

KATZENSCHRECK

PHYSIK UND BEWEGUNG

PROJEKTUNTERRICHT,
DEZEMBER 2017

Hans Kieninger, Paul Fouché und Zino Monsch

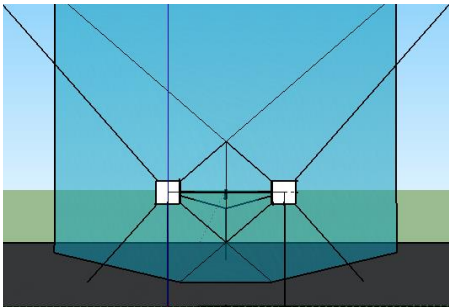
Inhaltsverzeichnis

1) Einleitung:	2
2) Berechnungen:	2 bis 7
3) Hardware:	8
4) Software:	9
a) Webseite:	10
b) Bildbearbeitung:	11
c) Position des Gegenstandes im Bild:	12
5) Fazit:	13

1. Einleitung:

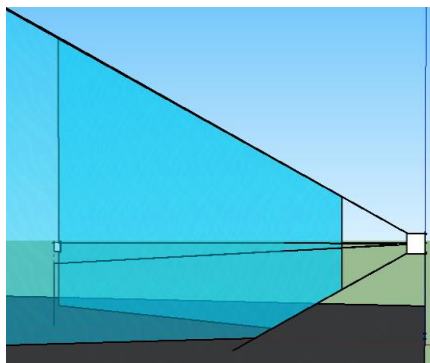
Im Projektunterricht der Kantonsschule Baden hatten wir die Idee ein Gerät zu entwickeln, welches einem Katzen vom Garten vertreiben kann. Die Grundidee bestand darin, eine Maschine zu bauen, die Katzen mit einem Wasserschlauch abspritzt. Da dies anfänglich zu ambitiös tönte, begnügten wir uns mit dem Bestimmen der Position eines farbigen Objektes (z.B. ein Ball). Allenfalls sollte das Gerät den Gegenstand mit einem Laser markieren und verfolgen. Wie das Ganze funktioniert, seht ihr im Laufe dieses Berichts.

2. Berechnungen

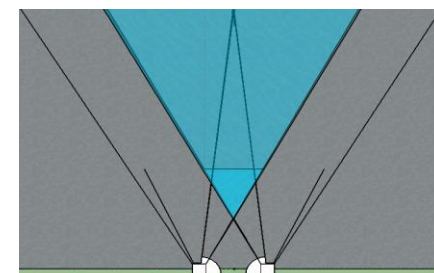


*Fig. 1:
Erste 3D-Modell auf SketchUp,
von hinter den beiden «Kameras»
betrachtet. Blaues Volumen ent-
spricht Überlappungsbereich der
beiden Bildern.*

Um die Berechnungen zur x-, y- und z-Koordinatenbestimmungen auszuführen brauchten wir am Anfang ein 3-D Modell, damit wir uns alles besser vorstellen konnten. Mit dem Konstruktionsprogramm SketchUp erstellten wir zwei gleich grosse Würfel auf gleicher Höhe, welche die Kameras repräsentieren sollten. Des weiteren stellten wir die Blickräume der Kameras mit je vier Linien dar, die sich vom Mittelpunkt der hinteren Seite des Würfels durch die vorderen Ecken ziehen. Der Überlappungsbereich dieser zwei Blickräume ist der Raum, worin sich das vordefinierte Objekt befinden sollte. Wenn es sich ausserhalb dieses Überlappungsbereichs befindet, können die x-, y- und z-Achsenabschnitte nicht bestimmt werden.



*Fig. 2:
Erste 3D-Modell auf SketchUp, von
der linken Seite aus betrachtet.
Blaues Volumen entspricht
Überlappungsbereich der beiden
Bildern.*



*Fig. 3:
Erste 3D-Modell auf
SketchUp, von oben be-
trachtet. Blaue Volumen
entspricht Überlappungs-
bereich der beiden Bildern.*

Wenn die Kameras stark zueinander geneigt sind, würde der Überlappungsbereich erstens nach einer kurzen Distanz nicht mehr vorhanden sein, da es einen abgeschlossenen Raum bildet, und zweitens wäre er sehr schmal. Das heisst der Raum müsste dem Objekt sehr präzise und schnell folgen können.

Folgende Skizzen stellen unsere Überlegungen zur Berechnung der Position des Gegenstandes im Raum abhängig von der Position in den Bildern dar.

