verglichen werden.
2. In der zweiten Darstellung (Abb. 5.2) wird für den linken und rechten Bereich in einem Diagramm dargestellt, wie viele Insekten sich wie lange in einem Bereich aufgehalten haben. Diese Auswertung dient dazu, einen allgemeinen Überblick zu gewinnen und die Aufenthaltsdauer der Insekten zu erfassen.

Zu Beginn der Darstellung 5.1 werden die Steller ausgerichtet, wodurch es zu Ungenauigkeiten kommen kann.

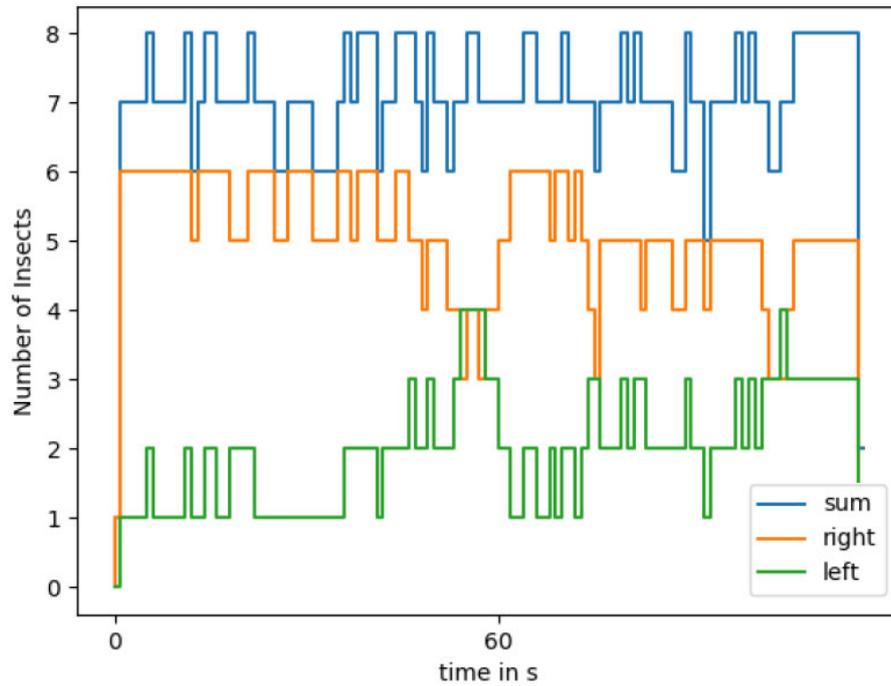


Abbildung 5.1: Zeitliche Auswertung der Positionen von Objekten in verschiedenen Bereichen

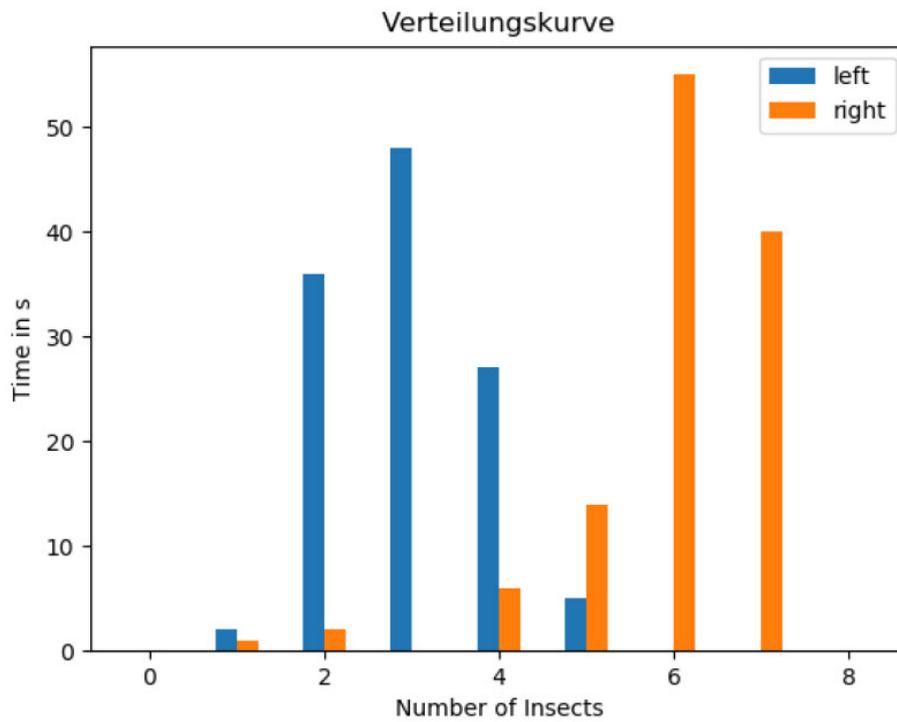


Abbildung 5.2: Verteilungsstatistik der Aufenthaltsdauer von Objekten in verschiedenen Bereichen

5.1 Testverfahren

Wie zuvor bereits beschrieben, halten sich die Insekten innerhalb des Testbereiches auf. Unterhalb des Bereiches befinden sich unterschiedliche Leuchtmittel, dessen Licht durch ein Diffusionsstoff gebrochen wird. Dies wird genutzt, um die Kamera auf der gegenüberliegenden Seite nicht zu blenden. Mit dem Programm werden nun die aufgenommenen Videos eingelesen und danach die Positionen der Insekten ausgewertet.

5.1.1 Grundlagentests

Die Grundlagentests dienen dafür herauszufinden ob die Insekten von Licht angelockt werden oder ob es sie nicht beeinflusst. Für diese Untersuchung standen die unterschiedlichen Leuchtmittel, welche in Kapitel 4 benannt sind, mindestens zwei mal abwechselnd pro Seite. Dies war nötig, um die Bewegung der Insekten zu dokumentieren. Hierbei ist wichtig, dass sich möglichst viele Insekten von der Seite aus dem ersten Test, zur anderen Seite bewegen. Also können hier für die erste und zweite Darstellungsart verwendet werden. Sollten diese kein genaues Ergebnis aufweisen, kann die dritte Darstellungsart genutzt werden, um zu erkennen, ob sich eventuell mehr Insekten länger in dem beleuchteten Bereich aufhielten.

5.1.2 Vergleichstests

Bei den Vergleichstests wurden jeweils in den linken und rechten Bereich ein Leuchtmittel platziert. Dadurch kann das Lockverhalten der verschiedenen Leuchtmittel untersucht werden. Die unterschiedlichen Tests beziehen sich hier auf die Lichttemperatur und die Art des Leuchtmittels

5.2 Tests allgemein und Grundlagentests mit Mücken

In dem folgenden Unterpunkten sind die Auswertungen der Grundlagentests bei Mücken aufgeführt. Es sind keine Vergleichstest vorhanden, da die Grundlagentest weitere Untersuchungen von Mücken in diesem Versuch unnötig machen.

Eine Aufnahme des Testbereiches in dem sich fünf Mücken befinden wird in Abbildung 5.3 dargestellt.



Abbildung 5.3: Testbildaufnahme von fünf Mücken im Testbereich

In der Abbildung 5.3 sind die Mücken nur schwer erkennbar. Diese sind, wie in den Grundlagen aufgeführt, 6mm groß, jedoch sehr dünn. Das führt bei der Bildererkennung zu Problemen. Der `ContourSizeMin` Regler steht auf eins, damit die Mücken erkannt werden. Trotz dessen ist die Erkennung nur der Mücken nicht möglich. Im linken Feld wird eine Mücke nicht erkannt. Hier kann kein Unterschied zwischen Mücke und Unreinheit gemacht werden, welches zu fehlerhaften Auswertungen führt. Daraus folgt, dass die Auflösung der Pi-Kamera nicht ausreichend ist, um die Körper der Mücken hinreichend darzustellen.

Auch bei Versuchen mit einer Smartphonekamera, war es nicht ohne weiteres umsetzbar, wie in Abbildung 5.4 zu erkennen ist. Die Aufnahme ist mit einer 20MP Kamera eines OnePlus5 in demselben Format 4:3 entstanden. Die Pixeldichte entsprach 720p. Das verwendete Leuchtmittel war die LED mit 6500K Lichttemperatur. Wie in der Abbildung 5.4 zu erkennen ist, zeigt das Bild Einzelheiten viel deutlicher als in Abbildung 5.3. Dies macht jedoch auch die Unreinheiten im Bild deutlich. Die Schrauben und Kanten sind genauer zu erkennen, die für die Konstruktion verwendet wurden oder entstanden sind. Die entstandenen Kanten stellen in der Auswertung Objekte da, wodurch sie verfälscht und unbrauchbar ist. Der Testbereich müsste annähernd makellos sein und die Bildererkennung müsste an die verwendete Kamera angepasst sein, um eine gute Auswertung zu generieren.

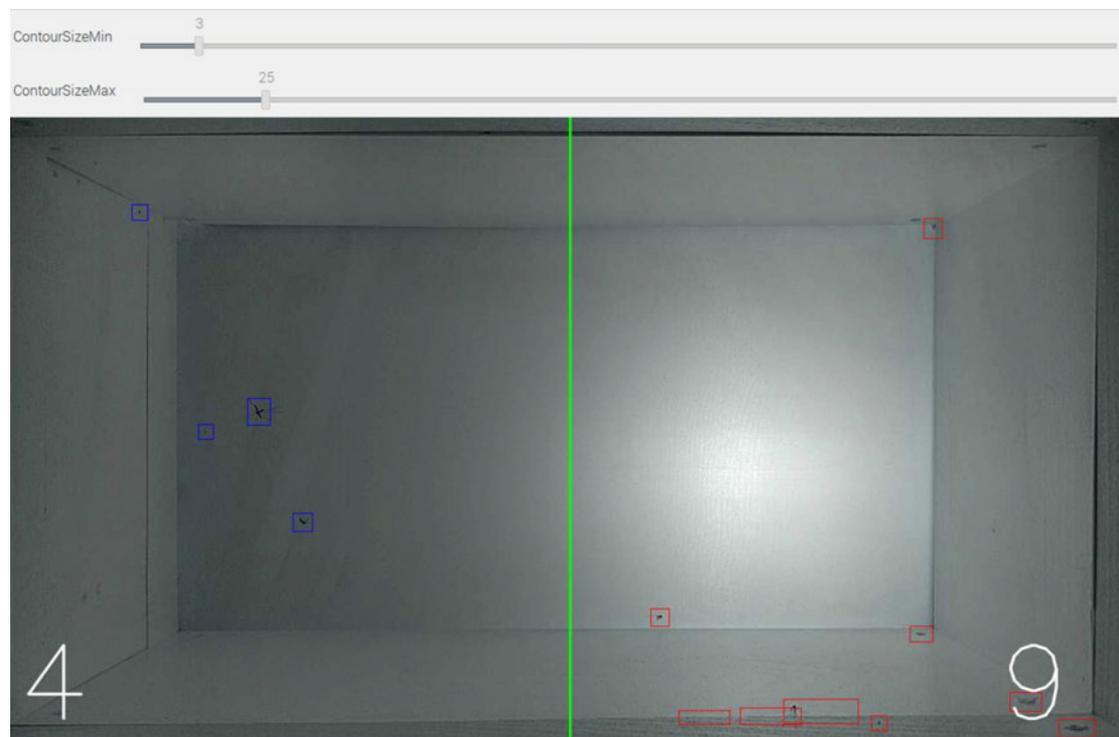


Abbildung 5.4: Testbildaufnahme von einem Handy, von fünf Mücken im Testbereich

Bei den Auswertungen des Verhaltens bei Mücken, ist besonders aufgefallen, dass diese sich sobald das Licht angeschaltet wurden, nicht mehr bewegt haben. Dabei gab es auch kein Unterschied, welches Leuchtmittel verwendet wurde. In der folgenden Abbildung 5.5 ist eine Auswertung von Mücken bei der LED 6500K zu erkennen.

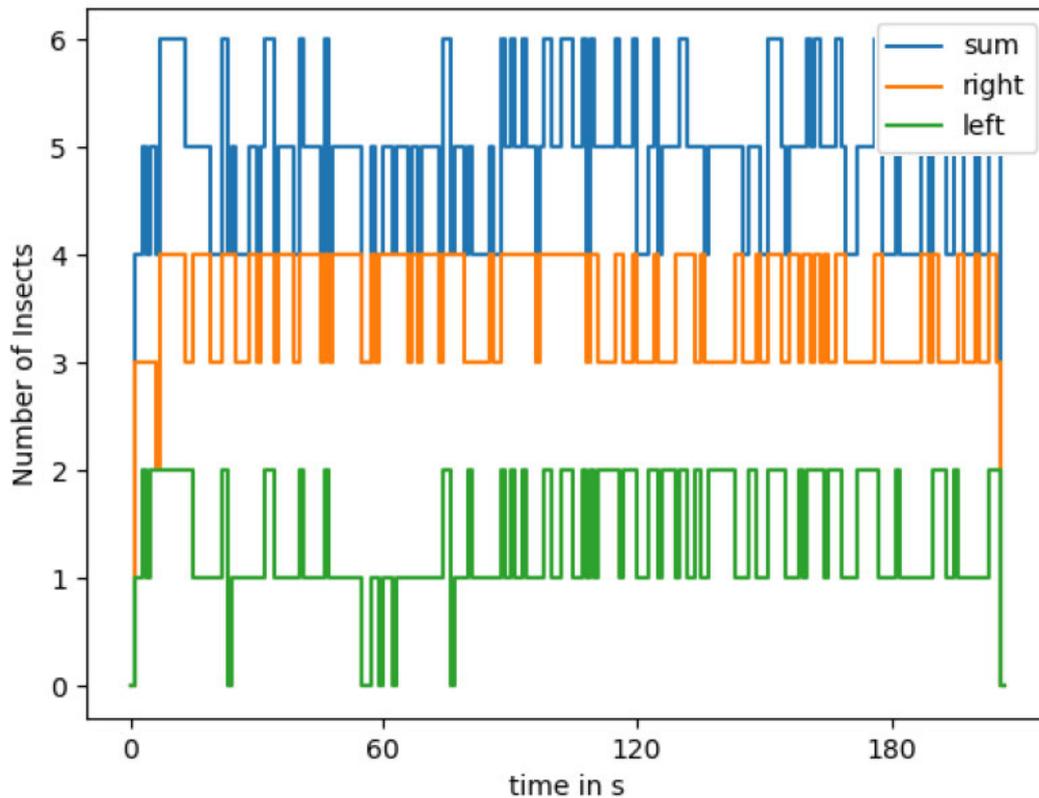


Abbildung 5.5: Auswertung von Mücken bei einer LED 6500K

Deutlich erkennbar ist, das starke schwanken der Messwerte, welche durch die zuvor beschreibenden Bedingungen entstanden ist. Solche Ergebnisse sind nicht aussagekräftig. Zu dem kommt, dass sich die ganze Zeit über drei Mücken rechts und zwei Mücken links befunden haben. In der Auswertung ist jedoch ein Trend zu erkennen, dass sich drei Mücken rechts jedoch nur eine links befunden haben. Die Ungenauigkeit am Anfang des Diagramms entstand durch die Positionierung der Regler.

Auch Auswertungen anderer Videos mit anderen Leuchtmitteln ergaben ähnliche Ergebnisse. Auffällig war, dass sich die Mücken nicht bewegt haben. Auch starkes Rütteln oder Klopfen an der Außenwand haben keine Wirkung gezeigt.

Aus dem Grund wurde eine Aufnahme über 14h gemacht, um sicherzustellen, ob eine längere Auswertung kein anderes Ergebnis liefert. Vielleicht benötigen die Mücken eine Eingewöhnungszeit an das Licht. Die Auswertung der Zeitreihe ist aufgrund der vielen Datenpunkte schwer zu analysieren, da durch den Wertewechsel der linken Seite nur ein breiter grüner Balken zwischen eins und zwei erkennbar ist. In der Darstellungsart zwei ist es jedoch ein Ergebnis zu erkennen.

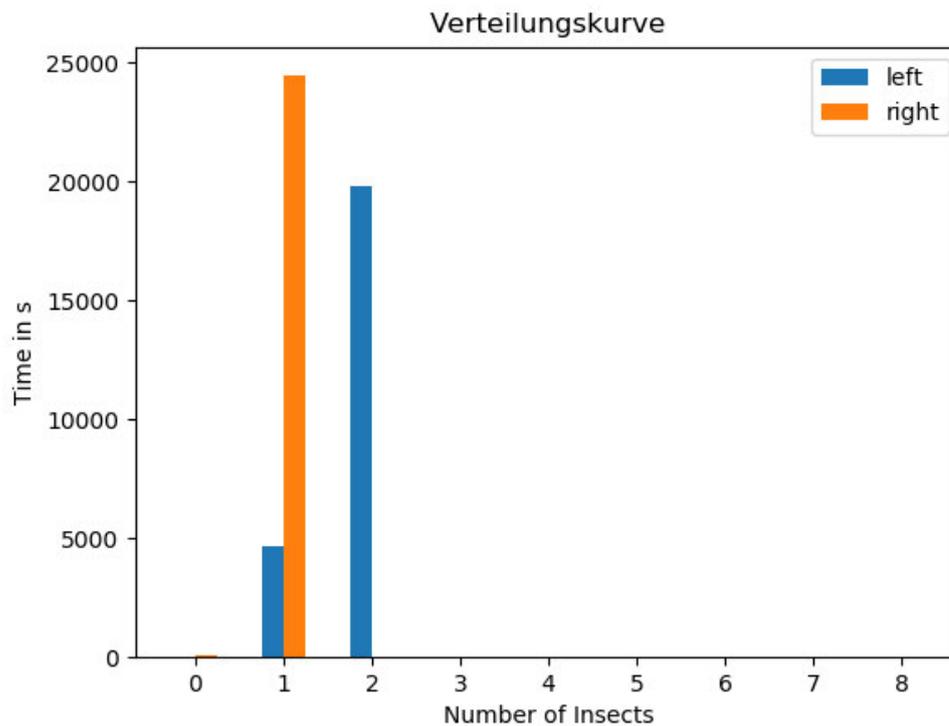


Abbildung 5.6: Verteilungsstatistik von 14h Video bei LED 3000K

Deutlich zu erkennen ist, dass sich auch bei einer längeren Aufnahme über Nacht keine Bewegung der Mücken zeigt. In der rechten Hälfte befindet sich den ganzen Zeitraum über nur eine Mücke. in dem rechten weitestgehend zwei, wobei einer häufiger nicht erkannt wurde. Daher wurden keine weiteren Versuche mit Mücken durchgeführt, weil dadurch keine neuen Aussagen getroffen werden können.

