



Department für Informatik
Abteilung Systemsoftware und verteilte Systeme

Masterstudiengang Informatik

Masterarbeit

Erweiterung eines automatischen Systems zur
Erstellung von Steckperlenbildern um Mechanismen zur
Fehlertoleranz und Qualitätskontrolle

Marius Hacker

27. Juni 2016

1. Gutachter: Prof. Dr.-Ing. Oliver Theel
2. Gutachter: Robert Schadek

Zusammenfassung

Das Ziel der vorliegenden Masterarbeit war es, ein automatisches System zum Erstellen von Steckperlenbildern um Mechanismen der Fehlertoleranz und Qualitätskontrolle zu erweitern. Um die am System auftretenden Fehler zu identifizieren, wurde eine Fehleranalyse durchgeführt. Das Auftreten der ermittelten Fehler wurde durch das Ersetzen von Bauteilen des System gesenkt. Diese Teile wurden mit einem 3D-Drucker erstellt. Zur Qualitätssicherung wurde ein Fehlerkorrekturprozess entworfen und implementiert. Dieser erkennt falsch gesetzte Steckperlen und korrigiert sie.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Hintergrund	1
1.2	Aufgabenstellung	1
2	Grundlagen	3
2.1	Aufbau und Funktionsweise der Beadbox	3
2.2	Überblick der Softwarearchitektur	5
2.3	Verwendete Technologien	6
3	Vorgehensweise und Aufbau dieser Arbeit	9
4	Ist-Analyse	10
4.1	Fehleranalyse	10
4.1.1	Fehler beim Vereinzelmehanismus	11
4.1.2	Fehler beim Setzrohr	12
4.1.3	Bedienfehler	14
4.1.4	Fehler am Druckbild	14
4.2	Vorhandene Fehlererkennung- und Fehlerkorrekturmechanismen	15
4.3	Fehlereinordnung	16
4.4	Prioritäten für Fehlerbearbeitung	18
5	Fehler vermeiden und korrigieren	21
5.1	Fertigungsgenauigkeit	21
5.2	Mechanische Verbesserungen	24
5.2.1	Schieber	25
5.2.2	Setzrohr	29
5.2.3	Perlenmagazin	34
5.3	Fehlerkorrektur	35
5.3.1	Planung	36
5.3.2	Kamerahalterung	36
5.3.3	Die Kamera	38
5.3.4	Das Bild ausrichten	40
5.3.5	Farberkennung	43
5.3.6	Korrekturdruck	57
5.3.7	Finaler Ablauf des Korrekturprozesses	58

6	Erweiterung der Architektur	61
6.1	Software zur Kamerakalibrierung	62
6.2	Fehlererkennung	62
6.2.1	Die Klasse bbqualitycontrol	62
6.2.2	Die Klasse bbimagetransformation	64
6.2.3	Die Klasse bbcolorfinder	64
6.3	Fehlerkorrekturdruck	65
6.4	Erweiterung der GUI	65
7	Bewertung der Beadboxerweiterungen	66
7.1	Mechanische Verbesserungen	66
7.2	Fehlerkorrektur	67
8	Benutzerhandbuch	69
8.1	Kamera kalibrieren	69
8.2	Einstellen des Beadbox-Ports	70
8.3	Druck mit Fehlerkorrektur	70
9	Fazit und Ausblick	73
A	Testbilder	75
B	Testläufe	76
C	Farberkennung Tests	79
D	Changelog der Modelle	82

Abbildungsverzeichnis

1	Übersicht Beadbox [Zie13]	3
2	Vereinzelungsmechanismus nach [Zie13]	5
3	Softwarearchitektur [Zie13]	6
4	Fehler V1	11
5	Fehler V2	12
6	Fehler S1	13
7	Fehler S2	13
8	Dimensionen einer Perle	22
9	Bügelperlen der Firma Nabbi	24
10	Stopper am alten Schieber	26
11	Schieber V 0.1	27
12	Fehler V1 Lösungsversuch 1	27
13	Neu auftretender Fehler	28
14	Neu auftretender Fehler: Problem	29
15	Neu auftretender Fehler: Lösung	29
16	Setzrohr Version 0.1	30
17	Fehler V1 am Setzrohr	31
18	Modulare Version vom Setzrohr	32
19	Fehler S1: Lösungsversuch	33
20	Unterlegscheibe	35
21	Ermittlung der Höhe der Unterlegscheibe	35
22	Ablaufdiagramm Fehlerkorrektur	37
23	Metallstreifen für Kamerahalterung	37
24	Adapter für Kamerahalterung	38
25	Fertige Kamerahalterung	39
26	Kamerakalibrierung	40
27	Kamerabild	40
28	Schablone	42
29	Transformationen	43
30	Kamerabild Ausschnitt	44
31	Checker Shadow Illusion [EH]	45
32	Kamerabild zugeschnitten	45
33	Durchschnittsfarbe für ein Perlenfeld	47
34	Pixel-Karte	48
35	Farberkennung	48
36	Farbpalette	49

37	Farberkennung mit zwei Beispielfarben	50
38	Verktordistanz mit Beleuchtung	51
39	Rundes Raster	52
40	Farberkennung anhand zwei Beispielfarben	53
41	K-means anhand einer Perle	53
42	K-means	54
43	K-Means mit zwei Beispielfarben	54
44	K-Means mit Referenzfarben	56
45	K-Means mit verbesserten Referenzfarben	56
46	Angepasste Farbpalette	57
47	Reduzierte Farbpalette	57
48	Ablaufdiagramm Fehlerkorrekturprozess	60
49	Architektur [Zie13]	61
50	GUI: Fehlerkorrektur Auswahl	70
51	GUI: Kamera ausrichten	71
52	GUI: Perlen entfernen	72
53	Testbilder	75
54	Farberkennung Test 1	79
55	Farberkennung Test 2	79
56	Farberkennung Test 3	80
57	Farberkennung Test 4	80
58	Farberkennung Test 5	80
59	Farberkennung Test 6	81
60	Changelog Schieber	82
61	Changelog Setzrohr	83
62	Changelog Unterlegscheibe	83

Glossar

Arduino Mikrocontroller-Board mit digitalen und analogen Eingängen. 3

BSD-Liezens Open-Source Lizenzmodell. 6

CAD Software Die englische Abkürzung CAD steht für *computer-aided design*. Also ist eine CAD-Software eine Software zum rechnerunterstützten Konstruieren. 7

Raspberry Pi Ein Einplatinencomputer mit ARM-Architektur auf dem ein Linux-Betriebssystem läuft. 3

STL-Datei Eine STL-Datei ist eine Datei, in der die Beschreibung eines dreidimensionalen Objekts gespeichert ist. Die *Standard Tessellation Language* kurz STL ist eine Beschreibungssprache für dreidimensionale Objekte, die als Standard in vielen CAD-Anwendungen Verwendung findet. 25

1 Einleitung

Im Rahmen seiner Bachelorarbeit „BeadBox: Entwicklung eines informatischen Systems zur automatischen Erstellung von Bügelperlenbildern“ [Zie13] hat Herr Ziegler ein System zum automatischen Setzen von Bügelperlenbildern entwickelt. Die vorliegende Arbeit beschäftigt sich mit der Erweiterung dieses Systems um Mechanismen zur Fehlertoleranz und Qualitätskontrolle. Im Folgenden wird ein kurzer Einblick in den Hintergrund der Arbeit gegeben und anschließend die Aufgabenstellung dargelegt.

1.1 Hintergrund

Bügelperlen werden vielen aus ihrer Kindheit bekannt sein. Es sind kleine farbige Plastikröllchen, die auf einem Stiftbrett zu Bildern zusammengesetzt werden. Anschließend werden sie mit Backpapier bedeckt und mit einem Bügeleisen verschmolzen. Das so entstandene Bügelperlenbild kann vom Stiftbrett gelöst werden und dient als Dekoration oder Untersetzer für Tee- oder Kaffeekannen. Das Setzen eines Steckperlenbilds soll bei Kindern ab drei Jahren die motorischen Fähigkeiten verbessern und die Kreativität fördern.

Das von Herrn Ziegler entwickelte System, die Beadbox, ist in der Lage diese Steckperlenbilder automatisch zu setzen. „Der Name *BeabBox* setzt sich aus dem englischen Wort für Bügelperlen *bead* und dem englischen Wort für Kasten/Gehäuse *box* zusammen.“ [Zie13, S. 1] Das System bekommt ein Bild als Eingabe. Da die Auflösung eines Bügelperlenbildes nicht besonders hoch ist - ein 25 mal 25 Perlen großes Bild hat die Dimensionen von 12,5 mal 12,5 Zentimetern - muss das Bild vorher skaliert werden. Außerdem kann die Beadbox nur mit maximal 14 verschiedenen Farben drucken, da sie nur 14 Perlenmagazine besitzt und es die Bügelperlen auch nicht in jeder Farbe zu kaufen gibt. Wurde auch der Farbraum des ursprünglichen Bildes auf diese 14 Farben angepasst, so kann die Beadbox das Bild setzen. Das System ist als Demonstrator für z.B. Schülerinformationstagen gedacht und soll dort bei Schülern das Interesse an der Informatik wecken.

1.2 Aufgabenstellung

Obwohl die BeadBox bereits über Fehlerkorrekturmechanismen verfügt, kommt es beim Setzen der Perlen häufig zu Fehlern. Die meisten Druckvorgänge können ohne das Eingreifen eines Menschen nicht korrekt beendet werden. Das

Ziel dieser Arbeit ist es, das vorliegende System um Fehlertoleranzkonzepte sowie Schritte der Qualitätskontrolle zu erweitern. Durch den Einsatz von Fehlertoleranzmechanismen soll ein weitestgehend fehlerfreier Druckvorgang der Beadbox erreicht werden. Beim Setzen von Bügelperlenbilder auftretende Fehler sollen identifiziert werden. Ihr Auftreten soll z.B. durch Austauschen und Verbessern der vorhandenen Hardware verringert werden. Durch Mechanismen zu Qualitätskontrolle soll ein weitestgehend fehlerfreies Druckergebnis erzielt werden. So kann z.B. nach Beenden des Drucks das Bild mit einer Kamera auf falsch gesetzte oder fehlende Perlen überprüft werden, um diese anschließend gezielt zu korrigieren.

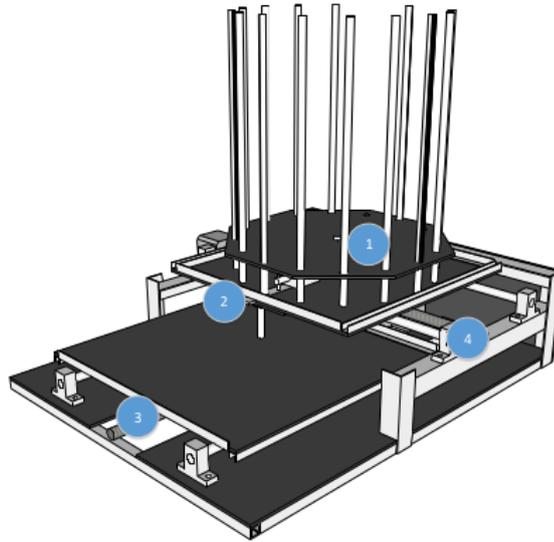


Abbildung 1: Übersicht Beadbox [Zie13]

2 Grundlagen

In diesem Kapitel wird zunächst die Funktionsweise und der Aufbau der Beadbox kurz beschrieben. Es werden alle für diese Arbeit wichtigen Bauteile benannt und ihre Funktionsweise erläutert. Anschließend wird ein Überblick über die Softwarearchitektur gegeben. Zuletzt werden in diesem Kapitel die verwendeten Technologien beschrieben. Die Technologien, die schon für die Entwicklung der Beadbox verwendet wurden, werden nur kurz beschrieben. Eine ausführlichere Beschreibung findet sich in [Zie13, S. 8f].

2.1 Aufbau und Funktionsweise der Beadbox

Die Beadbox erzeugt ein Bügelperlenbild, indem sie Bügelperlen auf einem Stiftbrett platziert. Die zylinderförmigen Perlen haben in der Mitte ein Loch, sodass sie auf die senkrecht auf dem Stiftbrett stehenden Stifte platziert werden können. In Abbildung 1 ist eine Übersicht der Beadbox zu sehen. Das Stiftbrett wird auf der Ablagefläche (3) platziert. Die zu platzierenden Perlen werden in die Perlenmagazine (1) gefüllt. Diese senkrecht angebrachten Röhren beinhalten jeweils Perlen einer Farbe.

Die Beadbox benutzt einen druckerähnlichen Mechanismus, um eine Perle zu platzieren. Wie beim Drucker der Druckkopf, so wird bei der Beadbox das Setzrohr auf der einen Achse bewegt. Durch das Setzrohr werden die Bügelperlen auf das Stiftbrett platziert. Die andere Achse wird beim Drucker durch Bewegung des Blattes verschoben. Bei der Beadbox wird ebenfalls die

zu bedruckende Fläche bewegt. In diesem Fall das Stiftbrett. So ist es möglich das Setzrohr über jedem Stift des Stiftbretts zu platzieren. Die Positionierung wird mithilfe von Schrittmotoren realisiert. Diese drehen jeweils eine Welle, auf der die Arbeitsfläche bzw. das Setzrohr bewegt wird. Das Perlenmagazin wird mithilfe eines Getriebemotors gedreht. Am unteren Ende des Setzrohrs ist eine Lichtschranke angebracht, mit der überprüft wird, ob eine Bügelperle gesetzt wurde. Für die Steuerung dieser Komponenten ist das Mikrocontroller-Board Arduino zuständig. Dieses ist mit dem Einplatinencomputer Raspberry Pi verbunden, der diesen wiederum steuert. Die Software auf dem Raspberry Pi ist für die Bildverarbeitung, die Benutzerschnittstelle und die Steuerung des Druckprozesses zuständig.

Ein Druckvorgang läuft in den folgenden Schritten ab.

1 Eingabebild anpassen Damit die Beadbox ein Perlenbild drucken kann, benötigt sie eine Vorlage, nach der sie die Perlen auf dem Stiftbrett platzieren kann. Dazu wird von der Software der Beadbox eine Bilddatei eingelesen. Das eingelesene Bild muss für den Druck angepasst werden.

2 Starten des Druckvorgangs Die folgenden Schritte werden für jede Perle wiederholt, bis das Bild komplett gedruckt wurde.

2.2 Positionierung des Setzrohrs Soll eine Perle gesetzt werden, so muss zunächst das Setzrohr über der richtigen Position des Stiftbrettes platziert werden. Die Beadbox verwendet dazu zwei Schrittmotoren, die das Setzrohr, bzw. das Stiftbrett in X- und Y-Position bewegen. Ist das Setzrohr ausgerichtet, muss als nächstes das richtige Perlenmagazin ausgewählt werden.

2.3 Auswahl der richtigen Perlenfarbe Die Beadbox ist mit 14 Perlenmagazinen ausgestattet, die jeweils ca. 60 Perlen derselben Farbe beinhalten. Die Magazine sind kreisförmig angeordnet und können durch eine Drehbewegung ausgewählt werden. Die Beadbox dreht das Perlenmagazin mit der benötigten Farbe über den Vereinzelnungsmechanismus und kann anschließend mit der Vereinzelnung der Perlen beginnen.

2.4 Vereinzelnungsmechanismus Der Vereinzelnungsmechanismus (3 in Abbildung 1) ist in Abbildung 2 detaillierter zu sehen. Der dort abgebildete Schieber hat eine Öffnung, mit der er eine Perle aus dem Perlenmagazin aufnehmen kann. Der Schieber wird bis zu dem Setzrohr nach vorne bewegt, bis die Perle

in diese hineinfällt. Ist die Perle durch das Setzrohr durchgerutscht, landet sie direkt auf dem vorgesehenen Stift auf dem Stiftbrett und der Vorgang kann mit den nächsten Perlen solange wiederholt werden, bis das Steckperlenbild fertig gesetzt wurde.

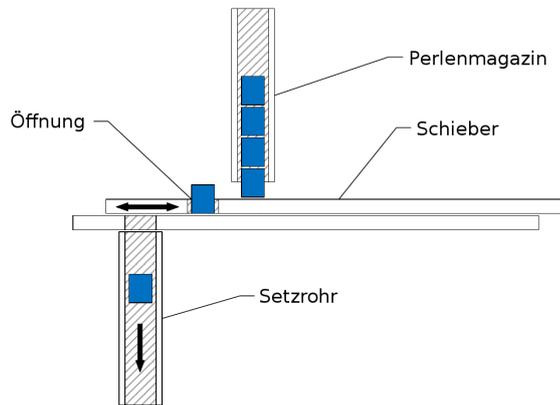


Abbildung 2: Vereinzelmehrsmechanismus nach [Zie13]

2.2 Überblick der Softwarearchitektur

Wie Herr Ziegler in seiner Arbeit [Zie13] beschreibt, besteht die Software aus zwei wesentlichen Komponenten, der Hardwarekontrollsoftware und der Steuerungssoftware. In Abbildung 3 ist der Aufbau der Softwarearchitektur dargestellt. Die Hardwarekontrollsoftware läuft auf einem Arduino und ist für die Ansteuerung der Schrittmotoren, Getriebemotoren sowie für die Abfrage der Lichtschranken zuständig. Sie kommuniziert mit der Steuerungssoftware über eine serielle Schnittstelle. Die Steuerungssoftware hingegen läuft auf einem Raspberry Pi. Sie ist in fünf Projekte unterteilt: *bbcore*, *bbcontrol*, *bbprocessing*, *bbgui* und *bbconsole*. Das Projekt *bbcore* „[...] stellt die Funktionen der Hardwareschnittstellenschicht als Bibliothek zur Verfügung.“ [Zie13, S. 21]. Das Programm *bbcontrol* „[...] bietet eine graphische Oberfläche zur direkten Steuerung der Hardware über die *bbcore*-Bibliothek.“ [Zie13, S. 21]. Die Bibliothek *bbprocessing* „[...] implementiert die Bildbearbeitungs- und Ablaufsteuerungsschicht.“ [Zie13, S. 21]. Das Projekt *bbgui* realisiert die graphische Benutzeroberfläche [Zie13, S. 21] und das Projekt *bbconsole* stellt „[...] eine einfache Konsolenanwendung zur Verfügung.“ [Zie13, S. 21].

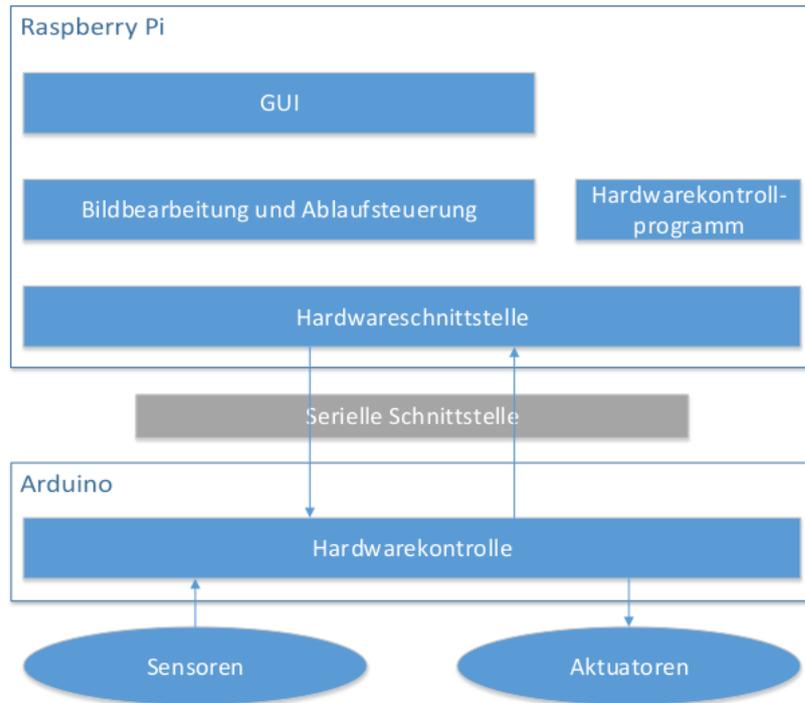


Abbildung 3: Softwarearchitektur [Zie13]

2.3 Verwendete Technologien

In diesem Abschnitt werden verwendete Technologien beschrieben. Es wird sowohl auf die Softwaretechnologien als auch auf die Hardwaretechnologien eingegangen.

QT QT ist ein C++-Framework zum Erstellen von grafischen Benutzeroberflächen. Es wurde bei der Entwicklung der Software für die Beadbox verwendet und wird auch für die Erweiterungen, die im Rahmen dieser Arbeit vorgenommen werden, benutzt. [qt2]

openCV ist eine C++-Bibliothek, die Funktionen im Bereich des maschinellen Sehens und Lernen bereitstellt. Darunter Algorithmen zum Erkennen von Gesichtern, zur Verfolgung von Kamerabewegungen, Objekterkennung und viele weitere. Wie auf ihrer Homepage [Gar] geschrieben, wird sie unter der BSD-Lizenz veröffentlicht und kann sowohl für akademische als auch kommerzielle Zwecke frei verwendet werden. *openCV* unterstützt eine Vielzahl von Betriebssystemen darunter Windows, Linux und Mac OS. Der Hersteller gibt an, dass die Bibliothek im optimierten C/C++ Code geschrieben ist und Mehrkernprozessoren unterstützt.

3D-Drucker Für diese Arbeit steht ein 3D-Drucker Ultimaker 2 der Firma *Ultimaker B.V.* zur Verfügung [B.Vb]. Wie der Hersteller angibt, kann der Ultimaker 2 Gegenstände bis zu einer Größe von 223 x 223 x 205 Millimetern drucken. Es wird eine 0,4 mm große Düse verwendet, mit der eine Schicht 20 bis 200 Mikrometer dick gedruckt werden kann. Die Verwendung eines 3D-Druckers hat gegenüber der Fertigung per Hand den Vorteil, dass die zu druckenden Objekte am Computer designt und innerhalb weniger Stunden gedruckt werden können. Durch die relativ kurze Druckzeit kann bei der Entwicklung ein iteratives Vorgehen verwendet werden, bei dem das Modell zuerst entworfen, anschließend gedruckt und zum Schluss getestet wird. Entspricht das getestete Modell nicht den Anforderungen, so wird dieser Vorgang wiederholt, bis diese erfüllt werden. Der 3D-Drucker bietet viele Einstellungsmöglichkeiten, die die Qualität des Drucks beeinflussen. Hierfür wird hauptsächlich die Schichtdicke, die Fülldicke und die Druckgeschwindigkeit des Druckers eingestellt. Die Schichtdicke gibt an, wie hoch eine gedruckte Schicht ist. Der Drucker druckt ein Objekt Schicht für Schicht beginnend mit der untersten Schicht. Um Material zu sparen und die Druckzeit zu verkürzen werden Objekte in der Regel beim Drucken nicht ausgefüllt, sodass diese von innen hohl sind. Im Inneren wird eine Gitterkonstruktion gedruckt, die dem Objekt genügend Stabilität verleiht. Die Fülldicke gibt an, wie fein und damit wie stabil diese Gitterkonstruktion ist. Als letzter Druckparameter kann die Druckgeschwindigkeit eingestellt werden. Je schneller gedruckt wird, um so ungenauer werden die einzelnen Bahnen und die Qualität des gedruckten Objekts nimmt ab.

Durch diese Einstellungsmöglichkeiten beim Drucken können bei der Entwicklung zunächst Prototypen mit geringerer Qualität gedruckt werden. Da diese Prototypen deutlich schneller gedruckt werden können, eignen sie sich, um zu überprüfen, ob das entworfene Objekt auch nach dem Drucken wie gewünscht funktioniert und aussieht.

Modellierungs Software Als Modellierungssoftware für die 3D-Modelle, die mit dem 3D-Drucker gedruckt wurden, wird die CAD Software *OpenSCAD* [Kin] verwendet. Im Gegensatz zu anderer CAD Software wird bei *OpenSCAD* das gewünschte Objekt in einer Script-Datei beschrieben. Dafür stehen einfache geometrische Objekte wie z.B. ein Würfel, ein Zylinder oder eine Kugel zur Verfügung. Ebenso wird auch die Möglichkeit gegeben, selbst Objekte mit eigener Form zu erstellen, sowie diese beliebig durch verschiedenen Operationen zu kombinieren. Ähnlich den Funktionen in Programmiersprachen gibt es

in *OpenSCAD* Module. Mit diesen können Teilobjekte erstellt und parametrisiert werden. Für diese Arbeit ist die Verwendung von *OpenSCAD* sinnvoll, da zum einen nur Objekte modelliert werden müssen, die sich aus einfachen geometrischen Formen zusammensetzen lassen und zum anderen schnell kleine Änderungen an den Objekten vorgenommen werden können.

3 Vorgehensweise und Aufbau dieser Arbeit

Da diese Arbeit die von Herrn Ziegler entwickelte Beadbox erweitern soll, war zunächst eine Einarbeitungsphase nötig, um sich mit der Hardware und Software der Beadbox auseinander zu setzen. Insbesondere die Einrichtung der Umgebung zum Erstellen der Software stellte, obwohl Herr Ziegler diese sehr gut dokumentiert hatte, eine Herausforderung dar. Dazu in Kapitel 6 mehr.

Nach der Einarbeitungsphase musste zunächst geklärt werden, welche Fehler und Probleme beim Drucken eines Bügelperlenbildes mit der Beadbox auftreten. Dazu wurde die, in Kapitel 4 beschriebene, Ist-Analyse durchgeführt. Während der Ist-Analyse wurde die Beadbox einer Fehleranalyse unterzogen. Dabei wurden Testdrucke durchgeführt und auftretende Fehler genau festgehalten. Anschließend wurden die Fehler in Kategorien eingeordnet und bewertet, welche dieser Probleme priorisiert gelöst werden sollen. Bei der Lösung der Probleme, war ein ingenieurmäßiges Vorgehen sehr wichtig. Bei dieser Vorgehensweise wurden zunächst Problemlösungen erarbeitet. Diese wurden anschließend umgesetzt. Nach der Umsetzung wurde ausführliche Tests durchgeführt, um zu überprüfen, wie erfolgreich die Lösung ist. Dieser Vorgang wird wiederholt, bis die Umsetzung den Erwartungen entspricht. Gerade bei den Problemen, die direkt mit der Hardware der Beadbox zu tun haben, war ein iteratives Vorgehen sehr wichtig. So wurde beispielsweise ein Ersatz für das Setzrohr mit dem 3D-Drucker gedruckt. Dieses wurde anschließend getestet, mögliche Schwächen ausgebessert und eine neue Version gedruckt. Welche Maßnahmen zum Beheben und korrigieren der gefundenen Fehler vorgenommen wurden, ist in Kapitel 5 beschrieben. Dort wird neben den Änderungen an der Hardware der Beadbox auch die Entwicklung eines Prozesses zur Fehlerkorrektur beschrieben. Dieser nutzt eine Kamera, um Fehler auf dem gedruckten Bügelperlenbild zu identifizieren. Identifizierte Fehler können im Anschluss mit einem Korrekturdruck behoben werden. In Kapitel 6 werden die vorgenommenen Erweiterungen an der Software beschrieben. Die Software wurde um den Fehlerkorrekturprozess erweitert. Dazu mussten Anpassungen an der GUI vorgenommen werden.