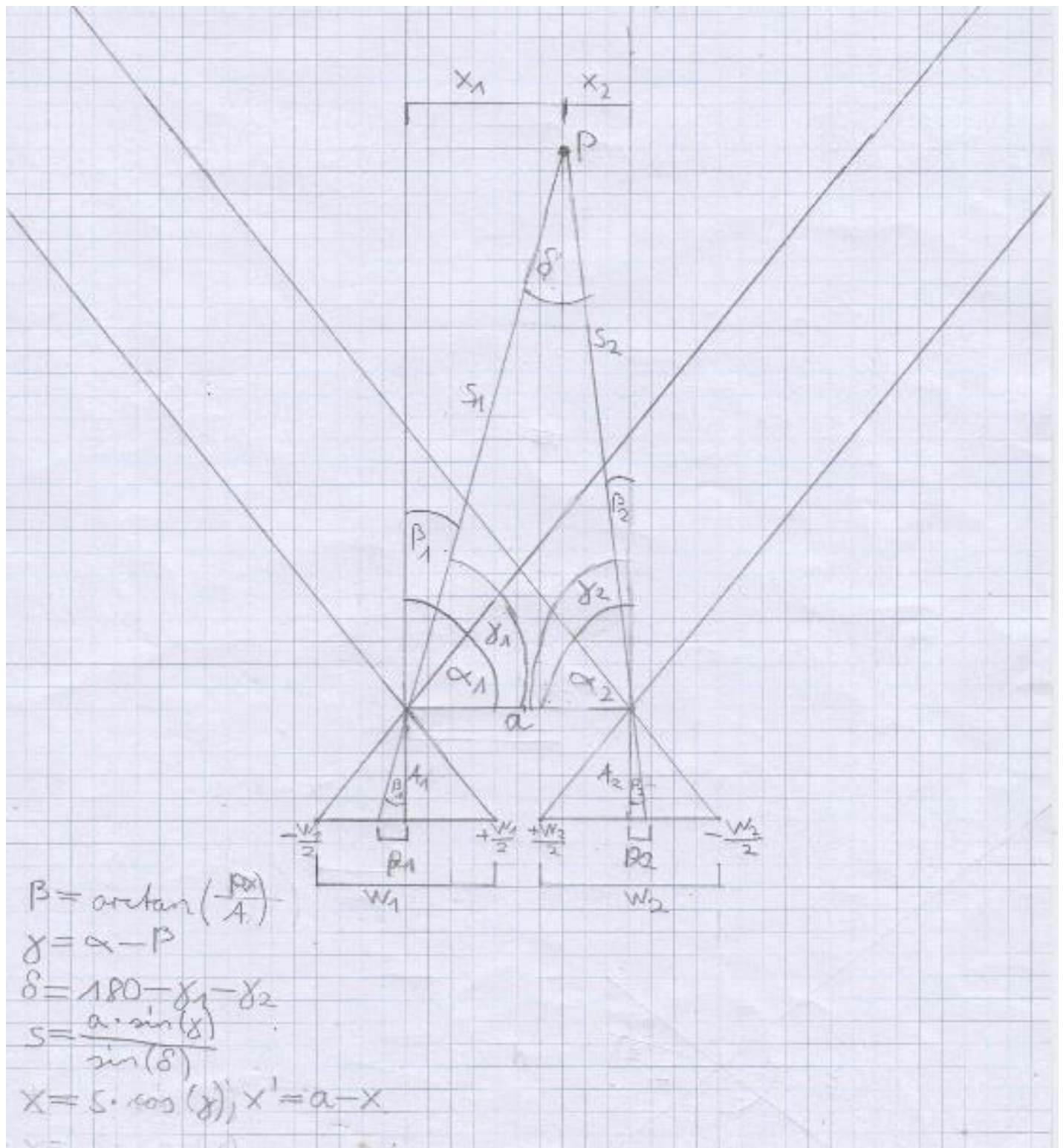


Fig. 4:
Skizze des Systems von oben.

