Titelseite



in Kooperation  
 mit

# Kurzfassung

In dem Projekt wurde ein Laser programmiert, ein rundes Objekt zu bestrahlen. Die Grundidee basiert auf einer Art Schutzprogramm, dass Menschen vor bösartigen Insekten schützten. Dieses Programm könnte anderweitig genutzt werden, um ebenfalls den Menschenschutz zu sichern. Wenn man das Programm an den Flughäfen anbringen würde, könnte man die Drohnen abschießen. Dies würde aber bei echten Versuchen an Tieren unsere ethische Grenze überstreiten. Deswegen haben wir Kügelchen und ausgeschnittenen Kunstinsekte verwendet. Um so etwas überhaupt Anrichten zu können, brauchte man einige Programmierungen, die jeweils für verschiedenstes verantwortlich sind. Dabei haben wir erkannt, dass die genaue Kalibrierung etwas komplizierter ist, doch das wurde auch effizient behandelt.

Inhaltsverzeichnis

[Kurzfassung 2](#_Toc124179647)

[1 Einleitung 1](#_Toc124179648)

[2 Material und Methoden 1](#_Toc124179649)

[2.1 Hough Circle Transform 1](#_Toc124179650)

[2.2 Canny Edge Detection 3](#_Toc124179651)

[2.3 Kalibrierung der Koordinatensysteme 3](#_Toc124179652)

[2.4 Versuch 1 3](#_Toc124179653)

[2.5 Versuch 2 3](#_Toc124179654)

[3 Ergebnisse 4](#_Toc124179655)

[4 Ergebnisdiskussion 4](#_Toc124179656)

[5 Schluss 4](#_Toc124179657)

[6 Quellen 4](#_Toc124179658)

[7 Unterstützungsleistungen 4](#_Toc124179659)

[8 Danksagung 4](#_Toc124179660)

# Einleitung

Durch den Klimawandel wird es in Mittel- und Nordeuropa immer wärmer. Dadurch wandern immer mehr Insektenarten aus südlicher gelegenen Ländern ein oder können hier überleben und Populationen bilden, wenn sie eingeschleppt werden. Manche von ihnen wie z.B. die Tigermücke können Krankheiten[[1]](#footnote-1) auf den Menschen übertragen, die bisher in Europa so nicht verbreitet wurden.

Würde man solche Arten mit Insektengift bekämpfen, so würde man auch einheimischen Arten schaden und damit Nahrungsketten und das ökologische Gleichgewicht stören. Wir haben uns überlegt ob es eine Giftfreie Möglichkeit gibt, solche Arten gezielt zu bekämpfen.

Unsere Idee ist es mithilfe einer Bilderkennungssoftware bestimmte Insektenarten zu erkennen. Dann wollen wir sie mit einem Laser treffen.

Natürlich ist dies nur ein Gedankenexperiment. Für unsere JuFo-Arbeit wollen wir Objekte erkennen, und anschließend mit einem Laser markieren. Damit dies funktioniert sind mehrere Teilschritte nötig. Zuerst muss dass Koordinatensystem der Kamera mit dem des Lasers in Einklang gebracht werden. Anschließend muss ein Objekt mit der Kamera erfasst werden. Zuletzt muss nun der Laser auf dieses Objekt ausgerichtet werden.

# Material und Methoden

## Hough Circle Transform

Eine der für unser Projekt meistbenutzten Funktionen von OpenCV[[2]](#footnote-2) ist die HoughCircles Methode. Diese wird benutzt, um Kreise in einem Bild zu finden. Sie funktioniert folgendermaßen: Zuerst werden im Originalbild die Kanten gesucht, vgl. Abbildung 1 .

Anschließend werden für alle möglichen Radien (von einem festgelegten Minimum bis normalerweise zur Länge der Diagonalen des Bildes) in separaten Arrays mit Dimensionen des Bildes Kreise mit dem zugehörigen Radius durch die Kanten übereinandergelegt. In Punkten, in denen sich 2 oder mehr Kreise überschneiden ist der Wert die zahl der Kreise durch diesen Punkt.