Im Kapitel 7 werden alle an der Beadbox vorgenommenen Erweiterungen bewertet. Dafür werden Testläufe durchgeführt, um die Häufigkeit der Fehler nach Verbesserung der Beadbox zu ermitteln. Die ermittelten Werte werden dann mit den in der Ist-Analyse ermittelten Werten verglichen und bewertet.

4 Ist-Analyse

Wie in der Aufgabenstellung in Kapitel 1 beschrieben, soll die Beadbox um Fehlertoleranzmechanismen erweitert werden und Probleme, die zu Fehlern beim Drucken führen sollen beseitigt werden. Dazu müssen die auftretenden Fehler zunächst ermittelt werden. Außerdem soll bestimmt werden, welche Fehlererkennungs- und Fehlerkorrekturmechanismen bereits in der Beadbox eingebaut sind. Dazu wird eine Ist-Analyse durchgeführt. Die im Rahmen dieser Analyse ermittelten Fehler, werden anschließend bewertet. Diese Bewertung dient als Entscheidungsgrundlage, welche der Fehler vorrangig behoben werden sollen.

4.1 Fehleranalyse

Bei der Fehleranalyse geht es um das Beobachten und Beschreiben von Fehlern, die beim Drucken von Bügelperlenbildern mit der Beadbox auftreten. Zunächst muss geklärt werden, was genau unter einem Fehler verstanden wird. Wenn in dieser Arbeit von einem Fehler gesprochen wird, so ist ein Zustand der Beadbox bzw. des Bügelperlenbilds gemeint, der ...

... den weiteren Druckvorgang beeinträchtigt, verzögert oder abbricht.

... zu einem falschen Bügelperlenbild führt.

Um herauszufinden welche Fehler auftreten, werden Testdrucke durchgeführt. Für diese wurden zwei Testbilder ausgewählt. Die benutzten Bilder sind in Anhang A zu sehen. Das eine Bild ist einfarbig und das andere Bild mehrfarbig. Beim Drucken des einfarbigen Testbildes wird ein Perlenmagazin durchgängig benutzt. Beim mehrfarbigen Testbild wird das Perlenmagazin hingegen häufig gewechselt. Falls beim häufigen Wechseln der Perlenmagazine andere Fehler auftreten, als bei der Nutzung nur eines Perlenmagazins, können bei den gewählten Testbildern alle diese Fehler beobachtet werden. Diese Bilder wurden nacheinander mit der Beadbox gedruckt und währenddessen wurden auftretende Fehler genau beobachtet. Im Folgenden findet sich eine ausführliche Beschreibung dieser Fehler. Die Fehler sind mit einer Kennung, die aus einem Buchstaben gefolgt von einer Zahl besteht, versehen und wurden den Komponenten zugeordnet, an denen sie auftraten oder von denen sie beeinflusst werden.

4.1.1 Fehler beim Vereinzelnmechanismus

Zum Vereinzelnmechanismus der Beadbox gehört der untere Teil des Perlenmagazins, der obere Teil des Setzrohrs, sowie der Schieber. Wie schon im Grundlagenkapitel in Abschnitt 2.1 beschrieben, nimmt der Schieber beim Vereinzeln eine Perle aus dem Perlenmagazin auf, wird mit ihr nach vorne in eine Position über dem Setzrohr bewegt und gibt diese an das Setzrohr ab. Die folgenden Fehler treten am Vereinzelnmechanismus auf.

V1 Perle steckt am Setzrohr fest Dieser Fehler tritt auf, nachdem der Schieber erfolgreich eine Perle aus dem Perlenmagazin aufgenommen und sich mit der Perle zum Setzrohr bewegt hat. Der Schieber befindet sich mit der Öffnung über der Öffnung des Setzrohrs. Die Perle ist in Schräglage und steckt im Schieber fest. Der untere Teil der Perle befindet sich bereits im Setzrohr. Dadurch, dass die Perle in Schräglage ist rutscht sie nicht, wie vorgesehen, in das Setzrohr, sondern verweilt in dieser Position. In Abbildung 4 ist eine so festsitzende Perle abgebildet.

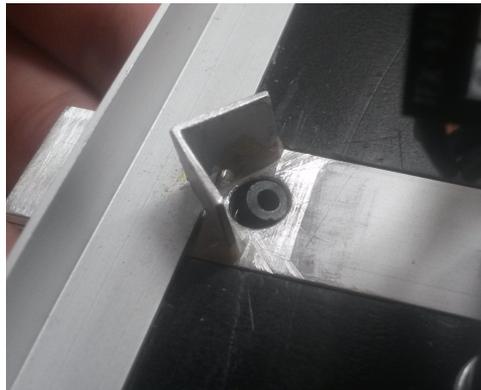


Abbildung 4: Fehler V1

V2 Blockierung des Schiebers Die Perlenmagazine sind kreisförmig an einer Rotationsscheibe angeordnet. Wird für das Drucken eine andere Farbe benötigt, so dreht sich die Rotationsscheibe und bewegt das gewünschte Magazin über den Schieber, sodass dieser aus dem Magazin die entsprechende Perlenfarbe entnehmen kann. Es wurde beobachtet, dass der Schieber nicht nach vorne bewegt werden konnte. Die Ursache hierfür ist, dass das angefahrne Perlenmagazin Kraft auf den hinteren Teil des Schiebers ausübt. Dies ist in Abbildung 5 zu sehen. Der Hubmagnet, der den Schieber nach vorne bewegen soll, hat zu wenig Kraft, um den Schieber frei zu bekommen.



Abbildung 5: Fehler V2

V3 Aufnahme einer zweiten Perle Tritt der in V1 beschriebene Fehler auf, so versucht die Software der Beadbox die festhängende Perle durch zurück- und vor-bewegen des Schiebers zu lösen. Es kann vorkommen, dass die Perle nicht so stark festklemmt und sich bereits beim Zurückfahren des Schiebers löst. Da der Schieber das Zurückfahren noch nicht beendet hat und die Perle ihn nicht mehr blockiert, wird er bis zum Perlenmagazin zurückgefahren, wo er eine weitere Perle aufnimmt. Diese gelangt beim nach vorne Fahren des Schiebers in das Setzrohr. Im unteren Ende des Setzrohrs verweilt sie anschließend, da sie durch die schon gesetzte Perle blockiert wird. Als nächstes fährt die Beadbox die Position zum Setzen der nächsten Perle an. Dabei rutscht die Perle aus dem Setzrohr auf ein freies Feld. Da der Software der Beadbox nicht bekannt ist, dass sich jetzt auf diesem Feld schon eine Perle befindet, wird versucht eine weitere Perle zu setzen. Diese bleibt im unteren Teil des Setzrohrs, da sie von der schon gesetzten blockiert wird. Die Beadbox kann nicht weiter drucken, bis diese Perle manuell entfernt wird.

4.1.2 Fehler beim Setzrohr

Das Setzrohr ist für das Absetzen der Perlen auf dem Stiftbrett zuständig. Dazu wird es auf eine Position genau über einem Stift gefahren und eine Perle rutscht durch das Setzrohr auf diesen Stift. Während die Perle durch das Setzrohr rutscht passiert sie eine Lichtschranke, die registriert, dass die Perle das Setzrohr verlassen hat. Folgende Fehler konnten im Zusammenhang mit dem Setzrohr beobachtet werden.

S1 Perlen werden blockiert Rutscht eine Perle durch das Setzrohr, so kann es vorkommen, dass sie durch schon gesetzte Perlen daran gehindert wird

dieses zu verlassen. Im Unterschied zu dem beim Fehler V3 auftretendem Fall wo eine Perle direkt unter dem Setzrohr gesetzt wurde, wird die Perle in diesem Fall von einer Perle auf einem angrenzenden Feld blockiert. Die Perle hängt dabei leicht schräg im unteren Ende des Setzrohrs und liegt mit der Kante auf einer angrenzenden Perle auf. In Abbildung 6 ist dieser Fall dargestellt. Das dieser Fehler auftritt, kann zwei Ursachen haben. Die Erste entsteht, weil die Perle in einer schrägen Position mehr Platz einnimmt, als in einer senkrechten. Deswegen liegt sie auf einer angrenzenden Perle auf. Würde sie das Setzrohr genau senkrecht verlassen, hätte sie genug Platz um das Setzrohr zu verlassen. Die andere Ursache kann sein, dass eine oder mehrere schon gesetzte Perlen einen zu großen Durchmesser haben. In diesem Fall kann die Perle auch dann nicht das Rohr verlassen, wenn sie genau senkrecht im Setzrohr sitzt.

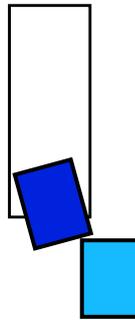


Abbildung 6: Fehler S1

S2 Umgeworfenen Perlen Am unteren Ende des Setzrohrs sind rechts und links die zwei Komponenten einer Lichtschranke angebracht. Diese werden bei der Positionierung des Setzrohrs mit bewegt. Ist eine Perle etwas höher als die normalen Perlen, so kann es passieren, dass die Komponenten der Lichtschranke diese bei der Bewegung des Setzrohrs umkippen.

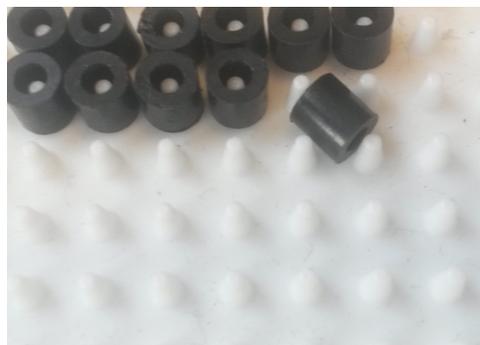


Abbildung 7: Fehler S2

4.1.3 Bedienfehler

Als Bedienfehler werden alle Fehler bezeichnet, dessen Ursache in der falschen Bedienung der Beadbox liegt. Damit ist weniger die Bedienung der Software, sondern die Bedienung der Hardware gemeint, wie z.B. das Nachfüllen der Perlenbehälter.

B1 Flasche Perlenfarbe Die verschiedenen Perlenmagazine werden mit je einer Perlenfarbe befüllt. Dieser Vorgang muss per Hand durchgeführt werden. Da die Beadbox keinen Mechanismus besitzt, um die Perlenfarbe zu überprüfen, führt ein falsch befülltes Perlenmagazin, z.B. gelbe Perlen im Magazin für schwarze Perlen, zu einem falschen Druckergebnis (gelbe Perlen, da wo schwarze sein sollten).

B2 Leeres Perlenmagazin Ein Perlenmagazin kann ca. 60 Perlen aufnehmen. Wird eins der für den Druck benötigten Perlenmagazine nicht befüllt, so erkennt die Beadbox dies erst, sobald eine Perle dieser Farbe aus dem Magazin entnommen werden muss und gibt eine entsprechende Meldung aus, dass das Magazin nachgefüllt werden muss. Der Drucker kann erst weiter drucken, wenn das entsprechende Magazin nachgefüllt wurde. Dass ein Magazin leer ist, weil alle 60 Perlen für den Druck benutzt wurden, kann auch während des Druckens von großen Bildern, oder Bildern mit vielen gleichfarbigen Pixeln geschehen. Dies wird als Fehler mit aufgeführt, da ein leeres Perlenmagazin den Druckvorgang unterbricht.

B3 Falsch ausgerichtete Beadbox Damit die Beadbox die einzelnen Stifte des Stiftbrettes anfahren kann, muss sie wissen, wo sich diese befinden. Dafür werden in einer Konfigurationsdatei für drei der vier Eckpunkte des Druckbereichs die benötigten Schritte der Schrittmotoren, um zu diesem Punkt zu gelangen, angegeben. Stimmen diese Angaben nicht mehr mit der tatsächlichen Position des Stiftbrettes überein, so können die Perlen nicht richtig auf diesem platziert werden. Im schlimmsten Fall werden sie neben den Stiften, oder sogar neben dem Stiftbrett platziert.

4.1.4 Fehler am Druckbild

Als letzte Art von Fehlern betrachten wir Fehler, die ein falsches Druckergebnis beschreiben. Diese können durch einen der schon aufgeführten Fehler ausge-

löst werden, oder ihre Ursache in Problemen haben, die in der Analyse nicht beobachtet werden konnten.

E1 Falsche Farbe einer gesetzten Perle Das gedruckte Bügelperlenbild beinhaltet Perlen der falschen Farbe. Auf einem Feld des Stiftbretts wurde zwar korrekt eine Perle gesetzt, dessen Farbe entspricht aber nicht der Farbe, die gedruckt werden sollte.

E2 Falsch gesetzte Perle Im gedruckten Bügelperlenbild befinden sich Perlen auf Feldern, die leer bleiben sollten.

E3 Nicht gesetzte Perle Beim gedruckten Bügelperlenbild sind Felder leer geblieben, auf denen Perlen gesetzt werden sollten.

4.2 Vorhandene Fehlererkennung- und Fehlerkorrekturmechanismen

Die Beadbox kann drei verschiedene Arten von Fehlern erkennen. Das sind die Fehler V1 „Perle steckt am Setzrohr fest“, S1 „Perlen werden blockiert“ und B2 „Leeres Perlenmagazin“. Zwischen den Fehlern V1 und B2 kann die Beadbox allerdings nicht unterscheiden. Zum Erkennen dieser Fehler benutzt die Beadbox eine Lichtschranke [Zie13, S. 15]. Diese ist am unteren Ende des Setzrohrs angebracht und wird unterbrochen, sobald sich eine Perle in diesem Teil des Setzrohrs befindet. Rutscht eine Perle durch das Setzrohr auf ein Stift auf dem Stiftbrett, so wird die Lichtschranke kurz unterbrochen. Die Beadbox nimmt diese Unterbrechung wahr und weiß, dass die Perle das Setzrohr verlassen hat, woraufhin sie beginnt die nächste Perle zu setzen. Tritt der Fehler V1 auf und eine Perle bleibt am Schieber stecken, so kann die Lichtschranke keine vorbeikommende Perle registrieren und die Beadbox geht davon aus, dass ein Fehler aufgetreten ist. Es muss sich um einen der Fehler V1 oder B1 handeln. In beiden Fällen wird zur Fehlerkorrektur der Schieber angesteuert. Blockiert eine Perle diese, so wird sie durch das Bewegen des Schiebers meist befreit und rutscht wie geplant durch das Setzrohr, der Druckvorhang kann fortgeführt werden. Geschieht dies nicht bei dem ersten Versuch, so wird der Schieber noch bis zu zwei weitere Male bewegt, um die Perle frei zu bekommen. Konnte die Perle bei dem dritten Versuch nicht gelöst werden, so gibt die Beadbox eine Fehlermeldung aus. Der andere Fehler, der aufgetreten sein kann, ist ein leeres Perlenmagazin. Ist das gerade benutzte Perlenmagazin leer, so kann der

Schieber keine Perle aufnehmen und die Lichtschranke wird keine Perle im Setzrohr detektieren. Auch in diesem Fall versucht die Beadbox den Schieber zu bewegen. Da das Magazin leer ist, kann keine weitere Perle aufgenommen werden und nach drei Versuchen gibt die Beadbox eine Fehlermeldung aus. Mit der Lichtschranke kann ebenfalls der Fehler S1 registriert werden. Bei diesem kann eine Perle das untere Ende des Setzrohr nicht verlassen, da sie von anderen Perlen behindert wird. Da die Lichtschranke genau dort angebracht ist, registriert sie die Perle. Um diesen Fehler zu korrigieren, bewegt die Beadbox das Setzrohr in vertikaler und horizontaler Richtung hin und her. Durch diese kreuzförmige Bewegung kann die Perle in den meisten Fällen gelöst werden. Diesen Vorgang wiederholt die Beadbox bis zu drei Mal, wurde der Fehler dadurch nicht behoben, so gibt sie auch hier eine Fehlermeldung aus.

4.3 Fehlereinordnung

Nachdem die auftretenden Fehler beobachtet und beschrieben wurden, werden diese bewertet, um eine Entscheidungsgrundlage zu geben, welche der Fehler priorisiert behoben werden. Als Entscheidungsgrundlage dient eine Bewertung der Fehler. Die Fehler werden mithilfe der drei Kriterien Fehlerhäufigkeit, Umsetzbarkeit und Beeinträchtigung des Druckvorgangs beurteilt. Die Fehlerhäufigkeit wird in Auftreten des Fehlers pro 100 gedruckter Perlen angegeben und als f_a abgekürzt. Um sie zu ermitteln werden Testdrucke durchgeführt und dabei die auftretenden Fehler gezählt. Für diese Messungen werden zwei Testbilder verwendet, mit denen je fünf Testdurchläufe durchgeführt werden. Die Testbilder wurden so gewählt, dass möglichst realistische Werte für die Fehlerhäufigkeit zustande kommen. Diese Testbilder sind im Anhang A zu sehen. Als zweites Kriterium wird die Umsetzbarkeit beurteilt. Für ihre Beurteilung wird der Aufwand einer Umsetzung eingeschätzt. Dazu wird betrachtet, ob für eine Umsetzung ...

... die Anpassung von Software ...

... die Neuentwicklung von Software ...

... die Anpassung von Hardware ...

... die Neuentwicklung von Hardware ...

... erforderlich ist. Die Umsetzbarkeit wird in drei Stufen unterteilt: leicht, mittel und schwer. Die Einteilung in diese Stufen geschieht aufgrund der eben

genannten vier Punkte, wobei die Anpassung der Soft- oder Hardware als einfacher eingestuft wird, als die Neuentwicklung.

Als drittes Kriterium zur Beurteilung wird die Beeinträchtigung des Druckvorgangs herangezogen. Dazu wird bewertet, ob der Fehler zum Abbruch des Druckvorgangs führt, ihn verzögert, oder ihn nicht beeinflusst. Als abgebrochen wird ein Druckvorgang bezeichnet, bei dem der Druckvorgang der Beadbox unterbrochen wird und sie nicht in der Lage ist diesen selbstständig fortzuführen.

Das Ergebnis der Bewertung ist in der Entscheidungsmatrix (siehe Tabelle 1) zu sehen. Diese zeigt die ermittelte Fehlerhäufigkeit, die Einschätzung der Umsetzbarkeit zu jedem beobachteten Fehler und die Beurteilung inwiefern er den Druckvorgang beeinflusst. Wie die Fehlerhäufigkeit ermittelt wurde, ist bereits beschrieben worden.

Die Umsetzbarkeit einer Lösung der Fehler V1, V3 wird als mittelschwer eingestuft. Diese kann aller Voraussicht nach durch Anpassen oder Austauschen vom Vereinzelnmechanismus gelöst werden. Der Fehler V2 wird als leicht lösbar eingestuft, da voraussichtlich nur einfache Änderungen an den Perlenmagazinen vorzunehmen sind.

Von den Fehlern, die dem Setzrohr zugeordnet sind, wird eine Lösung für S1 als schwer umsetzbar bewertet, da das Setzrohr ausgetauscht werden muss. Dieser Fehler führt nicht zum Abbruch, sondern lediglich zur Verzögerung des Druckvorgangs. Der Fehler S2 hingegen kann zum Abbruch führen, wenn z.B. eine umgeworfene Perle ein noch zu bedruckendes Perlenfeld blockiert. Dieser kann leicht korrigiert werden, da ein Versetzen der Lichtschranken diese bereits lösen dürfte.

Als nächstes betrachten wir die Bedienfehler. Zu ihnen gehören die Fehler B1, B2 und B3. Die Umsetzbarkeit einer Lösung von B1 und B3 wird als schwer eingeschätzt, da zum Beheben zunächst die Farbe einer Perle erkannt werden muss. Wurde eine falsch einsortierte Perle detektiert, so muss diese aussortiert werden. Dies erfordert die Neuentwicklung von Hardware. Um den Fehler B3 zu beheben, müsste die Beadbox automatisch ausgerichtet werden. Dies geschieht bereits mithilfe von zwei Lichtschranken. Diese automatische Ausrichtung ist allerdings nicht genau genug und erfordert eine weitere manuelle Ausrichtung. Auch zur Behebung dieses Fehlers ist die Neuentwicklung von Hardware nötig. Während der Fehler B1 den Druckvorgang nicht beeinflusst, kann B3 zum Abbruch des Druckvorgangs führen. Wenn die Beadbox nicht richtig ausgerichtet ist, so kann es passieren, dass die Perlen nicht auf die

Fehler	Fehlerhäufigkeit f_h			Umsetzbarkeit			Druckvorgang		
	$\leq \frac{1}{100}$	$< f_h <$	$\geq \frac{1}{10}$	leicht	mittel	schwer	kein	verz.	abbr.
V1			•		•			•	
V2	•			•					•
V3	•				•				•
S1			•			•		•	
S2	•			•					•
B1	-	-	-			•	•		
B2	-	-	-			•			•
B3	-	-	-			•			•
E1	-	-	-			•	•		
E2	-	-	-			•	•		
E3	-	-	-			•	•		

- Markierung
- Keine Daten erhoben

Tabelle 1: Entscheidungsmatrix

Stifte des Stiftbretts gesetzt werden. Bei einem leeren Perlenmagazin tritt der Fehler B2 auf. Das Erkennen dieses Fehlers kann durch den Einsatz einer Lichtschranke bewerkstelligt werden. Um den Fehler zu beheben wird ein größeres Perlenmagazin benötigt. Dieser Lösungsansatz ist schwer umzusetzen. Da die Perlenmagazine schon sehr hoch sind, können diese nicht einfach nach oben hin verlängert werden. Es müssten neuen Perlenmagazine entwickelt werden.

Die letzten drei Fehler in der Tabelle sind die Fehler am Druckbild. Sie verzögern den Druckvorgang nicht, da sie sich nur auf ein falsch gedrucktes Bügelperlenbild beziehen. Das Lösen dieser Fehler wird als schwer eingestuft, da zum Erkennen dieser die Analyse des gedruckten Bügelperlenbilds erforderlich ist.

4.4 Prioritäten für Fehlerbearbeitung

Im vorherigen Abschnitt wurden alle erkannten Fehler nach Häufigkeit, Umsetzbarkeit einer Lösung und Beeinflussung des Druckvorgangs bewertet. Diese erhobenen Werte werden nun genutzt um für die Fehler Prioritäten zu vergeben. Die Prioritäten werden in die drei Stufen niedrig, mittel und hoch unterteilt, wobei Fehler mit hoher Priorität unbedingt behandelt werden sollen, mit mittlerer Priorität wäre es wünschenswert, wenn diese Fehler behandelt werden und die mit niedriger können bearbeitet werden, wenn die Fehler höherer und mittlerer Prioritäten gelöst wurden. In Tabelle 2 sind die vergebenen

Prioritäten zu sehen.

• Fehler	Priorität		
	niedrig	mittel	hoch
V1			•
V2		•	
V3	•		
S1			•
S2		•	
B1	•		
B2		•	
B3	•		
E1			•
E2			•
E3			•

Tabelle 2: Prioritäten

Die Fehler V1 „Perlen stecken am Setzrohr fest “ und S1 „Perlen werden blockiert “ haben die Prioritäten hoch bekommen. Das liegt vor allem daran, dass diese eine sehr hohe Fehlerhäufigkeit haben. Durch beide Fehler kann der Druckvorgang stark verzögert werden. Außerdem kann der Fehler V1 den Fehler V3 „Aufnahme einer zweiten Perle “ zur Folge haben, der zum Abbruch des Druckvorgangs führt. Der Fehler V3 bekommt nur eine niedrige Priorität, da das Lösen des Fehlers V1 diese mit behebt und es deswegen sinnvoller ist den Fehler V1 direkt zu beheben.

Die Fehler V2 „Blockierung des Schiebers “ und S2 „Umgeworfene Perlen “ wurden mit der Priorität mittel eingestuft. V2 führt zum Abbruch des Druckvorgangs und tritt sehr selten auf. Da er aller Voraussicht nach leicht umzusetzen ist, sollte er behoben werden. Der Fehler S2 wurde aus denselben Gründen mit mittlerer Priorität eingestuft. Der Fehler B1 „Flasche Perlenfarbe “ bekommt die Priorität niedrig, da er als schwer umsetzbar eingestuft wird und den Druckablauf nicht verzögert. B2 „Leeres Perlenmagazin “ wird in der Priorität mittel eingeordnet. Eine Lösung für diesen Fehler ist schwer umsetzbar.

Der Fehler B3 „Falsch ausgerichtete Beadbox “ kann zum Scheitern des gesamten Druckvorgangs führen und beim Drucken möglicherweise andere Fehler hervorrufen. Wie im Benutzerhandbuch der Beadbox beschrieben [Zie13, S. 41], muss die Beadbox nach Änderung der Hardware neu ausgerichtet werden. Wie Herr Ziegler schreibt, ist das neuerliche Ausrichten der Beadbox während des regulären Betriebes nicht nötig. Aus diesem Grund bekommt der Fehler nur

eine niedrige Priorität. Als letztes werden die sonstigen Fehler E1 „Falsche Farbe einer gesetzten Perle“, E2 „Falsch gesetzte Perle“ und E3 „Nicht gesetzte Perle“ betrachtet. Diese sind alle drei mit hoher Priorität eingeordnet, obwohl diese den Druckvorgang nicht negativ beeinflussen und die Umsetzbarkeit als schwer eingestuft wurde. Diese Fehler sollte trotz alledem auf jeden Fall behoben werden, da ihr Auftreten bedeutet, dass das gedruckte Bügelperlenbild nicht korrekt gedruckt wurde.

