



Entwicklung eines portablen Abschreksystems gegen unliebsame Kleintiere

Studienarbeit

des Studiengang Technische Informatik- IT-Automotive
an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Levin Müller

14. Oktober 2022

Bearbeitungszeitraum
Matrikelnummer, Kurs
Ausbildungsfirma
Betreuer

zeitraum
7994341, TINF20-ITA
softwareinmotion GmbH, Schorndorf
Prof. Dr. Janko Dietzsch
Janko.Dietzsch@dhbw-stuttgart.de

Erklärung

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema: *Entwicklung eines portablen Abschrecksystems gegen unliebsame Kleintiere* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Welzheim, 14. Oktober 2022

Levin Müller

Zusammenfassung

Abstract normalerweise auf Englisch. Siehe: http://www.dhbw.de/fileadmin/user/public/Dokumente/Portal/Richtlinien_Praxismodule_Studien_und_Bachelorarbeiten_JG2011ff.pdf (8.3.1 Inhaltsverzeichnis)

Ein „Abstract“ ist eine prägnante Inhaltsangabe, ein Abriss ohne Interpretation und Wertung einer wissenschaftlichen Arbeit. In DIN 1426 wird das (oder auch der) Abstract als Kurzreferat zur Inhaltsangabe beschrieben.

Objektivität soll sich jeder persönlichen Wertung enthalten

Kürze soll so kurz wie möglich sein

Genauigkeit soll genau die Inhalte und die Meinung der Originalarbeit wiedergeben

Üblicherweise müssen wissenschaftliche Artikel einen Abstract enthalten, typischerweise von 100-150 Wörtern, ohne Bilder und Literaturzitate und in einem Absatz.

Quelle: <http://de.wikipedia.org/wiki/Abstract> Abgerufen 07.07.2011

Diese etwa einseitige Zusammenfassung soll es dem Leser ermöglichen, Inhalt der Arbeit und Vorgehensweise des Autors rasch zu überblicken. Gegenstand des Abstract sind insbesondere

- Problemstellung der Arbeit,
- im Rahmen der Arbeit geprüfte Hypothesen bzw. beantwortete Fragen,
- der Analyse zugrunde liegende Methode,
- wesentliche, im Rahmen der Arbeit gewonnene Erkenntnisse,
- Einschränkungen des Gültigkeitsbereichs (der Erkenntnisse) sowie nicht beantwortete Fragen.

Quelle: http://www.ib.dhbw-mannheim.de/fileadmin/ms/bwl-ib/Downloads_alt/Leitfaden_31.05.pdf, S. 49

Inhaltsverzeichnis

Abkürzungsverzeichnis	I
Abbildungsverzeichnis	II
Tabellenverzeichnis	III
1 Einleitung	1
1.1 Begrifflichkeiten und Systeme	1
1.2 Problemstellung	2
2 Aufgabenstellung	3
3 Grundlagen	5
3.1 Stand der Technik	5
3.2 Computer Vision	5
3.3 Dreidimensionales Zielsystem	5
4 Umsetzung	7
4.1 Bildererkennung und -Verarbeitung	7
4.2 Hardwarerealisierung	7
4.3 Dreidimensionales Zielsystem	8
5 Reflexion und Ausblick	9
Literatur	10

Abkürzungsverzeichnis

FFI Foreign-Function-Interface

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Begrifflichkeiten und Systeme

1.1.1 Rust

„Rust ist einer der beliebtesten Programmiersprachen und das 7 Jahre in Folge.“-[1]

Einer der Gründe dafür ist, dass Rust sich von dynamisch-typisierte Sprachen inspirieren lassen hat. Durch diese Inspiration ist es nicht nötig Variablen mit konkreten Datentypen zu deklarieren. Bei der Kompilierung werden diese automatisch erkannt und verwendet. Rust setzt allerdings ein stark typisiertes System ein. Bei der Kompilierung werden die dynamischen Datentypen aufgelöst und Datentypfehler direkt erkannt. Dadurch werden viele Datentypkonversionsfehler während der Kompilierung und somit in der Entwicklungsphase aufgedeckt. Ohne Erkennung führen sie zu Problemen in der Laufzeit eines Programms. Der Compiler verfügt zugleich über umfangreiche Checks, die vor weiteren Laufzeitfehlern schützen sollen. Ohne diese Checks muss ein Entwickler diese mittels eigener Tests finden und abfangen.