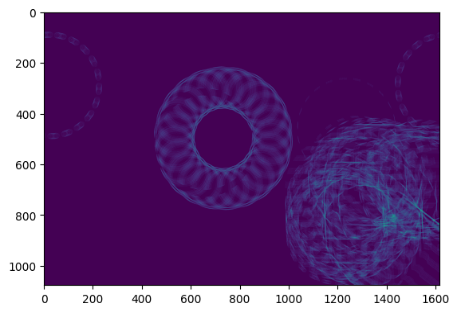
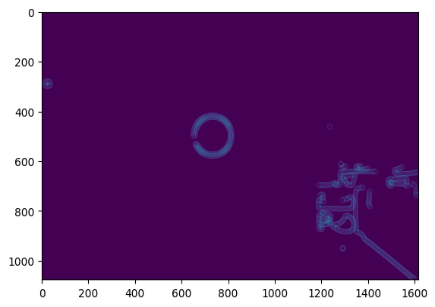


Abbildung : Kreise mit verschieden Radien

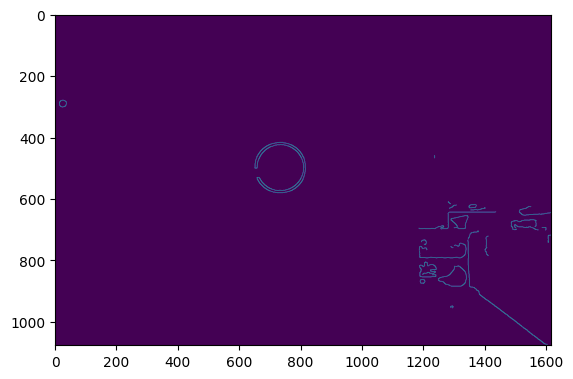
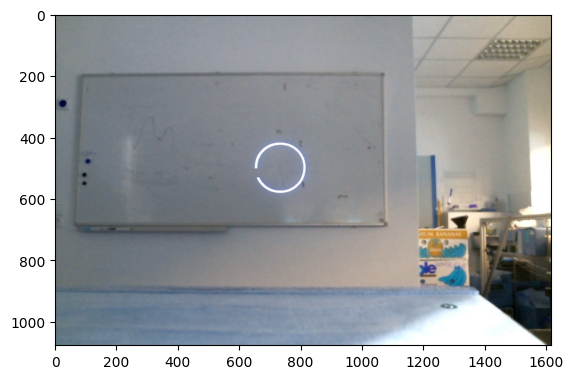


Abbildung : Originalbild und Kanten

Gehen nun für einen Radius viele dieser Kreise durch denselben Punkt, sind also Gleichweit von diesem Entfernt kann man annehmen, dass dieser Punkt Mittelpunkt eines Kreises mit besagtem Radius ist (vgl. Abbildung 3). Vorstellen kann man sich das Ganze so, dass man lokale Maxima in der Helligkeit des hellsten Punktes für jeden Radius sucht (Abbildung 4, Blaue Linie). Radius ist dann die x-Stelle dieses Maximums, x (Grüne Linie) und y (Orange Linie) sind Koordinaten des Mittelpunkts, einfach Koordinaten des hellsten Punktes bei diesem Radius. Anschließend kann man zur Kontrolle den Kreis in das Originalbild einzeichnen.