5.3 Grundlagentest mit Fliegen

Die Aufnahmen, welche für die Auswertung aufgenommen wurden, sind alle länger als 10 Minuten. Im Gegensatz zu den Ergebnissen der Mücken, sind die Ergebnisse der Fliegen weitaus interessanter und aussagekräftiger. Dies liegt unter anderen daran, dass sich die Fliegen auch während der Aufnahme bewegen und auf Grund ihrer Form deutlich leichter von Unebenheiten zu unterscheiden sind. Dies ist deutlich in der Abbildung 5.7 zu erkennen. Hier sind nur die Fliegen markiert. Im Gegensatz zu den Mücken, wo der `ContourSizeMin` -Regler auf maximal eins gestellt werden konnte, kann dieser hier auf vier oder höher gestellt werden. Das reicht aus, um Unreinheiten zu ignorieren.

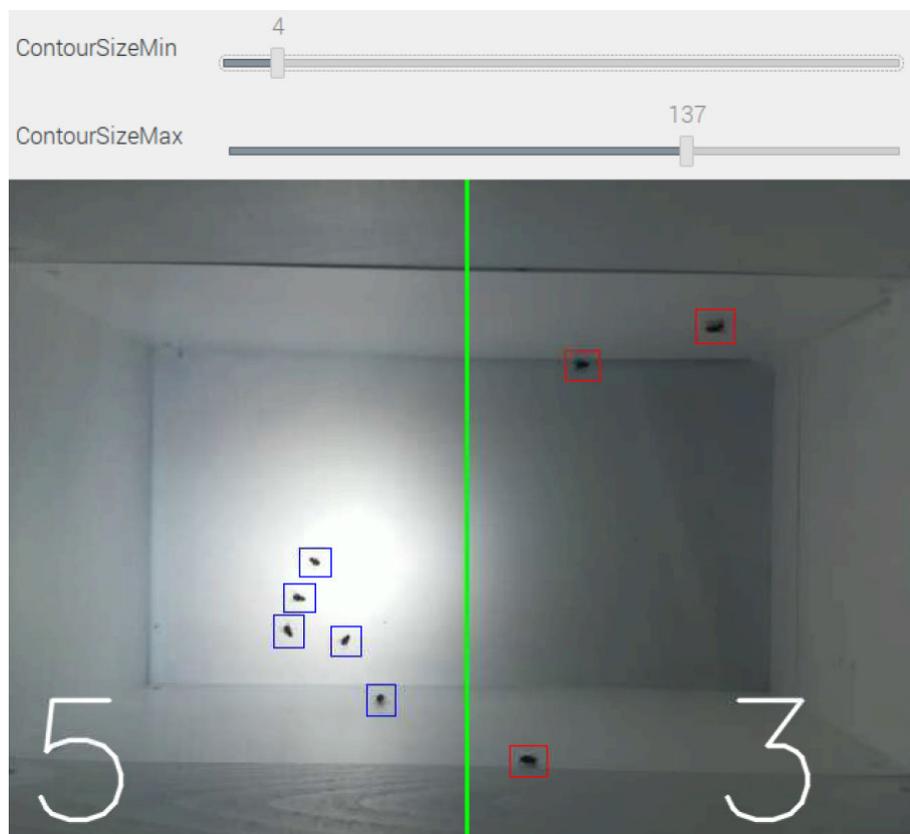


Abbildung 5.7: Testbildaufnahme von acht Fliegen im Testbereich

Bei den Messungen haben sich immer acht Fliegen in dem Testbereich befunden. Außerdem sollten es nicht zu viele sein, um zu vermeiden, dass sich mehrere Fliegen auf einem Fleck befinden und so das Ergebnis verfälschen.

Zu jedem Leuchtmittel wurden mehrere Videos aufgenommen und ausgewertet, welche im folgenden Verlauf aufgeführt sind. Hier werden pro Leuchtmittel jeweils zwei Tests dargestellt, die sich auch in den weiteren Tests weitgehend widerspiegeln.

5.3.1 Grundlagentests Leuchtmittel LED 6500K

Wie anhand der Auswertungen in den Abbildungen 5.9 zu erkennen ist, sind die Fliegen zu Beginn der Auswertung relativ verstreut. Das Leuchtmittel befindet sich unter dem linken Bereich. Im weiteren Verlauf der Aufnahme wird die Anzahl der Fliegen in der linken Hälfte größer und in der rechten weniger. Was bedeutet, dass sich die Fliegen in die beleuchtete Hälfte bewegen. Darüber hinaus setzen sich einige Fliegen genau über dem Leuchtmittel, wie in Abbildung 5.8 zu erkennen ist, was für ein hohes Lockverhalten spricht.

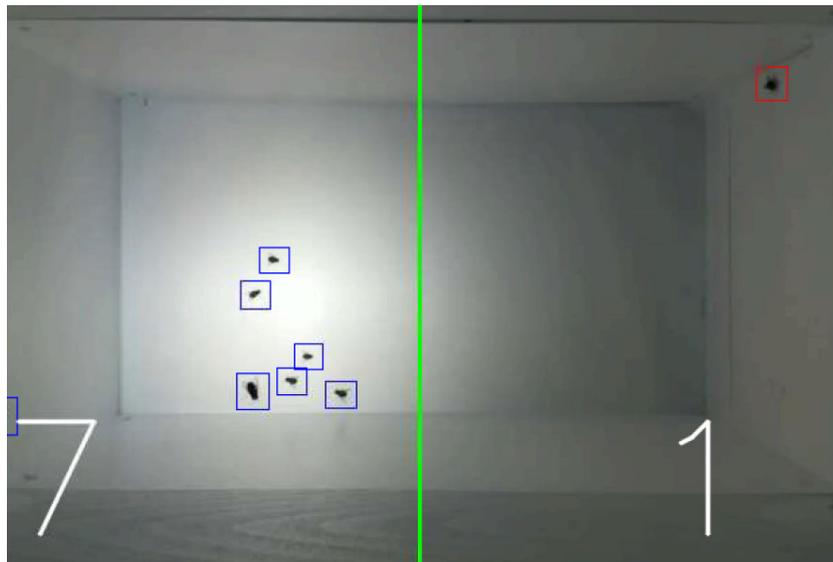


Abbildung 5.8: Position der Fliegen während der Auswertung von der LED 6500K

In der Zweitreihe (Abb. 5.9) ist auffällig, dass sich im mittleren die Gesamtzahl der Insekten auf 7 reduziert hat. Dies wird darauf zurückgeführt, dass sich zu dem Zeitpunkt zwei Fliegen nebeneinander befinden. Wodurch diese dann nur als eine erkannt werden. Zu dem sind zwei Fliegen kurzzeitig in den rechten Bereich gewechselt.

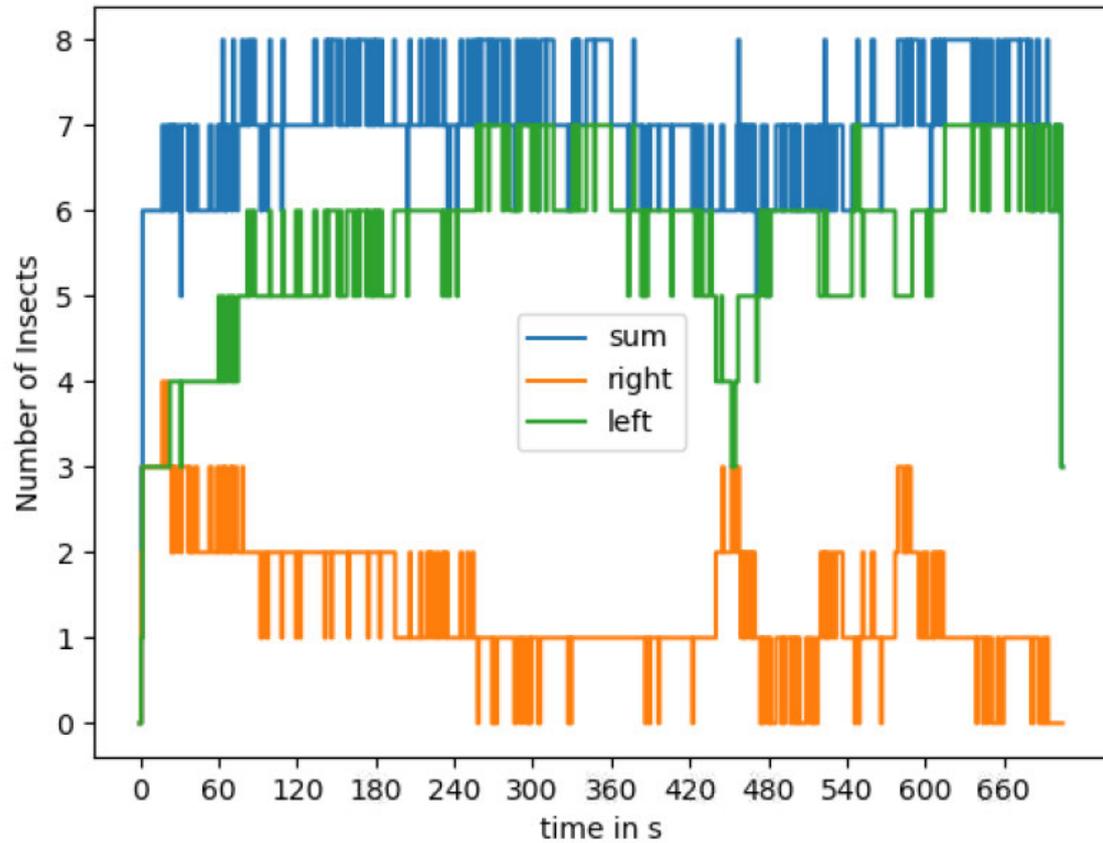


Abbildung 5.9: Zeitliche Auswertung von acht Fliegen mit LED 6500K links

Zur Sicherstellung des Ergebnisses wird das Leuchtmittel anschließend unter die rechte Seite des Bereiches platziert. Anfangs befinden sich noch mehr Fliegen in dem linken Bereich. Auch bis kurz vor Ende der Aufnahme ist ein ständiger Wechseln zu erkennen. Jedoch am Ende ist eine eindeutige Tendenz zur rechten Seite erkennbar. (Abb. 5.10) Auch hier befinden sich die Fliegen wieder über dem Leuchtmittel.

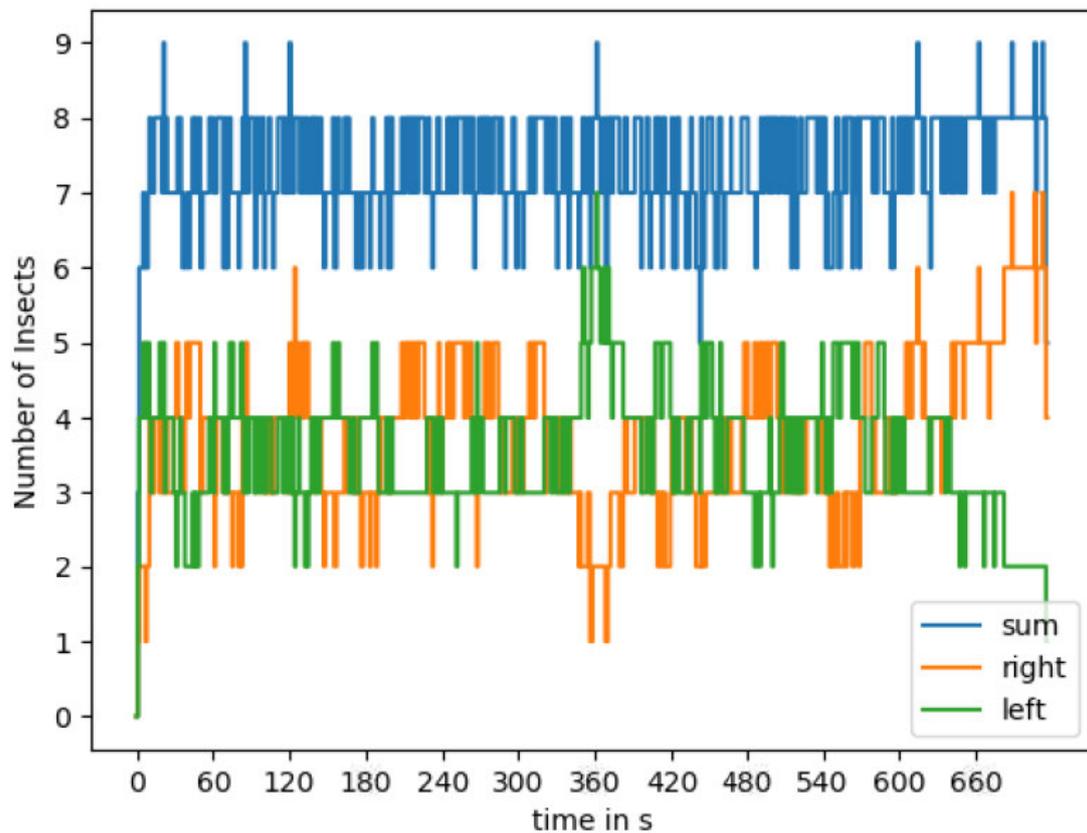


Abbildung 5.10: Zeitliche Auswertung von acht Fliegen mit LED 6500K rechts

Daraus lässt sich erschließen, dass das Leuchtmittel „Philips LED cool daylight“ ein starkes Lockverhalten auf die Fliegen hat, auch wenn diese eine Eingewöhnungszeit benötigen.

5.3.2 Grundagentests Leuchtmittel LED 3000K

Das nächste zu testende Leuchtmittel ist die „Kopos-LED BULB A60“. Bei der Auswertungsvariante eins (Abb. 5.11) ist kein eindeutiges Ergebnis zu erkennen. Die zeitliche Auswertung schwankt stark. Anhand der Auswertung ist zu erkennen, dass sich die meiste Zeit in beiden Seiten gleich viele Fliegen aufhalten und es keine entschiedenen Unterschiede gibt.

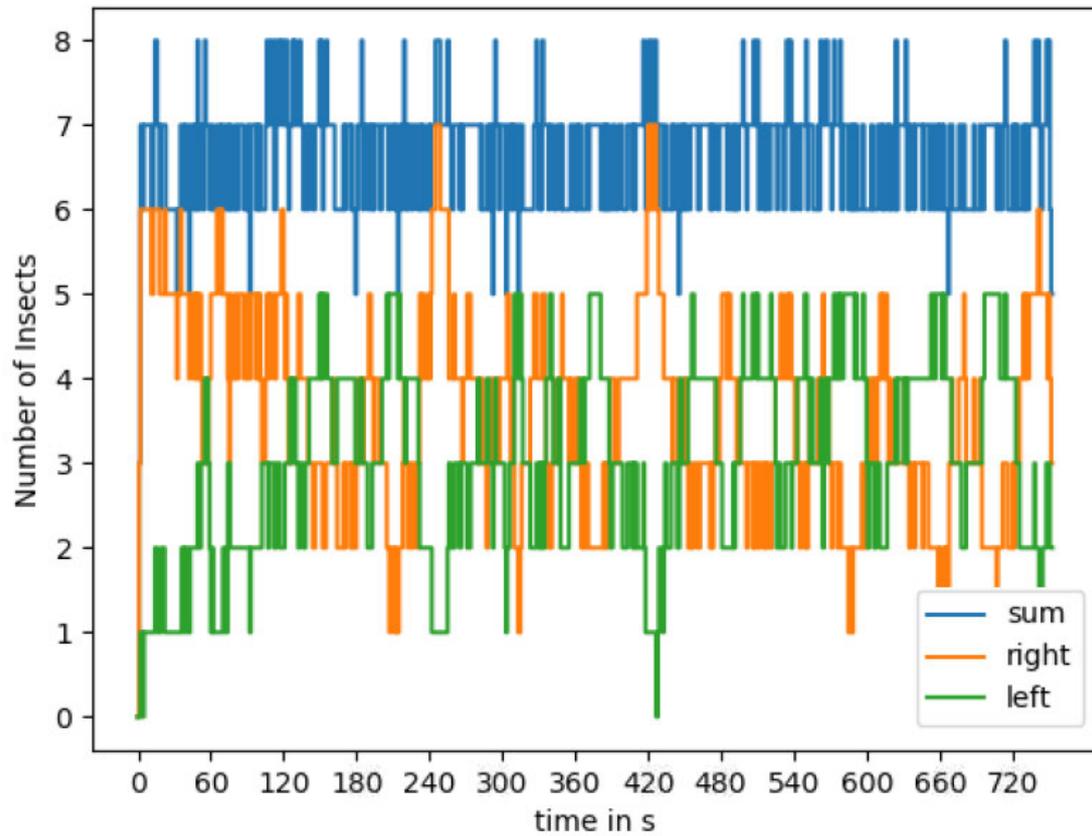


Abbildung 5.11: Zeitliche Auswertung von acht Fliegen mit LED 3000K rechts

Nach der Umpositionierung des Leuchtmittels auf die linke Seite ist ein deutlicher Trend zu dieser erkennbar (Abb. 5.12). Die Fliegen haben sich über die Aufnahmedauer hinweg, von der anfangs linken Seite, zur rechten Seite bewegt.