Dem Entwickler wird zusätzlich die Möglichkeit geboten zu entscheiden ob Daten auf dem Stack oder dem Heap gespeichert werden sollen. Durch eine bewusste Nutzung dieser Möglichkeit kann noch das letzte Bisschen Performanz aus dem Speichermanagement geholt werden.

Ein großer Vorteil entgegen C#, C/C++ oder Java liegt darin, dass der Compiler erkennen kann, wann Daten nicht länger benötigt werden und deren belegten Speicher automatisch freigibt. Dies ermöglicht ein effizientes Nutzen des Speichers ohne viel Aufwand dem Entwickler aufzubürden. Vergleichsweise viel Aufwand hat man in C/C++. Dort müssen verschiedene Speichernutzungen mit *malloc* und *free* manuell gemanagt.

Mit diesen Möglichkeiten kann Rust direkten *low-level* Code mit hohen Sicherheitsgarantien produzieren. Da Rust noch jung ist, sind Neuentwicklungen von Desktopanwendungen schwer umsetzbar. Durch zu wenige Features für die Entwicklung graphischer Oberflächen sind diese nur schwer zu realisieren. Dadurch werden häufig Performanz- und sicherheitskritische Abschnitte in anderen Sprachen mittels dem Foreign-Function-Interface (FFI) ausgetauscht. [2]

1.1.2 Raspberry Pi

1.1.3 Computer Vision-Stereo Vision

1.2 Problemstellung

Die meisten haben es schon miterlebt oder von einem Bekannten gehört, dass bei Ihnen einen Marder zugeschlagen hat oder die Mülltonnen von Waschbären verwüstet wurden. Nicht nur können sie dabei einen großen Schaden verursachen, sondern hinterlassen noch ihre eigenen Substanzen, in die die meisten hineintreten. Vor allem in den ländlicheren Regionen nahe den Wäldern kommt es häufiger zu unliebsamen Interaktionen.

Die Kleintiere können häufig schon mit einfachen Mitteln von den Missetaten auf dem eigenen Grundstück ferngehalten werden. Dazu gehören Abschreckmechaniken, die bei Bewegung einen Hochfrequenzton abspielen und das Licht einschalten, aber auch verschiedene teilweise teure Duftstoffe. Bedauerlicherweise haben die meisten davon ein großes Problem: Sie halten nicht lange. Die Tiere gewöhnen sich an die Töne, dem Licht und an die Duftstoffe. Wenn diese immer wieder versetzt werden, irritiert die Tiere das zwar, zögert die Gewöhnung aber lediglich um einige Wochen hinaus.

Das Ziel der Arbeit ist es, eine weitere Komponente hinzuzufügen. Das Wasser. Besonders Marder und Katzen, die ihre Hinterlassenschaften im Garten verteilen, meiden das Wasser und somit Sprinkleranlagen. Daher finden Sie neue Wege, bei denen Ihnen keine Sprinkler im Weg stehen.

Daher soll mit den anderen erprobten Mitteln ein Gesamtpaket entwickelt werden, welches überall einfach platziert werden kann und die Tiere vom Garten, sowie Auto fernhält. Da unschuldige Passanten nicht nass gespritzt werden wollen und Wasser gespart werden soll, soll das Gesamtpaket nur auf Kleintiere ihr Inventar loslassen.

Anpassung mit Bild Verwüstung des eigenen Gartens...

2 Aufgabenstellung

1. Auswahl der Hardware und Bibliotheken (Wasserpumpe, Kamera, Energieversorgung und Management, Rust-Crates).
2. Einarbeitung Opencv, Tensorflow und Rust alternativen (wenn in rust nicht vorhanden).
3. Hardwaretechnische Realisierung und Implementierung der einzelnen Hardwarecontroller. Eventuell Auslagerung der Motoransteuerung (Zielsystem) auf Arduino → Einfacherer Wechsel Pi/Jetson da nur Vertikaler und Horizontaler Winkel gestellt werden müssen.