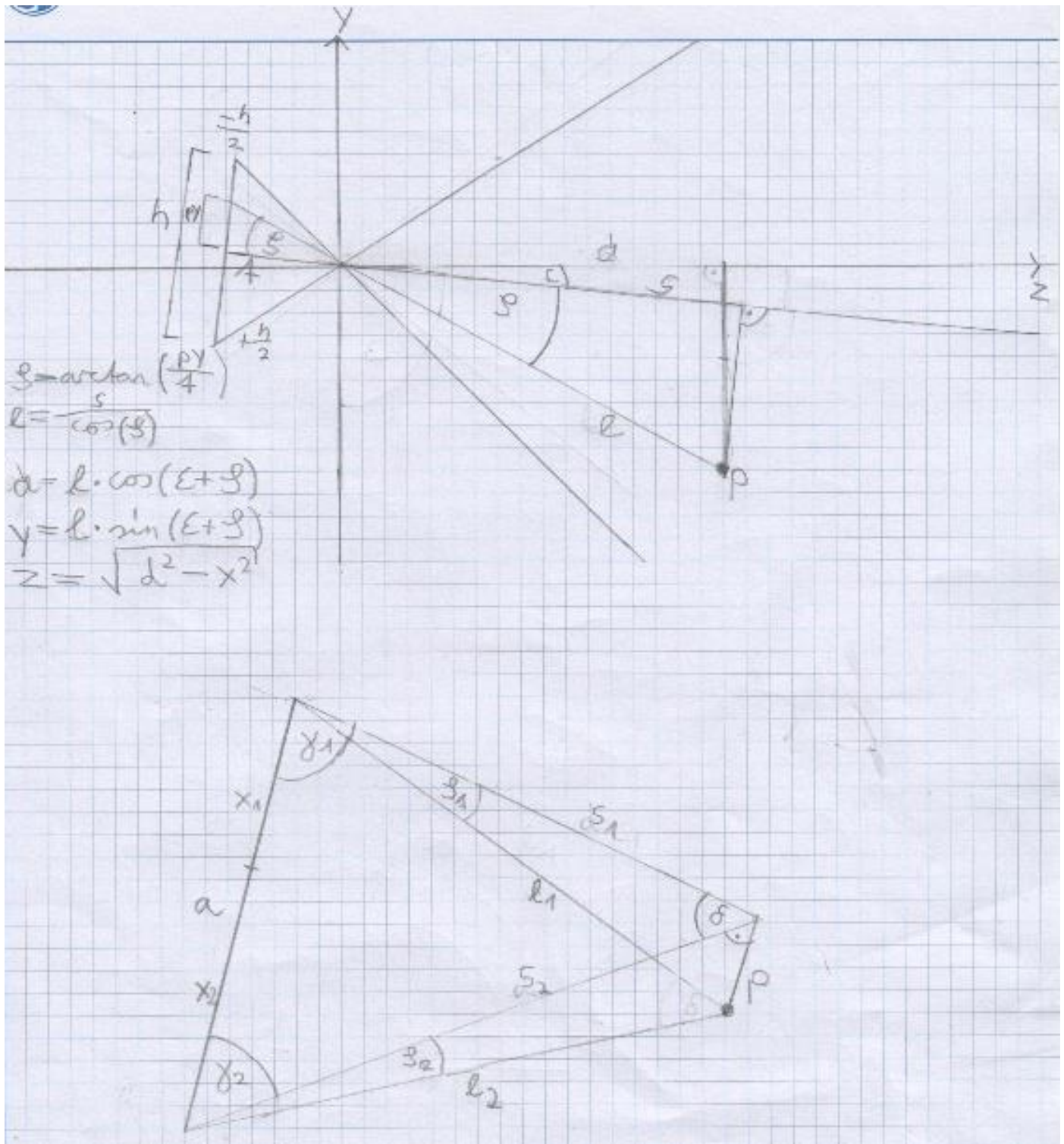


Fig. 5:
Skizze von der Seite.

$$f(p_{x1}, p_{x2}, p_{y1}) \rightarrow \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

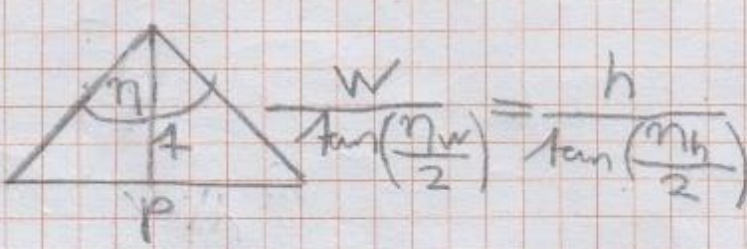
- 1.) $\gamma_1 = \alpha_1 - \arctan\left(\frac{p_{x1}}{A_1}\right)$
 $\gamma_2 = \alpha_2 - \arctan\left(\frac{p_{x2}}{A_2}\right)$
- 2.) $\delta = 180 - (\gamma_1 + \gamma_2)$
- 3.) $s_1 = \frac{a \cdot \sin(\gamma_2)}{\sin(\delta)}$
- 4.) $x_1 = s_1 \cdot \cos(\gamma_1)$
- 5.) $\beta_1 = \arctan\left(\frac{p_{y1}}{A_1}\right)$
- 6.) $\epsilon_1 = \frac{s_1}{\cos(\beta_1)}$
- 7.) $y = \epsilon_1 \cdot \sin(\epsilon_1 + \beta_1)$
- 8.) $z = \sqrt{(\epsilon_1 \cdot \cos(\epsilon_1 + \beta_1))^2 - x_1^2}$

Parameter

A_1, A_2 : über "Bildwinkel" bestimmbar
 $\alpha_1, \alpha_2, \epsilon_1$: über "kalibrieren" zu bestimmen

Bildwinkel: η

$$\frac{p}{2A} = \tan\left(\frac{\eta}{2}\right)$$

$$A = \frac{p}{2 \tan\left(\frac{\eta}{2}\right)}$$


$$\frac{W}{\tan\left(\frac{\eta}{2}\right)} = \frac{h}{\tan\left(\frac{\eta}{2}\right)}$$

Fig. 6:
Gleichungssystem für
Positionsbestimmung

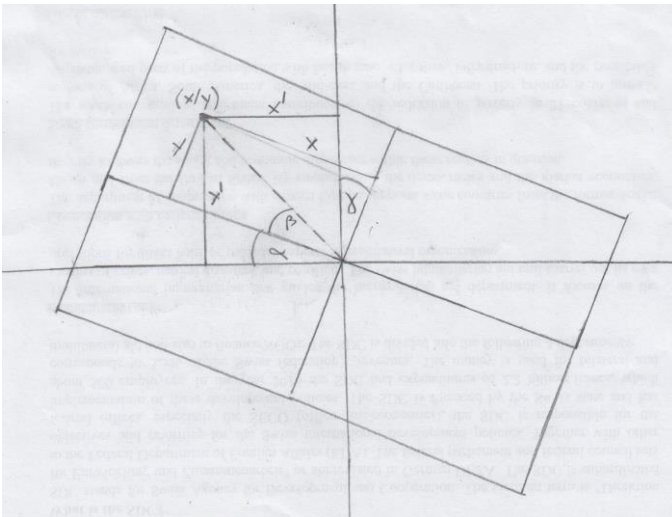


Fig. 7: Berücksichtigung von θ_1 und θ_2 , in diesem Fall α genannt.

Wir haben uns eine Art Funktion ausgedacht, die aus der Position des Gegenstandes in zwei Bildern, die räumlichen Koordinaten des Gegenstandes berechnen kann. In einem ersten Schritt müssen wir die Neigung relativ zur x-z-Ebene beachten. Als können die Winkel berechnen, indem die Lichtstrahlen in die Kamera einfallen. Nun können wir anhand der Trigonometrie und dem Abstand zwischen den zwei Kameras die x-, y- und z-Koordinaten berechnen. Die Neigung relativ zur x-y- und y-z-Ebene müssen auch berücksichtigt werden. Der Witz an der Sache ist, dass man aus einem Bild nur zwei Winkel relativ zum Objekt

herausfinden kann und anhand eines zweiten Bildes, das zwei weitere Winkel liefert, die 3-D Position errechnen kann.

Parameter

Folgende Parameter müssen in der Berechnung berücksichtigt werden.