5 Fehler vermeiden und korrigieren

In der Ist-Analyse wurden die Fehler ermittelt, die bei der Verwendung der Beadbox auftreten. In diesem Kapitel wird beschrieben, welche Lösungen erarbeitet wurden, um diese Fehler korrigieren oder zu beheben. Bei der Entwicklung dieser Lösungen wurden die für die Fehler vergebenen Prioritäten mit berücksichtigt. Es wurde mehr Aufwand in die Erarbeitung von Fehlerlösungen mit hoher Priorität gesteckt. Während es im Abschnitt 5.1 und Abschnitt 5.2 um Lösungen geht, die das Auftreten eines Fehlers vermeiden sollen, handelt der Abschnitt 5.3 von der Entwicklung eines Verfahrens, um schon aufgetretene Fehler zu erkennen und zu korrigieren.

5.1 Fertigungsgenauigkeit

Betrachtet man die für die Beadbox verwendeten Bügelperlen, so fällt auf, dass diese Unterschiede in Höhe und Durchmesser aufweisen. Werden die Perlen, wie vom Hersteller vorgesehen, per Hand auf dem Stiftbrett platziert, so stellen diese Unterschiede kein Problem dar. Beim maschinellen Setzen von Bügelperlen, können diese Unterschiede hingegen Probleme bereiten. Bei der Beadbox kann beispielsweise eine Perle mit einem zu großen Durchmesser im Setzrohr stecken bleiben. Solche Probleme können bei der Beadbox vor allem an den mechanischen Teilen, wie dem Vereinzelungsmechanismus oder dem Setzrohr auftreten. Diesen beiden Komponenten konnten die Fehler V1, V2, V3, S1 und S2 zugeordnet werden. Das Verwenden von Bügelperlen, die kleinere Unterschiede in Höhe und Durchmesser aufweisen, könnte diese Fehler beheben, oder die Häufigkeit mit der sie auftreten verringern. Aus diesem Grund soll zunächst geprüft werden, ob es Hersteller von Bügelperlen gibt, die diese mit höherer Genauigkeit herstellen. Dafür wurden folgende drei Hersteller identifiziert:

- Hama [A/S]
- SES [b.va]
- Nabbi [AB]

Die Perlen der Firma Hama werden bereits für die Beadbox verwendet. Für alle drei Hersteller soll nun die Fertigungsgenauigkeit ermittelt werden. Die Fertigungsgenauigkeit bezeichnet die Abweichung von den ermittelten zu den gewünschten Abmessungen eines gefertigten Gegenstandes. Dabei gilt: je geringer die Abweichung, desto genauer, je größer die Abweichung, desto ungenauer

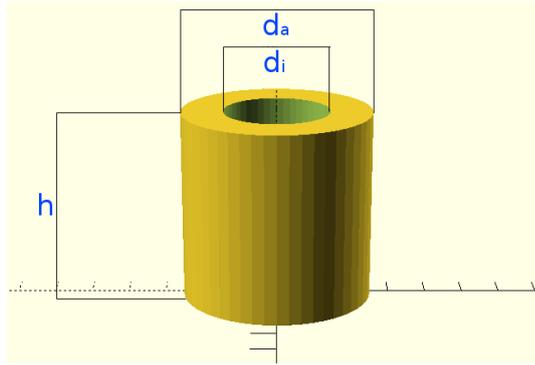


Abbildung 8: Dimensionen einer Perle

ist ein Gegenstand gefertigt. Ursachen für eine ungenaue Fertigung können z.B. die zur Herstellung verwendeten Maschinen oder das verwendete Material sein. Um die Fertigungsgenauigkeit einer Bügelperle zu bestimmen, muss zunächst ihre Form beschrieben werden. Eine Perle kann durch ihre Höhe h , den Innendurchmesser d_i und den Außendurchmesser d_a beschrieben werden (siehe Abbildung 8). Der Innendurchmesser ist für die weite Betrachtung nicht von Interesse, da dieser lediglich groß genug sein muss, damit die Perlen auf die Stifte des Stiftbretts passen. Das eine Perle einen zu kleinen Durchmesser hat, konnte bei keiner Perle festgestellt werden. Bei den meisten Herstellern können keine konkreten Angaben darüber gefunden werden, wie groß eine Bügelperle ist. Die Firma Hama gibt lediglich die Höhe ihrer Perlen mit fünf Millimetern an [A/S], während die Firma SES diese mit vier bis fünf Millimeter angibt [b.va].

Die Hersteller von Bügelperlen geben in der Regel keine Angaben zur Fertigungsgenauigkeit ihrer Produkte an. Das wird daran liegen, dass die Zielgruppe ihrer Produkte Kinder sind, die zum einen nicht viel mit den Angaben anfangen können und zum anderen ist für das Setzen der Bügelperlen per Hand keine hohe Fertigungsgenauigkeit nötig. Die Fertigungsgenauigkeit bezogen auf die Höhe der Perle F_h und bezogen auf den Durchmesser F_d kann wie folgt bestimmt werden:

$$F_h = h_s - h$$

$$F_d = d_s - d_a$$

h_s und d_s geben dabei die Sollwerte für Höhe und Durchmesser der Perlen an. Wie schon erwähnt, geben die Hersteller für Bügelperlen in der Regel keine genauen Werte nach denen die Perlen gefertigt werden an. Es muss eine andere

Möglichkeit gefunden werden die Genauigkeit der Fertigung bei den Perlen zu bestimmen, damit diese verglichen werden können. Dazu werden sie mit einer Schieblehre vermessen und anschließend der Durchschnittswert für Höhe und Durchmesser bestimmt. Um die Genauigkeit in der Perlenfertigung verschiedener Hersteller zu vergleichen, können die Standardabweichungen für Höhe und Durchmesser betrachtet werden. Um diese Messung durchführen zu können, wurde von den drei Perlenherstellern Hama, SES und Nabbi je 1000 Perlen bestellt. Aus jeder Menge wurden je 20 zufällige Stichproben gewählt und mit einer Schieblehre vermessen. Die Schieblehre kann, laut Herstellerangaben, auf 0,01 Millimeter genau messen. Die Ergebnisse dieser Messung sind in Tabelle 3 abgebildet.

Probe	Hama		SES		Nabbi	
	Höhe	Durchm.	Höhe	Durchm.	Höhe	Durchm.
1	5	4,61	4,94	4,26	4,95	4,85
2	4,96	4,62	5,61	4,56	4,79	4,72
3	4,99	4,62	5,12	4,63	4,9	4,98
4	4,78	4,72	5,01	4,36	4,97	4,78
5	4,89	4,63	5,20	4,63	5,03	4,78
6	4,76	4,66	5,04	4,62	4,92	4,85
7	4,83	4,6	5,08	4,41	5,04	4,74
8	4,94	4,7	5,45	4,46	4,91	4,67
9	5,07	4,61	4,95	4,37	4,91	4,92
10	5,01	4,63	5,04	4,48	4,99	4,80
11	5,17	4,63	5,22	4,47	4,89	4,81
12	4,92	4,67	5,22	4,48	4,99	4,83
13	5,01	4,68	5,04	4,59	5,01	4,85
14	5	4,61	5,68	4,66	4,78	4,75
15	4,97	4,69	5,15	4,42	5,08	4,82
16	5,03	4,57	5,30	4,54	4,96	4,83
17	4,98	4,71	4,92	4,62	4,81	4,88
18	4,83	4,63	4,76	4,49	4,89	4,81
19	4,96	4,66	5,68	4,6	4,77	4,73
20	4,9	4,49	4,94	4,34	4,84	4,83
Durchschnitt	4,95	4,637	5,1675	4,4995	4,9215	4,8115
Standardabw.	0,0963	0,0517	0,2537	0,1116	0,088	0,0694

Tabelle 3: Gemessene Perlengrößen

Bei jeder Perle sind gemessene Höhe und Durchmesser in Millimeter angegeben. Diese Werte sind jeweils auf zwei Nachkommastellen gerundet. Ebenfalls sind die Werte für Durchschnitt, Varianz und Standardabweichung angegeben. Diese sind auf vier Nachkommastellen gerundet. Zusätzlich zu den gemessenen

Werten konnte bei den Perlen der Firma Nabbi beobachtet werden, dass diese oft schief sind. Bei den in Abbildung 9 gezeigten Perlen ist gut zu erkennen, dass diese schräg sind.



Abbildung 9: Bügelperlen der Firma Nabbi

Bewertung der Messergebnisse Bei den Messergebnissen in Tabelle 3 fällt auf, dass die Perlen der Firma SES im Durchschnitt deutlich höher sind als die der Firmen Hama und Nabbi. An der Standardabweichung für die Höhe der Perlen, ist zusätzlich zu erkennen, dass die Werte der Perlenhöhe bei SES größere Abweichungen haben als bei den anderen beiden Firmen. Bei der Standardabweichung des Durchmessers ist dies ebenfalls der Fall. Die Bügelperlen der Firma SES haben somit eine niedrigere Fertigungsgenauigkeit als die der anderen beiden Hersteller. Die Werte der Standardabweichung für die Perlen der Hersteller Hama und Nabbi weichen nicht so stark voneinander ab. Während die Firma Nabbi einen etwas niedrigeren Wert bei der Höhe ihrer Perlen erzielt, hat Hama einen etwas niedrigeren Wert beim Durchmesser. Die Perlen dieser beiden Hersteller haben eine ähnliche Fertigungsgenauigkeit. Allerdings konnte, wie im vorherigen Abschnitt erwähnt, bei den Perlen der Firma Nabbi beobachtet werden, dass diese oft schief sind. Aus diesem Grund, sollten sie nicht mit der Beadbox verwendet werden. Die bereits verwendeten Perlen der Firma Hama eignen sich damit am besten, um mit der Beadbox verwendet zu werden. Aus diesem Grund werden sie auch weiterhin mit der Beadbox verwendet.

5.2 Mechanische Verbesserungen

In diesem Abschnitt wird beschrieben, welche Veränderungen an der Hardware der Beadbox vorgenommen wurden, um die Fehler zu beheben, die mit den mechanischen Teilen der Beadbox zusammenhängen. Zu diesen Fehlern gehören die Fehler V1, V2 und V3, die an dem Vereinzelungsmechanismus auftreten und die Fehler S1 und S2, die am Setzrohr auftreten. Der Vereinzelungsmechanismus besteht aus dem Schieber, dem oberen Ende des Setzrohrs und dem

unteren Ende des Perlenmagazins. Sowohl der Schieber als auch das Setzrohr wurden komplett ausgetauscht. Beide bestehen aus Metall. Die neu entwickelten Komponenten werden mit einem 3D-Drucker gedruckt und bestehen aus Plastik.

3D-Druck Bei der Entwicklung der Komponenten wurden zunächst die zu ersetzenden Komponenten mit einer Schieblehre vermessen. Aus diesen Messergebnissen wurde mit der Software *OpenSCAD* ein 3D-Modell entworfen. Um die erstellten Modelle am 3D-Drucker drucken zu können, wird das erstellte Modell als STL-Datei gespeichert. Diese Datei kann von der Software des 3D-Druckers eingelesen und in eine vom 3D-Drucker verständliche Datei umgewandelt werden.

Das Drucken der in dieser Arbeit entworfenen Modelle dauert, je nach Größe des Modells und eingestellter Druckqualität, ein bis vier Stunden. Dies ist allerdings nur die reine Druckzeit. Beim Drucken mit dem 3D-Drucker traten des Öfteren Probleme auf, die zum Abbruch des Drucks führten. Häufig lag es daran, dass die Düse, durch die das Plastik auf die Druckfläche aufgetragen wird, verstopft war. Diese Düse musste daraufhin gereinigt werden und der Druckvorgang von neuem gestartet werden. Nachdem eine neue Düse am 3D-Drucker installiert wurde, traten dieses Probleme nicht mehr so häufig auf.

Versionen der Modelle Da die neuen Komponenten schrittweise verbessert wurden, wurde jede Version einer Komponente mit einer Versionsnummer versehen. Dadurch kann leicht nachverfolgt werden, welche Änderungen vorgenommen wurden und falls diese nicht zum gewünschten Resultat führen, können sie leichter rückgängig gemacht werden. Die Versionsnummern beginnen mit der Version *0.1* und werden zu jeder neuen Version um *0.1* erhöht. Eine Änderungshistorie (engl. *changelog*) zu jeder gedruckten Komponente ist im Anhang D zu finden.

5.2.1 Schieber

Der Schieber nimmt die Perlen aus dem Perlenmagazin, vereinzelt sie und übergibt sie an das Setzrohr. Er hat somit eine zentrale Rolle in der Mechanik der Beadbox. Aus diesem Grund können hier auch die meisten mechanischen Fehler auftreten. Der Schieber ist mit einem Hubmagneten verbunden, der ihn bewegt. In der Ausgangsposition ist dieser ausgefahren, sodass die Öffnung im Schieber über der Öffnung im Setzrohr ist. Wie schon in Abschnitt 4.1.1

mit dem Fehler V1 beschrieben, können sich Perlen im Schieber verhaken. Um dieses Problem zu lösen, soll ein neuer Schieber entworfen und mit dem 3D-Drucker gedruckt werden. Im Folgenden wird die schrittweise Entwicklung dieses Schiebers beschrieben.

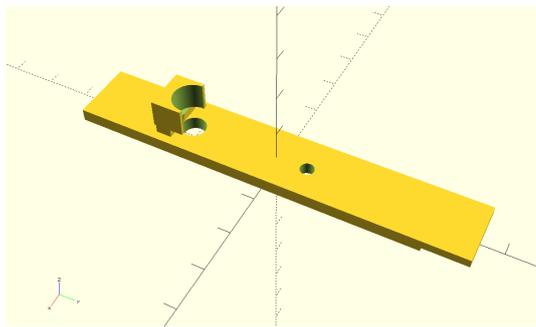
Ersatz des alten Schiebers Ziel der ersten Version ist es einen Schieber zu entwerfen, der den alten ersetzen kann. Dazu wurde dieser mit einer Schiebellehre vermessen und mit der Software *OpenSCAD* ein Modell des Schiebers angefertigt. Am alten Schieber befinden sich zwei Wände aus Metall, die ihn beim Zurückfahren am Rohr des Perlenmagazins stoppen und beim nach vorne fahren über der Öffnung zum Setzrohr anhalten. Diese Vorrichtung bezeichnen wir als Stopper. Der Stopper ist in der Abbildung 10 zu sehen.



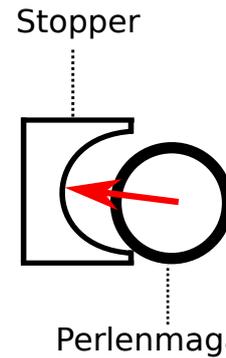
Abbildung 10: Stopper am alten Schieber

Seine Wände sind etwa 0,5 mm dick. Da der Stopper bei dem neuen Modell aus Plastik besteht, wären diese Wände nicht stabil genug um den beim Stoppen wirkenden Kräften Stand zu halten. Aus diesem Grund muss der Stopper verändert werden. In der neuen Version des Schiebers besteht er aus einem Kasten, der an der einen Seite eine rundliche Aussparung hat. In diese füt sich beim Zurückfahren genau das Rohr des Perlenmagazins. Dadurch wird das Rohr beim Zurückfahren zusätzlich über der Öffnung des Schiebers zentriert. Dieser Vorgang ist in Abbildung 11b schematisch dargestellt. Die Version 0.1 des entworfenen Schiebers ist in Abbildung 11a zu sehen.

Nachdem die Version 0.1 des Schiebers mit dem 3D-Drucker ausgedruckt wurde, konnte festgestellt werden, dass der Stopper noch nicht die richtige Form hat, damit er sich mit dem Rohr des Perlenmagazins beim Rückfahren zusammenfügt. In der Version 0.2 des Schiebers wurde der Stopper so verändert, sodass das Rohr in die Aussparung passt. Bei der zweiten Version wurde festgestellt, dass der hintere Teil des Schiebers beim Zurückfahren von unten gegen den Hubmagneten kommt. Dadurch kann der Schieber nicht ganz zurückgefahren werden. Um dieses Problem zu lösen, wurde in der Version 0.3



(a) Schieber



(b) Funktionsweise Stopper

Abbildung 11: Schieber V 0.1

am hinteren Teil des Schiebers etwas Material entfernt, sodass mehr Platz zwischen Schieber und Hubmagnet ist. Mit der Version 0.3 des Schiebers wurden alle Problem beseitigt, die verhindern, dass der Schieber eine Perle aufnehmen kann. Aus diesem Grund kann mit ihr ein erster Testdruck durchgeführt werden. Zum Testen wurde dasselbe Bild gedruckt, welches schon in der Ist-Analyse verwendet wurde. Dieses ist im Anhang A in Abbildung 53a zu sehen und besteht aus einem Feld mit sechs mal sechs schwarzen Perlen. Es wurden zwei Testdrucke durchgeführt, deren Ergebnisse in Tabelle 9 im Anhang B zu sehen sind. Die Fehlerhäufigkeit hat sich mit 26,7% gegenüber der des alten Schiebers nicht drastisch verändert. Damit kann diese Version als Ersatz des alten Schiebers angesehen werden. In den nächsten Versionen des Schiebers soll versucht werden, den Fehler V1 zu beheben.

Fehler V1 Der Fehler V1 kann darauf zurückgeführt werden, dass die aufgenommene Perle im Schieber in Schräglage gerät und sich am oberen Teil des Setzrohres verhakt. Dies ist in Abbildung 12a schematisch dargestellt.

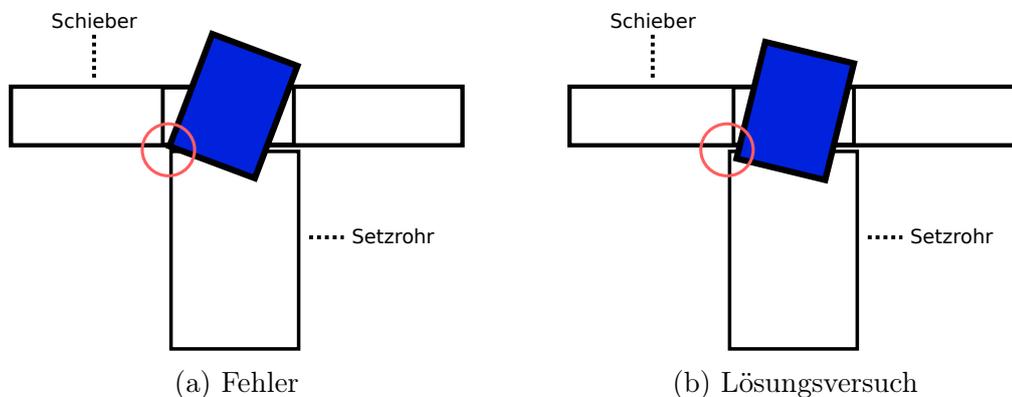


Abbildung 12: Fehler V1 Lösungsversuch 1

Wie in der Abbildung zu sehen ist, liegen die Öffnung und das Setzrohr nicht genau übereinander, wodurch eine Kante entsteht, an der die Perlen hängen bleiben. Damit dieser Fehler auftritt, muss die Öffnung nur minimal zum Setzrohr verschoben sein. Eine genauere Ausrichtung des Schiebers ist nur schwer umsetzbar. Deswegen wird die Öffnung des Schiebers verkleinert. Dadurch kann die Perle nicht auf der Kante aufliegen. Die Veränderung ist in Abbildung 12b dargestellt. Die Öffnung hatte einen Durchmesser von 6 Millimetern und wurde auf 5,6 Millimeter verringert. Obwohl die Öffnung jetzt kleiner ist, sollten die Perlen ohne Probleme in diese passen, da der im Abschnitt 5.1 ermittelten Durchschnittswert für den Perlendurchmesser bei 4,637 Millimetern liegt. Beim Testen diese Version des Schiebers, wurde festgestellt, dass die Perlen sich zwar jetzt nicht mehr am Setzrohr verhaken, dafür aber ein ähnliches Problem am Perlenmagazin auftritt. Dieses ist in Abbildung 13a dargestellt. Dadurch, dass die Öffnung des Schiebers verkleinert wurde, entste-

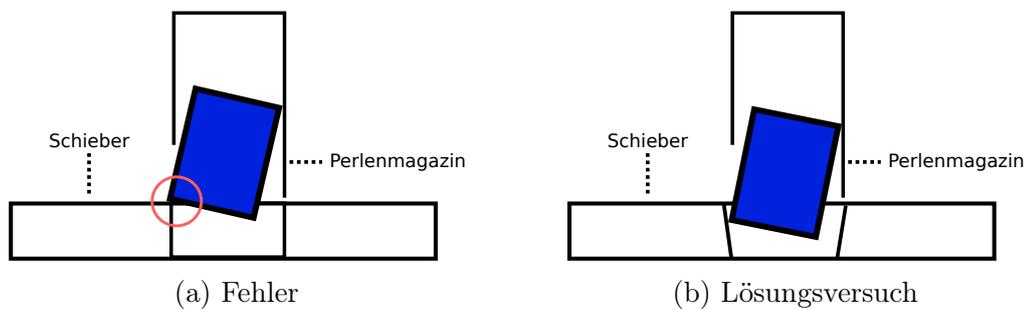


Abbildung 13: Neu auftretender Fehler

hen nun Kanten am Perlenmagazin, an denen die Perle hängen bleibt und den Schieber blockiert. Dieses Problem sollte gelöst werden, indem die Öffnung von einer Zylinderform in eine Trichterform geändert wurde. Wie in Abbildung 13b zu sehen ist entsteht so am Perlenmagazin keine Kante und die Perle kann ohne Probleme aufgenommen werden. Diese Änderung wurde in Version 0.5 des Schiebers vorgenommen. Nachdem das Modell mit dem 3D-Drucker gedruckt wurde, konnte es getestet werden. Beim Testen wurde der Schieber abermals am Setzrohr blockiert. Die Perlen blieben diesmal an der Schräge der trichterförmigen Öffnung hängen. Dies wird in Abbildung 14 gezeigt.

Eine trichterförmige Öffnung ist damit keine Lösung für das Problem. Es musste ein Anderer Lösung gefunden werden. In der Version 0.6 des Schiebers wurde dazu um die Öffnung eine Zentriereinheit angebracht. Diese funktioniert ähnlich dem Stopper, der das Perlenmagazin zentriert. Wie in Abbildung 15a dargestellt ist, drückt dieser die Perle an das hintere Ende des Perlenmagazins. Dadurch ist die Perle genau senkrecht im Perlenmagazin. In Abbildung 15b

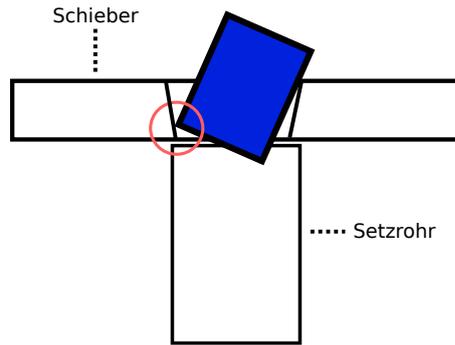


Abbildung 14: Neu auftretender Fehler: Problem

ist die Zentriereinheit von oben zu sehen. Wie dort zu erkennen ist, wird die Perle genau über der Öffnung zentriert.

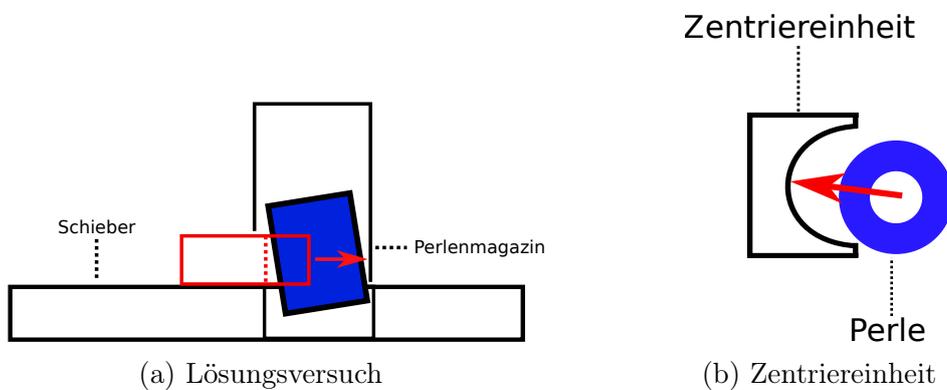


Abbildung 15: Neu auftretender Fehler: Lösung

Um zu überprüfen, ob diese Lösung das Problem V1 beheben kann, wurden erneut Testläufe durchgeführt. Die Fehlerhäufigkeit vom Fehler V1 ist deutlich gesunken auf 2,2%.

5.2.2 Setzrohr

Das Setzrohr ist am oberen Teil der Beadbox angebracht. Es wird zusammen mit dem Perlenmagazin und dem Vereinzelungsmechanismus bewegt, um die Perlen auf dem Stiftbrett zu platzieren. Am unteren Ende des Setzrohrs ist eine Lichtschranke befestigt, die Perlen beim Verlassen des Setzrohrs detektiert. Am Setzrohr treten die Fehler S1 „Perlen werden blockiert“ und S2 „Umgeworfene Perlen“ auf. Wie bei der Entwicklung des Schiebers, soll auch für das Setzrohr zunächst ein 3D-Modell entworfen werden, welches das alte Setzrohr ersetzen kann. Das alte Setzrohr ist mit der Beadbox verklebt. Da das neue Setzrohr wie auch der Schieber in mehreren Versionen entwickelt wird, ist es sinnvoll eine bessere Befestigungsmöglichkeit zu entwickeln, die ein einfaches

Austauschen des Setzrohrs ermöglicht. Außerdem sollte für die Lichtschranke eine neue Halterung entworfen werden, da diese auch am Setzrohr festgeklebt ist und ebenfalls ein Austauschen des selbigen erschwert.

Ersatz des alten Setzrohrs Wie in Abbildung 16 zu sehen ist, hat die erste Version des Setzrohrs eine zusätzliche Fläche am oberen Ende.