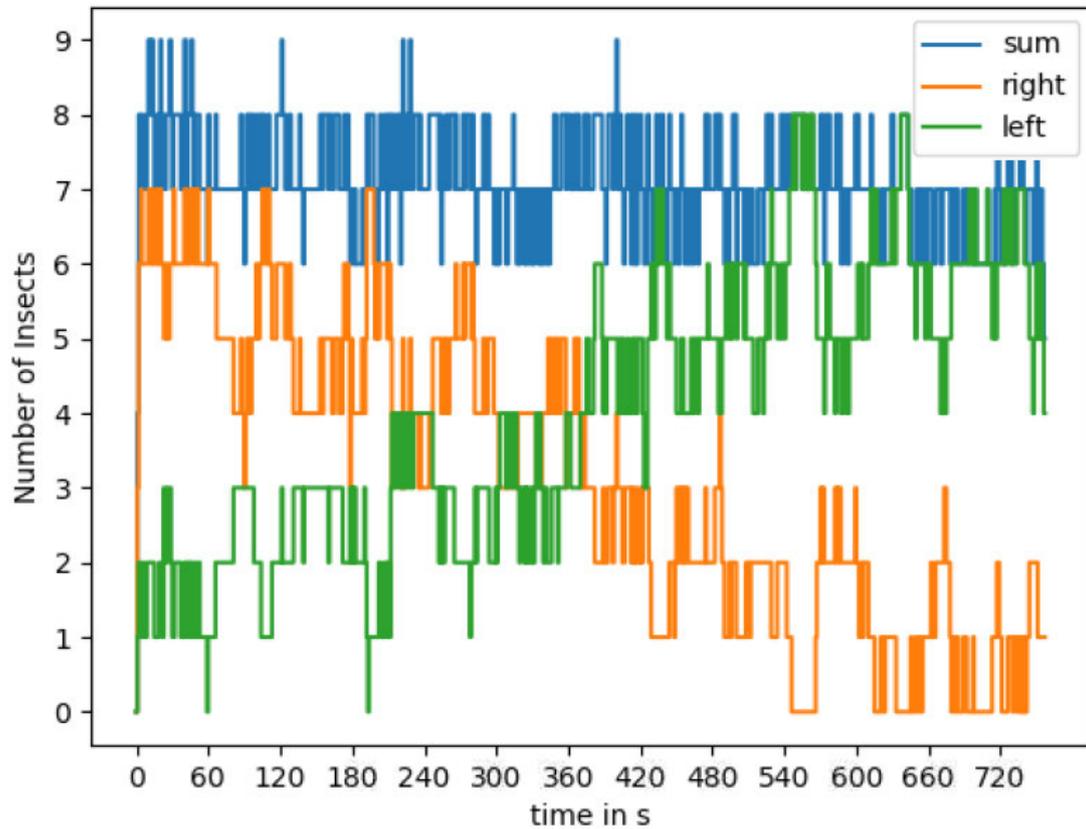


Abbildung 5.12: Zeitliche Auswertung von acht Fliegen mit LED 3000K links

Daraus ist zu schlussfolgern, dass die „Kopos-LED BULB A60“ auch ein schwaches Lockverhalten gegenüber den Insekten aufweist, wie besonders die Abbildung 5.12 zeigt

5.3.3 Grundlagentests Leuchtmittel LED 2000K

Um die Testreihe der LED's abzuschließen werden die Tests noch mit einer „Müller licht Retro LED“ durchgeführt. Die Aufnahmen haben nach der FL6500 stattgefunden. Aus dem Grund befinden sich die Insekten zu Beginn vermehrt auf der rechten Seite. Auch im späteren Verlauf ist kein relevanter Positionswechsel zu erkennen. Welches für kein starkes Lockverhalten spricht.

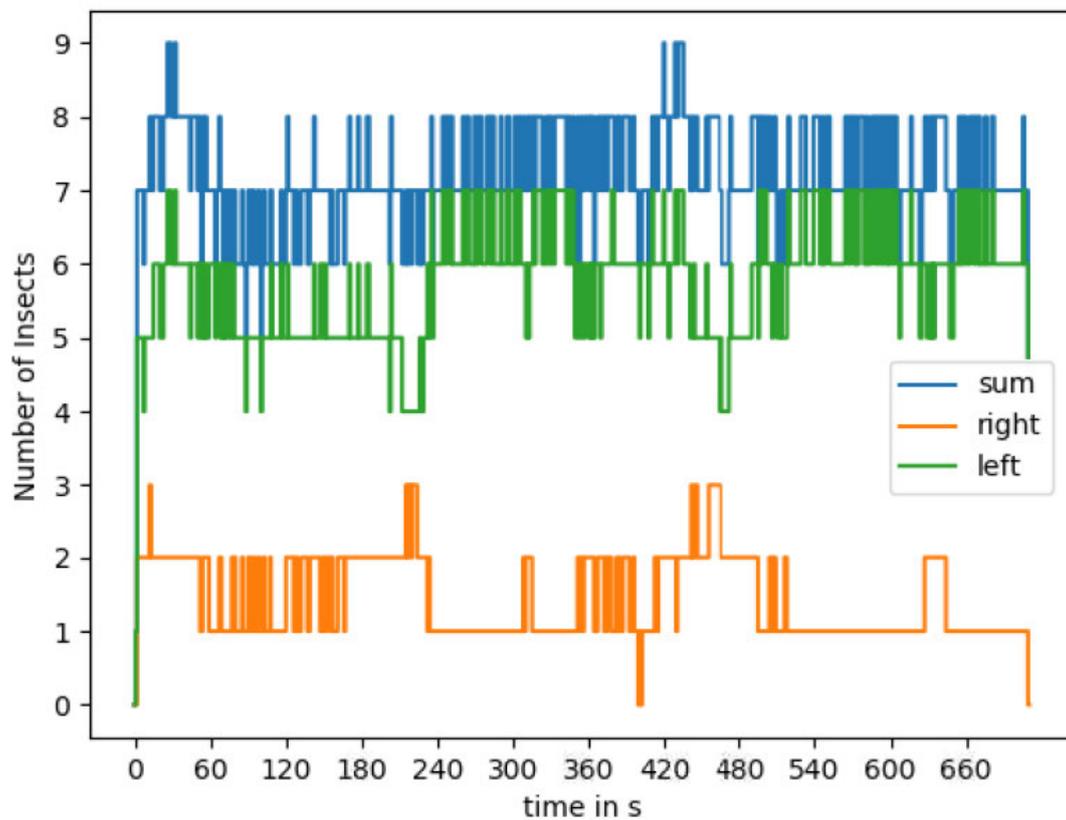


Abbildung 5.13: Zeitliche Auswertung von acht Fliegen mit LED 2000K links

Nach der Umpositionierung des Leuchtmittels auf die rechte Seite verstreuen sich die Fliegen in dem Testbereich. Es befinden sich dennoch überwiegend mehr Fliegen auf der linken Seite. Hier kam es bis zur Sekunde 450 zu Erkennungsschwierigkeiten, wodurch das Ergebnis nicht korrekt ist.

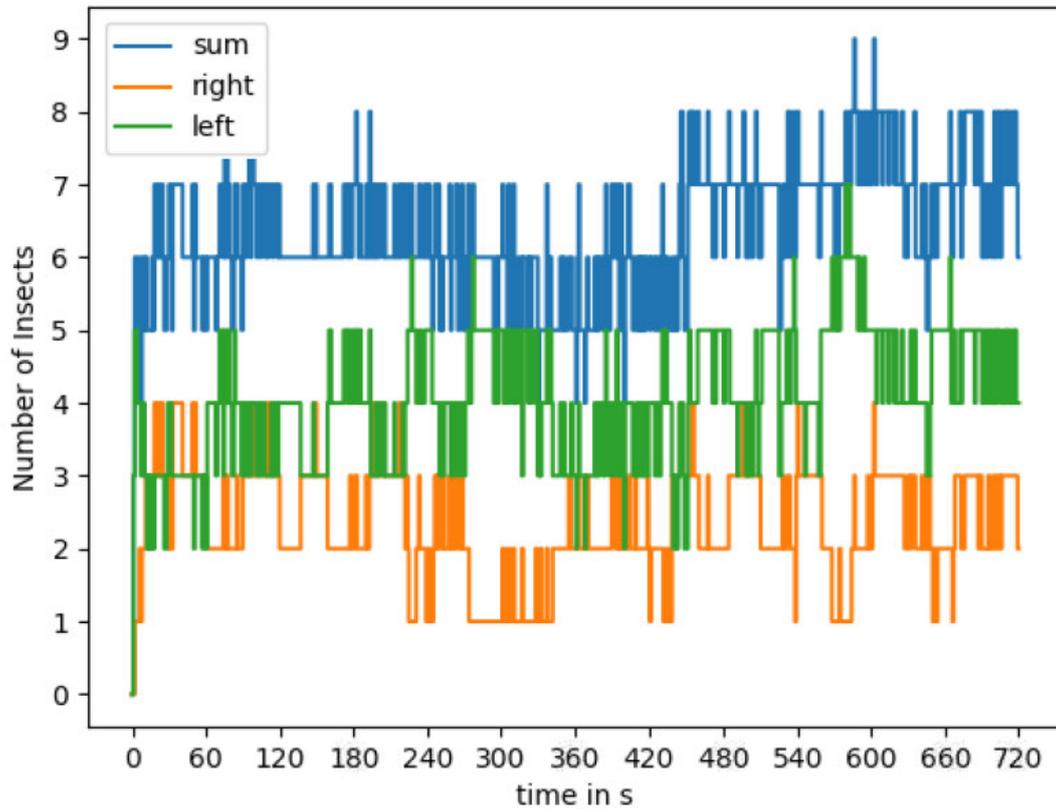


Abbildung 5.14: Zeitliche Auswertung von acht Fliegen LED 2000K rechts

Anhand der Auswertungen kann ein leichtes Lockverhalten interpretiert werden, da sich die Fliegen bei der zweiten Messung teilweise in den rechten Bereich bewegt haben. Im Vergleich zu ersten Auswertung ist mehr Bewegung zu erkennen.

5.3.4 Grundlagentests Leuchtmittel Energiesparlampe 6500K

Um einen aussagekräftigen Vergleich zwischen LED und Energiesparlampen zu erhalten, wird jetzt noch das Lockverhalten der „Philips cool daylight“ analysiert. Besonders auffällig ist hier, dass von Beginn an fast keine Fliege auf der rechten Seite befindet. Dies spricht für ein hohes Lockverhalten. Außerdem sind die Fliegen häufig dicht nebeneinander wodurch die Summenlinie geringer ist, was für eine dichte Bündelung spricht.

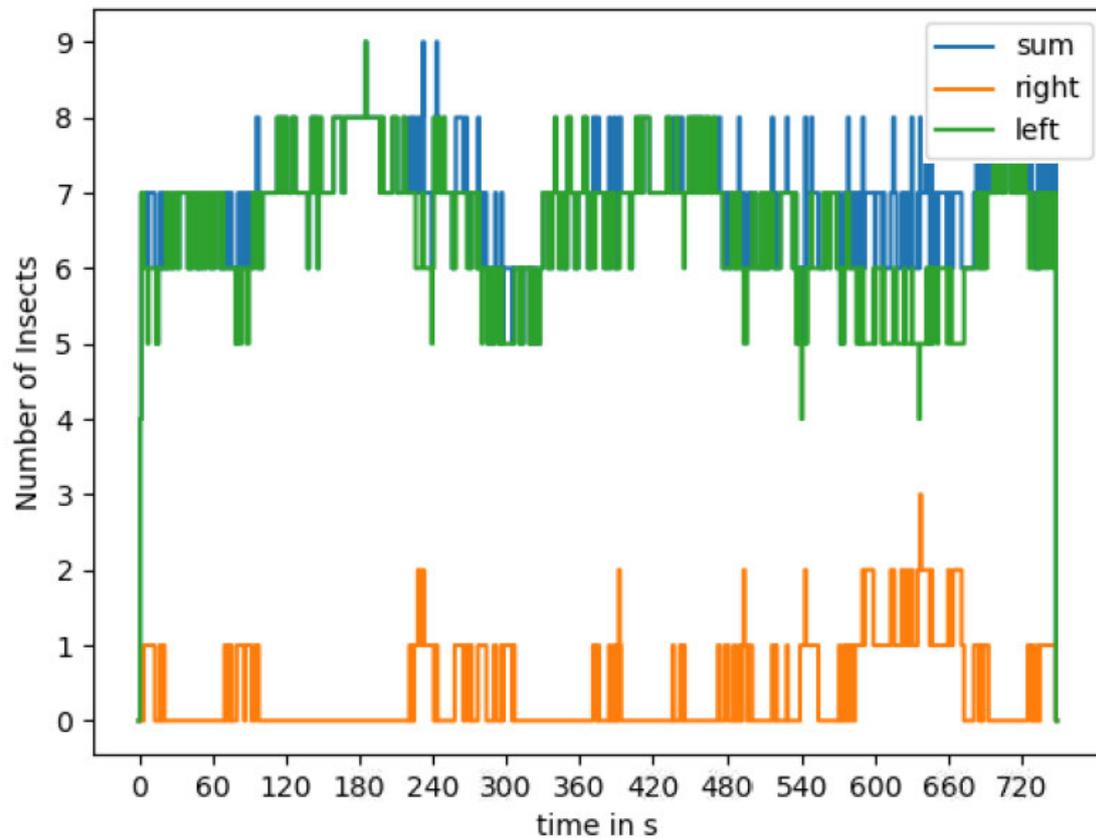


Abbildung 5.15: Zeitliche Auswertung von acht Fliegen mit FL 6500K links

Nach der Umpositionierung der Lampe auf die rechte Seite ist kein direkter Wechsel der Fliegen zu erkennen. (Abb. 5.16) Das dritte Auswertungsverfahren (Abb. 5.17) zeigt auch jetzt noch eine Tendenz zur rechten Seite.

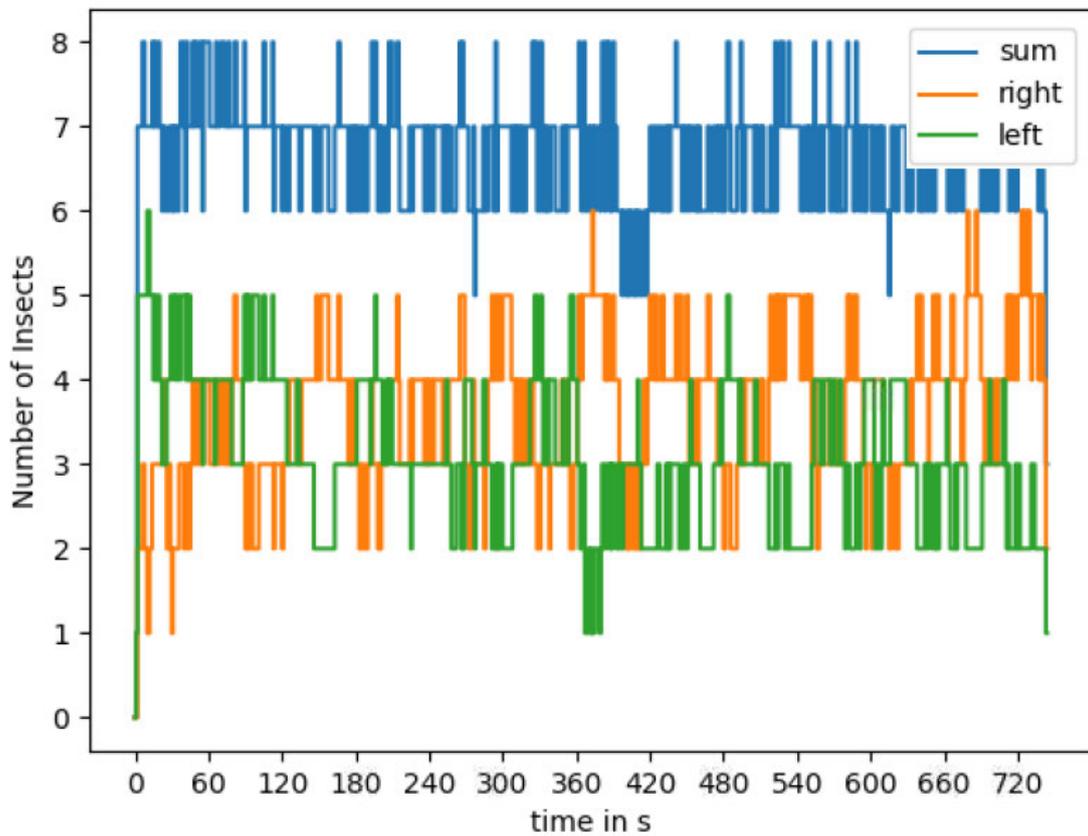


Abbildung 5.16: Zeitliche Auswertung von acht Fliegen mit FL 6500K rechts

Hier kann kein eindeutiges Ergebnis festgestellt werden. Anfangs sind die Fliegen eindeutig auf der Seite des Leuchtmittels. Jedoch nach dem Seitenwechsel ist kein eindeutiges Verhalten erkennbar.

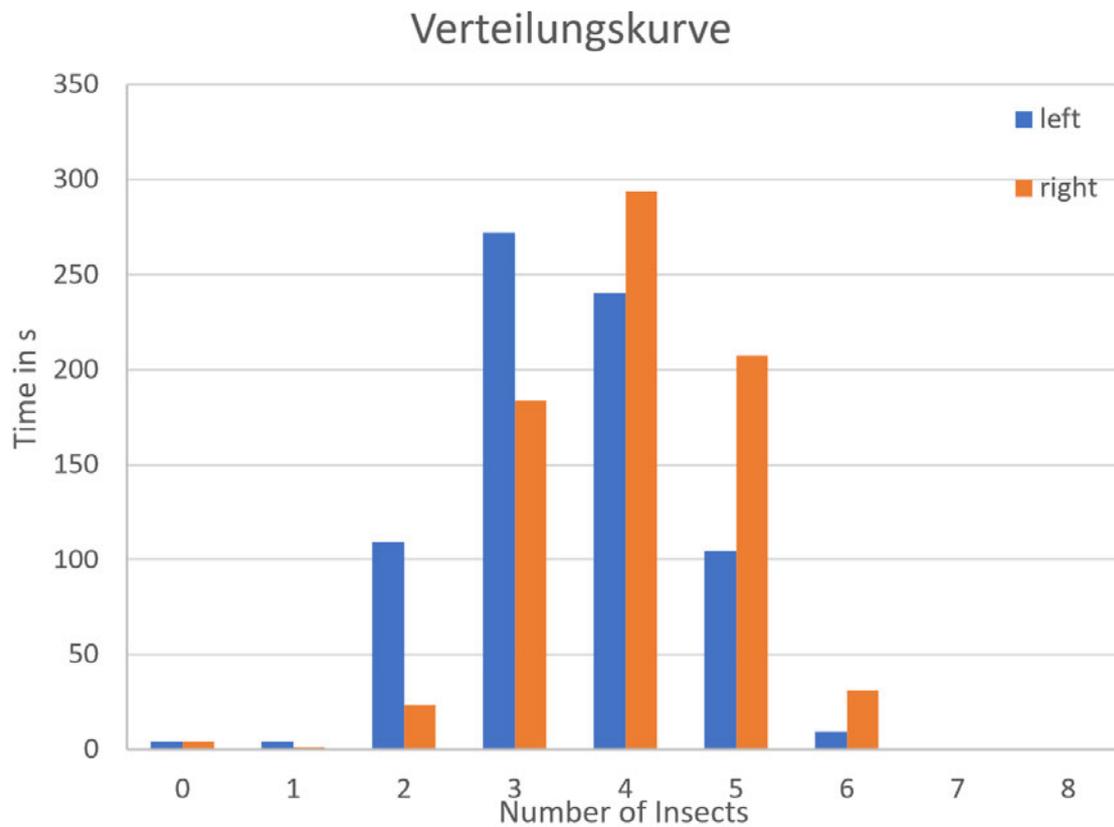


Abbildung 5.17: Verteilungsstatistik von acht Fliegen mit FL 6500K rechts

5.3.5 Grundlagentests Leuchtmittel Energiesparlampe 2700K

Bei der Energiesparlampe „Philips warm white“ ist ein stärkeres Lockverhalten zu erkennen, als bei der FL 6500K Lampe. Dies könnte an dem weit aus höherem Lichtstrom liegen, wodurch das Licht intensiver ist. In der Auswertung (Abb. 5.19), in dem sich das Leuchtmittel im rechten Bereich befindet, bewegen sich fünf bis sechs Fliegen nach ungefähr drei Minuten in den rechten Bereich und bleiben dort.