Hardware:

- (Demo?) Raspberry Pi 3b (PI 4 CM + IO/Jetson)
 - Unbestimmte Menge an Kabel
 - 1-2 Baugleiche Kameras (NoIR da Einsatz bei Nacht/unbeleuchtete Umgebung)
 - vielfältige 3D-Drucke Gehäuse/Zielsystem etc.
 - Energieversorgung:
12V kleine Autobatterie
Solarpaneel
 - Schrittmotoren/E-Motoren für Zielsystem (mit Positionssensor oder „einlernen“)
 - Pumpe mit Schläuche
 - Tonwiedergabe (+Verstärker?)
 - Lichtwiedergabe (Blitzlicht)
 - Relais für einschalten der Aktoren
MOSFETS durch nicht konstante Energiequelle (Batterie/Solaranlage) nicht direkt möglich
4. Demo- und Testentwicklung der Bildverarbeitungssoftware.
Batterie/Solar/PI 3 + Kamera in kleine Box für frühzeitiges erhalten von Real-World-Daten

5. Weiterentwicklung und Testen in der freien Wildbahn.
Alles zusammen in PI 4 oder Jetson

3 Grundlagen

3.1 Stand der Technik

Existierende Systeme beschreiben... Marderschreck und co.

3.2 Computer Vision

Befähigung Computerprogramme „sehen“ zu lassen.

3.2.1 Vision

Bilderkennungsmerkmale etc.

3.2.2 Stereo Vision

Tiefenwahrnehmung für anständiges Zielen (statt ML? und zu viele Tricks?)

3.3 Dreidimensionales Zielsystem

Mechanisches System beschreiben etc.

3.3.1 Strömungslehre

Pumpe → Nötigung anständiger Berechnung des Zielsystem

3.3.2 Tiefenberechnung

Zusätzlich zu betrachtende Effekte Luftwiderstand; Dispersion; Wind/Error-correction etc.
Bezug Error: Bilderkennung des Wasserstrahls → Regelungstechnik und Autokalibrierung
+ evtl. Einspielen der Tiefenwahrnehmung an neuer Position (nicht bei Stereo-Vision
nötig).

4 Umsetzung

4.1 Bilderkennung und -Verarbeitung

4.2 Hardwarerealisierung

Verkabelung, Aufbau, Herausforderungen, spezielle Lösungen

Auswahl PI -> Liefermangel Erstidee Multiplexer um zeitverzögerung zu reduzieren -> schlechte Kritik

Zweite Idee usb-> von vornerein "komplexer/durch zeitkacke -> irgendwie auch teuer

Dritte Idee esp32-CAM Module -> Hohe Latenzen und Zeitsynchronitätsaufwand

Viertens -> compute model + I/O shield auch teuer aber 2 csi Anschlüsse

5. Jetson mit 2 CSI Anschlüssen -> teuer und noch mehr Liefermangel

4.2.1 Wasserversorgung

integrierte Pumpe → Soll portable sein

Alternativ: Gartenschlauch mit Ventilsteuerung

Pumpe

Orientierung Gartenschlauch/Sprinkler: Durchfluss in etwa 20L/min; Druck bis 4 Bar; Düsenspritze ca. 1-2mm Durchmesser

Pumpe: Membranpumpe → Gleichbleibende Fördermenge bei hohen Druckunterschieden (Pumpe 1-4 Bar). Druck Vernachlässigen und Strömungslehre mit Fördermenge berechnen. Membranpumpe haben Druckschalter → wie ist ein/ausschaltverfahren? Delay etc. Auswahl Verwendung von Ventil zum Durchschalten oder von Versorgungsspannung der Pumpe???

Normale Pumpe: Komplexer durch Kennlinie → Berechnung des inneren Drucks nötig und Interpolation dieser. (Meiste Pumpen haben schon bei 1-2 Bar Probleme → Druckventil schaltet Pumpe aus.

Wassertank

Integriert oder Schlauch zu extern. Vor-/Nachteile Entscheidung?

Integriert:

Vollständig portable

;schwerer, Dichtigkeitsproblem

Extern:

Bedingt portable, einfacher zu realisieren, Systembetrachtung geringer

Wasserversorgung muss am Einsatzort möglich sein, Lange Strecken und Höhen für Pumpe nicht gut.

4.3 Dreidimensionales Zielsystem

Mit Vision Schwerpunkt der Arbeit teils/ganz in HW-Realisierung?

5 Reflexion und Ausblick

Literatur

- [1] *2022 Developer Survey*. 2022. URL: <https://survey.stackoverflow.co/2022/>.
- [2] Jake Goulding. *What is Rust and why is it so popular?* 2020. URL: <https://stackoverflow.blog/2020/01/20/what-is-rust-and-why-is-it-so-popular/>.