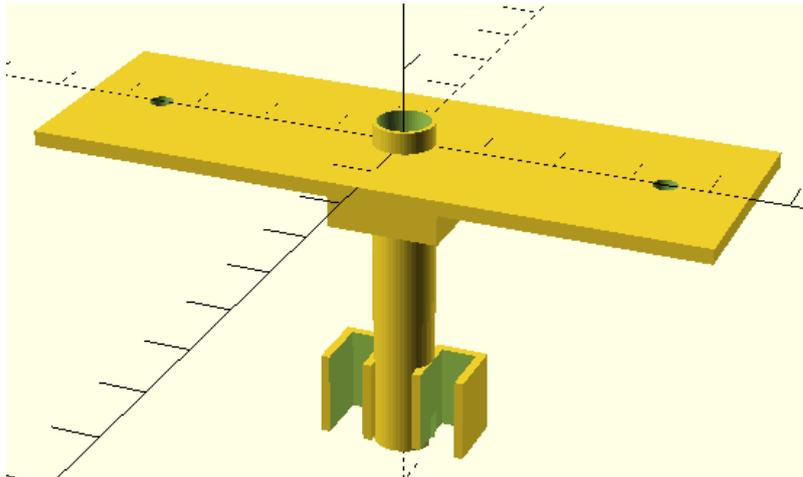


Abbildung 16: Setzrohr Version 0.1

In dieser sind zwei kleine Löcher, sodass die Fläche mit zwei schon vorhandenen Schrauben an der Beadbox befestigt werden kann. Für die Lichtschranke wurde am unteren Ende zwei Halterungen hinzugefügt. Diese sind so aufgebaut, dass die Komponenten der Lichtschranke einfach hineingeschoben werden und von der Spannung zwischen den Wänden der Halterung fixiert werden. Nachdem die erste Version des Setzrohrs mit dem 3D-Drucker gedruckt wurde, konnten noch einige Probleme an dem Modell festgestellt werden. Dabei der ersten Version die Löcher für die Schrauben an der falschen Stelle waren, mussten diese in einer zweiten Version korrigiert werden. Ebenso saß die Lichtschranke in der Halterung zu locker, sodass die Halterung ebenfalls angepasst wurde. Der richtige Abstand für die Lichtschrankenhalterung konnte erst nach dem Drucken einer weiteren Version korrekt gewählt werden. Diese Version 0.3 kann nun benutzt werden, um Testdrucke durchzuführen. Beim Testen wurde festgestellt, dass einige Perlen im Setzrohr stecken blieben. Das lag nicht daran, dass der Innendurchmesser des Rohrs zu klein ist, da dieser gleich dem Innendurchmesser des alten Setzrohrs ist. Der Grund dafür ist die Beschaffenheit der Oberfläche des gedruckten Setzrohrs. Dadurch dass der 3D-Drucker ein Modell Schicht für Schicht druckt, ist die Oberfläche nicht ganz eben. Durch die hohe Auflösung, mit der der Drucker druckt, sind diese Un-

ebenheiten nur sehr fein und mit dem bloßen Auge kaum erkennbar. Um die Oberfläche der Objekte zu glätten, wurde das Setzrohr mit Aceton behandelt. Aceton weicht die Oberfläche auf, sodass diese geglättet wird. Dadurch konnte das Problem gelöst werden. Bei allen weiteren gedruckten Versionen wird ebenfalls die Oberfläche mit Aceton behandelt. Die Perlen rutschen nun ohne Probleme durch das Setzrohr. Mit dieser Version wurden wie auch schon bei der Entwicklung des Schiebers Testdrucke durchgeführt, um zu überprüfen, wie sich das neue Setzrohr auf das Auftreten der Fehler auswirkt. Die Ergebnisse sind in Anhang B in Tabelle 10 zu sehen. Die Fehlerhäufigkeit von Fehler S1 liegt mit der neuen Version mit 71,6% leicht über dem gemessenen Wert mit dem alten Schieber. Der Fehler S2 trat bei der neuen Version deutlich öfter als bei der alten auf. Das liegt daran, dass das Setzrohr noch nicht die richtige Länge hat. Außerdem konnte bei den Testläufen festgestellt werden, dass der Fehler V1 abermals vermehrt auftrat. Dessen Fehlerhäufigkeit konnte durch den ausgetauschten Schieber deutlich gesenkt werden. Anscheinend hat das Austauschen des Setzrohrs diesen Fehler mit beeinflusst.

Fehler V1 Nach einigen Testläufen konnte die Ursache für das vermehrte Auftreten des Fehlers V1 gefunden werden. Das neue Setzrohr ist am oberen Ende zu kurz, sodass zwischen Schieber und Setzrohr ein Spalt entsteht. Dies ist in Abbildung 17 dargestellt. Durch diesen Spalt, hat die Perle genug Platz, um auf dem Rand des Setzrohrs aufliegen zu können. Um das Problem zu lösen, wurde das Setzrohr am oberen Ende verlängert, sodass diese Lücke geschlossen wird. Diese neue Version 0.4 des Setzrohrs wurde getestet. Wie der Tabelle 10 im Anhang B zu entnehmen ist, konnte die Fehlerhäufigkeit auf ein ähnliches Level, wie nach der Anpassung des Schiebers, gesenkt werden.

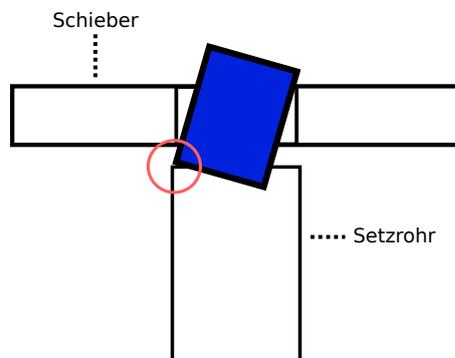


Abbildung 17: Fehler V1 am Setzrohr

Modulare Version Die Version 0.4 des Setzrohres zu drucken dauert etwa vier Stunden. Da die meisten Änderungen am Rohr des Setzrohrs vorgenommen werden, ist es sinnvoll das Modell des Setzrohrs in mehrere Komponenten aufzuteilen. Dadurch muss nicht das gesamte Modell neu gedruckt werden, wenn nur Änderungen am Rohr vorgenommen werden. Außerdem muss jedes Mal die Lichtschranke aus der dafür entworfenen Halterung entfernt und in die neue Version des Setzrohrs eingesetzt werden. Besser ist es, wenn die Halterung abgenommen und mit der Lichtschranke an dem neuen Rohr befestigt werden kann.

Das Setzrohr wurde in drei Komponenten aufgeteilt. Der Halterung, dem Setzrohr und der Lichtschrankenhalterung. Mit dem Begriff Setzrohr wird von nun an nur noch das eigentliche Rohr bezeichnet. Die Komponenten sind in Abbildung 18 zu sehen.

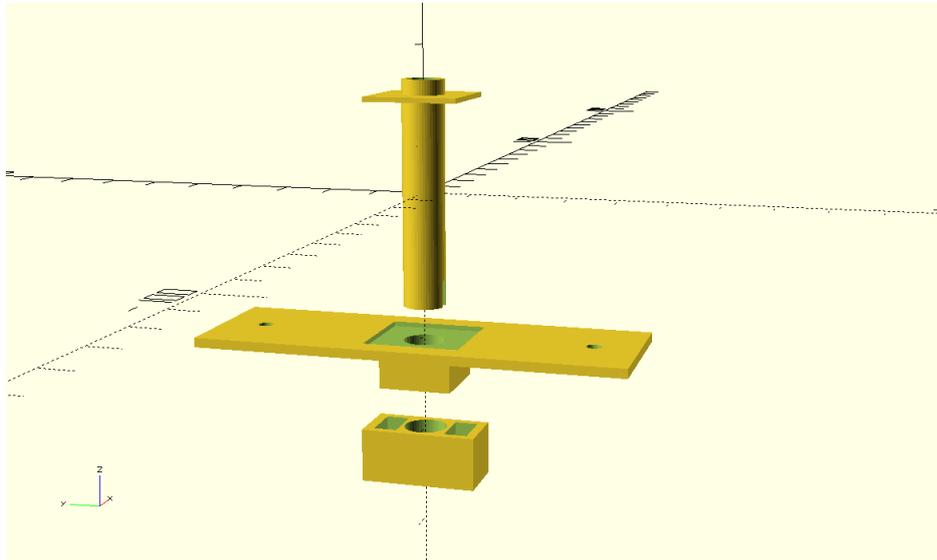


Abbildung 18: Modulare Version vom Setzrohr

Das Setzrohr wird von oben in die Halterung geschoben, sodass es nach dem Verschrauben der Halterung mit der Beadbox festsitzt. Die Lichtschrankenhalterung wird von unten auf das Setzrohr geschoben. Wie im Bild zu erkennen ist, mussten für die modulare Version, an der Lichtschrankenhalterung weitere Änderungen vorgenommen werden. Die Wände der neuen entworfenen Halterung sind dicker als die der Vorversion. Außerdem wurde die Halterung von vorne geschlossen, sodass die Lichtschranke jetzt von oben eingesetzt wird. Dadurch sitzt sie stabiler in der Halterung. Die Druckzeit des Rohrs beträgt etwa 1 Stunde. Werden nur Änderungen an diesem vorgenommen, so hat sich die Druckzeit auf ein Viertel reduziert.

Da das Modell nun in drei Modelle aufgeteilt wurde, die einzeln gedruckt werden können, bekommen diese Komponenten separate Versionsnummern. Alle drei Komponenten starten mit der Version 0.5. Wird nur die Lichtschrankenhalterung verändert, so wird auch nur ihre Version auf 0.6 erhöht.

Fehler S1 Bei dem Fehler S1 werden Perlen im unteren Teil des Setzrohrs durch Perlen, die schon gesetzt wurden, blockiert. Wie in der Fehleranalyse erklärt, gibt es zwei Möglichkeiten, warum die Perle auf einer anderen aufliegt. Der eine Grund ist, dass eine angrenzende Perle einen zu großen Durchmesser hat und die Perle deswegen aufsetzt. Zu große Durchmesser von Perlen hängen mit der Fertigungsgenauigkeit zusammen. Wie in Kapitel 5.1 geschrieben, konnten keine Perlen mit höherer Fertigungsgenauigkeit gefunden werden. Darum soll die zweite Ursache für diesen Fehler betrachtet werden. Wie in der Abbildung gezeigt, geraten Perle im Setzrohr in Schräglage und dadurch liegt ihre untere Kante auf einer nahen Perle auf. Dies ist in Abbildung 19a dargestellt.

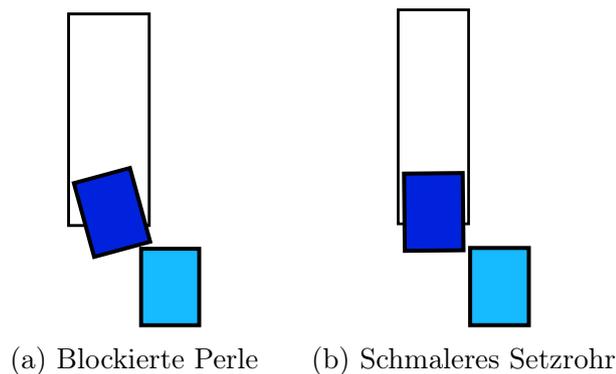


Abbildung 19: Fehler S1: Lösungsversuch

Um zu verhindern, dass die Perlen in Schräglage geraten, wird der Innendurchmesser des Setzrohrs verkleinert. Die Perlen haben dadurch seitlich nicht mehr so viel Platz und können nicht mehr so stark in Schräglage geraten. Dies ist in Abbildung 19b dargestellt. Der Innendurchmesser des Setzrohrs wurde von 6 auf 5,5 Millimeter reduziert. Da die Öffnung des Schiebers einen Durchmesser von 5,6 Millimeter hat, musste dieser auch etwas verkleinert werden, damit der Fehler V1 nicht von Neuem mit hoher Häufigkeit auftritt. Der Öffnungsdurchmesser wurde in der neuen Version 0.7 auf 5,3 Millimeter reduziert. Die neue Version 0.6 des Setzrohrs und die Version 0.7 des Schiebers wurden beide gedruckt und getestet. Beim Testen konnte festgestellt werden, dass der Innendurchmesser des Setzrohrs zu klein ist und einige Perlen im Setzrohr ste-

cken bleiben. Aus diesem Grund wurde der Innendurchmesser des Setzrohrs in einer neuen Version auf 5,8 Millimeter erhöht. Vom Schieber wird die Version 0.6 verwendet, die einen Öffnungsdurchmesser von 5,6 Millimetern hat. Mit diesen beiden Versionen wurden abermals Testdrucke durchgeführt. Wie der Tabelle 10 aus Anhang B zu entnehmen ist, konnte die Fehlerhäufigkeit des Fehlers S1 von etwa 71,6% auf 26,9% gesenkt werden.

Fehler S2 Der zweite am Setzrohr auftretende Fehler ist der Fehler S2. Bei diesem werden Perlen, die sich auf dem Stiftbrett befinden von der Lichtschranke umgeworfen. Dies konnte passieren, da die Lichtschranke etwas über die untere Seite des Setzrohrs hinausragte. Dadurch wurden hohe Perlen bei der Bewegung des Setzrohrs umgeworfen. Diese Fehler konnte bereits durch das Anfertigen einer modularen Version 0.5 des Setzrohrs behoben werden. Da die Lichtschrankenhalterung in dieser Version auf das Setzrohr geschoben wird, kann die Höhe der Lichtschranken verändert werden. Dazu muss lediglich die Lichtschrankenhalterung etwas höher geschoben werden. Wie die Testergebnisse aus dem letzten Testlauf zeigen, tritt dieser Fehler nicht mehr auf.

5.2.3 Perlenmagazin

Wie bei der Ist-Analyse mit dem Fehler V2 festgestellt, kann das Perlenmagazin den Schieber daran hindern nach hinten zu fahren. Soll eine Perle entnommen werden, so wird das entsprechende Perlenmagazin zu dem Schieber gefahren. Wie beim Fehler V2 beschrieben setzt das Perlenmagazin auf dem Schieber auf. Durch die ausgeübte Kraft (durch das Gewicht des Perlenmagazins) wird der Schieber blockiert. Die naheliegende Lösung für dieses Problem ist die Perlenmagazine etwas höher zu setzen, sodass der Abstand zwischen Perlenmagazin und Platte bzw. Schieber vergrößert wird. Dadurch können die Perlenmagazine nicht mehr auf der Platte schleifen oder auf dem Schieber aufsetzen. Um die Magazine etwas nach oben zu setzen, muss die Halterung für die Perlenmagazine etwas nach oben verschoben werden. Zunächst wurden hierfür zwei handelsübliche Unterlegscheiben benutzt. Diese haben eine Höhe von 1,65 Millimetern. Dadurch schleifen die Perlenmagazine nicht mehr, der Abstand zwischen Magazinen und Platte ist nun zu groß, sodass Perlen gelegentlich aus den Magazinen rutschen. Außerdem sitzen die Stifte, mit denen über eine Lichtschranke das aktuelle Perlenmagazin bestimmt wird zu hoch und es werden falsche Magazine ausgewählt. Aus diesem Grund wurde die Unterlegscheibe mit dem 3D-Drucker gedruckt. Sie ist in Abbildung 20 zu sehen. Der

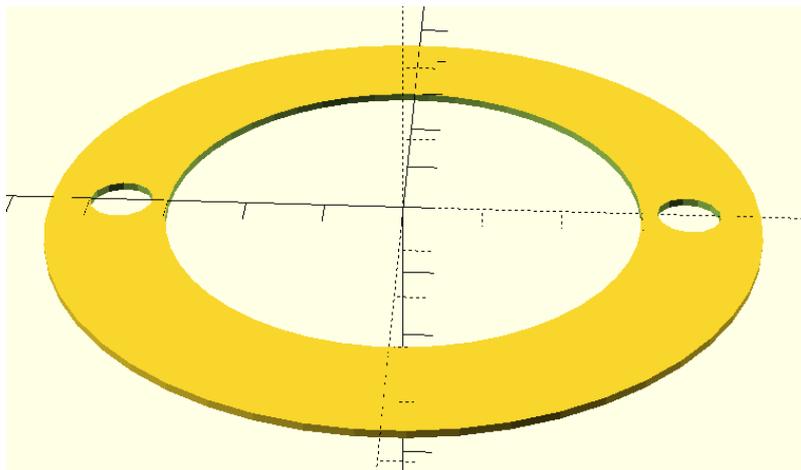


Abbildung 20: Unterlegscheibe

Vorteil dieser selbst gedruckten Unterlegscheibe gegenüber herkömmlichen ist, die Möglichkeit der genauen Abstimmung der Höhe. Der 3D-Drucker hat in der Höhe eine Auflösung von 0.1 Millimetern. Außerdem wurde die Unterlegscheibe so designet, dass das Gewicht des Perlenmagazins auf die Rotationsscheibe gleich verteilt wird. In der ersten Version der Unterlegscheibe ist diese 1,2 Millimeter dick. Beim Testen dieser Version wurde festgestellt, dass die Perlen im unteren Teil des Perlenmagazins zu viel Bewegungsfreiheit haben. Aus diesem Grund wurde die Höhe der Unterlegscheibe weiter verringert. Wie in Abbildung 20 zu sehen ist, musste dieser Wert mehrmals angepasst werden, bis die Unterlegscheibe den gewünschten Zweck erfüllt. Mit einer Höhe von nur 0,3 Millimetern erfüllt die Unterlegscheibe ihren Zweck. Bei dieser Höhe haben die Perlen wenig Bewegungsfreiraum und die Perlenmagazine können den Schieber nicht mehr blockieren.

Versionsnummer	Druchmesser
0.1	1,2 mm
0.2	0,9 mm
0.3	0,4 mm
0.4	0,3 mm

Abbildung 21: Ermittlung der Höhe der Unterlegscheibe

5.3 Fehlerkorrektur

Die Fehler E1, E2 und E3 beschreiben ein fehlerhaftes Bügelperlenbild. Beim Fehler E1 wurde eine Perle der falschen Farbe gesetzt. Bei den Fehlern E2 und E3 wurde entweder eine Perle auf ein Feld gesetzt, auf das keine Perle

gehört, oder auf ein Feld, auf das eine gesetzt werden sollte, keine gesetzt. Da diese Fehler den Zustand des Bügelperlenbildes nach dem Druckvorgang beschreiben, müssen diese auch nach dem Druckvorgang gelöst werden. Dazu wird eine Fehlerkorrektur durchgeführt.

5.3.1 Planung

Zunächst wird der Ablauf des Korrekturprozesses geplant. In Abbildung 22 ist ein Ablaufdiagramm dieses Fehlerkorrekturprozesses zu sehen. Der Korrekturprozess startet nach dem normalen Druckprozess. Auf dem Stiftbrett befindet sich ein gedrucktes Bild, welches nun auf Fehler überprüft werden kann und ggf. korrigiert werden soll. Dazu wird zunächst der Druckbereich mit der Kamera aufgenommen. Anschließend muss der Farbwert für jede Perle bestimmt werden. Dazu muss das System in der Lage sein, die einzelnen Perlen zu identifizieren. Ist dies geschehen, so werden die ermittelten Farbwerte mit den erwarteten Werten des Eingabebildes verglichen. Wie im Ablaufdiagramm zu sehen ist, endet der Korrekturprozess, sobald beide übereinstimmen. Stimmen sie nicht überein, so wird der rechte Teil des Ablaufdiagramms ausgeführt. Dort werden als erstes die Perlen bestimmt, die nicht richtig gesetzt, oder vergessen wurden. Anschließend wird der Benutzer aufgefordert alle Perlen die falsch gesetzt wurden zu entfernen. Dieser Vorgang wird mit einer erneuten Aufnahme und Überprüfung eines Kamerabildes kontrolliert. Wurden vom Benutzer nicht alle Perlen korrekt entfernt, so wird er aufgefordert dies zu tun, bis alle falschen Perlen entfernt wurden. Wurden alle falschen Perlen korrekt entfernt, so werden die vorher als fehlerhaft identifizierten Perlen gedruckt. Anschließend wird wieder zum Anfang zurückgesprungen und der Fehlerkorrekturprozess beginnt von vorne. Dieser Prozess wird solange ausgeführt, bis das Bild fehlerfrei gedruckt wurde.

5.3.2 Kamerahalterung

Damit ein Bild des Stiftbrettes mit einer Kamera aufgenommen werden kann, muss zunächst eine Kamera in geeignetem Abstand zum Stiftbrett angebracht werden. Eine gute Position für eine Kamera wäre direkt über dem Stiftbrett in etwa 15 bis 20 cm Abstand. Da an der Beadbox dort keine Möglichkeit besteht eine Kamera anzubringen, muss zunächst eine Kamerahalterung gebaut werden. Um die Kamera in der gewünschten Höhe anbringen zu können, wird ein Metallstreifen verwendet. Wie in Abbildung 23 zu sehen ist, hat dieser Metallstreifen mehrere Löcher und kann so leicht durch Schrauben befestigt

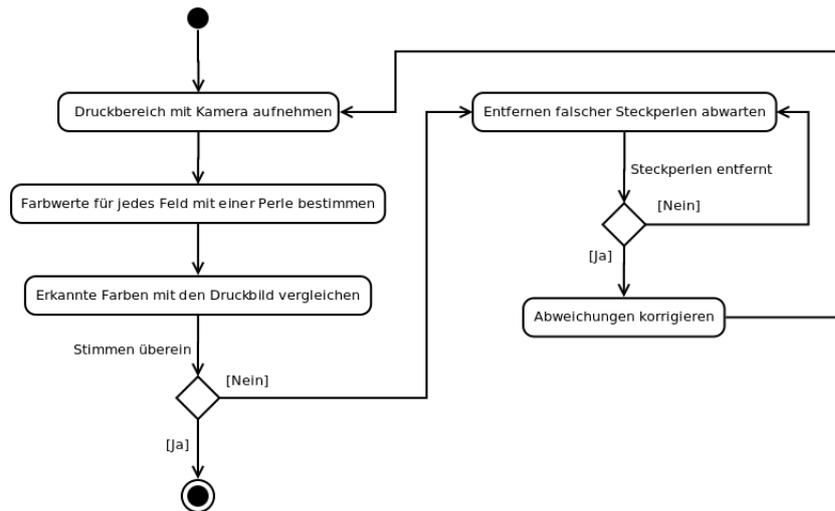


Abbildung 22: Ablaufdiagramm Fehlerkorrektur

werden.



Abbildung 23: Metallstreifen für Kameralhalterung

Für die Befestigung an der Beadbox wurde ein Adapter mit dem 3D-Drucker entworfen. Dieser ist in Abbildung 24 zu sehen.

Der Adapter wird mit dem Metallstreifen über vier Schrauben verbunden und mit zwei weiteren Schrauben oberhalb des Setzrohrs mit der Beadbox verschraubt. Die Kamera kann dadurch zusammen mit dem Setzrohr an jede Position des Stiftbretts gefahren werden. Der Adapter wurde, genau wie der Schieber und das Setzrohr in mehreren Iterationen entwickelt. Zunächst wurde ein Prototyp entwickelt, der den Metallstreifen mit der Beadbox verband. Dieser Prototyp hatte noch nicht genug Stabilität, um den Metallstreifen zu halten. Mehrere Prototypen waren nötig, bis der Adapter die benötigte Stabilität hatte und sowohl den Metallstreifen halten, als auch gut an der Beadbox befestigt war. Der Metallstreifen war etwas zu schwer, wodurch eine Leiste auf den Schieber drückte und diesen blockierte. Aus diesem Grund wurde er durch ein Holzbrett ersetzt. Die fertige Konstruktion ist in Abbildung 25 zu sehen.

Die Kamera ist etwa 30 Zentimeter oberhalb des Stiftbretts angebracht. In dieser Höhe erfasst die Kamera etwa ein Viertel des gesamten Stiftbretts. Damit die Kamera mehr vom Stiftbrett erfasst, müsste sie deutlich höher ange-

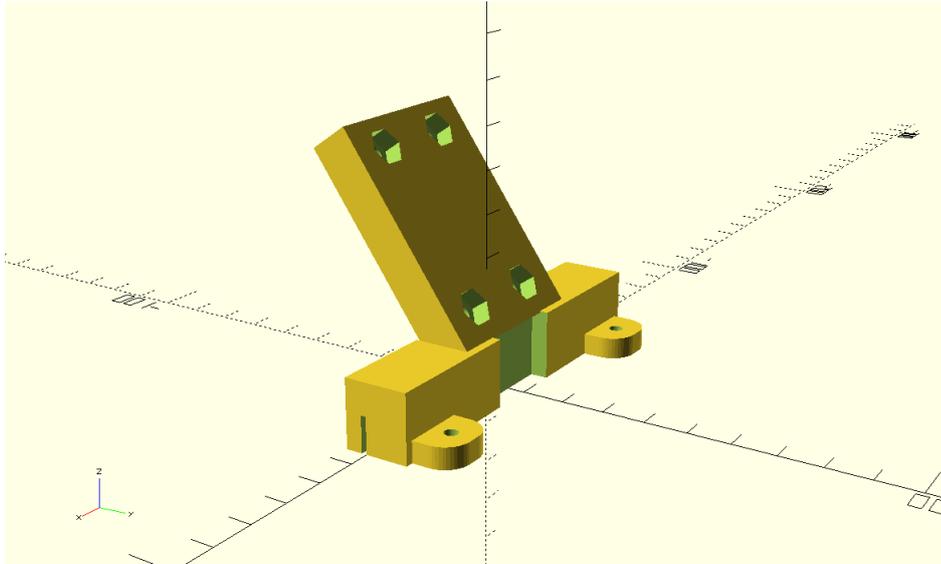


Abbildung 24: Adapter für Kamerahalterung

bracht werden. Da eine solche Konstruktion schwieriger zu realisieren ist, wird der Korrekturprozess auf Bilder der Größe 20 x 20 eingeschränkt. Die Software wird so entwickelt, dass eine einfache Anpassung an größere Druckbereiche möglich ist.

5.3.3 Die Kamera

Für diese Arbeit wird eine Logitech Kamera des Modells C270 verwendet. Diese Webcam kann mit einem USB-Kabel angeschlossen werden und hat eine Auflösung von 1280 x 720 Pixeln. Die Kamera besitzt eine automatische Lichtkorrektur, wodurch sie auch in schlechter ausgeleuchteten Umgebungen noch gute Bilder machen kann [Log]. Diese Kamera wurde ausgewählt, da sie sowohl unter den neueren Windows-Betriebssystemen, als auch unter den meisten Linux-Distributionen funktioniert [ide].