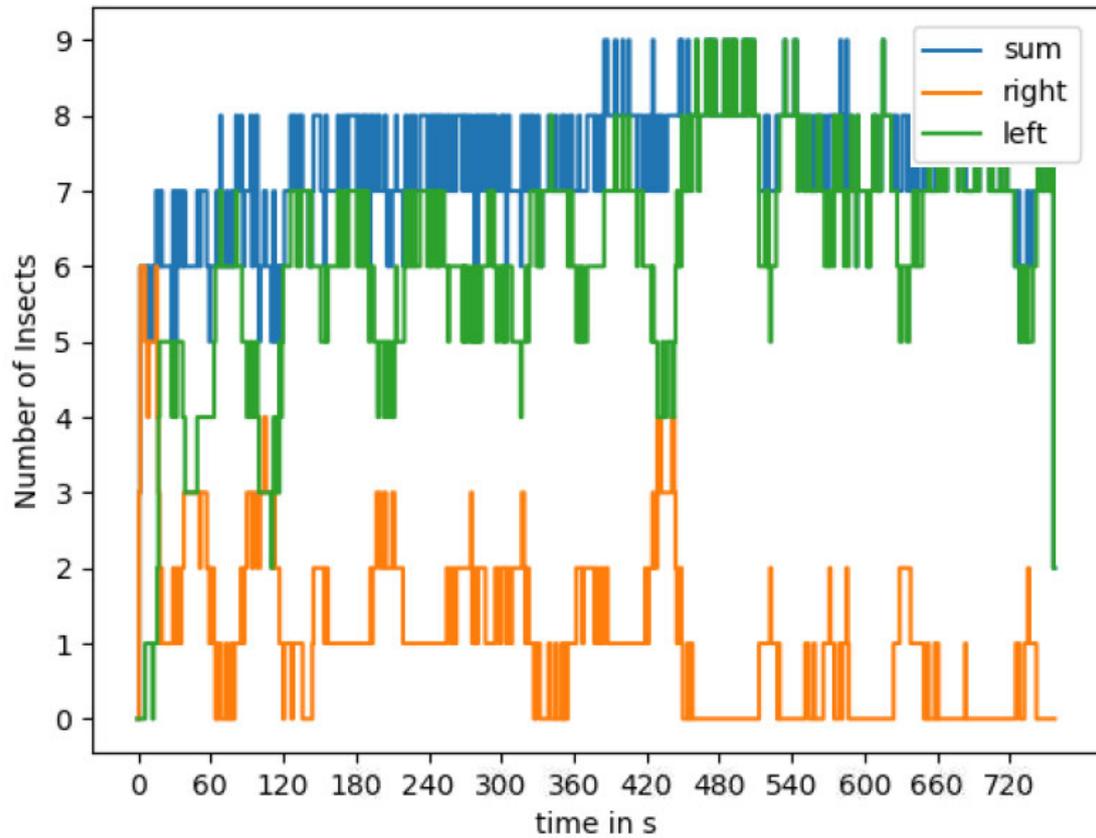


Abbildung 5.18: Zeitliche Auswertung von acht Fliegen mit FL 2700K links

Das Verhalten ist nach der Umpositionierung ähnlich wie zuvor. Nach ungefähr zwei Minuten befinden sich die meisten Fliegen auf der linken Seite. Später sind die Fliegen auf der linken Seite dicht gebündelt, wodurch es bei der Zählung zu Fehlern kommt.

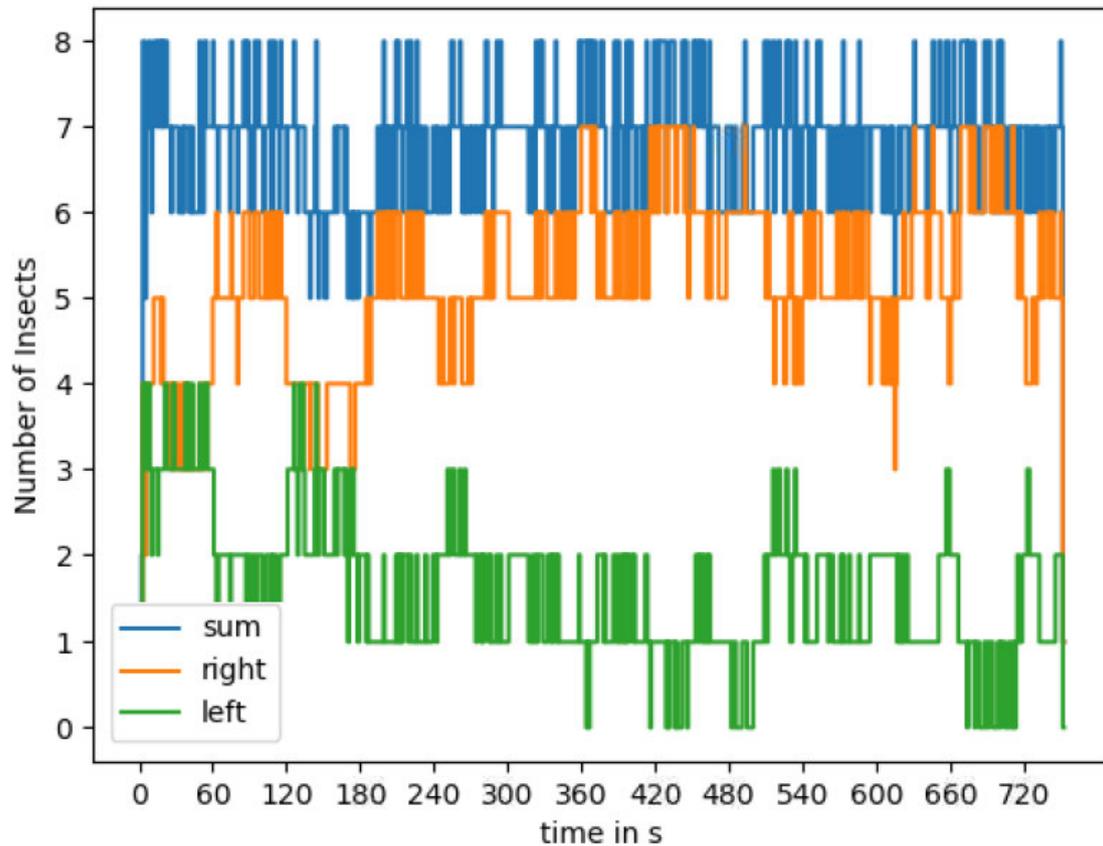


Abbildung 5.19: Zeitliche Auswertung von acht Fliegen mit FL 2700K rechts

5.3.6 Zusammenfassung

Zusammengefasst kann behaupten werden, dass sich Fliegen von Licht angezogen fühlen, auch wenn es abhängig vom Leuchtmittel und Lichtstrom unterschiedlich stark ist. Energiesparlampen haben ein hohes Lockverhalten. Der Lichtstrom ist dabei stark zu berücksichtigen. Bei den LED-Lampen hingegen wirkt die „Philips LED cool daylight“ mit der Lichttemperatur 6500K als sehr anziehend und die anderen LED-Lampen weisen keine eindeutigen Ergebnisse auf.

Tabelle 5.1: Ergebnisse der Grundlagentests

Leuchtmittel	Ergebnis	Signifikanz
Philips LED cool daylight	starkes Lockverhalten	erst eindeutig, nach Umpositionierung erst später
Kopos-LED BULB A60	mäßiges Lockverhalten	leichte Tendenz erkennbar
Müller licht Retro LED	geringes Lockverhalten	wenig Reaktion nach Umpositionierung
Philips cool daylight	mäßiges Lockverhalten	anfangs sehr stark, später nur eine Tendenz
Philips warm white	starkes Lockverhalten	eindeutiger Verlauf in Richtung der Lampe

5.4 Vergleichstests

Im Folgenden wird der Prototyp dafür verwendet, das Verhalten von Fliegen unter Anwendung von zwei unterschiedlichen Leuchtmitteln zu untersuchen. Dafür werden LED-Lampen und Energiesparlampen bei unterschiedlichen Lichttemperaturen untereinander und auch gegeneinander untersucht.

5.4.1 Vergleichstests mit Fliegen LED 6500k und FL 6500K

Die Vergleichstests dienen dazu, einen direkten Vergleich zwischen ausgewählten Leuchtmitteln aufzuzeigen. Hierbei werden die Lichtarten untersucht, ob zum Beispiel das Licht einer LED-Lampe anziehender wirkt als das einer Energiesparlampe oder andersrum. Bei der ersten Aufnahme befindet sich die LED 6500K im linken Bereich und die Energiesparlampe 6500K rechts. Die LED hat einen weitaus höheren Lichtstrom, wie unter Punkt vier beschrieben wurde.

In der Abbildung 5.20 ist zu erkennen, dass sich die Fliegen hauptsächlich auf der linken Seite befinden. Jedoch wechseln sie ab der Hälfte öfter die Seiten.

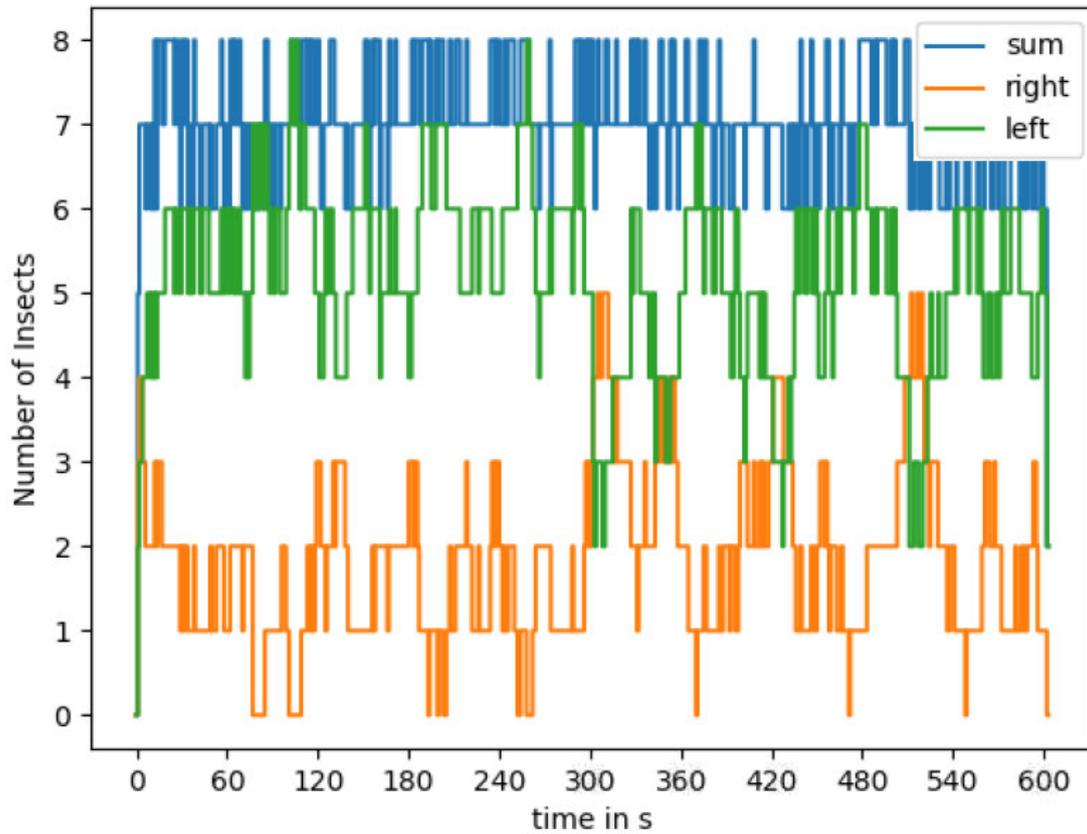


Abbildung 5.20: Zeitliche Auswertung von acht Fliegen mit links LED 6500K, rechts FL6500K

Anschließend werden die Positionen der Leuchtmittel getauscht. Hierbei ist in der Abbildung 5.21 keine eindeutige Seite festzustellen. Auch die Auswertung in Abbildung 5.22 zeigt kein direktes Ergebnis. Es entstehen in der Auswertung kurze Zeitabschnitte, in denen sich fünf bis sieben Fliegen in dem linken Bereich befinden, das ist jedoch nur ungefähr eine Minute.

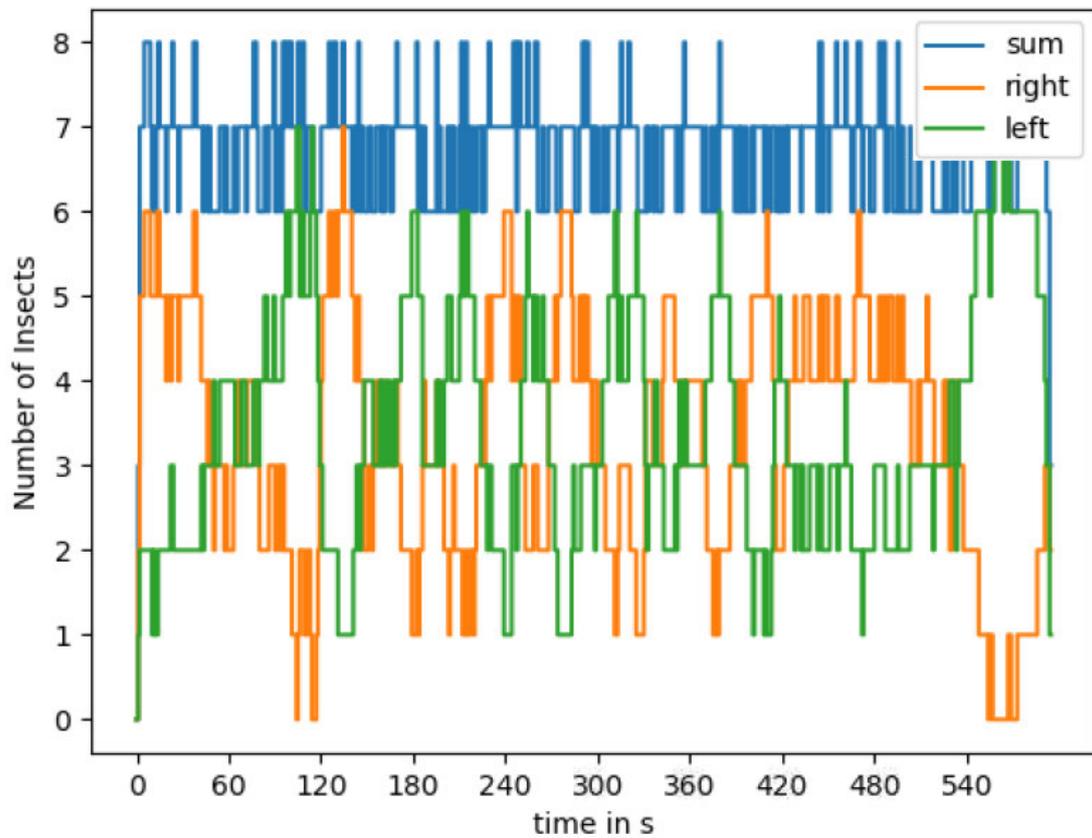


Abbildung 5.21: Zeitliche Auswertung von acht Fliegen mit rechts LED 6500K, links FL 6500K

Zusammengefasst weist die „Philips LED cool daylight“ ein höheres Lockverhalten auf als die „Philips cool daylight“. Was daran liegen könnte, dass der Lichtstrom der LED-Lampe um das Siebenfache höher ist als das der Energiesparlampe. Dadurch nimmt auch die Bestrahlungsstärke zu. Das Ergebnis stimmt mit den Erwartungen aus dem Kapitel 4 überein.

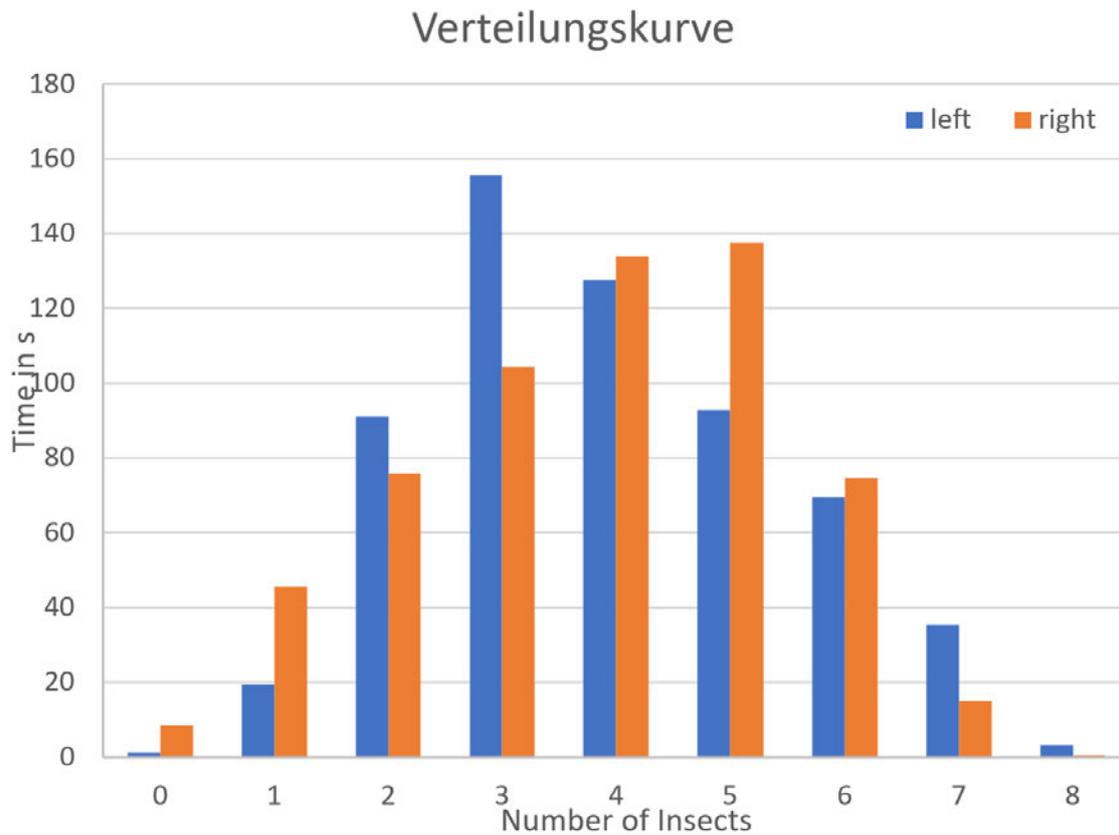


Abbildung 5.22: 3. Auswertungsverfahren rechts LED 6500K, links FL 6500K

5.4.2 Vergleichstests FL 2700K und FL 6500K

Des Weiteren wird das Lockverhalten von unterschiedlichen Lichttemperaturen auf Fliegen untersucht. Hierfür wird bei der ersten Messung unter den rechten Bereich die „Philips warm white“ mit 2700K und unter den linken Bereich die „Philips cool daylight“ mit 6500K platziert. In beiden Auswertungen (Abb. 5.23, 5.24) ist eine eindeutige Bewegung zur FL 6500K zu erkennen. Nur kurzzeitig sind die Bereiche ausgeglichen.