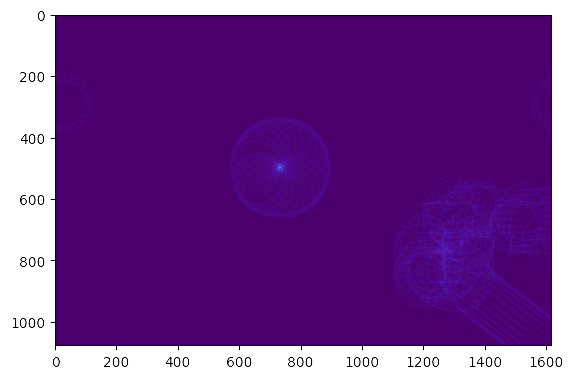


Abbildung : Array für korrekten Radius, Pfeil zeigt auf den Mittelpunkt

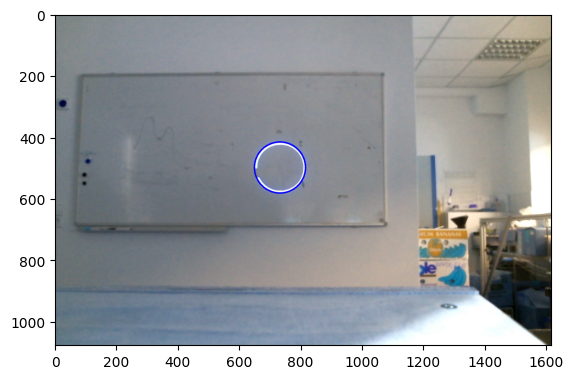


Abbildung : Originalbild mit markiertem Kreis

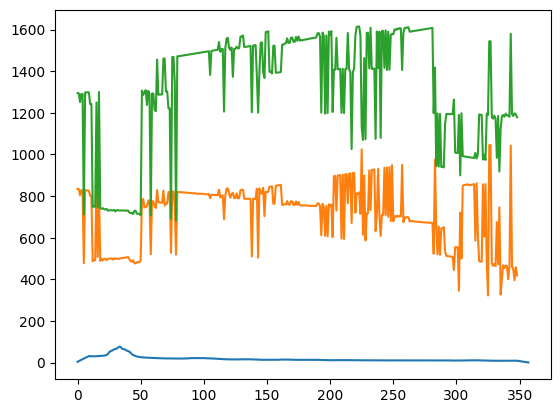


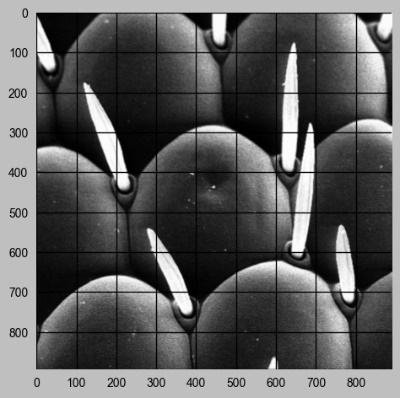
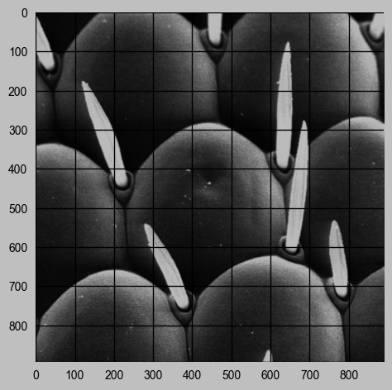
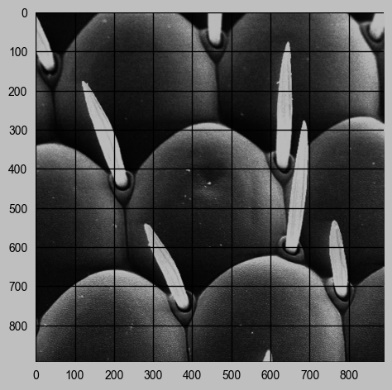
Abbildung : Helligkeit, x und y Koordinaten in Abhängigkeit vom Radius

## Canny Edge Detection[[3]](#footnote-3)

Die Methode, die wir z.B beim Finden von Kanten für die Houg Circles Methode verwenden, heißt Canny Edge Detection. Benannt nach ihrem Entwickler, John F. Canny, lässt sie sich logisch in 4 Schritte aufteilen, hier am Beispiel eines Fruchtfliegenauges dargestellt:

1. Grayscale und Noise Reduction (Gaussian). Dann haben wir ein Bild A. Dieses stellen wir uns als 2-dimensionalen Array vor, welcher nun die Helligkeiten der einzelnen Pixel speichert.