Kamera kalibrieren Die verwendete Kamera ist eine sogenannte Lochkamera. Diese Art von Kameras zeichnen sich durch ihren niedrigen Preis und sind deswegen auch die meist verbreitetste Art von Kameras [dev]. Leider kommt es bei Lochkameras oft zu Verzerrungen des aufgenommenen Bildes. Dieser Verzerrungseffekt wird *barrel*-Effekt (Fass-Effekt) genannt, da das Bild aussieht, als würde es um ein Fass oder eine Kugel gebogen werden. Dieser Effekt ist in Abbildung 26a mit einem Gittermuster dargestellt. Das abgebildete Gittermuster wird zur Mitte hin nach vorne gebogen. Wird diese Verzerrung nicht mit einbezogen, so führt sie zu verfälschten Ergebnissen, bei der Bestim-

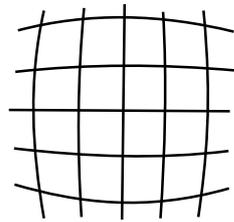


Abbildung 25: Fertige Kamerahalterung

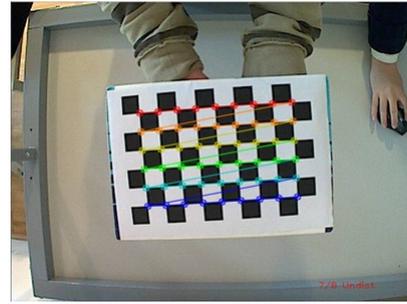
mung der Position von Objekten im Bild. Darum ist es wichtig diese Verzerrung zu korrigieren. Dafür stellt *openCV* die Funktion „`undistort()`“ bereit. Diese kann bei Übergabe der richtigen Funktionsparameter die Verzerrung herausrechnen. Wie in der Dokumentation von *openCV* [dev] beschrieben, werden dafür die *distortion*-Parameter benötigt. Sie werden in einer 1x5 Matrix angegeben:

$$\mathbf{Distortion}_{\text{coefficients}} = (k_1, k_2, p_1, p_2, k_3)$$

Diese Parameter müssen für jede Kamera einmalig bestimmt werden. Um sie zu bestimmen, wird in der Dokumentation [dev] ein Programm bereitgestellt. Damit das Programm die Parameter bestimmen kann, muss auf ein Blatt ein Muster gedruckt werden. Von dem Muster werden anschließend mit der Kamera Fotos aus verschiedenen Blickwinkeln gemacht. Das Programm erkennt die Muster im Bild und kann anhand dieser die Verzerrung bestimmen. Es können drei verschiedene Muster zur Kalibrierung verwendet werden: ein Schachbrettmuster, ein Muster mit symmetrischen Kreisen und ein Muster mit asymmetrischen Kreisen. Für diese Arbeit wird das Schachbrettmuster zur Kalibrierung verwendet. Dieses ist in Abbildung 26b zu sehen.



(a) Barrel-Effekt



(b) Schachbrettmuster

Abbildung 26: Kamerakalibrierung

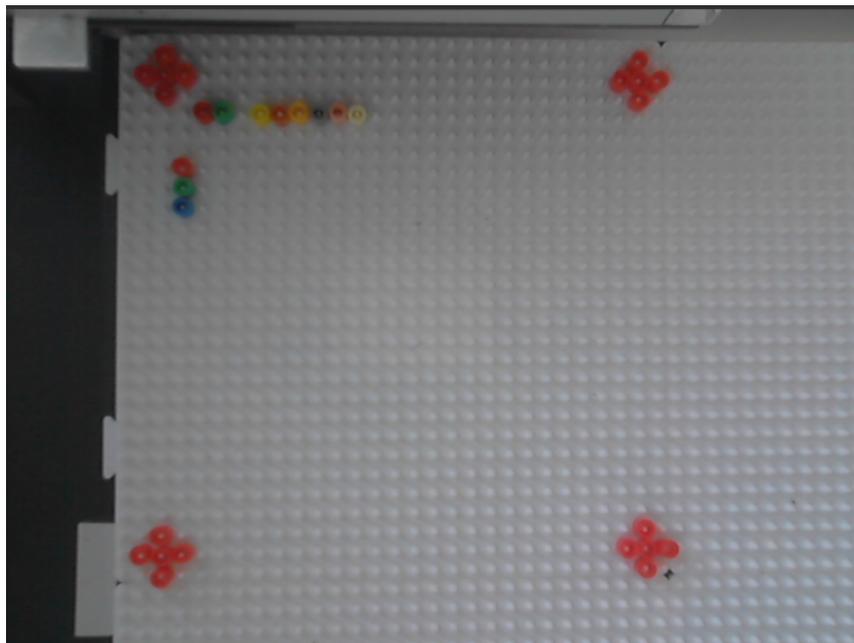


Abbildung 27: Kamerabild

Kamera positionieren Damit die Kamera den Druckbereich aufnehmen kann, muss diese zunächst mittig über dem selbigen positioniert werden. Die im Abschnitt Kamerahaltung beschriebene Halterung ist mittig über dem Setzrohr angebracht. Um die Kamera mittig über dem Druckbereich zu platzieren, muss nur das Setzrohr mittig platziert werden. Auf der anderen Achse muss das Setzrohr soweit zurückgefahren werden, dass es den Druckbereich nicht verdeckt. Die Position $(15, -2)$ erfüllt diese Bedingungen. Die Kamera wird bevor ein Bild gemacht wird zunächst an diese Position gefahren.

5.3.4 Das Bild ausrichten

Mit einer über dem Stiftbrett angebrachten Kamera wird ein Bild vom Stiftbrett aufgenommen. Ein solches Bild ist in Abbildung 27 zu sehen.

Wie in der Abbildung zu erkennen ist, so ist die Kamera nicht genau auf das Stiftbrett ausgerichtet. Sie ist leicht zur rechten Seite geneigt. Um die einzelnen Perlen im Bild genau identifizieren zu können, wäre es gut, wenn das aufgenommene Bild direkt von oben aufgenommen wurde und keine perspektivischen Verzerrungen vorhanden sind. Die Kamera so auszurichten gestaltet sich als schwierig und die Ausrichtung kann z.B. durch den Transport der Beadbox wieder verfälscht werden. Eine gewisse Toleranz gegenüber der Kameraausrichtung wäre wünschenswert. Um dies zu erreichen, muss zunächst bestimmt werden in welcher Lage sich das Stiftbrett zur Kamera befindet. Dies kann erreicht werden, indem auf dem Stiftbrett Markierungen angebracht werden, die anschließend im Bild wiedererkannt werden. Diese Markierungen sind in der Abbildung 27 zu sehen. Sie bestehen aus roten Steckperlen, die in einem Kreuz angeordnet wurden. Die vier Marker befinden sich in den Ecken des Stiftbrettes und grenzen die Druckfläche ein. Für die Erkennung von Objekten in einem Bild, stellt *opencv* mehrere Methoden bereit. Der SURF-Algorithmus [BTVG06] ist einer dieser Methoden. Das Akronym SURF steht für „Speeded Up Robust Features“. Der Algorithmus kann markante Stellen, die sogenannten *Features* in einem Bild ausfindig machen und diese mit einem anderen Bild vergleichen. So können ähnliche Objekte in einem Bild gefunden werden. Eine Andere etwas simplere Methode ist das Template-Matching, welches unter [teaf] beschrieben wird. Bei dieser Methode wird ein Bild Schritt für Schritt mit einer Schablone verglichen wird. Liegt der Wert der Übereinstimmung oberhalb einer vorher angegebenen Grenze, so wurde das gesuchte Objekt im Bild erkannt. Da der Template-Matching Algorithmus einfacher zu benutzen ist und mit ihm gute Ergebnisse erzielt werden können, wird dieser verwendet.

Template Matching Zur Erkennung der Marker wird die *Template Matching*-Methode aus der *opencv library* benutzt [teaf]. Ein *template*, ist ein Beispieldes gesuchten Objekts. Der Algorithmus ist in der Lage diese Beispieldes in einem anderen Bild wiederzufinden. In Abbildung 28 ist eine Schablone abgebildet. Diese zeigt einen der vier Marker, die in Abbildung 27 zu sehen sind. Die Marker sind mit roten Steckperlen auf das Stiftbrett gelegt und wurden mit einem Bügeleisen miteinander verschmolzen. Wie in [teaf] beschrieben, legt der Algorithmus, um die Markierungen zu finden, die Schablone zunächst oben links über das Bild und vergleicht sie mit dem darunterliegenden Ausschnitt. Da dieser Ausschnitt nicht sehr gut mit der Schablone übereinstimmt

berechnet der Algorithmus einen niedrigen Übereinstimmungswert. Die Schablone wird um ein Feld weiter nach rechts verschoben und der Vergleich wiederholt. Dieser Vorgang wird für das gesamte Bild fortgesetzt, sodass am Ende die Schablone an jede mögliche Position geschoben und der zugehörige Wert der Übereinstimmung berechnet wurde.



Abbildung 28: Schablone

Als Ergebnis liefert der Algorithmus die Koordinaten der Punkte an denen der Wert der Übereinstimmung größer als ein vorher definierter Schwellenwert ist. Je nach Angabe des Schwellenwerts und Aufbau des Markers, können mit diesem Verfahren auch leicht gedrehte oder geneigte Marker erkannt werden. Wurden genau vier Marker im Bild gefunden, so wird mit dem nächsten Schritt, der Transformation, fortgefahren. Werden weniger Marker gefunden, weil einer beispielsweise verdeckt ist, ist es nicht möglich den nächsten Schritt auszuführen und der Algorithmus muss die Überprüfung abbrechen und die Kamera manuell neu ausgerichtet werden. Werden hingegen mehr als vier Marker gefunden, weil im zu druckenden Bild selbst z.B. ähnliche Muster existieren, so stellt dies kein Problem dar. Die Funktion wurde so geschrieben, dass zu allen erkannten Markern der Abstand zu den jeweiligen Eckpunkten des Kamerabilds bestimmt werden. Anschließend werden die vier Ausgewählt, die der jeweiligen Ecke am nächsten sind. Dadurch ist gewährleistet, dass zum einen die Marker außerhalb des Druckbilds gefunden werden und zum anderen die Marker den richtigen Eckpunkten zugeordnet sind. Der letzte Punkt ist für die im folgenden beschriebene Transformation wichtig, damit das Bild nicht verdreht, gespiegelt oder gar verzerrt wird.

Transformation Nachdem nun die vier Marker im Bild erkannt wurden, wird das Bild so transformiert, dass es zur Kamera ausgerichtet ist. Im ersten Versuch wurde das Bild mit einer affinen Transformation von den vier Markern zu den vier Eckpunkten des Bildes transformiert. Bei einer affinen Transformation wird eine 2×3 Matrix zur Transformation benutzt [teaa]. Diese ermöglicht im 2-Dimensionalen Raum die vier linearen Operationen: Verschiebung, Rotation, Skalierung, Scherung. Betrachten wir das linke Bild in Abbildung 30. Der erkannte Bildausschnitt ist schwarz umrandet. Bei genauerer Betrachtung sehen wir, dass keine der eben genannten Operationen aus dieser Umrandung

ein Rechteck erzeugen kann. Eine der Eckpunkte würde immer entweder zu weit im Bild, oder außerhalb des Bildes liegen. Das liegt daran, dass das Bild perspektivisch verzerrt ist. Wird mit einer Kamera ein Bild aufgenommen, so sind Dinge die näher an der Kamera dran sind größer als Dinge die weiter weg sind. In Abbildung 29 ist der Unterschied zwischen affiner und perspektivischer Transformation noch einmal übersichtlich dargestellt. Wie zu sehen ist, bleiben bei einer affinen Transformation die Seitenlängen der jeweils gegenüberliegenden Seiten gleich lang. Dies ist bei der perspektivischen Transformation nicht der Fall.

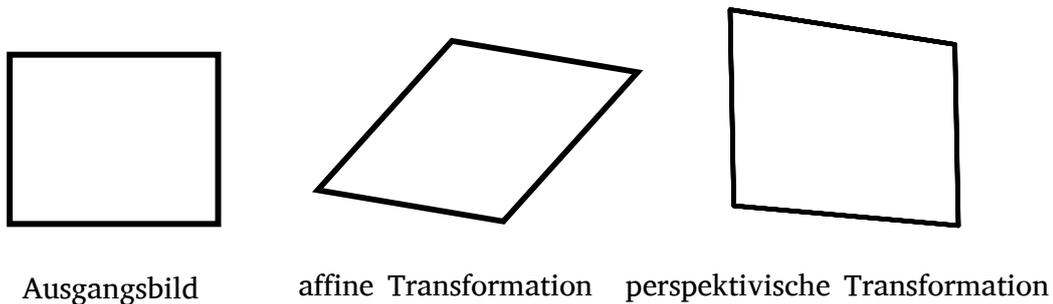


Abbildung 29: Transformationen

Mit einer affinen Transformation können wir unser Bild somit nicht korrigieren. Es wird eine Funktion benötigt, die eine perspektivische Transformation bereitstellt. Diese wird ebenfalls von der Library *opencv* unter der Funktion *warpPerspective* [teae] zur Verfügung gestellt. Bei einer perspektivischen Transformation wird im 2D-Raum eine 3x3 Matrix verwendet. Nach der perspektivischen Transformation ist das Bild wie gewünscht zugeschnitten. In Abbildung 30 ist gut zu erkennen, dass die jeweiligen Mittelpunkte der Marker genau in den Eckpunkten des Rechteckigen Bildausschnittes liegen.

Der erkannte Druckbereich wird in ein neues Bild transformiert.

5.3.5 Farberkennung

Wie Andrew J. Schofield in seinem Artikel [S⁺14] schreibt, sind Menschen gut darin zwischen räumlichen Veränderungen in der Beleuchtung und räumlichen Veränderungen von Reflexionen zu unterscheiden. Er verweist dabei auf das in Abbildung 31 gezeigte Bild von Edward H. Anderson mit dem Titel „Checker Shadow Illusion“. Betrachten wir zunächst das links abgebildete Schachbrett. Obwohl die mit A und B markierten Felder den selben Grauton haben, erscheint uns das Feld B deutlich heller als das Feld A. Wir beziehen völlig automatisch den Schattenwurf des Objektes mit in unsere Betrachtung ein und

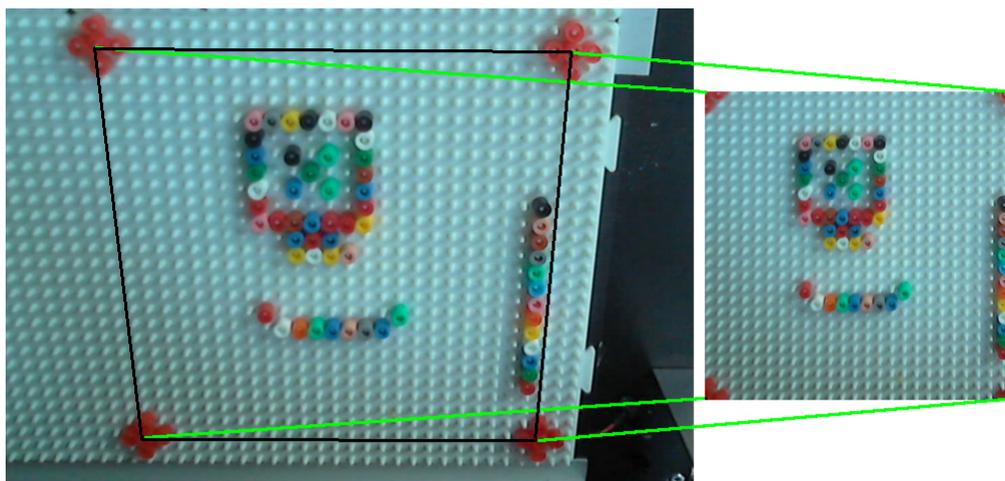
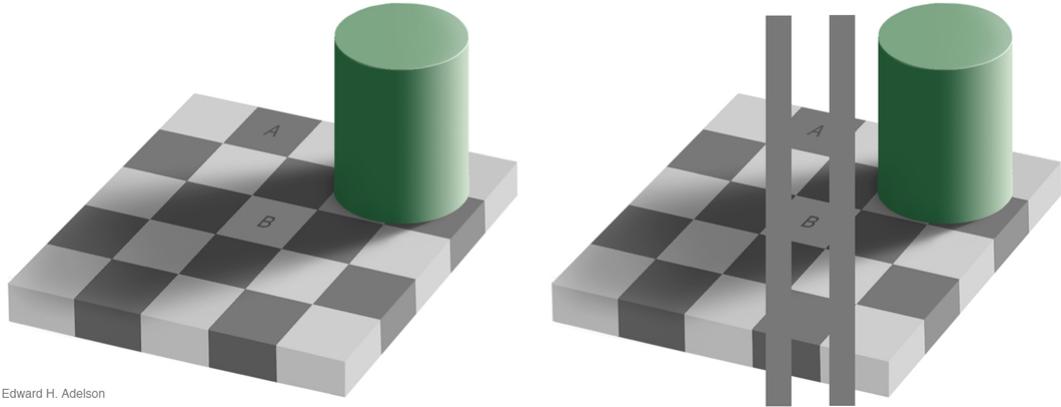


Abbildung 30: Kamerabild Ausschnitt

alle Felder, die auf dem Schachbrett diagonal in einer Reihe angeordnet sind, scheinen von derselben Farbe zu sein. Betrachten wir nun das auf der rechten Seite abgebildete Schachbrett. Die Abbildung ist genau die gleiche wie die auf der linken Seite, mit dem Unterschied, dass zwei graue Balken über das Bild gelegt wurden. Durch diese Balken ist es plötzlich möglich nur die Farbwerte der beiden Felder A und B zu betrachten. Wir sehen, dass die beiden Felder exakt den gleichen Grauton besitzen.

Dieses Beispiel soll verdeutlichen, wie wir Menschen ohne Probleme das Schachbrett als solches erkennen, obwohl der Schattenwurf des zylinderförmigen Objektes die Farben der Felder verändert. Wir können dies, da wir das Bild interpretieren. Wir erkennen, dass ein Objekt auf dem Schachbrett steht und einen Schatten auf dieses wirft. Da uns klar ist, dass dieser Schatten die Farben abdunkelt korrigiert unser Gehirn dieses und wir können das Muster korrekt wahrnehmen. Ein Computerprogramm hat diese räumlichen Informationen über die Objekte im Bild in der Regel nicht. Ohne diese Informationen würde das Computerprogramm die Felder A und B als Felder der gleichen Farbe erkennen.

Ähnliche Probleme ergeben sich bei der Farberkennung der Bügelperlen auf dem Stiftbrett. Auch hier spielen Beleuchtung und Schattenwurf eine wichtige Rolle. Im Folgenden wird erläutert, wie ein Verfahren entwickelt wurde, um die Perlenfarben zu bestimmen. Dazu wird das in Abbildung 32 gezeigte Bild als Beispiel verwendet. Dieses wurde wie im vorherigen Abschnitt beschrieben zugeschnitten und jetzt sollen die Farben der Perlen bestimmt werden.



Edward H. Adelson

Abbildung 31: Checker Shadow Illusion [EH]



Abbildung 32: Kamerabild zugeschnitten

Raster Nachdem das Bild nun genau zugeschnitten ist, muss zunächst die Position der einzelnen Steckperlenfelder im Bild bestimmt werden. Jedes Perlenfeld des Stiftbretts kann über seine Position auf der X- und Y-Achse identifiziert werden. Der Nullpunkt dieses Koordinatensystems liegt in der oberen linken Ecke des Stiftbretts. Die X-Achse liegt in horizontaler Richtung, die Y-Achse in vertikaler Richtung über dem Stiftbrett. Der in Abbildung 30 gezeigte Bildausschnitt zeigt 23 Perlenfelder in X-Richtung und 23 in Y-Richtung. Der Druckbereich umfasst somit 529 Perlenfelder. Das Bild ist 552 x 552 Pixel groß. In den Ecken dieses Bilds befinden sich die Mittelpunkte der Marker, die zur Transformation verwendet wurden. Im Mittelpunkt der Marker ist je ein Stift des Stiftbretts zu sehen. Zwischen je zwei Ecken befinden sich genau 24 Stifte. Die Position eines Steckperlenfelds im Bild kann diesen Werten leicht bestimmen werden. Dazu berechnen wir zunächst die Größe eines Perlenfelds in Pixel des Bilds. Die Pixelgröße einer Steckperlenfelds SF_g kann berechnet werden indem die Bildauflösung $A_x = 552$ (ist vertikal und horizontal gleich) durch die Anzahl Stifte $Anz(Stifte) = 24$. Da an den Kanten des Bilds je eine Reihe Stifte liegen, was einem halben Steckperlenfeld entspricht, werden bei der Berechnung 24 und nicht 23 Perlenfelder benutzt.

$$SF_g = A_x / Anz(Stifte)$$

$$SF_g = 552 \text{ Pixel} / 24 \text{ Perlen}$$

$$SF_g = 23 \text{ Pixel pro Perle}$$

Das Feld einer Perle ist somit 23×23 Pixel groß. Da die halben Randfelder ausgelassen werden, befindet sich das erste Perlenfeld an der Position $P_1 = (12, 12)$ im Bild, wobei der erste Wert die horizontale X-Achse beschreibt und der zweite Wert die vertikale Y-Achse. Der Nullpunkt liegt in der oberen linken Ecke. Da jetzt alle Parameter für das Raster bekannt sind, können die einzelnen Felder bestimmt werden. Eine Umrechnung von Stiftbrettfeld zu Rasterfeld kann durchgeführt werden. Ein Feld wird durch seinen Ankerpunkt (obere linke Ecke des Perlenfelds) und seine Ausmaße in x - und y -Richtung bestimmt. Der Ankerpunkt wird wie folgt berechnet:

$$Ankerpunkt(x_f) = 12 + 23 * x_f$$

$$Ankerpunkt(y_f) = 12 + 23 * x_f$$

Das Feld (4, 8) hat seinen Ankerpunkt im Bild auf dem Punkt (104, 196). Das mit einem Raster versehene Bild ist in Abbildung 34a zu sehen.

Farbwertdurchschnitt berechnen Nachdem mithilfe des Rasters die Felder des Stiftbrettes bestimmt wurden, muss als nächstes der Farbwert dieser Felder ermittelt werden. Wie schon beschrieben, hat ein Perlenfeld Ausmaße von 23×23 Pixeln und umfasst somit 529 Pixel. Jedes dieser Pixel hat eine Farbe, die im RGB-Farbraum dargestellt wird. Im RGB-Farbraum werden Farben durch ihren Rot-, Grün- und Blauanteil dargestellt. Die Farbanteile können ganzzahlige Werte im Bereich von Null bis 255 annehmen. Eine Farbe im RGB-Farbraum kann durch ein Tupel dieser drei Werte dargestellt werden:

(Anteil Rot, Anteil Grün, Anteil Blau)

Die Farbe Gelb hat beispielsweise einen hohen Rot- und Grünanteil und keinen Blau-Anteil. Sie kann mit dem Tupel (255, 255, 0) dargestellt werden. Um nun den Farbwert eines Felds zu ermitteln, bestimmen wir das arithmetische Mittel der Farben in diesem Feld. Das Ergebnis dieser Berechnung ergibt die Durchschnittsfarbe. Diese kann mit einer Funktion $mean()$ für das Feld F für jede Farbanteil $c \in \{Rot, Grn, Blau\}$ berechnet werden:

$$mean(F_c) = \sum_{i=0}^{529} c_i / 529$$

Die Durchschnittsfarbe für das Feld F kann mit dem Tupel ($mean(F_{\mathbf{Rot}})$, $mean(F_{\mathbf{Grün}})$, $mean(F_{\mathbf{Blau}})$) dargestellt werden. In Abbildung 33 ist auf der linken Seite ein Perlenfeld mit einer gelben Perle abgebildet. Auf der rechten Seite ist die berechnete Durchschnittsfarbe für dieses Feld abgebildet.

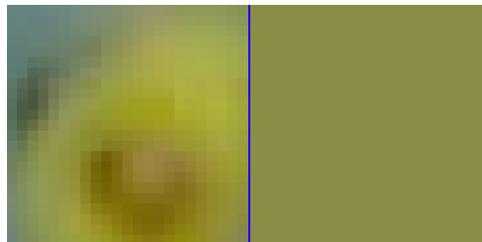


Abbildung 33: Durchschnittsfarbe für ein Perlenfeld

Die Durchschnittsfarbe hat den Farbwert (137, 141, 70). Die Perle im Bild hat einen helleren Gelbton, als die berechnete Durchschnittsfarbe. Das liegt vor allem daran, dass im Bild mit der Perle oben links ein Teil des Stiftbretts

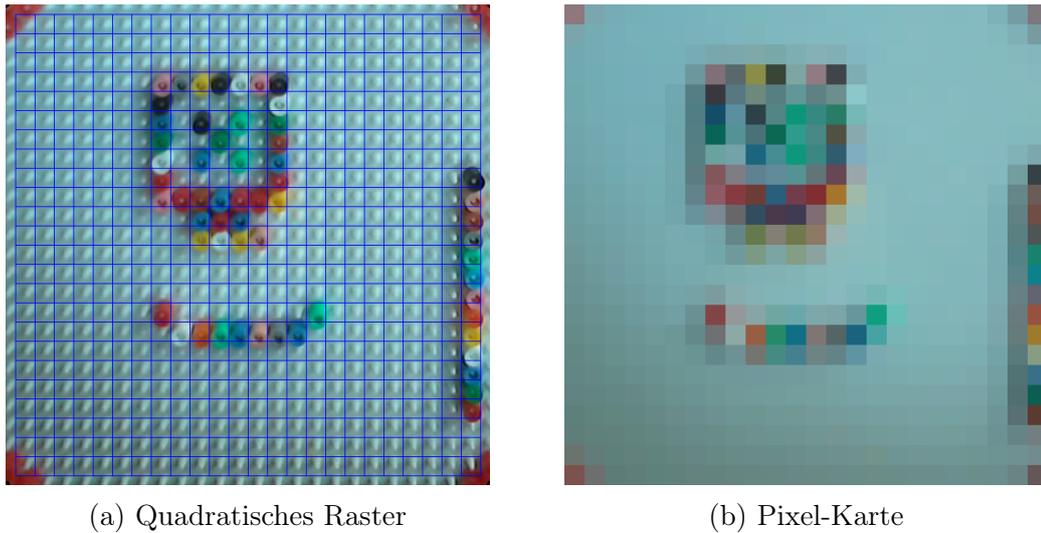


Abbildung 34: Pixel-Karte

mit zu sehen ist und in der Mitte der Perle die Perlenfarbe durch Schattenwurf verdunkelt wird. Berechnet man die Farben für jedes in Abbildung 34a dargestellte Raster, so erhält man 529 Mittelwerte. Diese können ebenfalls als ein Bild angeordnet werden. Dieses Bild bezeichnen wir als Pixel-Karte. Jedes Feld im Raster hat einen zugehörigen Pixel in der Pixel-Karte, der den Durchschnittsfarbwert dieses Felds angibt. In Abbildung 34b ist die Pixel-Karte für das Raster aus Abbildung 34a zu sehen.