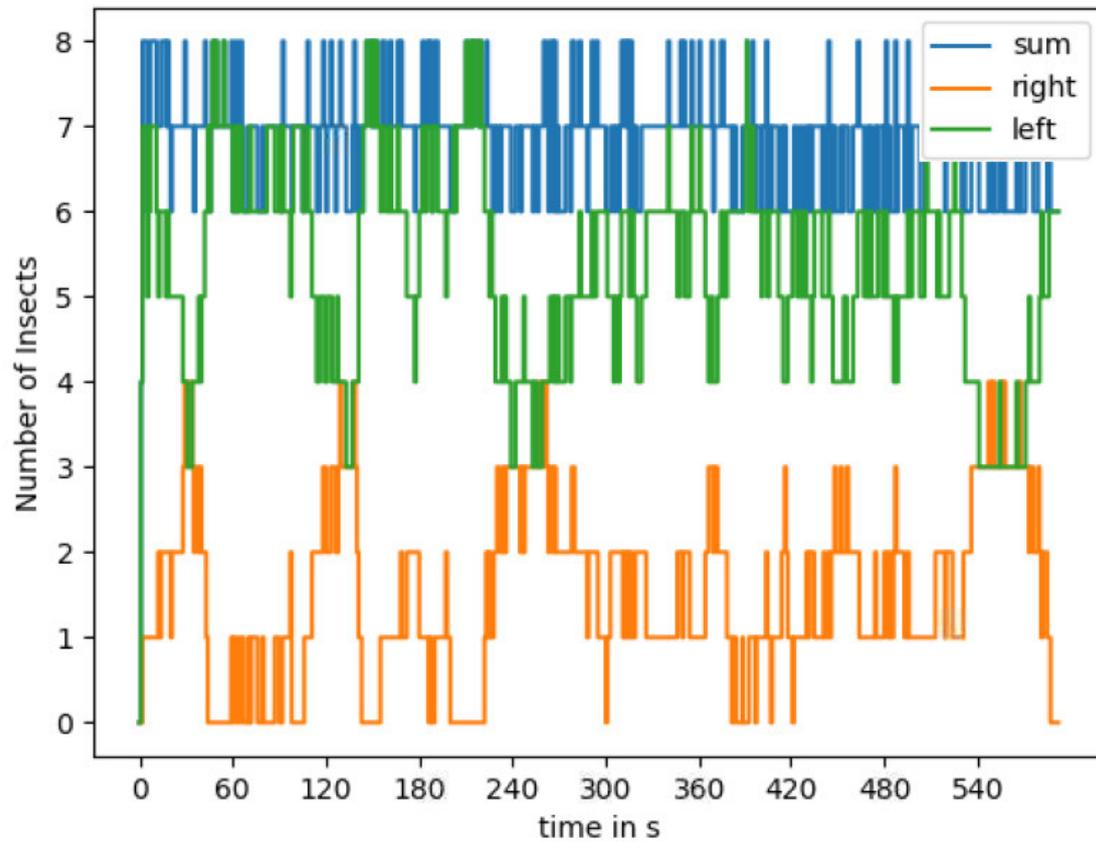


Abbildung 5.23: Zeitliche Auswertung von acht Fliegen mit rechts FL 2700K, links FL 6500K

Nach dem Tauschen der Positionen ist in Abbildung 5.26 eine leichte Tendenz zur rechten Seite erkennbar. In Abbildung 5.25 hingegen ist ein ständiger Wechsel aufzuzeigen. Hier ist keine eindeutiges Ergebnis zu erkennen.

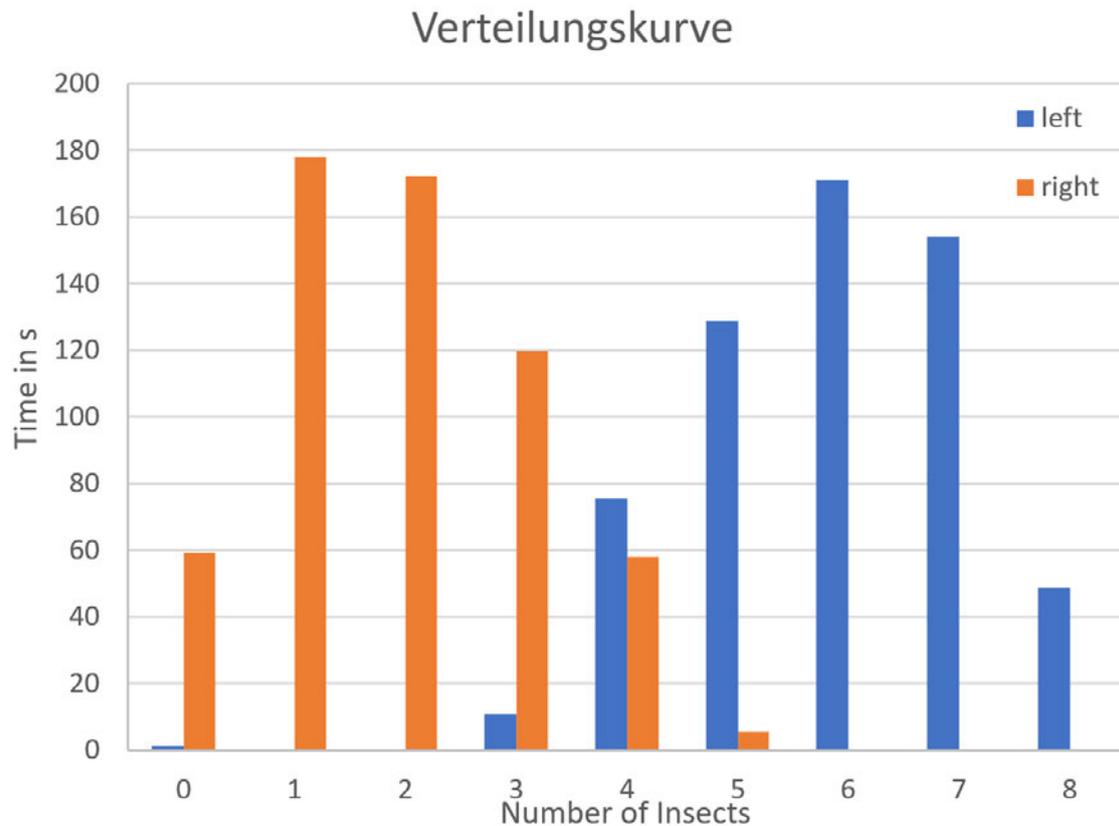


Abbildung 5.24: 3. Auswertungsverfahren mit rechts FL 2700K, links FL 6500K

Bei diesem Test ist keine eindeutige Aussage zu treffen. In der ersten Auswertungen weist die FL 6500K ein leicht höheres Lockverhalten auf. Bei der zweiten Auswertung wird keine Seite den Fliegen vorgezogen. Ausschlag geben hierfür ist wie in den Tabellen 4.4 und 4.5 aufgezeigt die Maxima, der Beleuchtungsstärke. Wie dort zu erkennen und beschreiben ist, sind die Werte annähernd gleich und es kann keine Aussage zu Lockverhalten getroffen werden.

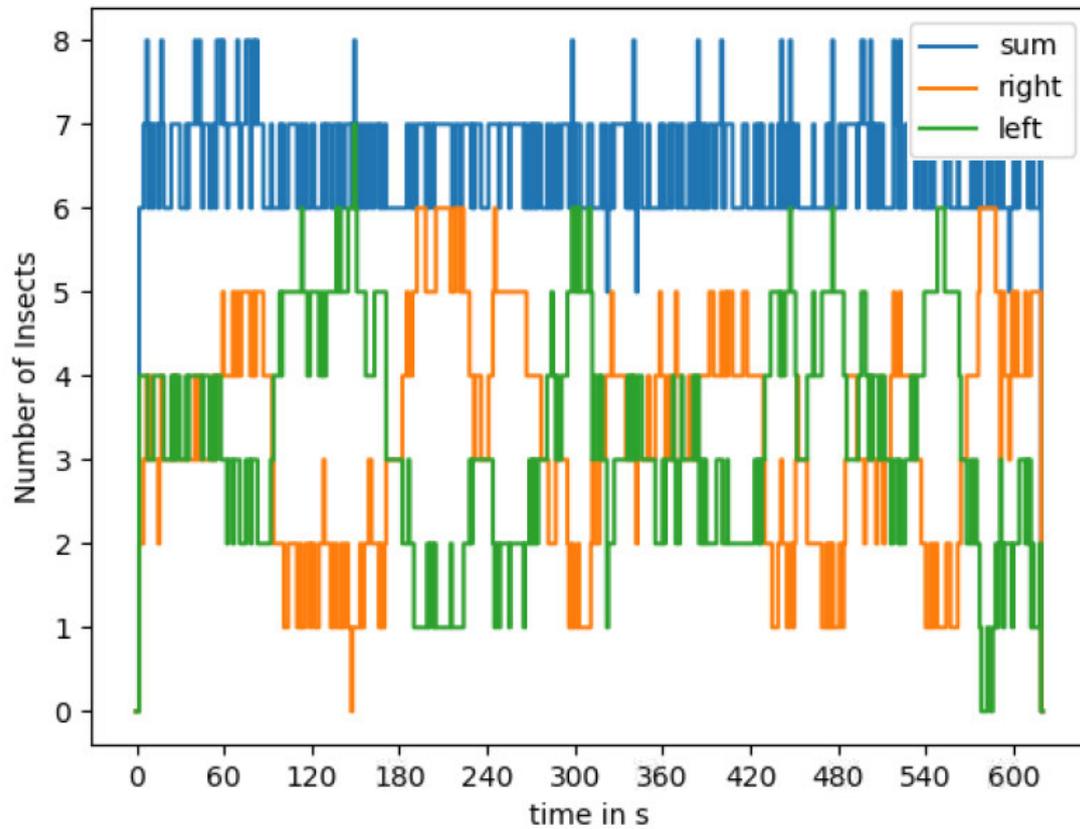


Abbildung 5.25: Zeitliche Auswertung von acht Fliegen mit links FL 2700K, rechts FL 6500K

5.4.3 Vergleichstests LED 2000K und LED 6500K

Abschließend werden die Lichttemperaturen der LED-Lampen untersucht. Hierfür werden die „Müller licht Retro LED“ und die „Philips LED cool daylight“ verglichen um den größtmöglichen Unterschied festzustellen. Anhand der Abbildungen 5.27 ist eindeutig zu erkennen, dass sich die Fliegen über der rechten, also LED 6500K befinden. Die LED 2000K zeigt keine Wirkung auf die Fliegen und ist neben der LED 6500K kaum zu erkennen (Abb. 5.29).

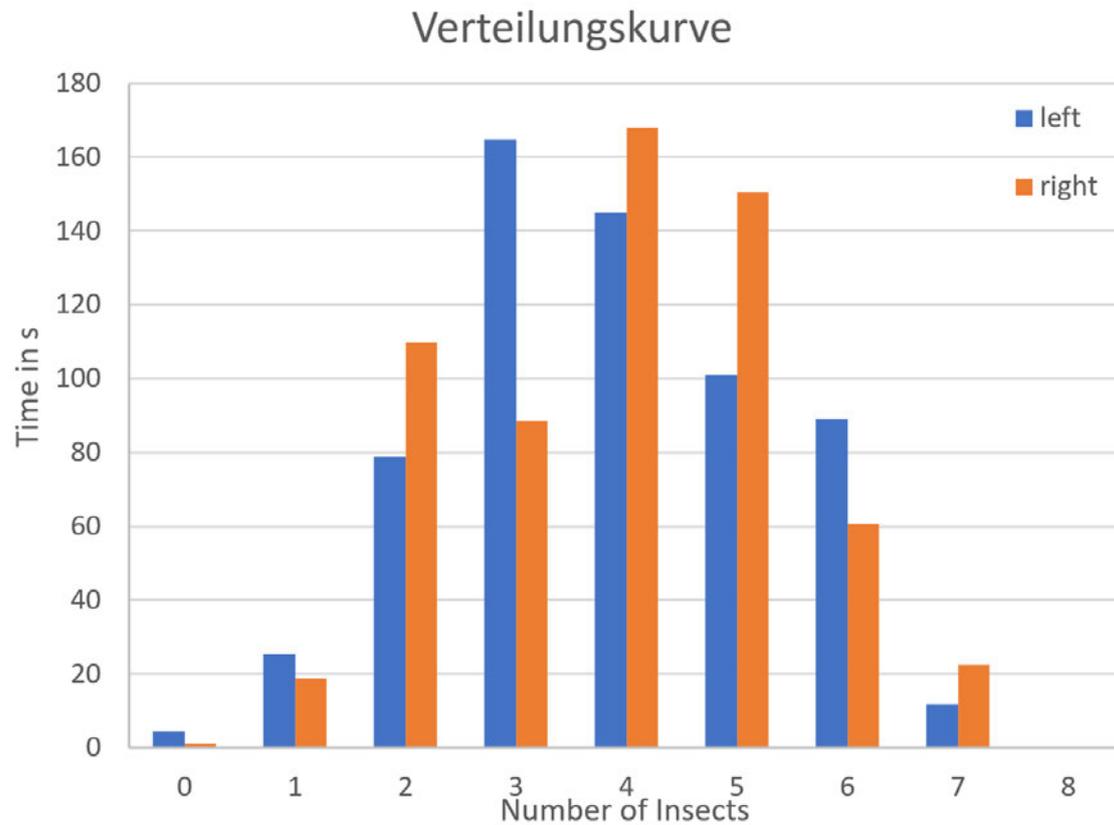


Abbildung 5.26: Verteilungsstatistik von acht Fliegen mit links FL 2700K, rechts FL 6500K

Auch nach dem Tausch der Leuchtmittel bewegen sich die Mehrzahl der Fliegen mit dem Licht der LED 6500K mit und bleiben dort bis zum Ende der Aufnahme. (Abb. 5.28)
Auch hier zeigt die 2000K LED keine Wirkung.

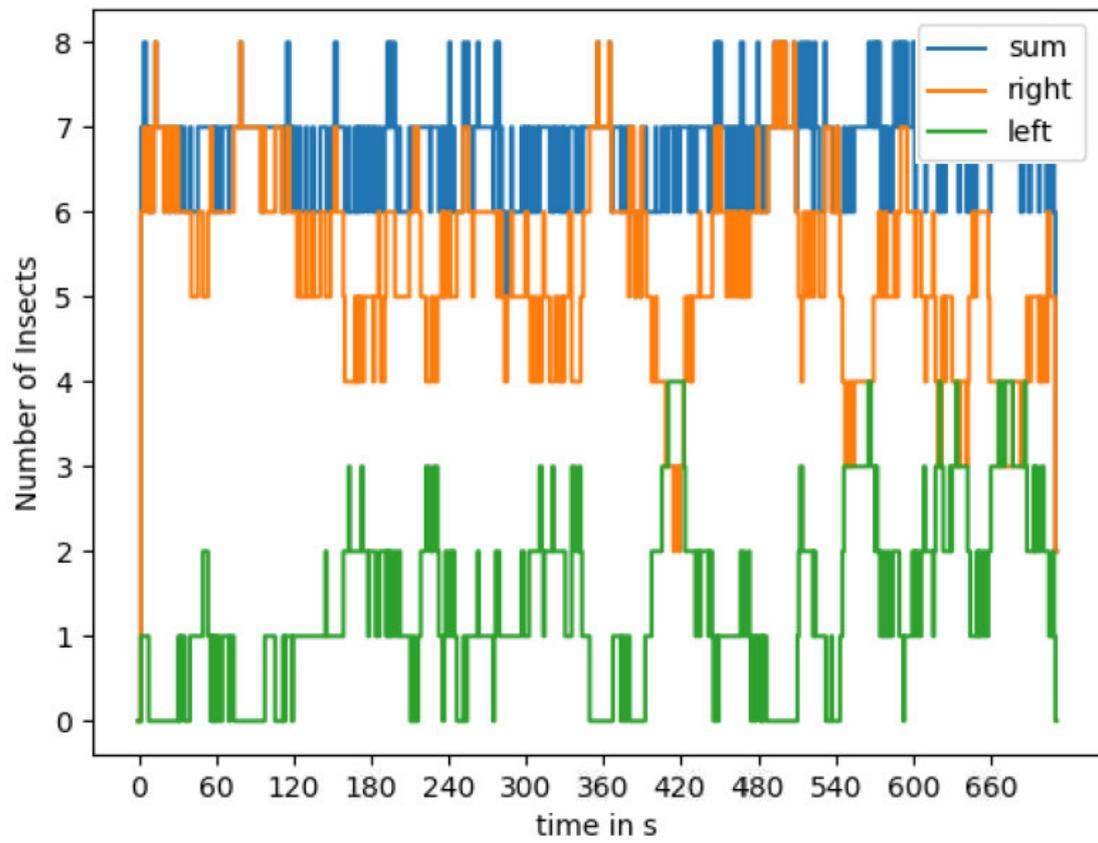


Abbildung 5.27: Zeitliche Auswertung von acht Fliegen mit links LED 2000K, rechts LED 6500K

Hier ist das Lockverhalten der Fliegen sehr gut erkennbar und nachvollziehbar. Wie schon im 4. Kapitel vermutet ist das Lockverhalten der LED 6500K entscheidend und die LED 2000K zeigt keine Wirkung.