Abbildung : Drosophilaauge, REM-Bild: Klaus Steiner



2. Intensity Gradients: Hier werden Sobeloperatoren in x- und y-Richtung angewendet.

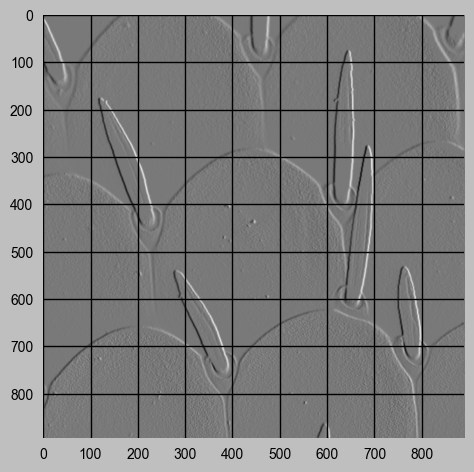


Abbildung : Gx

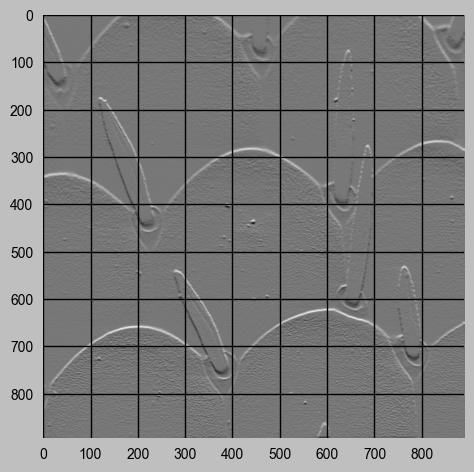


Abbildung : Gy

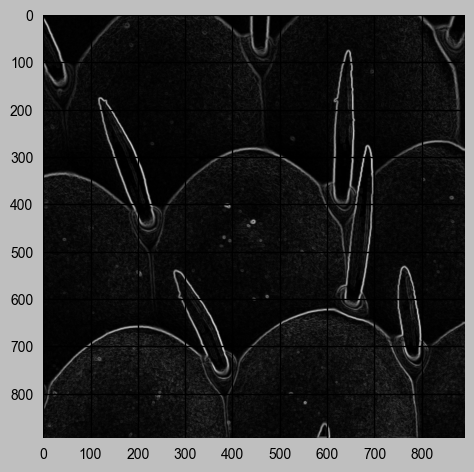


Abbildung : G

Anschließend wird die Kantenstärke berechnet mit . Die Richtung der Kante berechnet man mit Theta=atan2(Gx,Gy) und rundet auf einen der 4 Winkel 0°, 45°, 90° oder 135°.

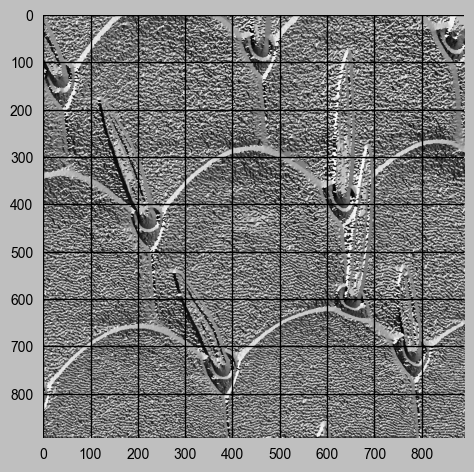


Abbildung : Theta

3. Non Maximum Suppression: Hier wird die Kantenbreite auf einen Pixel limitiert, dazu werden alle Werte, die keine lokalen Maxima von G horizontal zur Kante sind auf 0 gesetzt.

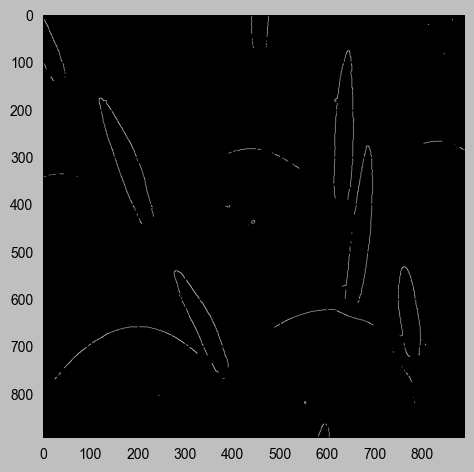


Abbildung : Non Maximum Suppression

4. Hysteresis: Arbeitet mit 2 Werten, T1 < T2,   
Für G>T2: Kante (sicher)   
Für G<T1: Keine Kante (sicher)   
Für T1<G<T2 (unsicher): Falls für mindestens einen Nachbarn dieses Pixels gilt G\*>T2, dann weist man diesem Pixel den Wert G=T2 zu. Nun kann dieser auch zur Aktivierung von benachbarten unsicheren Pixeln verwendet werden kann. (Diese Bilder wurden mit folgendem Python-Programm ausgewertet). Weil der Vorletzte Schritt hier sehr gut funktioniert hat, kann man zum letzten Bild keinen Unterschied sehen.