Farberkennung Ansatz Vektordistanz In Abbildung 35a ist das Druckbild für die in Abbildung 34b ermittelten Durchschnittswerte zu sehen. Ver-

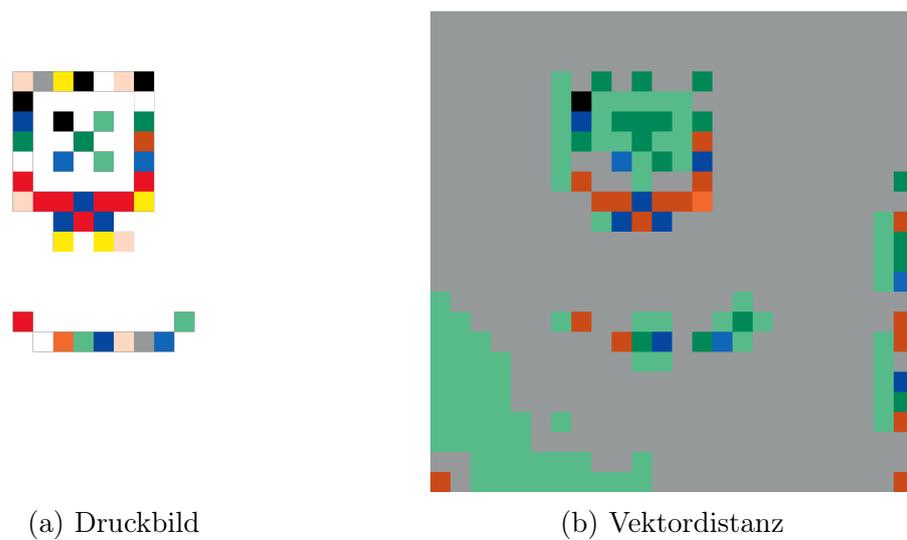


Abbildung 35: Farberkennung

gleicht man die beiden Bilder, so fällt auf, dass Werte der Durchschnittsfarben nicht den Werten der gedruckten Farben entsprechen. Das liegt zum einen daran, dass wir den Mittelwert der Perlen berechnet haben und dadurch, wie schon vorher beschrieben, die Farbwerte verändert werden. Außerdem werden die Farben, wie am Anfang des Abschnitts beschrieben, durch unterschiedliche Beleuchtung, sowie die Qualität des Bildsensors in der Kamera beeinflusst. Um die richtige Farbe aus der Pixel-Karte zu bestimmen, wird eine Zuordnung der berechneten Durchschnittswerte zu den Perlenfarben benötigt. Die verschiedenen Perlenfarben sind in Abbildung 36 abgebildet.



Abbildung 36: Farbpalette

Diese Farben können als Farbpalette für ein Perlenbild bezeichnet werden. Eine Farbpalette umfasst alle Farben, die in einem Bild benutzt werden können. Für die Farberkennung wird eine Zuordnung gesucht, sodass alle Felder mit einer Perle genau einer Farbe dieser Farbpalette zugeordnet wird. Außerdem soll die zugeordnete Farbe der tatsächlichen gesetzten Perlenfarbe entsprechen. Da wir noch keine Möglichkeit haben, die Felder mit einer Perle von denen ohne zu unterscheiden, ordnen wir vorerst jedem Perlenfeld eine Farbe der Farbpalette zu.

Um die ermittelten Durchschnittswerte der Farbpalette zuzuordnen betrachten wir jede Farbe als Vektor im dreidimensionalen Raum. Die Farbanteile für Rot, Gelb und Grün stehen dabei je für einen Freiheitsgrad in diesem Raum:

$$\vec{v} = \begin{pmatrix} Rot \\ Gelb \\ Blau \end{pmatrix}$$

Die Farbe Gelb (255, 255, 0) hat beispielsweise den zugehörigen Vektor $\vec{g} = (255, 255, 0)^T$. Da die Koordinaten eines Vektors mit den Farbanteilen zusammenhängen, liegen Farben deren RGB-Wert ähnlich ist dicht beieinander. Um zu bestimmen, welcher Farbe ein berechneter Durchschnittswert zugeordnet werden soll, kann der Farbwert ausgewählt werden, zu dem die Distanz am niedrigsten ist. Um die Distanz im dreidimensionalen Raum zu bestimmen, wird der euklidische Abstand zwischen zwei Punkten berechnet. Der Abstand zwischen zwei Punkten x und y kann wie folgt bestimmt werden:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

												
233	35	83	300	269	219	237	89	98	138	194	264	155
157	138	176	244	161	113	158	173	99	92	103	191	205

Abbildung 37: Farberkennung mit zwei Beispielfarben

Der Abstand der Farben $f_1 = (100, 3, 40)$ und $f_2 = (80, 5, 55)$ ist beispielsweise 25, 08. Für jeden Pixel der Pixel-Karte wird nun der Abstand zu jeder Farbe der Farbpalette berechnet. Anschließend wird die Farbe der Palette ausgewählt, bei der die Distanz minimal ist und ersetzt die Durchschnittsfarbe mit der Ausgewählten. Wendet man dieses Verfahren auf die Pixel-Karte aus Abbildung 34b an, so erhält man das in Abbildung 35b zu sehende Bild. Vergleichen wird diese beiden Bilder, so fällt zum einen auf, dass Feldern ohne Perlen einen Farbwert haben, der nicht in jedem Fall der weißen Perlenfarbe entspricht. Da das Stiftbrett eine weiße Farbe hat, wäre diese Zuordnung naheliegend. Des Weiteren ist zu erkennen, dass die Felder auf denen sich Perlen befinden nicht immer den richtigen Farben zugeordnet wurden. Von insgesamt 45 gesetzten Perlen wurden gerade mal zehn Perlen der richtigen Farbe zugeordnet. In Abbildung 37 wurden die berechneten Werte für zwei aus dem Bild ausgewählten Farben aufgeführt. Die erste Perle ist grün und wird korrekt zugeordnet. Die zweite Perle ist gelb und wird der Farbe Grau zugeordnet. Die Farben der Pixel-Karte sind zu stark verfälscht, um die richtigen Farben zuordnen zu können. Das Verfahren zum Erstellen dieser Pixel-Karte soll im Folgenden weiter angepasst werden, um bessere Ergebnisse erzielen zu können.

Anpassung der Lichtverhältnisse Wie am Anfang des Kapitels mit dem Schachbrettbeispiel erläutert, beeinflussen Reflexionen und Schatten die mit einer Kamera aufgenommenen Farbwerte. Um das Ergebnis der Farberkennung mit der Vektordistanz zu verbessern, soll der Druckbereich mit einer Lichtquelle beleuchtet werden. Es wurde eine Lichtquelle über dem Stiftbrett angebracht und auf den Druckbereich ausgerichtet. Das Ergebnis ist in Abbildung 38 zu sehen. In Bild 38a ist gut zu erkennen, dass das Licht nicht gleichmäßig verteilt wird.

Das Licht ist in der Mitte des Stiftbretts am hellsten und wird zu den Ecken hin dunkler. Damit dieser Effekt gut zu erkennen ist, wurden für die Aufnahme alle anderen Lichtquellen weitestgehend ausgeschaltet. Außerdem kann im Bild erkannt werden, dass sowohl auf dem Stiftbrett, als auch auf den Bügelperlen

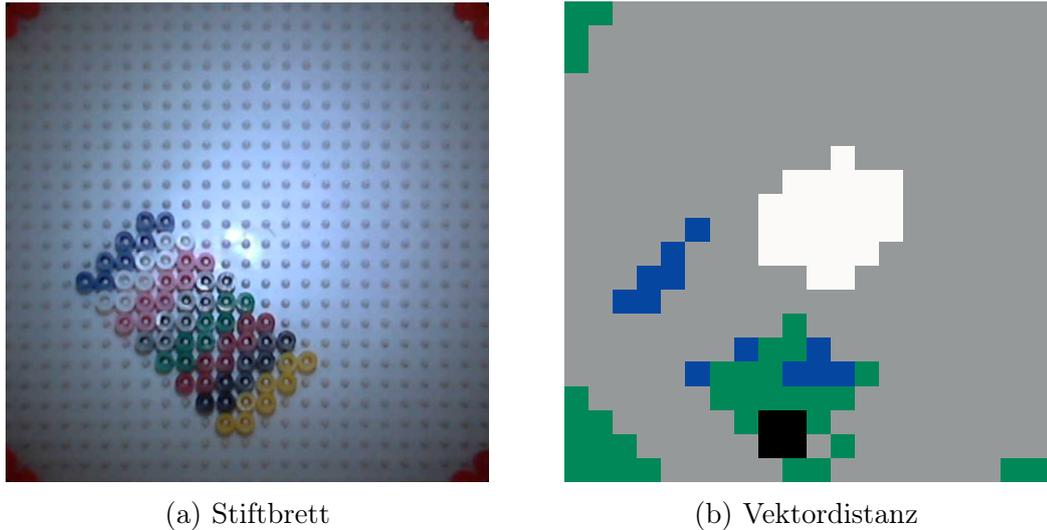


Abbildung 38: Vektordistanz mit Beleuchtung

Reflexionen entstehen. Auch mit vorhandener Umgebungsbeleuchtung wirkt sich dieser Effekt noch sehr stark auf die Farberkennung aus und verschlechtert diese deutlich. Auch durch die Verwendung von mehreren Lichtquellen konnten keine guten Ergebnisse erzielt werden. Aus diesem Grund wird nach weiteren Möglichkeiten gesucht die Farberkennung zu verbessern.

Anpassung des Rasters Um die Ergebnisse der Durchschnittsberechnung zu verbessern wird das Raster angepasst. Das ursprüngliche Raster hatte quadratische Felder. Die Bügelperlen selbst sind von oben betrachtet kreisförmig. Da eine Bügelperle ein quadratisches Feld nicht ganz ausfüllt, ist es besser das Raster aus runden Felder aufzubauen. Dies ist auch auf der vorher schon gezeigten Abbildung 33 zu sehen. Je nachdem wie die Kamera die Perle aufnimmt, sind Teile des Stiftbretts mit in diesem Feld. Es macht daher mehr Sinn ein Raster aus Kreisen über das Bild zu legen und nur die Pixel innerhalb der Kreise für die Berechnung der Durchschnittsfarben zu verwenden. Im ersten Ansatz sollte ein Algorithmus zur Erkennung von Kreisen angewandt werden. Das automatische erkennen der Bügelperlen hat gegenüber der Methode mit dem Raster den Vorteil, dass die Felder auf denen auch tatsächlich Perlen sind identifiziert werden. Das *openCV*-Framework bietet dazu die Funktion *HoughCircles* an. Diese benutzt eine Hough Transformation, welche in [teac] und [tead] beschrieben wird. Der Funktion *HoughCircles* werden als Parameter der minimale und maximale Radius der zu erkennenden Kreise mitgegeben. Diese Parameter wurden an die Kreisgröße der Steckperlen angepasst. Beim Testen der Funktion stellte sich heraus, dass diese zum Erkennen der Bügelperlen nur

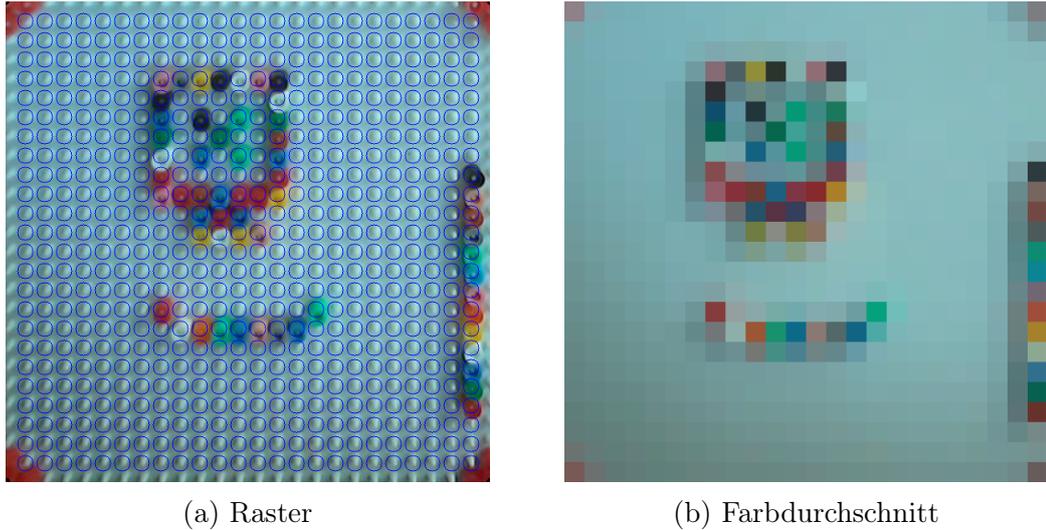


Abbildung 39: Rundes Raster

bedingt funktioniert. Es wurde stets nur ein Teil der Perlen erkannt. Es wird vermutet, dass der Grund hierfür darin liegt, dass die Perlen wie in Abbildung 33 zu erkennen ist, auf dem Kamerabild nicht immer kreisförmig aussehen.

Nachdem das automatische Erkennen von Kreisen zu keinen nutzbaren Ergebnissen führte, werden die Kreise in der Mitte von jedem Feld platziert. Das Ergebnis ist in Abbildung 39a zu sehen. Nachdem nun die runden Felder bestimmt wurden, muss die Berechnung der Durchschnittsfarben so angepasst werden, dass diese die runden Felder zur Berechnung benutzt. Während es relativ einfach war das Bild in Quadrate zu unterteilen und den Durchschnitt für jedes dieser Felder zu berechnen, ist es etwas schwieriger das Bild in runde Felder einzuteilen. Um dies zu bewerkstelligen, wird jedes quadratische Feld mit einer Maske addiert. Die Maske hat einen runde Ausschnitt. Beim Berechnen des durchschnittlichen Farbwerts werden nur die Pixel berücksichtigt, die nicht von der Maske verdeckt werden. Das Ergebnis dieser Berechnung ist in Abbildung 39b zu sehen. Vergleicht man die Durchschnittsberechnung mit quadratischem Raster (Abbildung 34b) mit der des runden Rasters, so kann man eine leichte Verbesserung bei den Farbwerten erkennen. Dies ist auch an unseren Beispielfeldern sichtbar. In Tabelle 40 sind die Werte für die beiden Farben aufgeführt. Das Feld mit der grünen Perle wird weiterhin korrekt als grün erkannt. Das Feld mit der gelben Perle wird nun nicht mehr der Farbe Grau zugeordnet, sondern der Farbe Braun. Insgesamt werden 10 von 45 Perlenfarben richtig erkannt.

												
233	35	83	300	269	219	237	89	98	138	194	264	155
154	139	180	253	161	112	166	179	108	102	98	200	200

Abbildung 40: Farberkennung anhand zwei Beispielfarben

Bessere Farbermittlung mit K-Means Weiter oben wurde das Verfahren zur Berechnung des Durchschnittsfarbwertes eines Perlenfelds vorgestellt. Dieses Verfahren hat den Nachteil, dass in den gebildeten Durchschnittswerten Pixel mit einbezogen werden, deren Farbe durch Reflexionen und Schatten verfälscht wurde. Um dieses Problem zu lösen wird ein Cluster-Algorithmus eingesetzt. Beim sogenannten *Clustering* werden Objekte die ähnlich zueinander sind in Gruppen eingeordnet. Wir wollen die einzelnen Pixel in einem Perlenfeld gruppieren. Dabei soll erreicht werden, dass eine Gruppe all die Pixel beinhaltet, die zu einer Steckperle gehören und nicht von Schatten überdeckt sind, oder stark reflektieren. Ist eine Gruppe ähnlicher Pixel identifiziert, kann der Mittelwert mit ihnen berechnet werden.

Der verwendete *Cluster*-Algorithmus heißt *K-means* [teab] und ist in *openCV* unter der Funktion *kmeans* implementiert. Als Eingabeparameter bekommt der Algorithmus die Anzahl der zu bildenden *Cluster*. Mit drei *Clustern* konnten die besten Ergebnisse erzielt werden. In Abbildung 41 ist in Teil a eine Bild einer Perle mit rundem Raster zu sehen. Wird der K-means Algorithmus

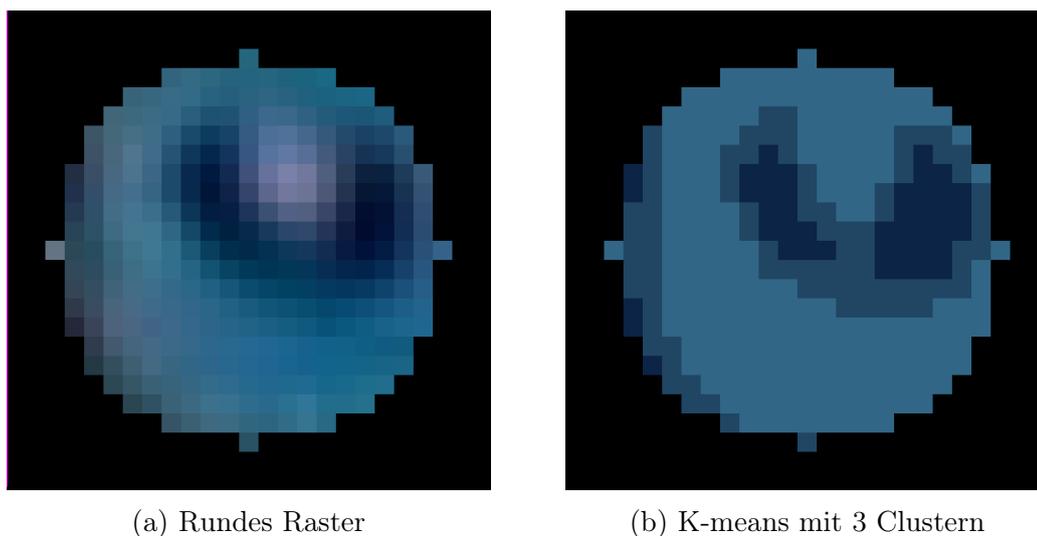


Abbildung 41: K-means anhand einer Perle

darauf angewandt, so entsteht die in Teil b dargestellten *Cluster*. Alle Pixel,

die zu einem *Cluster* gehören werden im Bild mit der selben Farben dargestellt. Die gewählte Farbe entspricht dem durchschnittlichen Farbwert des jeweiligen *Clusters*. Der entwickelte Algorithmus zählt die Anzahl der zugehörigen Pixeln zu jedem *Cluster* und nimmt den Farbwert des *Clusters* mit den meisten Pixeln. Wie in Abbildung 41b zu erkennen ist, wird dadurch der dunkle Bereich in der Mitte der Perle nicht mit in die Berechnung mit einbezogen. In Abbildung 42a ist das Ergebnis mit *K-means* und dem Durchschnittswert zu sehen. Zum Vergleich ist in Abbildung 42b der berechnete Durchschnittswert ohne den *K-means*-Algorithmus dargestellt. Es ist deutlich zu sehen, dass die Farben verschwommen wirken, während die Farbwerte mit *K-means* deutlich heller sind und eher den echten Farbwerten der Perlen entsprechen. Trotz der Verbesserung durch den K-means können nur 14 der insgesamt 45 Perlen richtig erkannt werden.

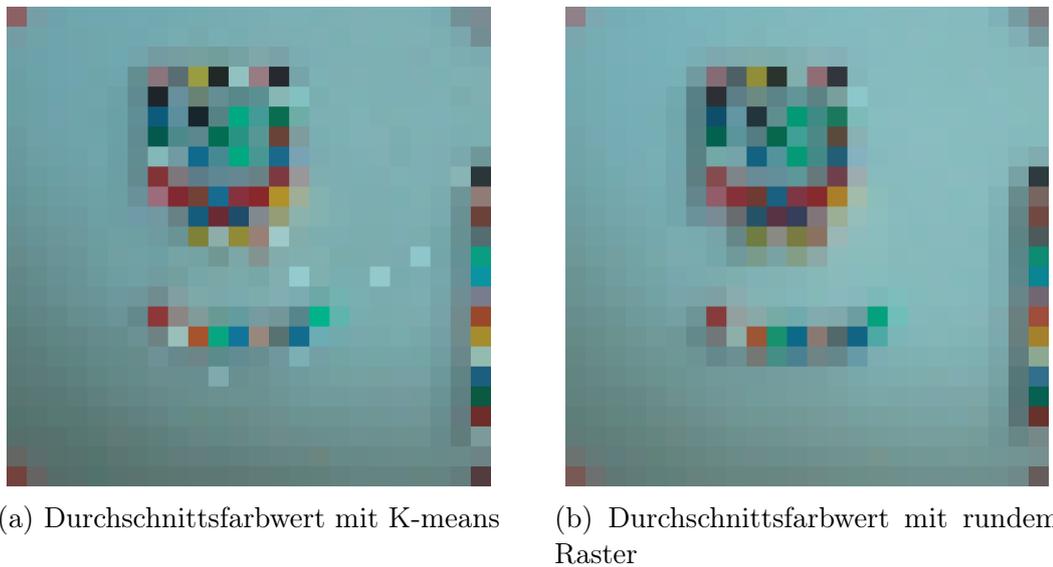


Abbildung 42: K-means

Referenzfarben Da das Verfahren mit der Verkördistanz der Farben noch nicht zu brauchbaren Ergebnissen führte, war die nächste Idee das Bild mit Referenzperlen zu kalibrieren. Dazu wurden auf dem Stiftbrett drei Perlen

240	34	86	326	284	233	261	102	123	162	202	289	133
161	137	176	240	161	116	156	171	95	89	107	188	208

Abbildung 43: K-Means mit zwei Beispielfarben

platziert. Je eine Perle in einer der drei Grundfarben: rot, grün und blau. Diese werden mit fotografiert und ihr Mittelwert bestimmt. Anschließend wird, wie im vorherigen schon beschrieben, die Vektordistanz zur erwarteten Farbe bestimmt. Ziel ist es den gesamten Farbraum um einen Vektor zu verschieben, der die Perlenfarben möglichst dicht an die erwarteten Farben anpasst. Dazu müssen wir aus den drei gemessenen Vektoren \vec{r} , \vec{g} und \vec{b} einen Mittelwertvektor bilden. Die Punkte auf die Vektoren verweisen bilden zusammen im dreidimensionalen Raum ein Dreieck. Wir suchen den Vektor, der auf den Schwerpunkt dieses Dreiecks verweist. Dieser Vektor \vec{m} ist unser gesuchter Mittelwert. Er kann wie folgt berechnet werden. Zunächst addieren wir die Vektoren und teilen sie durch drei:

$$\vec{m}_{rgb} = \frac{\vec{r} + \vec{g} + \vec{b}}{3}$$

Der resultierende Vektor \vec{m}_{rgb} verweist auf den Mittelpunkt des von \vec{r} , \vec{g} und \vec{b} beschriebenen Dreiecks. Anschließend bestimmen wir den Schwerpunkt des Dreiecks, das die erwarteten drei Farben bilden:

$$\vec{m}_e = \frac{\vec{r}_e + \vec{g}_e + \vec{b}_e}{3}$$

Werden nun die beiden Vektoren voneinander abgezogen, so ergibt sich der gesuchte Verschiebungsvektor \vec{m}_v :

$$\vec{m}_v = \vec{m}_{rgb} - \vec{m}_e$$

Dieser Vektor wird nun auf alle Farben der Farbpalette aufaddiert. Die dadurch entstandenen Farbwerte nennen wir die Referenzfarben. Diese werden in der Referenzfarbpalette gespeichert und können so der entsprechenden Farbe in der Farbpalette zugeordnet werden. Sollen nun für die berechneten Durchschnittsfarbwerten die zugehörige Farbe ermittelt werden, so wird die Vektordistanz nicht zu den Farben der Farbpalette berechnet, sondern zu denen der Referenzfarbpalette. In Abbildung 44 ist die Farbpalette und direkt da drunter die Referenzfarbpalette zu sehen. Wie an den Distanzwerten unter den Farben zu erkennen ist, so wird die gelbe Perle immer noch nicht richtig zugeordnet. Der Abstand zur gelben Farbe hat sich dafür verringert. Im gesamten Bild werden mit diesem Verfahren jetzt 17 von 45 Perlenfarben richtig erkannt.

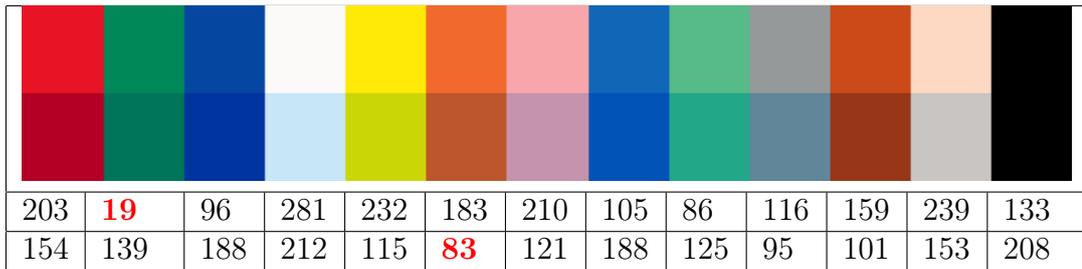


Abbildung 44: K-Means mit Referenzfarben

Erweiterung der Referenzfarben Da die Benutzung der drei Referenzfarben nur eine leichte Verbesserung zur Folge hatte, soll dieses Verfahren weiter verbessert werden. Statt, wie im Vorherigen beschrieben, drei Perlen als Referenzfarben zu nehmen, werden alle verwendeten Perlenfarben neben dem Druckbereich platziert. Von jeder wird der Farbwert bestimmt und direkt in der Referenzfarbpalette gespeichert. Dadurch, dass alle Farben als Referenzperlen benutzen werden, ist es nicht mehr nötig einen Schwerpunktvektor zu bestimmen. Die Farben können direkt als Referenz gespeichert werden. Wie auch vorher schon, wird bei der Berechnung der Distanz die Referenzfarbe benutzt. Wie in Abbildung 45 zu sehen ist, wird nun auch die gelbe Perle korrekt erkannt. Insgesamt wurden 32 von 45 Perlen im Testbild richtig erkannt. Das ist eine deutliche Steigerung zum Referenzverfahren mit drei Referenzperlen.