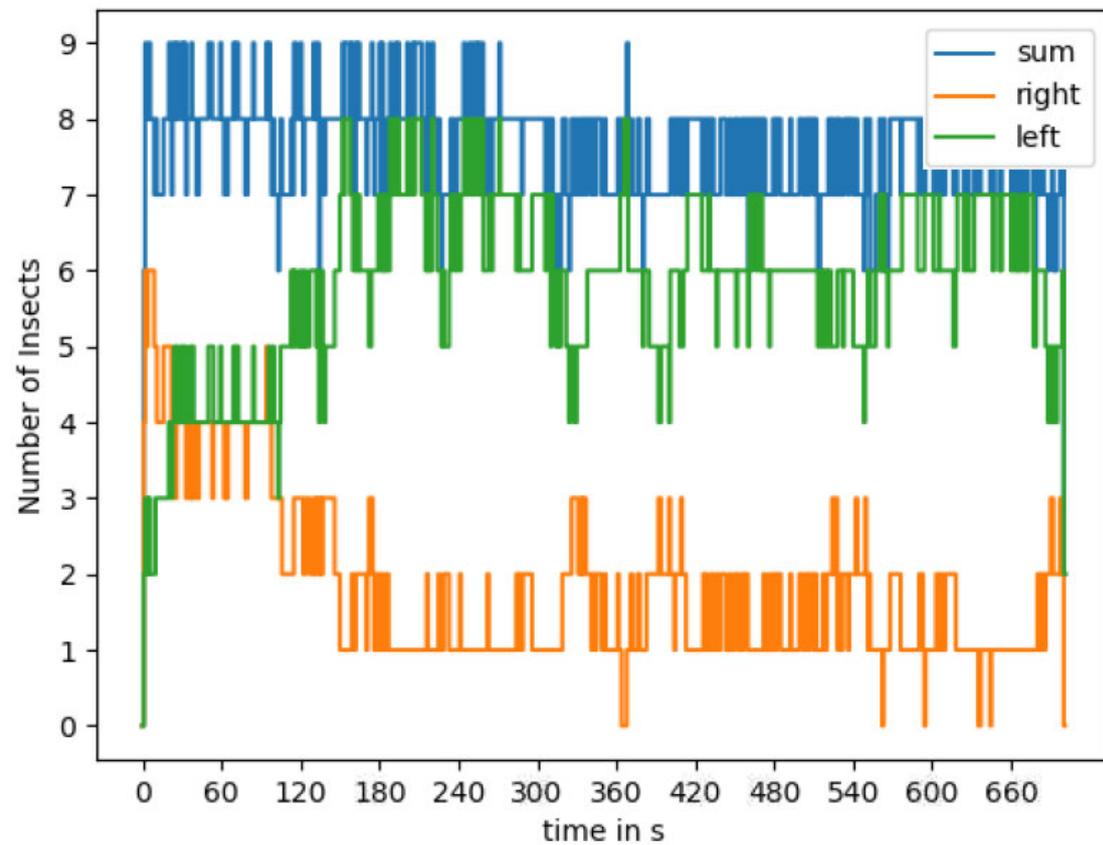


Abbildung 5.28: Zeitliche Auswertung von acht Fliegen mit rechts LED 2000K, links LED 6500K

5.4.4 Zusammenfassung

Die Ergebnisse der Experimente sind in der Tabelle 5.2 zusammengefasst.



Abbildung 5.29: Aufnahme beim Vergleich zwischen LED 2000 K und LED 6500K

Tabelle 5.2: Ergebnisse der Vergleichstests

Nr.	Lampe 1	Lampe 2	Ergebnis	Signifikanz
1	LED 6500k	FL 6500K	LED 6500K leicht höheres Lockverhalten	häufiger Seitenwechsel, jedoch leichte Tendenz erkennbar
2	FL 2700K	FL 6500K	kein eindeutiges Ergebnis	häufiger Seitenwechsel der Fliegen
3	2000K	LED 6500K	LED 6500 höheres Lockverhalten	klare Bewegung in Richtung der LED 6500K

6 Fazit

In dieser Arbeit wird unter Anwendung eines Prototyps einer vorangegangenen Bachelorarbeit, das Verhalten von Insekten unter Beeinflussung von verschiedenen Lichtquellen aufgenommen, dokumentiert und ausgewertet. Damit die Frage beantwortet werden kann, ob dieser Prototyp für solche Auswertungen geeignet ist und sich Insekten von Licht beeinflussen lassen?

Damit eine Vielzahl von Aufnahmen generiert werden kann, ist ein möglichst robuster Versuchsaufbau nötig. Aufgrund dessen wurde ein Aufbau aus Holz generiert. Die Aufnahmen werden über ein Raspberry Pi mit einer PI-Kamera aufgenommen. Auch die Auswertung wird auf diesem durchgeführt. Wegen unterschiedlicher Versionen der Software kann es bei der Ausführung des Programms zu Problemen führen.

Für die Tests wurden ursprünglich Stechmücken vorgesehen, da diese nachtaktiv und relativ leicht zu beschaffen sind. Auf Grund ihrer Fähigkeiten blaue bis hin zu ultraviolettem Licht wahrzunehmen, können nachtaktive Insekten in dem Mond- und Sternenlicht sehen. Diese geraten in eine Art Starre sobald die verwendeten Leuchtmittel eingeschaltet werden. Das zeigt auch eine 14 Stunden Aufnahme. Außerdem sind die Körper zu klein um von der PI-Kamera ausreichend groß für die Objekterkennung dargestellt zu werden. Auch der Versuch, Erfolge durch eine Handycamera zu erzielen, gelang nicht, da Unreinheiten des Versuchsaufbaus auch als Objekt erkannt werden. Aus den Gründen sind größere Insekten notwendig, die sich auch bei der Beleuchtung bewegen. Die Wahl viel hier auf Stubenfliegen. Diese sind aufgrund ihres rundlichen Körpers leichter zu erkennen und haben sich auch während der Aufnahmen bewegt, des weiteren ist die Beschaffung einfacher.

Die verwendeten Leuchtmittel weisen alle unterschiedliche Eigenschaften auf. Sie bestehen aus Energiesparlampen und LED-Lampen mit unterschiedlichen Lichttemperaturen. Die Auswahl ist jedoch verbesserungswürdig, denn die Energiesparlampe 6500K und die

LED 2000K haben ein weitaus geringeren Lichtstrom, als die anderen Leuchtmittel. Wodurch das Licht bei direkten Vergleichen nicht effektiv zur Geltung kommt.

Die Objekterkennung ist nicht voll ausgereift. Es entstehen häufig Schwankungen der erkannten Objekte, wodurch es auch zu Schwankungen der Auswertungen in den Diagrammen kommt. Dadurch ist nur eine Tendenz des Verhaltens erkennbar, jedoch selten die richtige Anzahl der Objekte pro Feld. Aus dem Grund und weil Auswertungen von längeren Videos schwer auf einem Diagramm erkennbar sind, ist die Auswertung auch in einem Balkendiagramm dargestellt. Indem die Anzahl der Objekte pro Zeit aufgeführt sind. Dennoch sind Tests und Auswertung möglich. Auch bei nicht exakten Ergebnissen

Wie sich bei den Tests gezeigt hat, stimmen die Vermutungen weitestgehend mit den Testergebnissen überein. Die Leuchtmittel, bei denen die kürzeren Wellenlängen intensiver sind, also eher blaues Licht aufweisen, locken die Fliegen mehr an. Hingegen längere Wellenlängen, also rötliches, keinen großen Einfluss auf die Fliegen zeigt. Unterschiede zwischen LED und Energiesparlampe können jedoch nicht eindeutig festgestellt werden, da der Lichtstrom des Leuchtmittels zu unterschiedlich ist, um diese genau zu vergleichen.

Zusammengefasst ist es möglich, mit dem aktuellen Prototypen und der damit verbundenen Bildererkennung automatisch und zeitaufgelöste Auswertungen von Fliegen und anderen Objekten zu erstellen. Auch unter Verwendung von unterschiedlichen Leuchtmitteln sind Unterschiede in der Auswertung festzustellen. Bei kleineren Objekten ist der Prototyp nicht ausreichend präzise. Auch das Zählen in den Bereichen funktioniert größtenteils. Damit jedoch saubere Diagramme erstellt werden können, sollte die Hardware und die Software angepasst werden.

7 Ausblick

Die in der Arbeit durchgeführten Versuche können zum Erfolg von einfachen Tests und Testobjekten führen. Sollten die Ergebnisse jedoch genauer werden oder die Anzahl der Objekte höher ist, sind weitere Bearbeitung des Prototyps notwendig.

Der Prototyp ist darauf ausgelegt, eine feste Anzahl an Testobjekten zu überwachen, daher sollte diese in der Software fest eingetragen werden können. Zudem wäre es von Vorteil, die Objekte zu markiert und gegebenenfalls zu nummeriert, um die Objekte in diesem Bereich wieder erkennen zu können.

Zur Verwendung einer besseren Hardware bietet sich eine Handykamera an. Diese ist fast immer vorhanden und bietet hochauflösende Bilder. Die damit erstellten Videos haben jedoch andere Pixeldichten und Bildwiederholungsraten, wodurch das Programm diesbezüglich angepasst werden muss. Dadurch wäre die Verwendung eines Raspberry Pi nicht mehr notwendig und das Programm könnte auf einem beliebigen Computer ausgeführt werden.

Für weiter Arbeiten solle der Prototyp diesbezüglich angepasst werden, um die Bilderkennung genauer zu machen.

Literaturverzeichnis

- [1] Bildverarbeitung als KI-Vorreiter. In: *automatica*. – URL <https://automatica-munich.com/de/newsroom/news/kuenstliche-intelligenz/bildverarbeitung-als-ki-vorreiter/>
- [2] Medizinische Bildverarbeitung (MBV). In: *Noukwe Medizintechnik*. – URL <https://www.noukwe.com/mbv/>
- [3] ALISTER, Danny M.: *Konzeption und Realisierung eines Prototyps zur automatisierten Zählung von Insekten mittels Bilderkennungssoftware*, HAW Hamburg, Dissertation, 03.2019
- [4] BARMER: Künstliche Intelligenz in der Medizin - Intelligente Bilderkennung kann helfen, auffällige Befunde in Röntgenaufnahmen zu markieren. In: *Internetredaktion BARMER*. – URL <https://www.barmer.de/gesundheitsverstehen/gesellschaft/kuenstliche-intelligenz-in-der-medizin-107156#:~:text=KünstlicheIntelligenzinderMedizin-IntelligenteBilderkennungkannhelfen,auffällige,BefundeinRöntgenaufnahmenzumarkieren&text=Software,dielernt,entwickeltsich,künstlicherIntelligenzinderPraxis.>
- [5] DEMUTH, Bernd: Klima- und Naturschutz: Hand in Hand. In: *Bundesamt für Naturschutz* (Juni 2018)
- [6] FOLKERS, Andreas: *Grundlagen des Deep Learning*. S. 3–13. In: *Steuerung eines autonomen Fahrzeugs durch Deep Reinforcement Learning*. Wiesbaden : Springer Fachmedien Wiesbaden, 2019. – URL https://doi.org/10.1007/978-3-658-28886-0_2. – ISBN 978-3-658-28886-0
- [7] HEINZE, Dipl.-Inform. N.: ABUL - Automatisierte Bildauswertung für Unbemannte Luftfahrzeuge. In: *Fraunhofer ISOB*. – URL <https://www.iosb.fraunhofer.de/servlet/is/59702/>

- [8] KIT: Automatische Sichtprüfung und Bildverarbeitung. In: *Institut für Industrielle Informationstechnik*. – URL <https://www.iiit.kit.edu/bildverarbeitung.php>
- [9] MARCUS HEBEL, Dr. rer. nat.: Änderungsdetektion in urbanen Gebieten durch objektbasierte Analyse und schritthaltenden Vergleich von Multi-Aspekt ALS-Daten. In: *Fraunhofer ISOB*. – URL <https://www.iosb.fraunhofer.de/servlet/is/23029/>
- [10] MEHLHORN, Heinz ; MEHLHORN, Birgit: *Insekten*. S. 87–163. In: *Zecken, Milben, Fliegen, Schaben ...: Schach dem Ungeziefer*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2020. – URL https://doi.org/10.1007/978-3-662-61542-3_6. – ISBN 978-3-662-61542-3
- [11] MESSER, Wolfgang: Im Test: „Candlelight“-LED-Lampe von LEDON – dezentes Stimmungslicht. (2020). – URL <http://fastvoice.net/2014/12/13/im-test-candlelight-led-lampe-von-ledon-dezentes-stimmungslicht/>
- [12] QUACK, Alice: *Maschinelles Lernen: So geht industrielle Bildverarbeitung heute*. 22.01.2020. – URL <https://www.vdi.de/news/detail/maschinelles-lernen-so-geht-industrielle-bildverarbeitung-heute>
- [13] SONNTAG, Daniel: Künstliche Intelligenz in der Medizin – Holzweg oder Heilversprechen? In: *HNO* 67 (2019), May, Nr. 5, S. 343–349. – URL <https://doi.org/10.1007/s00106-019-0665-z>. – ISSN 1433-0458

A Anhang: Programmcode

```

1  import sys
2  import time
3  import cv2
4  import numpy as np
5  from operator import add
6  from matplotlib import pyplot as plt
7  from tkinter import *
8  from tkinter import filedialog
9  from tkinter import messagebox
10 from array import *
11
12 #----- declaration-----
13
14 cv2.namedWindow('image') # create window
15
16 modus=0
17 areal =1
18 filename=""
19
20 #-----function-----
21
22 def testareal():
23     global areal
24     if areal ==1:
25         ButtonAreal.configure(bg = "#01DF01")
26         areal =2
27     else:
28         ButtonAreal.configure(bg = "#BDBDBD")
29         areal =1
30
31
32 #function for changing mode after button clicked
33 def mode(value):
34     global modus
35     modus=value
36     if value ==1:
37         ButtonMode2.configure(bg = "#DF013A")
38         ButtonMode3.configure(bg = "#DF013A")
39         ButtonMode1.configure(bg = "#01DF01")
40         ButtonGetStart.configure(bg = "#DF013A")
41         ButtonGetvideofile.configure(bg = "#BDBDBD")
42         ButtonGetPicturefile.configure(bg = "#BDBDBD")
43     if value ==2:
44         ButtonMode1.configure(bg = "#DF013A")
45         ButtonMode3.configure(bg = "#DF013A")
46         ButtonMode2.configure(bg = "#01DF01")
47         ButtonGetStart.configure(bg = "#DF013A")
48         ButtonGetvideofile.configure(bg = "#BDBDBD")
49         ButtonGetPicturefile.configure(bg = "#BDBDBD")
50     if value ==3:
51         ButtonMode1.configure(bg = "#DF013A")
52         ButtonMode2.configure(bg = "#DF013A")
53         ButtonMode3.configure(bg = "#01DF01")
54         ButtonGetStart.configure(bg = "#01DF01")
55         ButtonGetvideofile.configure(bg = "#DF013A")
56         ButtonGetPicturefile.configure(bg = "#DF013A")
57
58 #function for getting videofile after button clicked
59 def getvideofile():
60     global modus
61     if modus !=2:
62         messagebox.showerror("Error", "Zunaechst den Modus Video waehlen!")
63     else:
64         global filename
65         filename = filedialog.askopenfilename(title = "Select file",filetypes =
66         (("video files","*.avi"),("all files","*.*")))
67         ButtonGetvideofile.configure(bg = "#01DF01")
68         ButtonGetPicturefile.configure(bg = "#DF013A")
69         ButtonGetStart.configure(bg = "#01DF01")
70
71 #function for getting picture file after button clicked
72 def getpicturefile():

```

```

72     global modus
73     if modus !=1:
74         messagebox.showerror("Error", "Zunaechst den Modus Einzelbild waehlen!")
75     else:
76         global filename
77         filename = filedialog.askopenfilename(title = "Select file",filetypes =
78             (("png files","*.png"),("all files","*.")))
79         ButtonGetvideofile.configure(bg = "#DF013A")
80         ButtonGetPicturefile.configure(bg = "#01DF01")
81         ButtonGetStart.configure(bg = "#01DF01")
82
83 #function for start the algorithm and close gui after button clicked
84 def start():
85     global modus
86     global filename
87
88     if ((modus ==1 or modus ==2) and filename == "") or modus == 0:
89         messagebox.showerror("Error", "Zunaechst den Modus und Datei waehlen!")
90     else:
91         gui.destroy()
92
93 #-----GUI-----
94 #initialize gui and framecontour
95 gui = Tk()
96 gui.title("Automatische Zaehlung von Insekten")
97 gui.minsize (width = 600, height =200)
98 gui.config(background = "#737373")#backgroundcolour
99 leftFrame = Frame(gui, width=200, height = 300)
100 leftFrame.grid(row=0, column=0, padx=15, pady=3)
101 leftFrame.pack ()
102
103 #simple textlabel
104 TextMode = Label(leftFrame, text="Optional: 2 Zonen Modus")
105 TextMode.grid(row=0, column=0, padx=10, pady=3)
106
107 #Button for set 2 seperate Areal
108 ButtonAreal=Button(leftFrame, text="2 Zonen Testareal", bg="#BDBDBD", width=15,
109 command= testareal)
110 ButtonAreal.grid(row=1, column=0, padx=10, pady=3)
111
112 Space = Label(leftFrame, text="") #just a empty textfield for a space
113 Space.grid(row=2, column=0, padx=10, pady=3)
114
115 TextMode = Label(leftFrame, text="1. Modus waehlen")
116 TextMode.grid(row=4, column=0, padx=10, pady=3)
117
118 #Buttons for choosing Mode
119 ButtonMode1=Button(leftFrame, text="Einzelbild", bg="#BDBDBD", width=15, command=
120 lambda *args:mode (1))
121 ButtonMode1.grid(row=5, column=0, padx=10, pady=3)
122 ButtonMode2=Button(leftFrame, text="Video", bg="#BDBDBD", width=15, command=lambda
123 *args:mode (2))
124 ButtonMode2.grid(row=6, column=0, padx=10, pady=3)
125 ButtonMode3=Button(leftFrame, text="Live-Video", bg="#BDBDBD", width=15,
126 command=lambda *args:mode (3))
127 ButtonMode3.grid(row=7, column=0, padx=10, pady=3)
128
129 Space = Label(leftFrame, text="")
130 Space.grid(row=8, column=0, padx=10, pady=3)
131
132 TextMode1 = Label(leftFrame, text="2. Datei waehlen")
133 TextMode1.grid(row=9, column=0, padx=10, pady=3)
134
135 # Buttons for getting video or picture file
136
137 ButtonGetPicturefile=Button(leftFrame, text="Einzelbild auswaehlen", bg="#BDBDBD",
138 width=15, command=getpicturefile)
139 ButtonGetPicturefile.grid(row=10, column=0, padx=10, pady=3)
140
141 ButtonGetvideofile=Button(leftFrame, text="Video auswaehlen", bg="#BDBDBD",
142 width=15, command=getvideofile)
143 ButtonGetvideofile.grid(row=11, column=0, padx=10, pady=3)