Name	Beschreibung	Bestimmung	Optimaler Wert
a	Abstand zwischen den zwei Kameras	Messung	220 [mm]
t_1, t_2	Art Brennweite der Kameras in Pixel (abhängig vom Bildwinkel)	Messung	Gemessener Wert: 1850 und 1889 Pixel
ε	Winkel der linken Kamera, relativ zur x-y-Ebene	Gleichungssystem, kalibrieren	0°
α_1 und α_2	Winkel der Kameras relativ zur y-z-Ebene	Gleichungssystem, kalibrieren	90°
θ_1 und θ_2	Winkel relativ zur x-z-Ebene	Messung	0°

Für die Kalibrierung haben wir 12 Messungen gemacht. Mit der Methode der kleinsten Fehler Quadrate konnten die Werte bestimmt werden. Eine andere Möglichkeit wäre, ein Gleichungssystem mit drei unbekannten Parametern zu erstellen.

Genauigkeit

Je weiter weg der Gegenstand sich von den Kameras befindet, desto ungenauer werden die Koordinatenangaben. Ausserdem wird der Gegenstand auf grosser Distanz zu klein, so dass die Kameras ihn nicht mehr erfassen können. Während die Verschiebung um ein Pixel bei zwei Metern Distanz ungefähr zwei Zentimetern in der z-Richtung bedeuten, sind es bei acht Metern fast 35 Zentimetern.

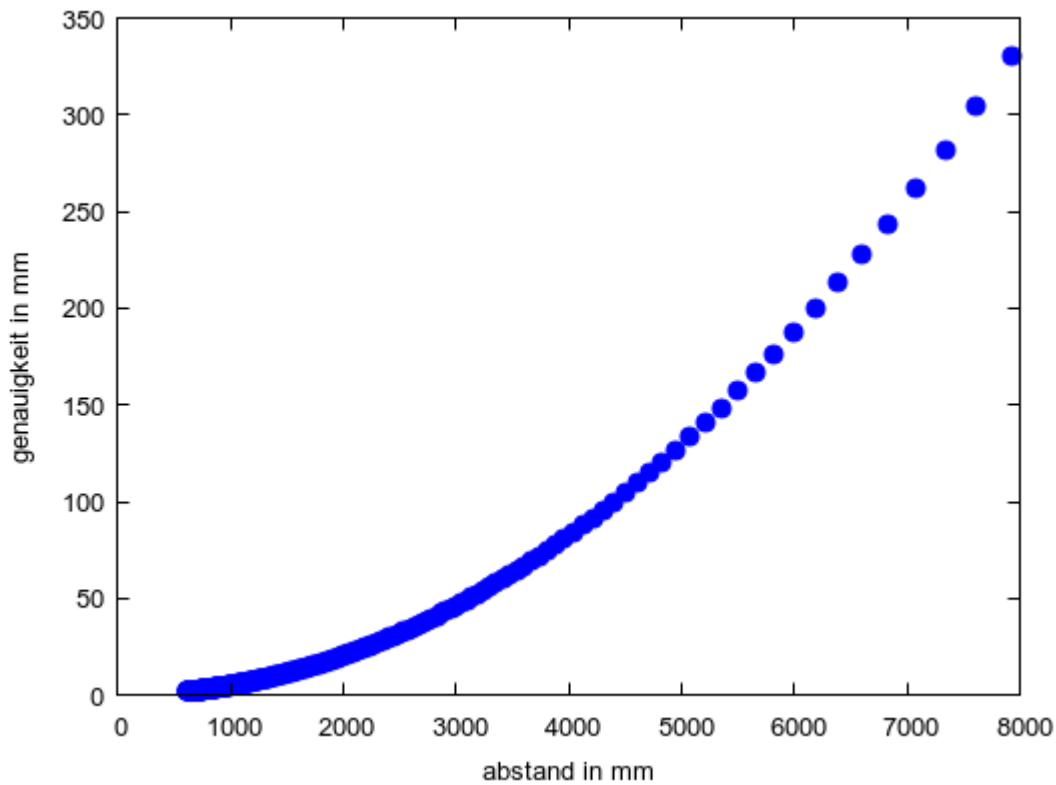


Fig. 8: Maximale Genauigkeit der z-Koordinate in Abhängigkeit des Abstands

3. Hardware

Das ganze System ist in vier verschiedene Bereiche einzuteilen. Der oberste Bereich besteht aus einem Stück Holz, welches 25cm lang und 5cm breit ist. Auf der linken und der rechten Seite ist je eine Kamera montiert. Die Kameras sind parallel zu einander und schauen in die gleiche Richtung. Leicht links von der Mitte ist ein kleines Stück Holz befestigt, worin sich einen Schrittmotor befindet. Dieser Schrittmotor ist mit einem anderen kleinen Holzstück verbunden, welches einen Laser enthält. Der Schrittmotor ist dafür verantwortlich, den Laser um die horizontale Achse zu drehen. Der Laser befindet sich möglichst genau in der Mitte der grossen Holzplatte und schaut auch in die Richtung der Kameras.

Der mittlere Bereich setzt sich folgendermassen zusammen: Ein Eisenstab ist an der unteren Seite in der Mitte der Holzplatte befestigt. Dieser Eisenstab wird mit zwei Kugellagern an einer anderen Holzplatte festgehalten. Die Kugellager erlauben, dass sich die obere Holzplatte um die vertikale Achse drehen kann.

Der unterste Bereich ist eine grosse und dicke Holzplatte, welche für die Stabilisation des gesamten Systems verantwortlich ist. In der Mitte dieser grossen Holzplatte ist ein zweiter Schrittmotor versenkt. Der Schrittmotor ist mit der Eisenstange des mittleren Bereiches verbunden. Somit kann die Drehung der oberen Holzplatte gesteuert werden. Ein Problem ist, dass die Schrittmotoren viel Spiel haben. Deswegen erreichen wir mit dem Laser keine hohe Präzision.