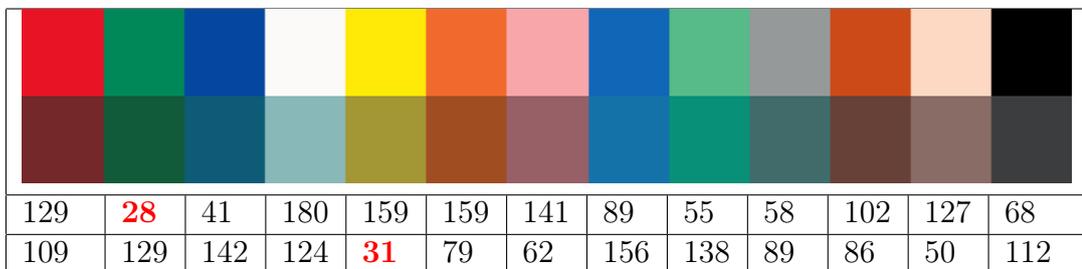


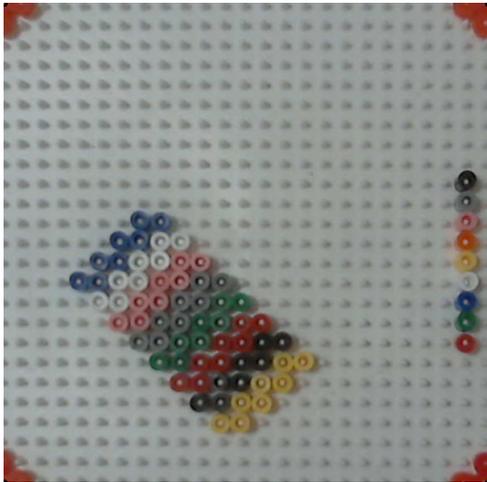
Abbildung 45: K-Means mit verbesserten Referenzfarben

Farbpalette anpassen Betrachtet man die Farben, die falsch erkannt werden, so fällt auf, dass oft ähnliche Farben verwechselt werden. Der Farbunterschied zwischen den hellblauen und blauen Perlen ist recht gering, sodass oft eine blaue Perle der Farbe Hellblau zugeordnet wird und umgekehrt. Um die Farberkennung zu verbessern, wird jeweils nur eine der ähnlichen Farben in die Farbpalette aufgenommen. Die neue Farbpalette ist in Abbildung 46 zu sehen. Um die Anpassung zu Testen wurde ein neues Testbild mit der Beadbox gesetzt. Dieses ist in Abbildung 47a zu sehen und benutzt nur Farben der

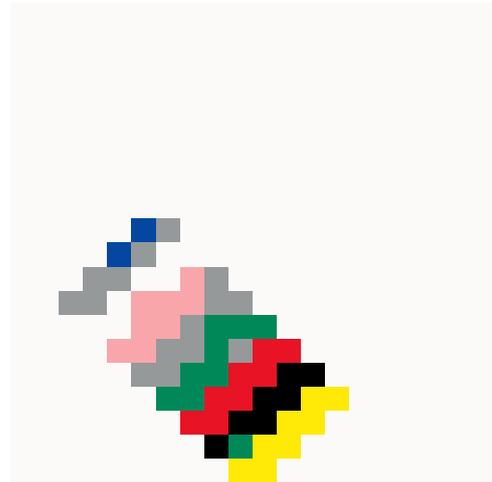


Abbildung 46: Angepasste Farbpalette

angepassten Farbpalette.



(a) Zugeschnittenes Kamerabild



(b) Farberkennung mit reduzierten Farben

Abbildung 47: Reduzierte Farbpalette

In Abbildung 47b sind die durch die Farberkennung zugeordneten Farben zu sehen. Es wurden 53 von 64 Farben richtig erkannt.

5.3.6 Korrekturdruck

Nachdem im vorangegangenen Abschnitt beschrieben wird, wie die Farberkennung abläuft, wird nun erläutert, wie das eigentliche Korrigieren der Bügelperlen funktioniert. Zunächst muss für die Korrektur eine Liste mit fehlerhaften Perlen erstellt werden.

Korrekturliste erstellen Damit die Beadbox gezielt die fehlerhaften Perlen korrigieren kann, benötigt sie die Position dieser Perlen und die richtige Perlenfarbe an dieser Stelle des Bügelperlenbilds. Um diese Liste zu erstellen, wird das Bild mit den erkannten Farben mit dem Druckbild verglichen. Bei beiden werden Pixel für Pixel die Farbwerte abgeglichen. Die Farbwerte können direkt verglichen werden, da nach der Farberkennung die Bilder dieselbe Farbpalette benutzen. Die Pixel, die im Druckbild transparent sind (nicht gesetzt werden sollen), werden beim Vergleich ausgelassen. Dadurch werden nur Felder verglichen, auf denen auch eine Perle gesetzt werden sollte. Das ist nö-

tig, da leere Felder auf dem Stiftbrett oft fälschlicherweise als Felder mit Perlen erkannt werden. Leider werden durch diese Maßnahme keine Felder erkannt, bei denen auf einem Feld eine Perle platziert wurde, welches leer bleiben sollte. Wird beim Vergleich ein Fehler erkannt (Farbwerte der verglichenen Pixel sind unterschiedlich), so wird die Position der fehlerhaften Perle, sowie die richtige zugehörige Farbe in der Korrekturliste aufgenommen.

Korrekturdruckvorgangs Der nächste Schritt ist das Starten eines Korrekturdruckvorgangs. Bevor dieser gestartet werden kann, müssen die falsch gesetzten Perlen entfernt werden. Dieser Vorgang wird per Hand ausgeführt. Einen Mechanismus zum automatisierten Entfernen der Perlen zu entwickeln, wäre im Rahmen dieser Arbeit zu aufwendig gewesen. Als Hilfestellung beim Entfernen der Perlen gibt die Beadbox die zu entfernenden Perlen am Bildschirm aus. Nachdem die entsprechenden Perlen entfernt wurden, beginnt die Beadbox damit alle Perlen auf der Korrekturliste auf die jetzt freien Felder zu setzen.

5.3.7 Finaler Ablauf des Korrekturprozesses

Im Abschnitt 5.3.1 wurde der Ablauf des Fehlerkorrekturprozesses geplant. Wir wollen nun den finalen Ablauf dieses Prozesses betrachten. Dieser ist in Abbildung 48 als Ablaufdiagramm dargestellt. Wie in dem Diagramm zu erkennen ist, sind gegenüber dem Ablauf im Planungsdiagramm 22 mehr Schritte nötig. Bevor das Kamerabild benutzt werden kann, muss zunächst der *barrel*-Effekt aus dem Bild entfernt werden. Dies wurde im Abschnitt 5.3.3 beschrieben. Anschließend muss das Bild transformiert werden, damit mit dem Druckbereich gearbeitet werden kann. Dazu müssen zunächst die Marker auf dem Stiftbrett im Bild wiedererkannt werden. Anschließend wird das Bild mit den Positionswerten der Marker transformiert. Wie dies geschieht, wurde in Abschnitt 5.3.4 erläutert. Um die Farbwerte zu ermitteln, müssen daraufhin im transformierten Bild die Perlenfelder bestimmt werden. Dazu wird ein Raster über das Bild gelegt. Sind die Perlenfelder bestimmt, wird zu jedem der durchschnittlichen Farbwert berechnet. Um einen besseren Farbwert bestimmen zu können, werden die Farben zunächst mit dem K-means-Algorithmus in *Cluster* aufgeteilt. Der Farbwert wird anhand des *Clusters* mit den meisten Pixeln im Bild berechnet. Für die Farberkennung werden die Referenzfarben ermittelt und den Farben der Farbpalette zugeordnet. Im Anschluss daran wird mit dem Vektordistanzverfahren zu jedem Perlenfeld die zugehörige Perlenfarbe bestimmt.

Dieser Ablauf wird im Abschnitt 5.3.5 genauer erläutert.

Nachdem die Farben der Perlen bestimmt wurden, werden diese mit den Farben des Druckbildes verglichen. Stimmen die Farben an einer Stelle nicht überein, so wird diese Perle in die Korrekturliste aufgenommen. Ist die Fehlerliste nach dem Vergleich aller Felder leer, so wurde das Bild richtig gedruckt und der Korrekturalgorithmus terminiert. Ist dies nicht der Fall, so muss ein Korrekturdruck durchgeführt werden. Dazu müssen zunächst die falsch gesetzten Perlen auf der Korrekturliste per Hand entfernt werden. Sind alle Perlen entfernt, so werden die Perlen der Korrekturliste erneut gedruckt. Der Korrekturdruck wird in Abschnitt 5.3.6 erläutert.

Nachdem der Korrekturdruck abgeschlossen ist, beginnt der Algorithmus von vorne. Es wird abermals ein Bild mit der Kamera aufgenommen und alle notwendigen Schritte durchgeführt, bis das korrigierte Bügelperlenbild mit dem Druckbild verglichen werden kann.

Sind beim Korrekturdruck erneut Fehler aufgetreten, so wird ein weiterer Korrekturdruck vorgenommen. Dieser Ablauf wird solange wiederholt, bis das Bügelperlenbild korrekt gedruckt wurde.

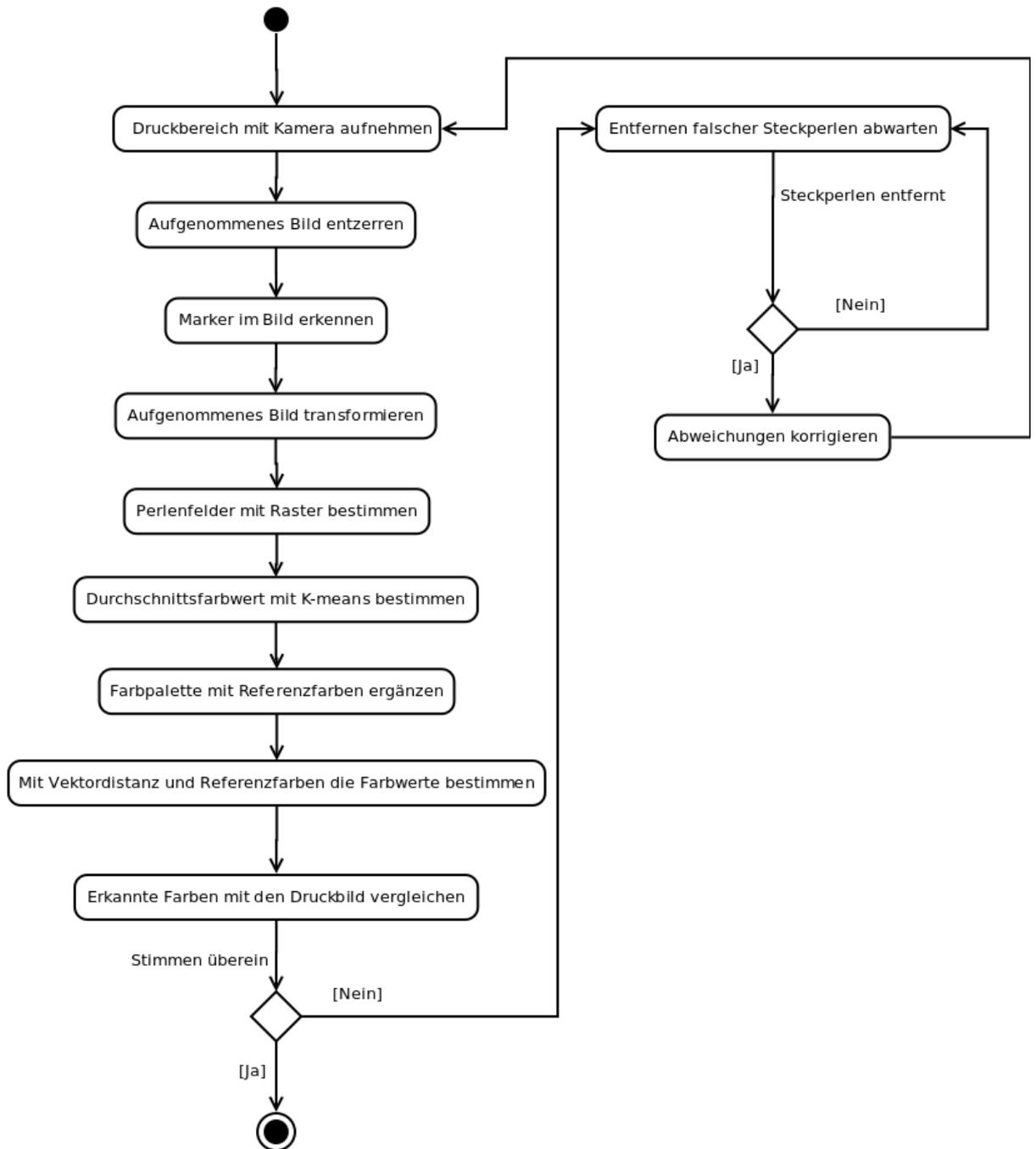


Abbildung 48: Ablaufdiagramm Fehlerkorrekturprozess

6 Erweiterung der Architektur

Die Architektur der Beadbox ist in Abbildung 49 zu sehen. Im Folgenden wird beschrieben, wie diese Architektur erweitert wird. Zunächst wird erläutert, warum die Software nicht mehr auf dem Raspberry Pi ausgeführt wird. Anschließend wird das benutzte Programm zur Kalibrierung der Kamera kurz beschrieben. Die eigentliche Erweiterung der Architektur wird im Abschnitt „Fehlererkennung“ und „Fehlerkorrekturdruck“ dargestellt.

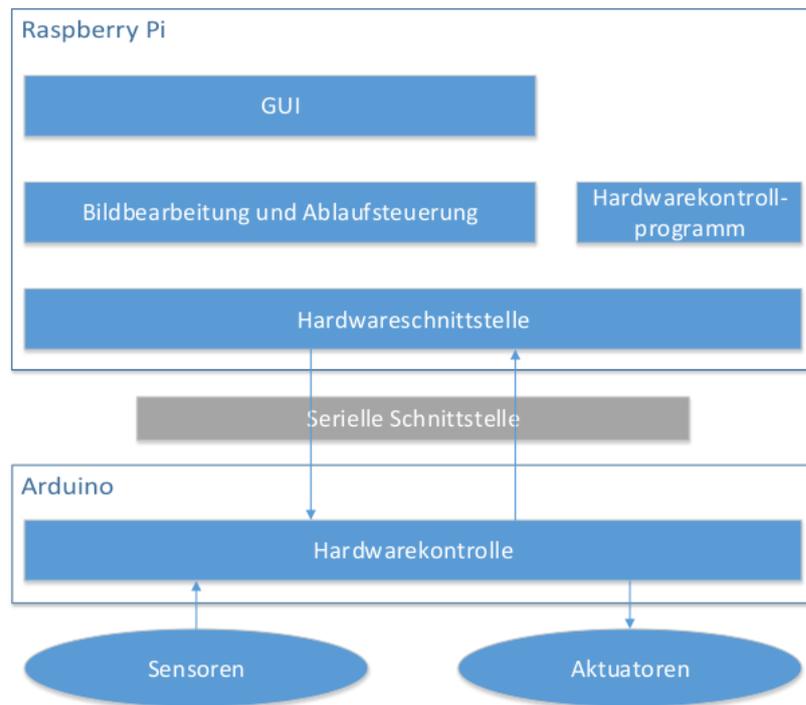


Abbildung 49: Architektur [Zie13]

Vom Raspberry Pi zum PC Die Steuerungssoftware für die Beadbox wird auf einem Raspberry Pi ausgeführt [Zie13]. Dieser ist an einem Touchscreen angeschlossen. Für diese Arbeit wird zusätzlich zu den von Herrn Ziegler für die Beadbox verwendeten Technologien das *openCV* Framework verwendet. *openCV* unterstützt die Betriebssysteme Windows, Linux, Android und MacOS [Gar]. Für den Raspberry Pi muss das Framework aus dem Quellcode kompiliert werden. Da der Raspberry Pi nur mit 700MHz taktet, dauert das Kompilieren mehr als zehn Stunden. Nach einigen Versuchen konnte *openCV* für den Raspberry Pi erfolgreich kompiliert werden. Da es Probleme mit der für die Beadbox verwendeten *QT*-Version gab, musste *QT* ebenfalls neu kompiliert werden. Auch dieser Vorgang stellte sich als kompliziert heraus. Es traten viele

Fehler auf, die zuerst behoben werden mussten. Es gelang zwar beides zu kompilieren, aber die gemeinsame Nutzung gestaltete sich als schwierig, da einige Fehler auftraten. Daher wurde entschieden die neue Software nicht direkt für den Raspberry Pi zu entwickeln und stattdessen einen Linux-Rechner mit einer 64-Bit Architektur zu verwenden. Auf diesem ließen sich sowohl *QT* als auch *openCV* ohne Probleme einrichten. Die Entscheidung auf einen Leistungsstärkeren PC umzusteigen hat ebenfalls den Vorteil, dass mehr Rechenleistung für die Funktionen des *openCV*-Frameworks bereitsteht, was den rechenaufwendigen Prozess der Bildverarbeitung erheblich beschleunigt.

Damit die BeadBox-Software auf einem Linux-PC lauffähig ist, muss lediglich in der Konfigurationsdatei *settings.ini* der richtige Port zum Ansteuern des Arduinos angegeben werden. Bei der Benutzung des Raspberry Pi's kann davon ausgegangen werden, dass sich der Port nicht ändert, da nur die zur Beadbox gehörenden Geräte angeschlossen sind. Wird ein beliebiger Linux-PC verwendet, so kann es vorkommen, dass der Pfad zum Port des Arduinos ein anderer ist. Wie der Port anzupassen ist, kann im Benutzerhandbuch unter Abschnitt 8.2 nachgelesen werden.

6.1 Software zur Kamerakalibrierung

Die in Abschnitt 5.3.3 beschriebene Software zur Kalibrierung der Kamera befindet sich in *openCV*-Installationspfad. Wie dort geschrieben, muss das Programm für die verwendete Kamera leicht angepasst werden. Eine angepasste Version ist in Programmverzeichnis der Beadbox zu finden.

6.2 Fehlererkennung

In diesem Abschnitt wird beschrieben, wie die Architektur der Beadbox um die Funktionalität der Fehlerkorrektur erweitert wird. Für die Bildverarbeitung wurden die folgenden drei Klassen *bbqualitycontrol*, *bbimagetransformation* und *bbcolorfinder* angelegt. Diese sollen im Folgenden beschrieben werden.

6.2.1 Die Klasse *bbqualitycontrol*

Die Klasse *bbqualitycontrol* stellt alle Funktionen zum Aufnehmen und Transformieren von Kamerabildern, sowie zur Erkennung der Perlenfarben bereit. Sie greift dazu auf Funktionen der beiden Klassen *bbimagetransformation* und *bbcolorfinder* zu. Die wichtigsten Funktionen werden im Folgenden kurz beschrieben:

int initializeCam() Diese Funktion initialisiert die Kamera. Dabei wird eine Verbindung zur Kamera aufgebaut und die Auflösung dieser eingestellt.

void getCameraImage() Veranlasst die Kamera ein Bild aufzunehmen und speichert dieses für die weitere Verwendung innerhalb der Klasse.

cv::Mat getFrame() Liefert das mit *getCameraImage()* aufgenommene oder mit *setTestimage(cv::Mat img)* gesetzte Bild zurück.

void setTestimage(cv::Mat img) Funktion zum Setzen eines Testbildes. Kann zu Testzwecken verwendet werden, wenn gerade keine Kamera angeschlossen ist oder ein bestimmtes Bügelperlenbild getestet werden soll. Wurde dieses Bild gesetzt, so liefert der Aufruf der Funktion *getFrame()* statt dem aktuellen Kamerabild das Testbild.

void setUndistortion(std::string filename) Diese Funktion gibt die Datei mit den Parametern zur Kalibrierung der Kamera an. In dieser Datei befinden sich die in Abschnitt 5.3.3 beschriebenen *distortion*-Parameter. Wurde eine Datei angegeben, so wird jedes durch die Funktion *getCameraImage()* aufgenommene Bild automatisch entzerrt.

void setPrintArea(int x, int y) Mit dieser Funktion wird die Größe des Druckbereichs angegeben. Für die Variablen *x* und *y* muss die Anzahl der Perlenfelder in Horizontaler und Vertikaler Richtung eingegeben werden.

void setTemplate(std::string filename) Mit dieser Funktion kann das zum Ausrichten des Bildes benötigte Template eingestellt werden.

int transformImage(double tempThreshold) Die Funktion bestimmt zuerst die Position der auf dem Stiftbrett liegenden Marker und schneidet anschließend das Kamerabild entsprechend dem in Abschnitt 5.3.4 beschriebenen Verfahren zu. Die Variable *tempThreshold* gibt den *Threshold* für das *Templatematching* an.

void getMeanColors(bool circlemask, bool kmeans) Berechnet die Durchschnittsfarbwerte der Perlenfelder. Als Parameter kann bestimmt werden, ob eine Kreismaske für die Felder verwendet und, ob der K-Means Algorithmus benutzt wird.

void findColors(bool useReferenzColors) Diese Funktion ordnet mit dem im Kapitel 5.3.5 beschriebenen Vektordistanzverfahren den Perlenfeldern die entsprechenden Perlenfarben zu. Es erstellt eine Pixel-Karte mit diesen Farbwerten.

void setPrintImage(cv::Mat prImage) Mit dieser Funktion kann das von der Beadbox erstellte Druckbild an die Klasse *bbqualitycontrol* übergeben werden. Danach kann es mit der Funktion *void comparePixelMaps()* mit den erkannten Farbwerten verglichen werden.

int comparePixelMaps() Diese Funktion vergleicht die Werte des Druckbildes mit den der erkannten Farben und speichert nicht übereinstimmende Felder in der Fehlerliste.

6.2.2 Die Klasse *bbimagertransformation*

Die Klasse *bbimagertransformation* ist für die Transformation des Kamerabildes zuständig. Dafür wird die folgende Funktion bereitgestellt:

int checkPattern(cv::Mat* templ, cv::Mat* frame, double tempThreshold, cv::Size size); Die Funktion *checkPattern* sucht in dem als Parameter mitgegebenen Bild *frame* nach einem Template, welches mit dem Parameter *templ* der Funktion übergeben wurde. Der Parameter *tempThreshold* gibt den Threshold für das Templatematching an und der Parameter *size* die Größe des Druckbereichs in Perlenfeldern. Als Rückgabewert liefert die Funktion, wenn sie vier Marker gefunden hat eine 1 und wenn nicht eine 0 zurück. Wurden vier Marker gefunden, wird außerdem das mitgegebene Bild transformiert.

6.2.3 Die Klasse *bbcolorfinder*

Die Klasse *bbcolorfinder* stellt alle Funktionen bereit, die zum Erkennen der Farbe einer Bügelperle benötigt werden. Ihre wichtigsten Funktionen sind die Folgenden:

cv::Scalar meancolor(cv::Mat beadfield, bool nomask) Diese Funktion berechnet den Durchschnittsfarbwert für das als Parameter übergebene Perlenfeld zurück. Der zweite Parameter gibt an, ob das Perlenfeld mit einer Maske versehen ist, oder nicht. Diese Maske wird für ein Raster mit runden Feldern verwendet.

cv::Scalar meanKmeans(cv::Mat beadfield, int clusterCount) Die Funktion *meanKmeans* liefert den mit dem Algorithmus K-means bestimmten Durchschnittsfarbwert zurück. Als Parameter werden ein Perlenfeld und die Anzahl der mit K-means zu bestimmenden Cluster mitgegeben.

6.3 Fehlerkorrekturdruck

Im vorherigen Abschnitt wurden die Klassen vorgestellt, die für das Erkennen der Perlenfarben benötigt werden. Diese wird von dem Fehlerkorrekturdruck benutzt, damit dieser die entsprechenden Perlen korrigieren kann. Der Fehlerkorrekturdruck wird als neues Druckverfahren implementiert. Dafür wurde in der Bibliothek *bbprocession* [Zie13, S. 21] eine neue Druckermethode angelegt. Diese neue Druckermethode heißt *ErrorCorrectionPlotterMethod*. Sie kann wie eine der anderen Druckermethode benutzt werden. Diese Methode druckt zunächst das Druckbild mit dem zeilenweisen Druckverfahren [Zie13, S. 34] und leitet anschließend den im Abschnitt 5.3.6 beschriebenen Fehlerkorrekturprozess ein.

6.4 Erweiterung der GUI

Damit der Fehlerkorrekturprozess bei einem Druck gestartet werden kann, wurde die GUI der Beadbox angepasst. Wie im Benutzerhandbuch in Abschnitt 8.3 zu sehen ist, wurden in der Oberfläche mit den Druckeinstellungen die Optionen für die Fehlerkorrektur hinzugefügt. Außerdem wurde eine Ansicht zur Ausrichtung der Kamera (*CameraPage.qml*) erstellt. Die Ansicht (*BeadCorrectionPage.qml*) wurde ebenfalls angelegt. In dieser Ansicht werden die Perlen, die für den Korrekturdruck entfernt werden müssen dargestellt.

7 Bewertung der Beadboxerweiterungen

In diesem Kapitel werden die im Kapitel 5 beschriebenen Verfahren zum Vermeiden und Korrigieren von Fehlern bewerte.

7.1 Mechanische Verbesserungen

Ein Ziel dieser Arbeit ist es, die beim Druckprozess der Beadbox auftretenden Fehler zu reduzieren und so den Druckvorgang zu verbessern. Dazu wurden einige Veränderungen an der Hardware vorgenommen und ein Mechanismus zur Fehlerkorrektur entwickelt. Im Abschnitt 5.2 wurden die Veränderungen an der Hardware beschrieben. Nun soll überprüft werden, ob durch die Veränderungen die auftretenden Fehler reduziert werden konnten. Dazu wird die Häufigkeit der auftretenden Fehler bei der verbesserten Version der Beadbox bestimmt. Diese Werte können anschließend mit den, während der Ist-Analyse erhobenen, Werten der Fehlerhäufigkeiten verglichen werden. Damit die Ergebnisse beider Testläufe vergleichbar sind, wurden identische Testbilder verwendet. Die Testbilder sind in Abbildung 53a und Abbildung 53b im Anhang A zu sehen. Wie beim ersten Testlauf, wird jedes Testbild fünfmal gedruckt. Die Ergebnisse des Testlaufes sind im Anhang B in der Tabelle 11 für das Bild mit den schwarzen Perlen und in Tabelle 12 für das bunte Bild aufgeführt. Diese wurden in der Tabelle 4 zusammengefasst. Dort sind ebenfalls die Werte aus den Testläufen der Ist-Analyse aufgeführt.