```

```

137
138 Space = Label(leftFrame, text="")
139 Space.grid(row=12, column=0, padx=10, pady=3)
140
141 TextMode2 = Label(leftFrame, text="3. Auswertung Starten")
142 TextMode2.grid(row=13, column=0, padx=10, pady=3)
143
144 # Button for start algorithm and close gui
145 ButtonGetStart=Button(leftFrame, text="Start", bg="#DF013A", width=15, command=start)
146 ButtonGetStart.grid(row=14, column=0, padx=10, pady=3)
147
148 Space = Label(leftFrame, text="")
149 Space.grid(row=15, column=0, padx=10, pady=3)
150
151
152 Hinweis = Label(leftFrame, text="Beende Algortihmus/Video mit der Taste'Q' ")
153 Hinweis.grid(row=15, column=0, padx=10, pady=3)
154
155
156 gui.mainloop()
157
158
159
160 #-----Graph paramters-----
161
162 graphcountleft=[0]
163 graphcountright=[0]
164 graphcountareal=[0]
165 graphxaxis=[0]
166 numberofinseksleft=[0]
167 numberofinseksright=[0]
168 #-----single picture-----
169 if modus == 1:
170
171     framergb = cv2.imread(filename,1) # read picture from file in colour
172     framegray = cv2.imread(filename,0)# read picture from file in gray
173
174 #-----video-----
175 if modus == 2:
176
177     cap = cv2.VideoCapture(filename) # read video from file
178
179 #-----LIVE-----
180 if modus == 3:
181
182     cap = cv2.VideoCapture(0) #Defolt index 0  ider -1
183     fourcc = cv2.VideoWriter_fourcc(*'XVID')
184     out = cv2.VideoWriter('output.avi', fourcc, 20.0, (640,480))
185
186 #-----algorithm for single picture-----
187
188
189
190 if modus == 1:
191     cmax=10 # maximum contour size
192     cmin=2 # minimum contour size
193     #filtering by bilateral filter
194     blur=cv2.bilateralFilter(framegray,3,75,175)
195
196     #edges by canny algorithm
197     framefiltered = cv2.Canny(blur,40,120)
198
199     #contour detection by findContours
200     contours, hierarchy = cv2.findContours(framefiltered, cv2.RETR_EXTERNAL,
201     cv2.CHAIN_APPROX_SIMPLE)
202
203     if areal ==2 :#if the areal is seperated
204         height, width = framegray.shape
205         countleft = 0
206         countright =0
207         for c in contours:

```

```

208
209     if cv2.contourArea(c) < cmax and cv2.contourArea(c) > cmin:#Only if Area
of Object is between cmin and cmax
210         (x, y, w, h) = cv2.boundingRect(c)
211         if x <= width/2:#if object is on the left side of the border
212             countleft = countleft +1
213             cv2.rectangle(framergb, (x-7, y-7), (x + w+7, y + h+7), (255, 0,
0), 1)
214
215             if x > width/2: #if object is on the right side of the border
216                 countright = countright+1
217                 cv2.rectangle(framergb, (x-7, y-7), (x + w+7, y + h+7), (0, 0,
255), 1)
218
219
220     font = cv2.FONT_HERSHEY_SIMPLEX
221     cv2.putText(framergb,str(countleft),(10,height-30), font,
4,(255,255,255),2,cv2.LINE_AA)
222     cv2.putText(framergb,str(countright),(width-150,height-30), font,
4,(255,255,255),2,cv2.LINE_AA)
223     cv2.line(framergb, (int(width/2), 0), (int(width/2), height), (0,255,0), 2)
224 else: # if the areal is not seperated
225     countglobal=0
226     for c in contours:
227         #sumc = c+1
228         #print (sumc)
229         if cv2.contourArea(c) < cmax and cv2.contourArea(c) > cmin:#Only if Area
of Object is between cmin and cmax
230             (x, y, w, h) = cv2.boundingRect(c)
231             countglobal = countglobal +1
232             cv2.rectangle(framergb, (x-7, y-7), (x + w+7, y + h+7), (255, 0, 0),
1)
233
234
235     font = cv2.FONT_HERSHEY_SIMPLEX
236     cv2.putText(framergb,str(countglobal),(10,460), font,
4,(255,255,255),2,cv2.LINE_AA)
237
238     cv2.imshow('image',framergb) #Show Video in Window
239     cv2.waitKey(0)
240
241
242
243 #-----algorithm for video or live-----
244
245
246 if modus == 2 or modus == 3:
247
248     def nothing(x):
249         pass
250     # create trackbars for size change
251     cv2.createTrackbar('ContourSizeMin','image',0,50,nothing)
252     cv2.createTrackbar('ContourSizeMax','image',0,200,nothing)
253
254     cmin=0
255     cmax=0
256
257     print(filename)
258     framecount=0# counter for grafik evaluation
259
260     while (True):
261
262         # capture frame
263         ret,frame = cap.read()
264
265         if ret == False: # if the captured frame is empty the algorithm stopped
266             break
267
268         frame = cv2.flip(frame,0)
269         time.sleep(0.1)
270         framegray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #put frame from BGR to
gray

```

```

271     framergb = cv2.cvtColor(frame,cv2.COLOR_BGR2RGB) #put frame from BGR to RGB
272
273     framecount=framecount+1 #framecounter for evaluation graph
274
275     #filtering by bilateral filter
276     blur=cv2.bilateralFilter(framegray,3,75,175)
277
278     #edges by canny algorithm
279
280     #-----Contours-----
281     def canny(image, var):# automatic canny algorithm with including the median
        from the grayscale of the video
282
283         median = np.median(image)
284         lower = int(max(0, (1.0 - var) * median))
285         upper = int(min(255, (1.0 + var) * median))
286         filtered = cv2.Canny(image, lower, upper)
287         return filtered
288
289     canny_var = 0.25 # variable for canny algorithm, should be adapt to the
        captured video
290     framefiltered = canny(blur,canny var)
291
292     #contour detection by findContours
293     contours, hierarchy = cv2.findContours(framefiltered, cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
294
295     #Count and create bounding boxes
296     countleft = 0
297     countright = 0
298
299     if areal == 2:
300         height, width, channel = framergb.shape # frame shape for putting area
            in two zones
301         countleft = 0
302         countright =0
303         for c in contours:
304
305             if cv2.contourArea(c) < cmax and cv2.contourArea(c) > cmin:#Only if
                Area of Object is between cmin and cmax
306                 (x, y, w, h) = cv2.boundingRect(c) # get the positon from contour
307                 if x <= width/2:#if object is on the left side of the border
308                     countleft = countleft +1
309                     cv2.rectangle(framergb, (x-7, y-7), (x + w+7, y + h+7),
                        (255, 0, 0), 1)
310
311                 if x > width/2: #if object is on the right side of the border
312                     countright = countright+1
313                     cv2.rectangle(framergb, (x-7, y-7), (x + w+7, y + h+7), (0,
                        0, 255), 1)
314
315         font = cv2.FONT_HERSHEY_SIMPLEX
316         cv2.putText(framergb,str(countleft),(10,height-30), font,
            4,(255,255,255),2,cv2.LINE_AA)
317         cv2.putText(framergb,str(countright),(width-150,height-30), font,
            4,(255,255,255),2,cv2.LINE_AA)
318         cv2.line(framergb, (int(width/2), 0), (int(width/2), height), (0,255,0),
            2)
319
320         # put actual count value in graphcounter lists
321         graphcountleft.append(countleft)
322         graphcountright.append(countright)
323
324
325     else:
326         countglobal=0
327         for c in contours:
328
329             if cv2.contourArea(c) < cmax and cv2.contourArea(c) > cmin:#Only if
                Area of Object is between cmin and cmax
330                 (x, y, w, h) = cv2.boundingRect(c)
331                 countglobal = countglobal +1

```

```

332         cv2.rectangle(framergb, (x-7, y-7), (x + w+7, y + h+7), (255, 0,
333         0), 1)
334
335         font = cv2.FONT_HERSHEY_SIMPLEX
336         cv2.putText(framergb, str(countglobal), (10, 460), font,
337         4, (255, 255, 255), 2, cv2.LINE_AA)
338         # put actual count value in graphcounter lists
339         graphcountareal.append(countglobal)
340
341     cv2.imshow('image', framergb) #Show Video in Window
342
343     # Get Trackbar Values
344     cmin=cv2.getTrackbarPos('ContourSizeMin','image')
345     cmax=cv2.getTrackbarPos('ContourSizeMax','image')
346
347     # put actual count of frames in x axis lists
348     graphxaxis.append(framecount)
349
350     if cv2.waitKey(1) & 0xFF == ord('q') or not cap.isOpened():
351         break
352
353     cap.release()
354     cv2.destroyAllWindows('image')
355
356     #----- Evaluation -----
357
358     n = 20 # Frames per sec
359
360     # build the average of all counter values over 20 frames per secs
361     if areal == 2:
362         averagesleft = [sum(graphcountleft[i:i+n])/n for i in
363         range(0, len(graphcountleft), n)]
364         averagesright = [sum(graphcountright[i:i+n])/n for i in
365         range(0, len(graphcountright), n)]
366         sumobject=list( map(add, averagesleft, averagesright) )
367
368         #put the Auswertung in a text document
369         with open("Auswertung links.txt", "w") as output:
370             output.write(str(graphcountleft))
371         with open("Auswertung rechts.txt", "w") as output:
372             output.write(str(graphcountright))
373
374         #count how many insects have been in the area
375         for number in range(0, 8):
376             numberofinseksleft.append(averagesleft.count(number))
377             numberofinseksright.append(averagesright.count(number))
378         print(*numberofinseksleft)
379         print(*numberofinseksright)
380
381         #plots
382         plt.figure(1)
383         plt.subplot(311)
384         plt.title("Left Area")
385         plt.plot(averagesleft, drawstyle='steps-pre')
386         plt.xlabel("time in s")
387         plt.ylabel("Number of Insects")
388         plt.xticks(np.arange(0, len(sumobject)+1, 60))
389         plt.yticks(range(min(averagesleft), max(averagesright)+1))
390
391         plt.subplot(312)
392         plt.title("Right Area")
393         plt.plot(averagesright, drawstyle='steps-pre')
394         plt.xlabel("time in s")
395         plt.ylabel("Number of Insects")
396         plt.xticks(np.arange(0, len(sumobject)+1, 60))
397         plt.yticks(range(min(averagesright), max(averagesright)+1))
398
399         plt.subplot(313)
400         plt.title("Complete Area")
401         plt.plot(sumobject, drawstyle='steps-pre')
402         plt.xlabel("time in s")

```

```

400     plt.ylabel("Number of Insects")
401     plt.xticks(np.arange(0,len(sumobject)+1,60))
402     plt.yticks(range(min(sumobject),max(sumobject)+1))
403
404     plt.figure(2)
405     plt.plot(sumobject,drawstyle='steps-pre', label='sum')
406     plt.plot(averagesright,drawstyle='steps-pre', label='right')
407     plt.plot(averagesleft,drawstyle='steps-pre', label='left')
408     plt.xticks(np.arange(0,len(sumobject)+1,60))
409     plt.yticks(range(min(sumobject),max(sumobject)+1))
410     plt.xlabel("time in s")
411     plt.ylabel("Number of Insects")
412
413     width = 0.25
414     x = [0,1,2,3,4,5,6,7,8] #number of insects in area
415     x_indexes =np.arange(len(x))
416     plt.figure(3)
417     plt.bar(x_indexes - width/2, numberofinseksleft, width=width, label='left')
418     plt.bar(x_indexes + width/2, numberofinseksright, width=width, label='right')
419     plt.title("Verteilungskurve")
420     plt.xlabel("Number of Insects")
421     plt.ylabel("Time in s")
422
423     else:
424         sumobject = [sum(graphcountareal[i:i+n])/n for i in
425             range(0,len(graphcountareal),n)]
426
427         plt.figure(1)
428         plt.plot(sumobject,drawstyle='steps-pre',label='sum')
429         plt.xticks(np.arange(0,len(sumobject)+1,10))
430         plt.yticks(range(min(sumobject),max(sumobject)+1))
431         plt.xlabel("time in s")
432         plt.ylabel("Number of Insects")
433
434     plt.legend()
435
436     # plt.xticks(ticks= x_indexes, labels=x)
437     plt.show()
438
439     cv2.destroyAllWindows()#close all windows

```

B Anhang: Bilder des Prototyps



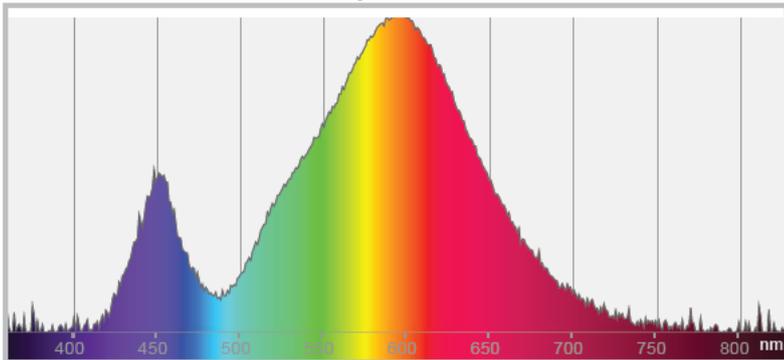


C Anhang: Messungen Leuchtmittel

Color parameters:

Color temperature	Color rendering index	Color quality scale	Color fidelity	Color gamut	Television Lighting Consistency Index	Standard Deviation of Colour Matching	Color coordinate cie 1931	Color coordinate cie 1931	Color coordinate	Color coordinate
CCT	CRI	CQS	TM30 Rf	TM30 Rg	TLCI	SDCM	x	y	u	v
3065 K	71,8	71,0	74,8	94,3	46	n/a	0,432	0,401	0,249	0,346

Spectra

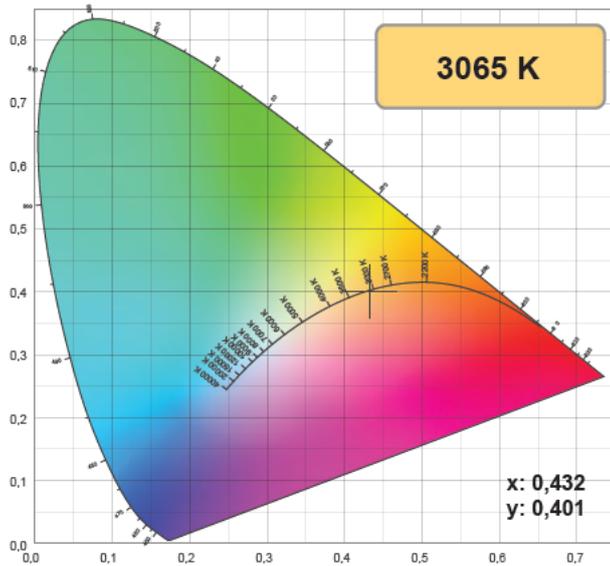


Dominant Wavelength: 584 nm

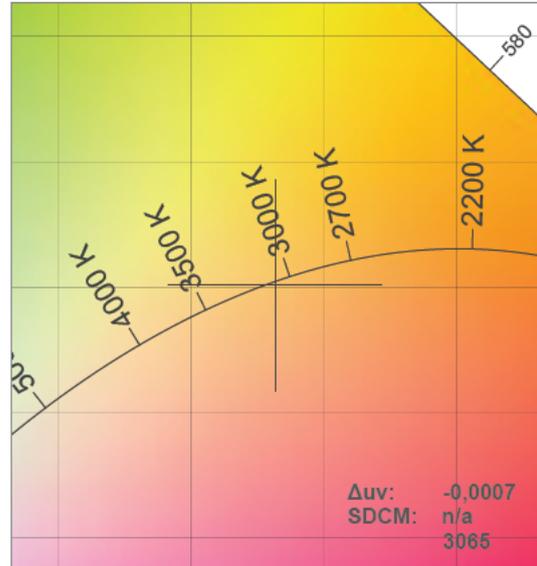
Irradiance:

400 nm:	0,45m W/m ² /nm
450 nm:	4,60m W/m ² /nm
500 nm:	1,72m W/m ² /nm
550 nm:	6,09m W/m ² /nm
600 nm:	9,33m W/m ² /nm
650 nm:	4,52m W/m ² /nm
700 nm:	1,25m W/m ² /nm
750 nm:	0,17m W/m ² /nm
800 nm:	0,00m W/m ² /nm

CIE 1931 diagram



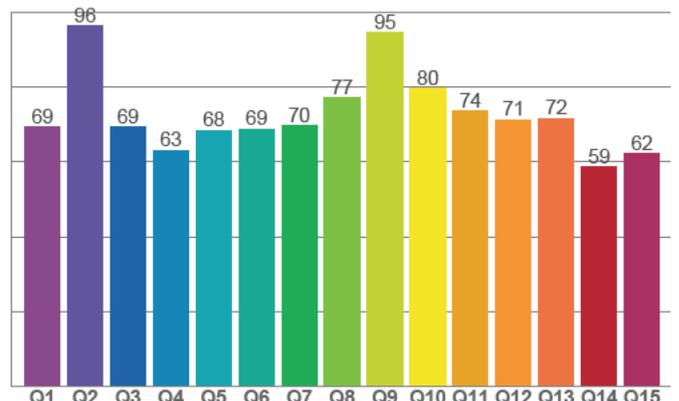
Zoomed CIE 1931 diagram with SDCM Ellipse



CRI: 71,8 (R1-R8)



CQS: 71,0



CRI R values, only R1-R8 are used to calculate final CRI value

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
68,2	81,9	92,8	67,4	66,8	74,2	78,2	45,3	-29,7	57,5	60,9	49,3	70,7	95,8	60,7

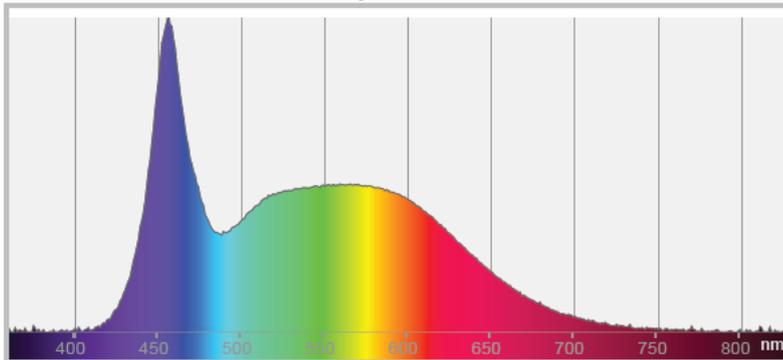
CQS Q values

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
69,5	96,3	69,3	63,1	68,4	68,8	69,6	77,2	94,6	79,7	73,6	71,3	71,7	58,8	62,3