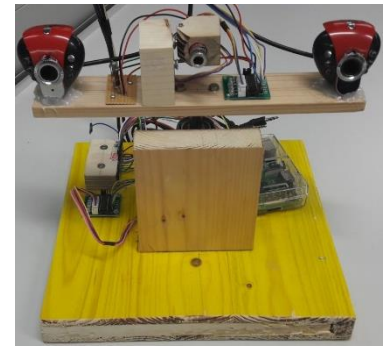


Abbildung 1 : Die Vorderseite des Roboters.

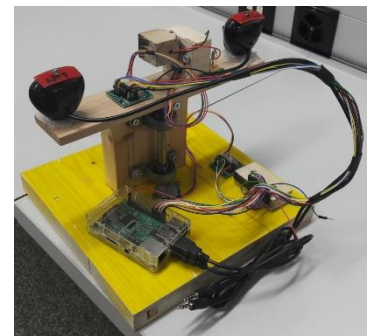
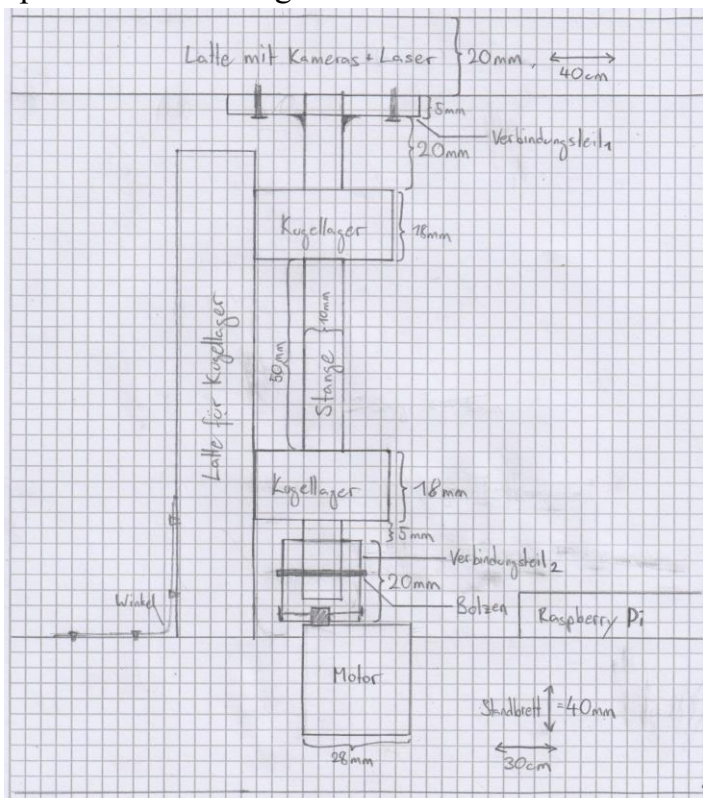


Abbildung 2 : Die Hinterseite des Roboters.



Der letzte Bereich ist die gesamte Elektronik der Kameras, des Lasers und den zwei Schrittmotoren, welche mit einem RaspberryPi verbunden sind. Der Laser wird über einem Transistor gesteuert, da der RaspberryPi nicht ausreichend Strom liefern kann.

Das Holz und die Kabel für die Elektronik erhielten wir von der Schule. Die Kameras, den Laser, die Schrittmotoren, die Eisenstange, die Kugellager, die Circuit-boards und den RaspberryPi besorgten wir selber. Die Gesamtkosten, inklusive dem Raspberry Pi, würden etwa 70 Franken betragen.

Fig. 9: Skizze des geplanten Prototyps.
(Weicht vom Endprodukt leicht ab)