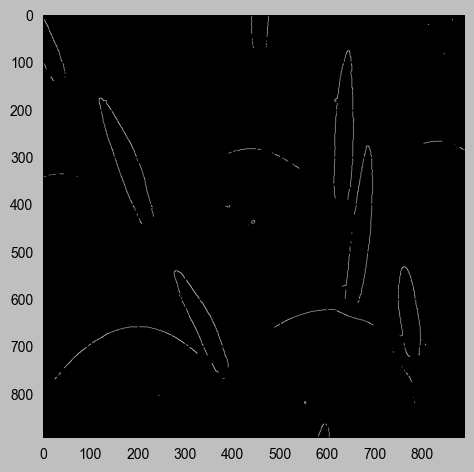


Abbildung : Kanten

## Versuch 1

Das Objekt wird von der Kamera erkannt und umrandet. Durch die Transformation werden die aufgenommenen Koordinaten der Position des runden Objekts in dem Koordinatensystem der Kamera, in die Koordinaten der Position in dem Koordinatensystem des Lasers umgerechnet. Somit schießt der Laser mit einem kleinen Kreis auf das erkannte, runde Objekt ab. Doch die war ein gutes, jedoch nicht das gewünschte Resultat. Zu Optimierungszwecken wurde es noch ein wenig umgeschrieben.

## Versuch 2

Um das Programm zu optimieren, umrandete die Kamera das runde Objekt nicht mehr, es hatte dieselbe Effizienz wie davor. Die Form, die der Laser zeichnete, hat sich auch verändert. Zuvor malte der Laser einen kleinen Kreis, doch nach der Optimierung malte der Laser eine horizontal gedrehte Sanduhr. Die Kamera erkennt die Bewegung des Objekts und die Umrechnung der neuen Koordinaten erfolgt immer noch gut und die Funktionsweise ist die ähnliche.

# Ergebnisse

# Ergebnisdiskussion

# Schluss

Durch die vorliegende Arbeit, konnte man sehen, dass die Programmierung innovativ, komplex als auch interessant ist. In Bezug auf die ursprüngliche Idee, kann sich sehen lassen, dass man viele Teile der Programmierungen miteinander verbinden muss, wie eine Art Puzzle. Man muss die Koordinaten der Kamera und des Lasers optimieren und übereinstimmen lassen, während man dafür sorgen muss, dass die Kamera das insektenähnliche, runde Objekt finden kann und der Laser einen Strahl auf die Koordinaten des Objekts hinaufschießt. Natürlich sollte der Laser die genügende Leistung aufbringe, damit es überhaupt ein Lebewesen verbrennen kann. Die Anwendungen könnten auch anderweitig genutzt werden, als eine Art Schutzprogramm oder ähnliches, wenn man zum Beispiel die Dauer des Programmes optimieren würde.

# Quellen

<https://www.ndr.de/ratgeber/klimawandel/Wie-sich-der-Klimawandel-auf-die-Tierwelt-auswirkt,artenvielfalt112.html> (Zuletzt abgerufen am 09.01.23)

<https://numpy.org/> (Zuletzt abgerufen am 09.01.23)

# Unterstützungsleistungen

Matthias Kesenheimer hat uns seine selbstgebaute Laseranlage sowie ein passendes Python-Interface zur Verfügung gestellt und stand uns jederzeit als kompetenter Ansprechpartner zur Verfügung.

# Danksagung

1. <https://www.gesundheitsamt-bw.de/lga/de/kompetenzzentren-netzwerke/arbo-baden-wuerttemberg/gesundheitsgefahren/> (zuletzt abgerufen am 09.01.23) [↑](#footnote-ref-1)
2. <https://docs.opencv.org/> () [↑](#footnote-ref-2)
3. Nach <https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123> (Zuletzt abgerufen am 09.02.23) [↑](#footnote-ref-3)