Color parameters:

Color temperature	Color rendering index	Color quality scale	Color fidelity	Color gamut	Television Lighting Consistency Index	Standard Deviation of Colour Matching	Color coordinate cie 1931	Color coordinate cie 1931	Color coordinate	Color coordinate
CCT	CRI	CQS	TM30 Rf	TM30 Rg	TLCI	SDCM	x	y	u	v
6819 K	85,6	82,6	84,0	92,9	75	n/a	0,308	0,326	0,195	0,311

Spectra

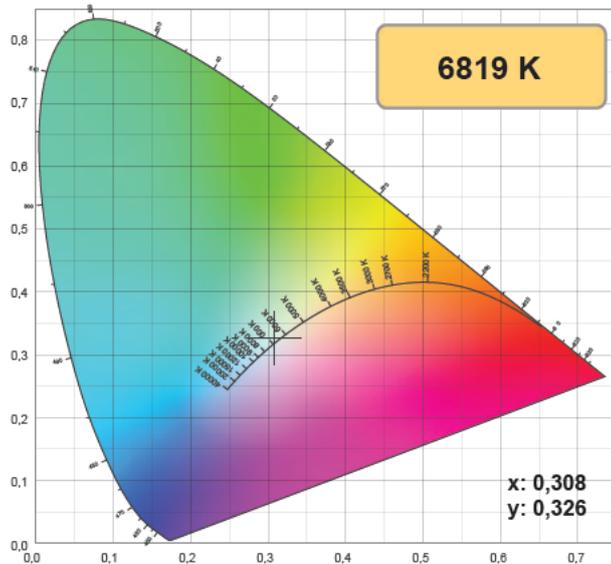


Dominant Wavelength: 484 nm

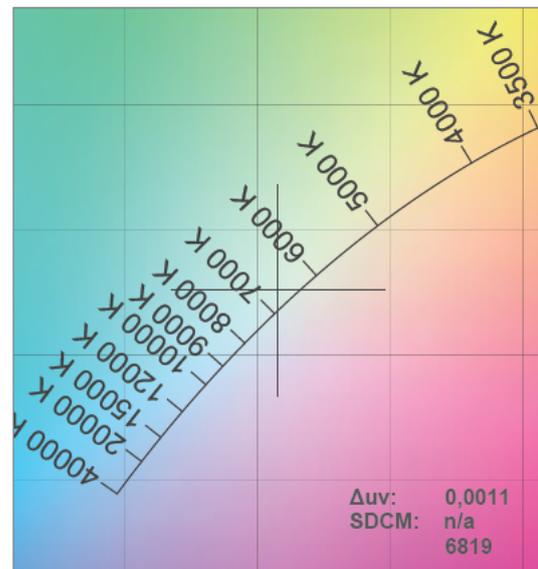
Irradiance:

400 nm:	0,5m W/m ² /nm
450 nm:	34,2m W/m ² /nm
500 nm:	15,2m W/m ² /nm
550 nm:	19,8m W/m ² /nm
600 nm:	18,0m W/m ² /nm
650 nm:	8,2m W/m ² /nm
700 nm:	2,1m W/m ² /nm
750 nm:	0,4m W/m ² /nm
800 nm:	0,0m W/m ² /nm

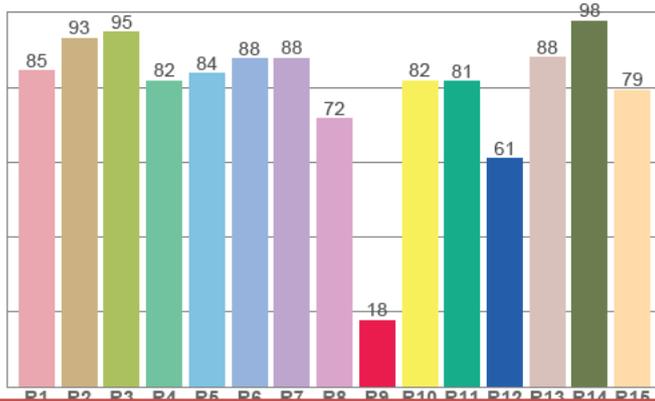
CIE 1931 diagram



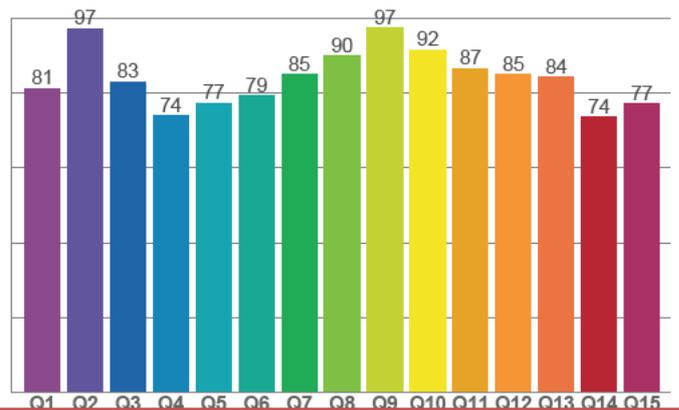
Zoomed CIE 1931 diagram with SDCM Ellipse



CRI: 85,6 (R1-R8)



CQS: 82,6



CRI R values, only R1-R8 are used to calculate final CRI value

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
84,5	93,1	94,7	81,7	83,7	87,8	87,7	71,8	17,8	81,8	81,5	60,9	87,9	97,7	79,2

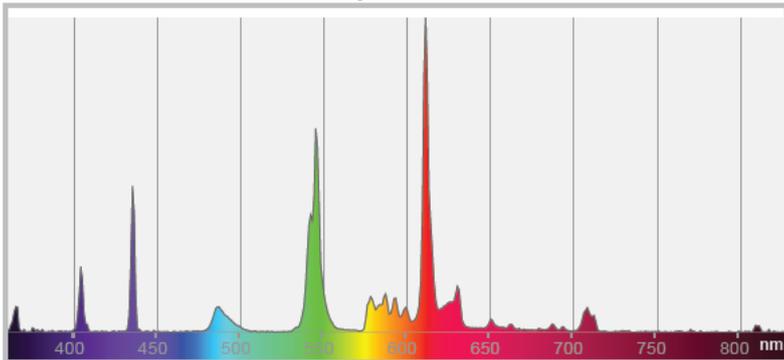
CQS Q values

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
81,2	97,2	83,0	73,9	77,4	79,3	85,0	90,1	97,5	91,6	86,5	85,0	84,3	73,8	77,3

Color parameters:

Color temperature	Color rendering index	Color quality scale	Color fidelity	Color gamut	Television Lighting Consistency Index	Standard Deviation of Colour Matching	Color coordinate cie 1931	Color coordinate cie 1931	Color coordinate	Color coordinate
CCT	CRI	CQS	TM30 Rf	TM30 Rg	TLCI	SDCM	x	y	u	v
2787 K	80,5	74,1	72,7	102,6	33	n/a	0,458	0,418	0,258	0,353

Spectra

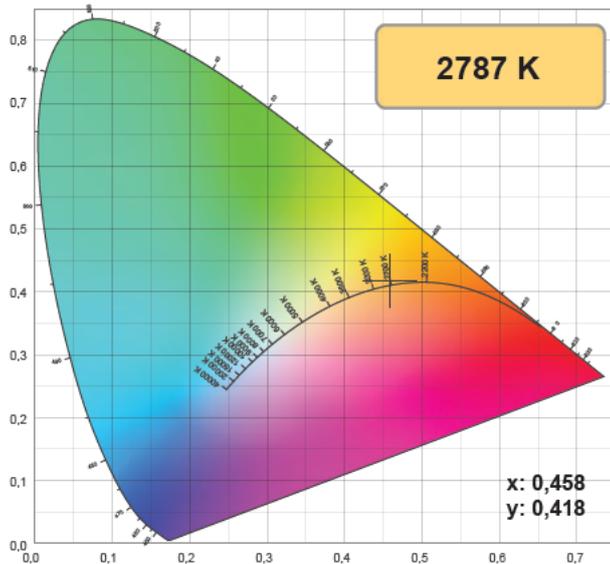


Dominant Wavelength: 583 nm

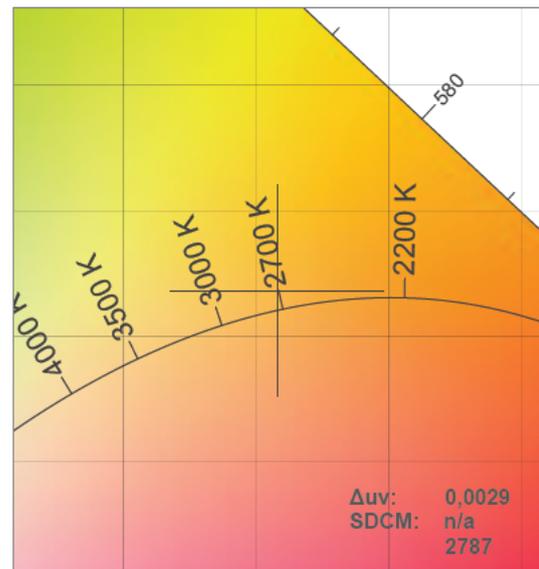
Irradiance:

400 nm:	0,6m W/m ² /nm
450 nm:	0,1m W/m ² /nm
500 nm:	1,1m W/m ² /nm
550 nm:	8,7m W/m ² /nm
600 nm:	5,0m W/m ² /nm
650 nm:	3,0m W/m ² /nm
700 nm:	0,4m W/m ² /nm
750 nm:	0,1m W/m ² /nm
800 nm:	0,0m W/m ² /nm

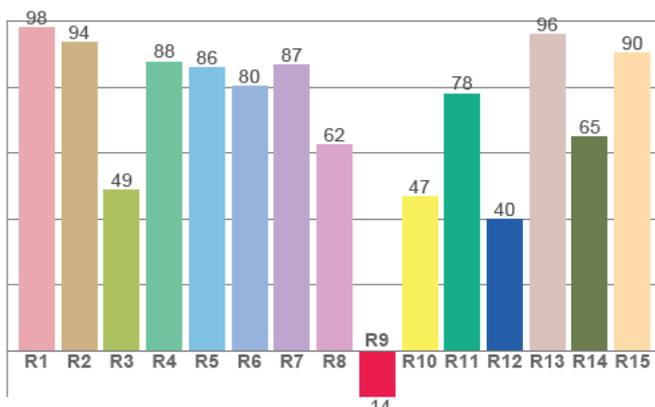
CIE 1931 diagram



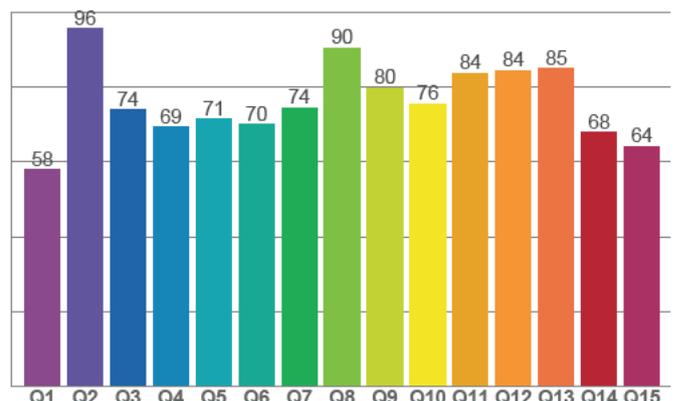
Zoomed CIE 1931 diagram with SDCM Ellipse



CRI: 80,5 (R1-R8)



CQS: 74,1



CRI R values, only R1-R8 are used to calculate final CRI value

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
98,1	93,7	48,9	87,7	85,8	80,2	86,8	82,5	-13,7	46,8	78,0	39,9	96,0	64,9	90,4

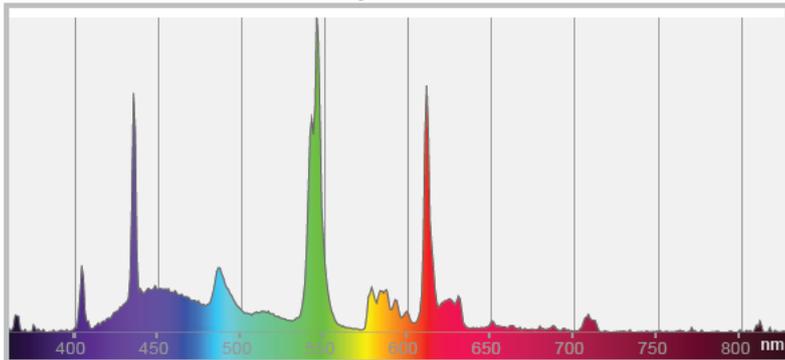
CQS Q values

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
57,9	95,6	74,0	69,4	71,4	70,0	74,4	90,5	79,9	75,6	83,7	84,4	85,0	68,0	64,3

Color parameters:

Color temperature	Color rendering index	Color quality scale	Color fidelity	Color gamut	Television Lighting Consistency Index	Standard Deviation of Colour Matching	Color coordinate cie 1931	Color coordinate cie 1931	Color coordinate	Color coordinate
CCT	CRI	CQS	TM30 Rf	TM30 Rg	TLCI	SDCM	x	y	u	v
6547 K	80,1	80,1	81,2	97,8	59	n/a	0,310	0,341	0,192	0,316

Spectra

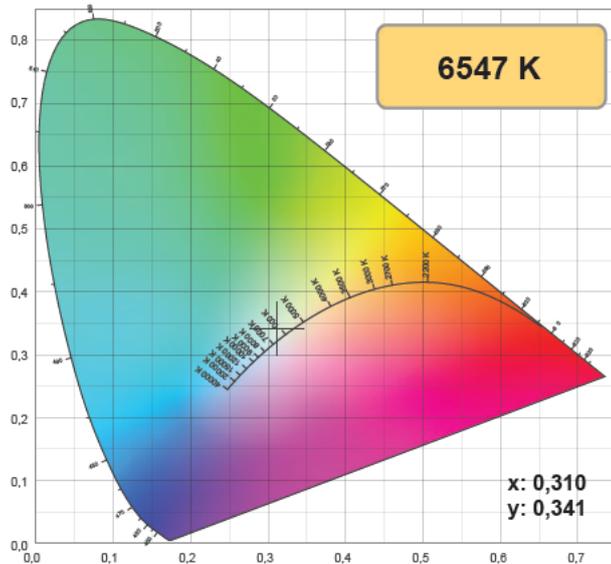


Dominant Wavelength: 540 nm

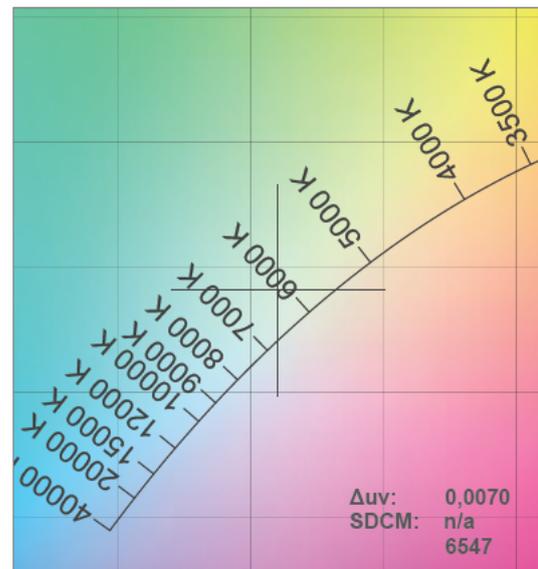
Irradiance:

400 nm:	0,3m W/m ² /nm
450 nm:	2,6m W/m ² /nm
500 nm:	1,3m W/m ² /nm
550 nm:	4,1m W/m ² /nm
600 nm:	1,1m W/m ² /nm
650 nm:	0,7m W/m ² /nm
700 nm:	0,1m W/m ² /nm
750 nm:	0,0m W/m ² /nm
800 nm:	0,0m W/m ² /nm

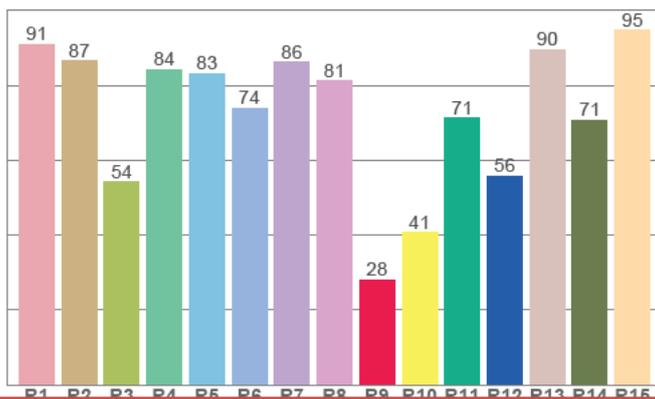
CIE 1931 diagram



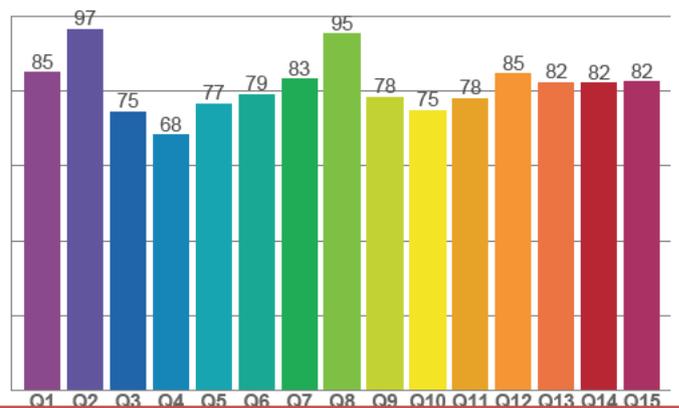
Zoomed CIE 1931 diagram with SDCM Ellipse



CRI: 80,1 (R1-R8)



CQS: 80,1



CRI R values, only R1-R8 are used to calculate final CRI value

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
91,0	88,5	54,3	84,3	83,1	74,0	86,1	81,2	28,1	40,6	71,2	55,8	89,6	70,7	94,6

CQS Q values

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
85,0	96,6	74,5	68,3	76,6	79,2	83,3	95,3	78,5	74,8	78,0	84,6	82,3	82,2	82,4

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	_____
Ort	Datum	Unterschrift im Original