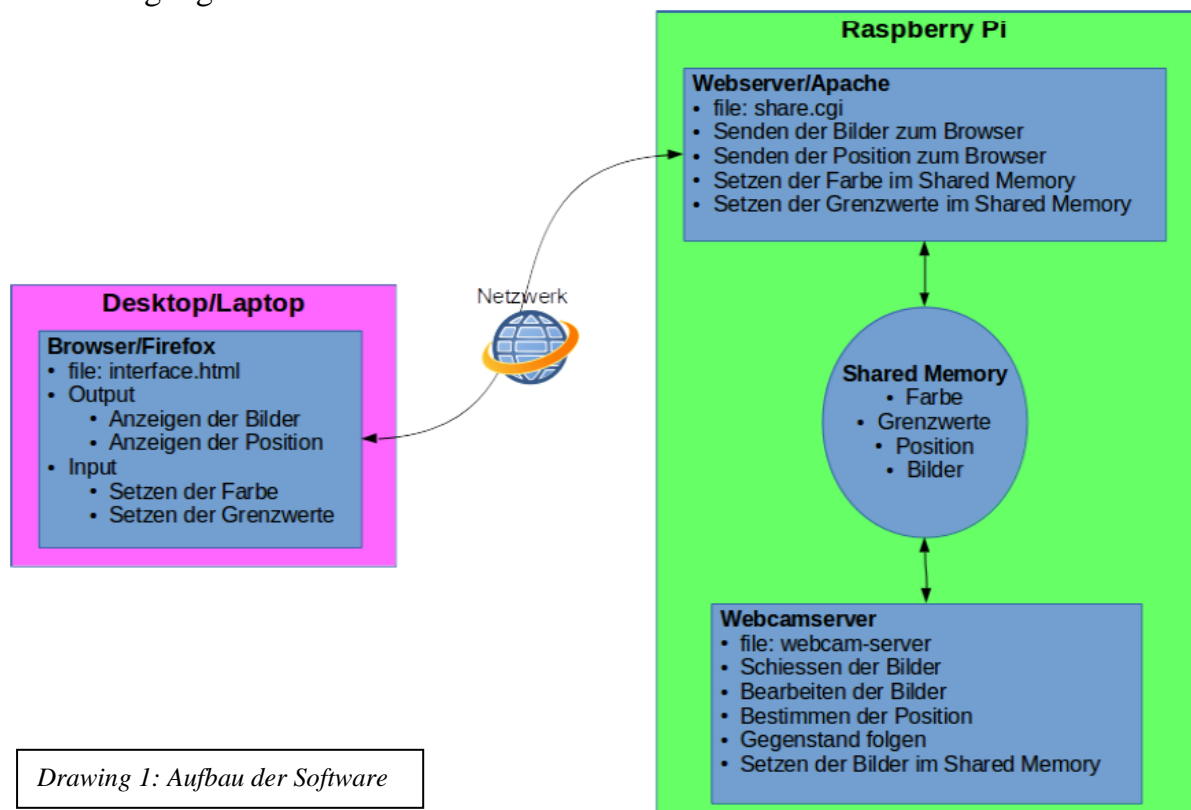
4. Software

Die Software implementiert die Überlegungen zur Berechnung der Position und enthält eine grafische Benutzeroberfläche. Die Oberfläche wurde als Webseite realisiert und läuft dadurch auf allen Betriebssystemen mit einem Webbrowser. Dies bedeutet, dass das Gerät nicht nur von Desktop und Laptops aus gesteuert werden kann, sondern auch vom Smartphone oder Tablet.

Aufbau der Software

Die Software ist eine Server-Client Anwendung und kann in folgende 3 Teile eingegliedert werden.

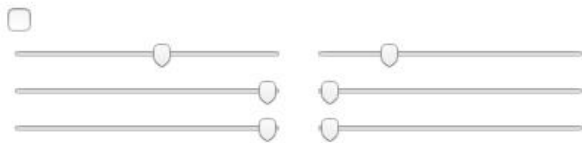
1. **Browser + Webseite:** Die Webseite läuft auf dem Computer des Benutzers. Sie zeigt die Resultate der Berechnungen und die Bilder der Webcams an. Des weiteren dient sie zum Auswählen des Gegenstandes.
2. **Webserver + CGI-Programm:** Der Webserver befindet sich auf dem RaspberryPi. Er liefert in einem ersten Schritt dem Browser des Benutzers die Webseite aus. Anschließend fordert die Webseite in regelmässigen Abständen die Resultate und die Bilder an. Gleichzeitig sendet die Webseite dem Webserver die Eingaben des Benutzers. Die angeforderten Daten werden aus einem Shared Memory Segment gelesen und der Webseite über das Netzwerk zugeschickt. Die Eingaben des Benutzers werden in das Shared Memory Segment geschrieben.
3. **Webcamserver:** Der Webcamserver ist für das Schiessen der Bilder, das Bearbeiten der Bilder, die Bestimmung der Position des Gegenstandes und dem Folgen des Gegenstandes zuständig. Er liest die Information über die Farbe des Gegenstandes aus dem Shared Memory Segment und stellt die Bilder und Resultate in diesem dem Webserver zur Verfügung.



Drawing 1: Aufbau der Software

a. Webseite

Die Webseite dient der Ein- und Ausgabe für das Programm. Ein Ankreuzfeld erlaubt das Folgen des Gegenstandes ein- und auszuschalten. Die 6 Schieberegler nützen dem Selektieren des Gegenstandes durch setzen der Wertebereiche für die Farbkomponenten (hue, saturation, value). Darunter befindet sich eine Anzeige für die Koordinaten des Gegenstandes im Raum. Die Werte werden in Millimeter angegeben und die Position des Lasers entspricht dem Origo. Ausserdem werden die Bilder der Webcams und ihre bearbeiteten Versionen angezeigt. In den bearbeiteten Versionen wird der Gegenstand violett mit weissem Rand markiert. Die blauen Flecken mit den grünen Rändern entsprechen zwar dem ausgewählten Wertebereich werden aber nicht zum Gegenstand hinzugerechnet.



x: 3 / y:-190 / z: 2142

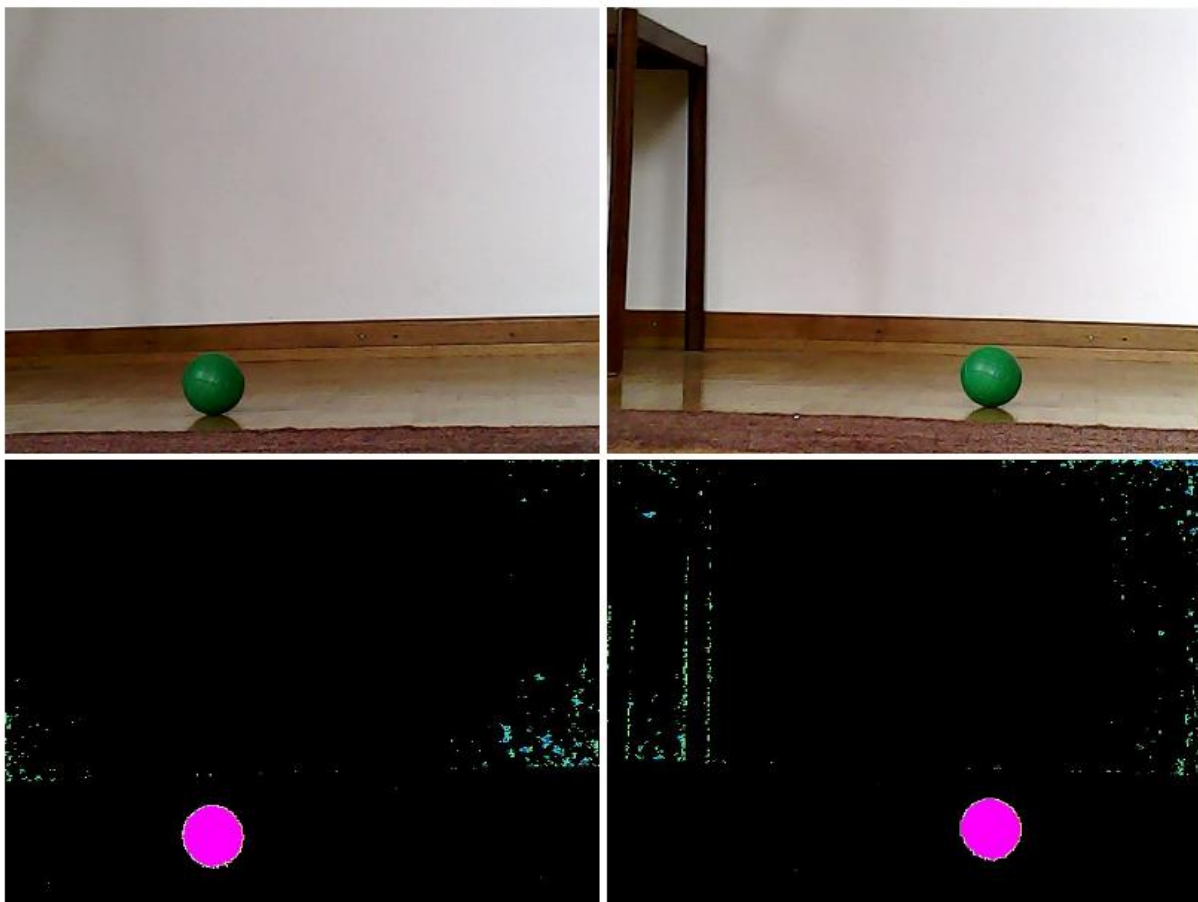
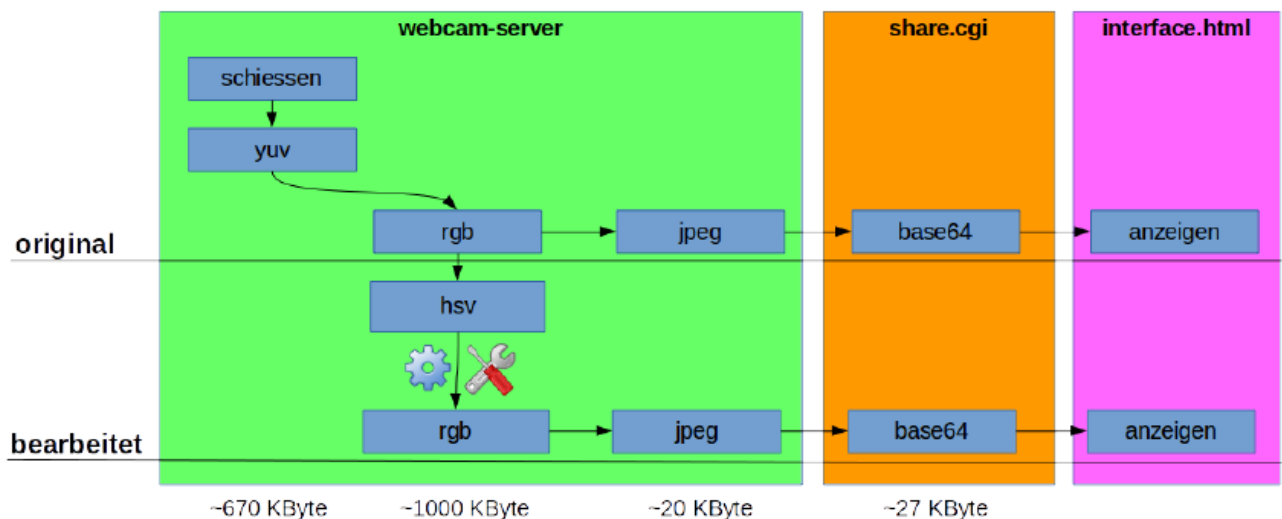


Illustration 1: Screenshot der Webseite

b. Bildbearbeitung

Damit die Berechnungen für die Position eines Gegenstandes im Raum stimmen, ist es zentral, dass die Position des Gegenstandes im Bild präzise angegeben werden kann. Dafür muss das Bild in verschiedene Farbformate umgerechnet werden. Die Webcams liefern die Bilder im YUV(4:2:2)-Format¹. Dieses Farbmodell nutzt die Eigenschaft aus, dass das menschliche Auge Farbe in einer tieferen Auflösung als Helligkeit wahrnimmt. Deswegen teilen sich in diesem Format jeweils zwei Pixel die Farbinformation, während sie eine eigene Helligkeitsinformation besitzen. Das Bild wird dann in einem ersten Schritt ins RGB-Format umgerechnet. Eine Kopie davon wird als JPEG komprimiert im Shared Memory Segment abgelegt. Um zu bestimmen ob ein Pixel Teil des Gegenstandes ist, wird das Bild in das HSV-Format¹ umgewandelt. Der HSV-Farbraum ist für den Menschen intuitiv, was das setzen des Wertebereichs erleichtert. Das Bild wird dann weiterbearbeitet. Als Resultat kristallisiert sich die Position des Gegenstandes im Bild und ein Bild im RGB-Format mit dem markierten Gegenstand heraus. Das resultierende Bild wird ebenfalls zu einem JPEG komprimiert und in das Shared Memory Segment geschrieben.



Drawing 2: Bildbearbeitungs-Prozess

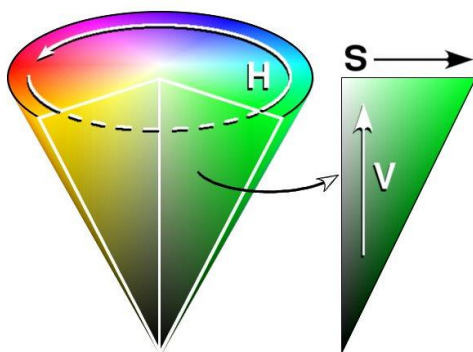


Illustration 2: HSV-Farbraum,
Quelle: <https://de.wikipedia.org/wiki/HSV-Farbraum>

¹ YCbCr wäre eigentlich die richtige Bezeichnung, YUV bezeichnet normalerweise ein ähnliches Farbmodell für das analoge Farbfernsehen

c. Position des Gegenstandes im Bild

Nachdem das Bild in das HSV-Format umgewandelt wurde, iteriert das Programm über alle Pixel. Die Pixel, welchen dem Wertebereich (Grenzwerte) entsprechen, werden markiert. Nun könnte bereits der Durchschnitt der x- und y-Koordinate aller markierten Pixel berechnet und als Position genommen werden. Das Problem bei dieser Lösung ist, dass bei suboptimalen Hintergrund sich ein Rauschen im Bild befindet, das die Position verfälscht. Aus diesem Grund nehmen wir den Durchschnitt der Koordinaten vom grössten System zusammenhängender, markierten Pixel. Um das grösste System zusammenhängender Pixel zu finden, folgt der Algorithmus die Umrisse im Bild und vergleicht die zugehörigen Flächen. Die grösste Fläche wird violett markiert und die anderen blau. Schlussendlich wird überprüft, ob die Koordinaten stimmen können. Die Grösse der Fläche muss in einem gewissen Bereich liegen und soll im linken und rechten Bild ähnlich sein. Ausserdem sind im Idealfall die y-Koordinaten in beiden Bildern gleich.

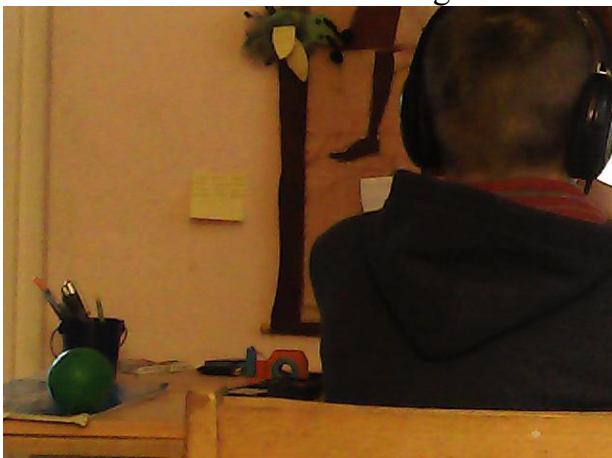


Illustration 3: originales Bild mit suboptimalem Hintergrund

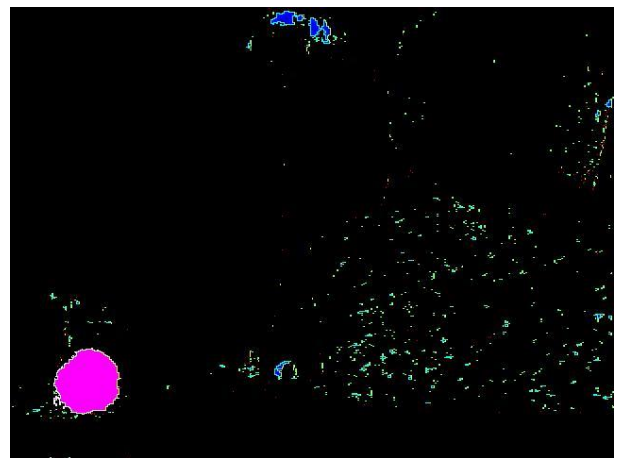


Illustration 4: bearbeitetes Bild, blau: Rauschen, violett: Gegenstand

Verbesserungsmöglichkeiten

Die Umrechnung von RGB zu HSV und das Finden des Umrisses mit der grössten zugehörigen Fläche ist rechenaufwendig. Da der RaspberryPi 3 über 4 schwache Rechenkerne (ARM Cortex-A53, 1.2GHz) verfügt und nur einer ausgenützt wird, erreicht das Programm ein Drittel der möglichen Anzahl an Bildern pro Sekunde. Durch das alleinige Auswerten von Bildbereichen in denen man den Gegenstand erwartet, könnte der Rechenaufwand minimiert werden. Ausserdem wäre es möglich das linke und rechte Bild parallel in verschiedenen Threads zu bearbeiten und somit zwei Rechenkerne statt einen auszunützen.

Des weiteren sollte der Schrittmotor, der die vertikale Komponente des Laser steuert, anhand der Bilder der Kameras kalibriert werden.

5. Fazit

Am Ende des Projektes haben wir folgende Ziele erreicht:

1. Position eines Gegenstandes auf zwei Meter Distanz mit ungefähr 5 Zentimeter Genauigkeit angeben.
2. Durch eine Drehung dem Gegenstand im Horizontalen folgen. Bei zu grosser Geschwindigkeit ist das Gerät nicht schnell genug.
3. Den Gegenstand anhand eines graphischen Benutzerinterfaces von einem beliebigen netzwerkfähigen Gerät auszuwählen.
4. Das Gerät kann den Gegenstand mit dem Laser anvisieren. Falls der Gegenstand zu klein ist, kann der Laserpunkt leicht danebenliegen.

Die Grundidee, eine Katze mit einem Wasserstrahl abzuschliessen, wurde wie erwartet nicht erreicht. Dazu müsste ein Algorithmus ausgedacht werden, der eine Katze von einem anderen Tier unterscheidet. Ausserdem bräuchten wir eine Vorrichtung, um einen Wasserstrahl zu steuern. Um eine höhere Präzision zu erreichen, müsste man sich bessere Kameras zulegen. Da der Wasserstrahl leicht gestreut ist, würde unsere Präzision reichen.