Fehlerart	Auftreten des Fehlers bei 900 Perlen	
	ohne Veränderungen	mit Veränderungen
V1	259 (28,8%)	50 (5,6%)
V2	4 (0,4%)	0
V3	5 (0,5%)	0
S1	610 (67,8%)	104 (11,6%)
S2	4 (0,4%)	0

Tabelle 4: Testläufe Gesamtergebnisse

Betrachten wir zunächst den Fehler V1. Bei diesem verhaken sich Perlen im Schieber. In den Testläufen der Ist-Analyse trat dieser Fehler bei 28,8% er gesetzten Perlen auf. Durch die Anpassungen an Schieber und Setzrohr tritt dieser Fehler nur noch bei 5,6% der Perlen auf. Der Fehler konnte zwar nicht vollständig behoben, die Fehlerhäufigkeit aber deutlich gesenkt werden.

Beim Fehler V2 wird der Schieber vom Perlenmagazin blockiert. Dieser Fehler konnte während der Ist-Analyse insgesamt viermal beobachtet werden.

Er trat in den neuen Testläufen kein einziges Mal auf. Um ihn zu beseitigen wurden die Perlenmagazine mit einer extra dafür entworfenen Unterlegscheibe höher gesetzt. Durch diese Maßnahme konnte der Fehler behoben werden.

Auch der Fehler V3 konnte bei den neuen Testläufen nicht beobachtet werden. Dieser tritt, wie in Abschnitt 4.1.1 beschrieben, in Folge des Fehlers V1 auf. Da der Fehler V1 nicht behoben wurde, sondern nur seltener auftritt, kann davon ausgegangen werden, dass auch der Fehler V3 seltener auftritt. Da keine Maßnahmen zur Behebung des Fehler V3 vorgenommen wurden, kann dieser auch nicht als behoben angesehen werden.

Bei dem Fehler S1 wird eine Perle von einer schon gesetzten daran gehindert das Setzrohr zu verlassen. Wie schon der Fehler V1 konnte dieser zwar nicht behoben, die Fehlerwahrscheinlichkeit aber gesenkt werden. Während er bei der Ist-Analyse noch bei etwa zwei Dritteln aller Perlen beobachtet werden konnte, so tritt er bei der verbesserten Beadbox nur noch bei 11,6% aller Perlen auf.

Der letzte Fehler S2 tritt auf, wenn die Lichtschranke am Setzrohr eine Perle umwirft. Durch das Versetzen der Lichtschranke konnte dieser Fehler behoben werden. Dies zeigen auch die ermittelten Werte der Fehlerhäufigkeit.

7.2 Fehlerkorrektur

Die Fehler E1, E2 und E3 sind Fehler, die in einem falschen Bügelperlenbild auftreten. Um diese zu korrigieren wurde ein Fehlerkorrekturprozess entwickelt. Wie gut dieser funktioniert hängt davon ab, wie gut er auftretende Fehler erkennt. Um dies zu ermitteln, wurde die Fehlererkennung mit den im Anhang C abgebildeten Bildern getestet. In den Abbildungen ist unter (a) das von der Kamera aufgenommene und zugeschnittene Bild zu sehen. In Teil (b) ist jeweils ein Bild mit den berechneten Durchschnittsfarbwerten zu sehen. Diese wurden wie in Abschnitt 5.3.5 beschrieben berechnet. In Teil (c) ist schließlich ein Bild mit den erkannten Farben zu sehen. Dieses Bild wurde mit dem Druckbild verglichen und so die korrekt erkannten Perlenfarben bestimmt. Die Ergebnisse aller durchgeführten Testläufe sind in Tabelle 5 zu sehen. Es konnten 84,6% der Perlenfarben richtig erkannt werden.

Nicht behobenen Fehler Der Fehler B2, der besagt, dass ein Perlenmagazin nicht ausreichend befüllt wurde. Um diesen Fehler zu beheben wurden Überlegungen zu einem Perlensortierer angestellt, der gleichzeitig die Perlenmagazine befüllen kann. Die Entwicklung eines solchen Perlensortierer konnte

TestNr.	Erkannte Perlen / Perlen
1	37 / 37
2	30 / 36
3	26 / 28
4	27 / 35
5	86 / 100
6	53 / 70
	259 / 306 (84,6%)

Tabelle 5: Farberkennung Testläufe

im Rahmen dieser Arbeit aus zeitlichen Gründen nicht mehr umgesetzt werden. Auch für den Fehler B3 konnte keine Lösung mehr erarbeitet werden. Um diesen zu beheben, ist eine präzise automatische Ausrichtung der Beadbox erforderlich.

Der Fehler B1 wurde ebenfalls nicht behoben. Da ein falsch befülltes Perlenmagazin beim Drucken zu dem Fehler E1 führt und dieser durch den Fehlerkorrekturprozess korrigiert wird, kann auch der Fehler B1 als korrigiert angesehen werden.

Ergebniszusammenfassung In Tabelle 6 ist das Ergebnis dieser Bewertung zusammengefasst. Die Fehler V2 und S2 konnten behoben werden. Bei den Fehlern V1, V3 und S1 konnte die Fehlerhäufigkeit reduziert werden. Die Fehler B2 und B3 konnten nicht behoben werden. Da die Fehlererkennung des Fehlerkorrekturprozesses nur bei 84,6% liegt, können die Fehler B1, E1, E2 und E3 nur als teilweise korrigiert angesehen werden.

Fehlerart	Ergebnis
V1	Fehlerhäufigkeit um ca. 20% reduziert
V2	Fehler behoben
V3	Fehlerhäufigkeit stark reduziert
S1	Fehlerhäufigkeit um ca. 50% reduziert
S2	Fehler behoben
B1	teilweise korrigiert
B2	nicht behoben
B3	nicht behoben
E1	teilweise korrigiert
E2	teilweise korrigiert
E3	teilweise korrigiert

Tabelle 6: Ergebnis der Bewertung

8 Benutzerhandbuch

Dieses Benutzerhandbuch ergänzt das von Herrn Ziegler erstellte Benutzerhandbuch [Zie13, S. 40ff] um die durch diese Arbeit geschaffenen Erweiterungen. Für die Farberkennung wird eine Kamera benutzt. Es wird beschrieben, wie diese zu kalibrieren ist. Außerdem wird die Software nicht mehr auf dem Raspberry Pi ausgeführt, weswegen möglicherweise der Beadbox-Port angepasst werden muss. Im letzten Abschnitt wird beschrieben, wie ein Druck mit Fehlerkorrektur ausgeführt wird.

8.1 Kamera kalibrieren

Zur Kalibrierung der Kamera wird das Kalibrierungsbeispiel von *openCV* verwendet. Der Quellcode ist auf der Webseite http://docs.opencv.org/3.0-beta/doc/tutorials/calib3d/camera_calibration/camera_calibration.html oder im *openCV*-Installations-Verzeichnis unter `samples/cpp/tutorial_code/calib3d/camera_calibration/` zu finden. Das Programm muss leicht angepasst werden, damit die Kalibrierung bei voller Kameraauflösung funktioniert.

```
#include <opencv2/highgui/highgui.hpp>
#include <unistd.h>

if (inputType == CAMERA) {
    inputCapture.open(cameraID);
    inputCapture.set(CV_CAP_PROP_FRAME_WIDTH, 1280);
    inputCapture.set(CV_CAP_PROP_FRAME_HEIGHT, 720);
    sleep(2);
}
```

Anschließend kann das Programm kompiliert und ausgeführt werden. Es öffnet sich ein Fenster in dem das Bild der Kamera angezeigt wird. Nun muss die Kamera auf das ausgedruckte Schachbrettmuster gerichtet werden. Diese ist unter der Adresse <http://docs.opencv.org/3.1.0/pattern.png> heruntergeladen werden. Dann wird der Kalibrierungsprozess mit der Taste „g“ gestartet. Es werden automatisch mehrere Bilder von dem Schachbrett gemacht. Dabei sollte man die Kamera leicht bewegen, damit das Schachbrett aus verschiedenen Winkeln aufgenommen wird. Ist die Kalibrierung erfolgreich abgeschlossen, so befindet sich im Programmverzeichnis eine Datei mit dem Namen `out_camera_data.xml`. In dieser Datei befinden sich die Kalibrie-

rungsdaten. Sie muss in *cameracalibration.xml* umbenannt werden und in das Hauptverzeichnis der Beadbox-Software kopiert werden.

8.2 Einstellen des Beadbox-Ports

Damit die Beadbox an einem PC verwendet werden kann, muss der Software der Port der Beadbox bekannt sein. Dieser Port kann in der Konfigurationsdatei der Beadbox (*settings.ini*) angegeben werden. Der Eintrag, mit dem dieser Port angegeben wird, sieht wie folgt aus:

```
[ Arduino ]  
Port=/dev/ttyACM0
```

Der Teil hinter dem Schlüsselwort *Port* gibt den Pfad zu dem selbigen an. In diesem Fall */dev/ttyACM0*.

8.3 Druck mit Fehlerkorrektur

Um einen Druckvorgang mit Fehlerkorrektur zu starten, muss zunächst, wie im Benutzerhandbuch der Beadbox [Zie13, S. 44ff] beschrieben, ein Bild ausgewählt werden. Hier besteht die Möglichkeit entweder eine Bildvorlage auszuwählen, oder selbst ein Bild zu malen. Ist dieser Schritt abgeschlossen, so erscheint die in Abbildung 50 gezeigte Oberfläche. Hier können Einstellungen

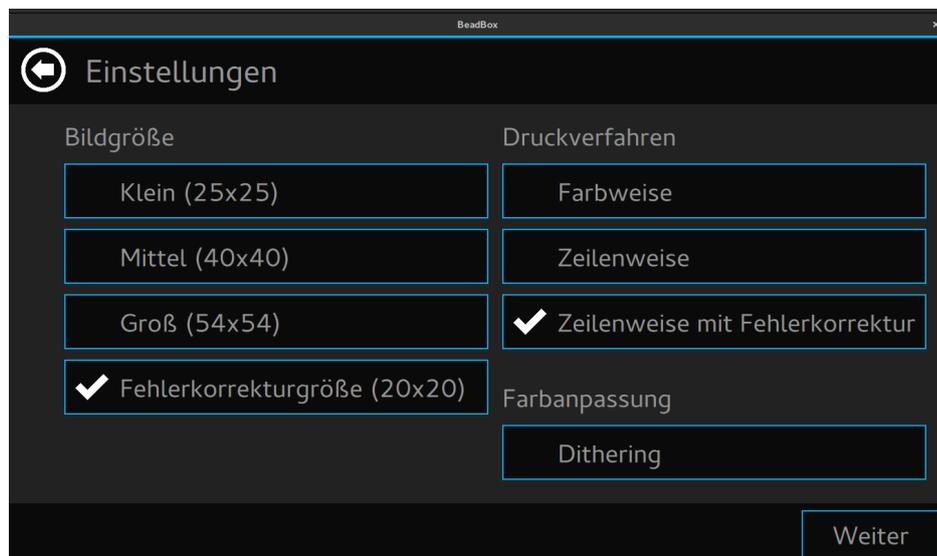


Abbildung 50: GUI: Fehlerkorrektur Auswahl

für den Druck festgelegt werden. Es kann entschieden werden, in welcher Größe das Bügelperlenbild erstellt wird und welches Verfahren zum Drucken benutzt

wird. Außerdem kann eingestellt werden, ob eine Farbanpassung durchgeführt wird. Für einen Druck mit Fehlerkorrektur muss unter Bildgröße die Option *Fehlerkorrekturgröße (20x20)* und unter Druckverfahren *Zeilenweise mit Fehlerkorrektur* ausgewählt werden.

Durch betätigen der Schaltfläche *Weiter* kann der Druckvorgang wie in Benutzerhandbuch der Beadbox [Zie13, S. 44ff] beschreiben gestartet werden. Nachdem das Bügelperlenbild gedruckt wurde, wird der Korrekturprozess gestartet. Dazu wird zunächst das Kamerabild angezeigt, damit überprüft werden kann, ob diese richtig ausgerichtet ist. Dies ist in Abbildung 51 zu sehen. Wird, wie in der Abbildung zu sehen ist, nur der Ausschnitt des Druckbereichs angezeigt, so ist die Kamera richtig ausgerichtet. Wird hingegen der gesamte Kameraausschnitt angezeigt, so muss die Kamera erneut ausgerichtet werden. Dazu wird die Kamera solange bewegt, bis die vier roten Marker im Bild der Kamera sind. Werden alle vier Marker erkannt, wird das Bild automatisch zugeschnitten und der Korrekturprozess kann gestartet werden.

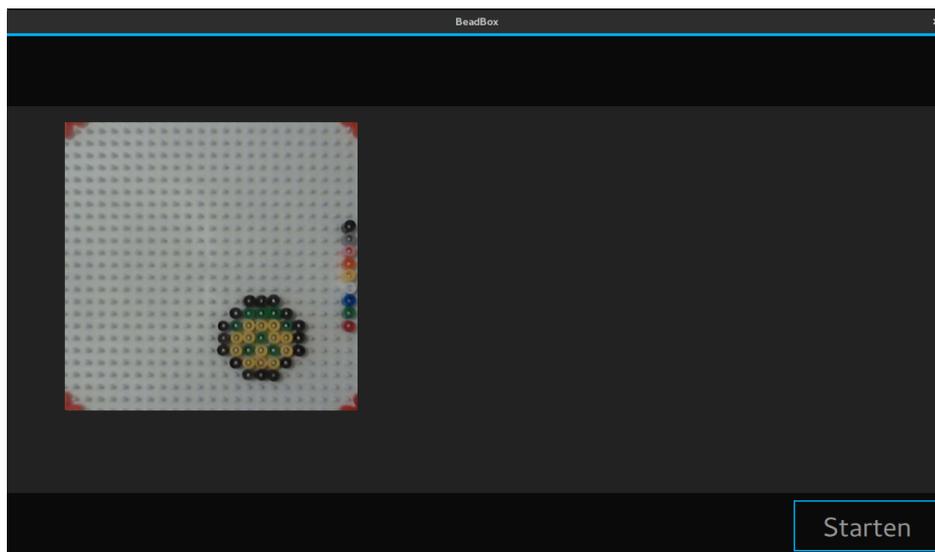


Abbildung 51: GUI: Kamera ausrichten

Wurde der Korrekturvorgang gestartet, wird das Perlenbild erkannt und die Farben werden mit dem Druckbild verglichen. Wurde das Bügelperlenbild korrekt gedruckt, so wird der Druck beendet und es kann ein neues Bild gedruckt werden. Wurden Fehler erkannt, so wird die in Abbildung 52 gezeigte Oberfläche angezeigt. Die hier angezeigten Perlen müssen per Hand entfernt werden, bevor der Korrekturdruck gestartet werden kann.

Wurden die angezeigten Perlen entfernt, kann der Korrekturdruck mit klicken auf die Schaltfläche *Weiter* gestartet werden. Im Anschluss an den Kor-

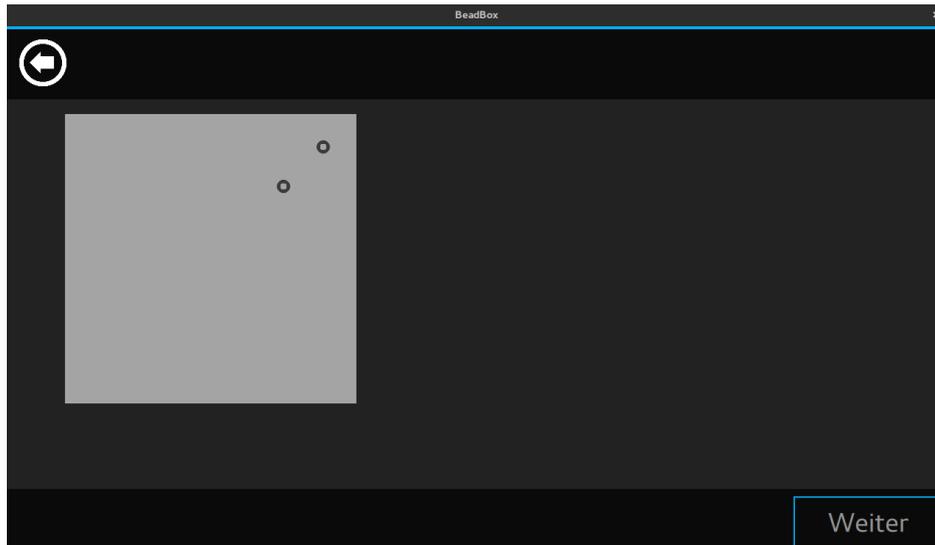


Abbildung 52: GUI: Perlen entfernen

rekturendruck wird eine erneute Fehlerüberprüfung durchgeführt und ggf. der Korrekturprozess abermals ausgeführt.

9 Fazit und Ausblick

Ziel der Arbeit ist das Erweitern der Beadbox um Mechanismen zur Fehlertoleranz und Qualitätskontrolle. Dadurch soll die Wahrscheinlichkeit auf einen problemlosen Druckablauf und ein fehlerfreies Druckergebnis erhöht werden. Um dieses Ziel zu erreichen, wurde zunächst eine Ist-Analyse durchgeführt, um alle auftretenden Fehler zu ermitteln. Zu den ermittelten Fehlern wurden verschiedene Lösungen erarbeitet. Einige der Fehler konnten durch anpassen der Beadbox-Hardware behoben, oder die Häufigkeit ihres Auftretens reduziert werden. Teile der Hardware wurden dazu komplett ausgetauscht. Die neu entwickelten Komponenten wurden mit einem 3D-Drucker erstellt. Die am Computer entworfenen Komponenten werden mit dem 3D-Drucker innerhalb von Stunden gedruckt. Änderungen an den Komponenten konnten dadurch schnell getestet werden.

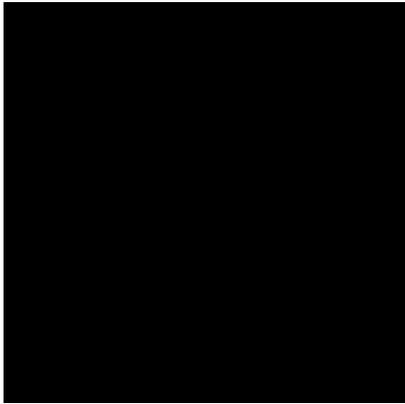
Um die Qualität des Druckergebnisses zu erhöhen, wurde ein Korrekturverfahren entwickelt. Diese Verfahren bestimmt mit Hilfe einer Kamera die Abweichungen vom Druckbild zum Druckergebnis. Aus den Erkannten Abweichungen wird eine Fehlerliste erstellt. Die falsch gesetzten Perlen müssen entfernt werden, damit ein Korrekturdruck gestartet werden kann. Beim Korrekturdruck werden auf den nun leeren Felder Perlen der richtigen Farbe gesetzt und so das Druckbild korrigiert. Im Anschluss an den Druck wird das Bild erneut auf Fehler untersucht. Ist das Bild noch nicht Fehlerfrei, so kann der Korrekturdruck abermals ausgeführt werden. Dieser Vorgang wird solange wiederholt, bis ein korrektes Druckergebnis erkannt wird. Um das Druckbild mit dem Druckergebnis vergleichen zu können, wird zunächst ein Bild mit der Kamera aufgenommen. In diesem Bild wird die Position der einzelnen Perlenfelder bestimmt. Um die Position möglichst genau zu bestimmen, werden auf dem Stiftbrett angebrachte Markierungen von der Software automatisch erkannt. Anschließend kann die Position des gesamten Druckbereichs bestimmt und mit einer Transformation das Bild digital ausgerichtet werden. Nachdem dies geschehen ist, kann die Position zu jedem Perlenfeld bestimmt werden. Anschließend wird von jeder Perle die Farbe ermittelt. Wie sich im Verlauf der Arbeit herausstellte, ist das Ermitteln einer Perlenfarbe auf Grundlage eines Kamerabildes keine einfache Aufgabe. Es wurden verschiedene Ansätze ausprobiert und ein Verfahren zum erkenne der Perlenfarben entwickelt. Dieses Verfahren wurde im Verlauf der Arbeit immer weiter verbessert. Trotz dieser zahlreichen Verbesserungen, kann nur bei etwa 84,6% der Perlen die richtige Farbe bestimmt werden. Damit der Fehlerkorrekturvorgang eingesetzt werden

kann, muss dieses Erkennungsverfahren noch weiter verbessert werden. Im Verlauf der Arbeit wurde versucht das Erkennungsverfahren durch Beleuchtung des Stiftbretts zu verbessern, es konnten aber keine guten Ergebnisse erzielt werden. Mit einer gleichmäßigen möglichst reflexionsfreien Beleuchtung könnte die Farberkennung noch verbessert werden.

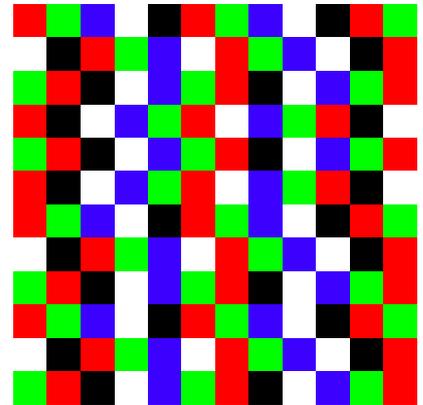
Auch an den mechanischen Teilen der Beadbox sind noch weitere Verbesserungen möglich. Die Fehlerhäufigkeit einiger Fehler könnte durch zukünftige Verbesserungen noch weiter gesenkt werden.

Zusammenfassend kann gesagt werden, dass die beim Setzen eines Bügelperlenbildes auftretenden Fehler deutlich verringert werden konnten. Leider erkennt der entwickelte Fehlerkorrekturprozess nicht alle Fehler im Bügelperlenbild. Während der Arbeit stellte sich heraus, dass das Erkennen der Farbe einer Bügelperle mithilfe eines Kamerabildes eine schwere Aufgabe darstellt. Obwohl das entwickelte Erkennungsverfahren nicht alle Farben der Perlen richtig erkennt, konnte die Erkennungsrate im Verlauf der Arbeit stark gesteigert werden. Durch zukünftige Arbeiten könnte die Erkennungsrate noch verbessert werden, sodass Korrekturprozess alle Fehler erkennen und korrigieren kann.

A Testbilder



(a) Schwarzes 6 x 6 Perlen



(b) Buntes 12 x 12 Perlen

Abbildung 53: Testbilder

B Testläufe

Testlauf	Aufgetretene Fehler (von 36 Perlen)				
	V1	V2	V3	S1	S2
1.	10	0	0	21	0
2.	7	0	0	27	0
3.	6	1	0	26	0
4.	8	0	0	26	1
5.	10	0	0	21	0
Gesamt	41 (22,8%)	1 (0,6%)	0	121 (67,2%)	1 (0,6%)

Tabelle 7: Testläufe Schwarz 6x6

Testlauf	Aufgetretene Fehler (von 144 Perlen)				
	V1	V2	V3	S1	S2
1.	32	0	2	85 (4)	0
2.	47	1	0	110	0
3.	46	0	1	97	1
4.	53	0	3	102	2
5.	40	2	0	105	0
Gesamt	218 (30,3%)	3 (0,4%)	5 (0,7%)	489 (67,9%)	3 (0,4%)

Tabelle 8: Testläufe Bunt

Version	Testlauf	Aufgetretene Fehler (von 36 Perlen)		
		V1	V2	V3
0.3	1.	13	0	0
	2.	9	0	0
	3.	7	0	0
	4.	9	0	0
	5.	10	0	0
0.6	1.	1	0	0
	2.	2	0	0
	3.	0	0	0
	4.	1	0	0
	5.	0	0	0

Tabelle 9: Testläufe Schieber Schwarz 6x6

Version	Testlauf	Aufgetretene Fehler (von 36 Perlen)		
		S1	S2	V1
0.3	1.	29	1	-
	2.	25	2	-
	3.	26	1	-
	4.	23	0	-
	5.	26	2	-
0.4	1.	24	0	1
	2.	19	1	1
	3.	27	2	0
	4.	26	1	2
	5.	26	0	3
0.7	1.	7	0	2
	2.	15	0	0
	3.	11	0	2
	4.	5	0	1
	5.	9	0	1

Tabelle 10: Testläufe Setzrohr Schwarz 6x6

Testlauf	Aufgetretene Fehler (von 36 Perlen)				
	V1	V2	V3	S1	S2
1.	2	0	0	4	0
2.	0	0	0	8	0
3.	1	0	0	7	0
4.	0	0	0	13	0
5.	1	0	0	9	0
Gesamt	4 (2,2%)	0	0	41 (22,8%)	0

Tabelle 11: Testläufe Schwarz 6x6

Testlauf	Aufgetretene Fehler (von 144 Perlen)				
	V1	V2	V3	S1	S2
1.	18	0	0	49	0
2.	6	0	0	24	0
3.	4	0	0	56	0
4.	15	0	0	31	0
5.	3	0	0	27	0
Gesamt	46 (6,4%)	0	0	63 (26%)	0

Tabelle 12: Testläufe Bunt 12x12

C Farberkennung Tests

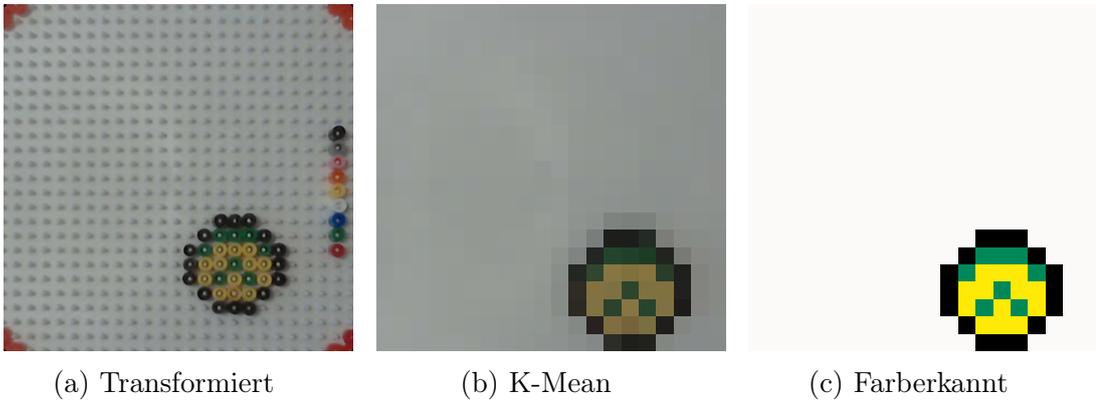


Abbildung 54: Farberkennung Test 1

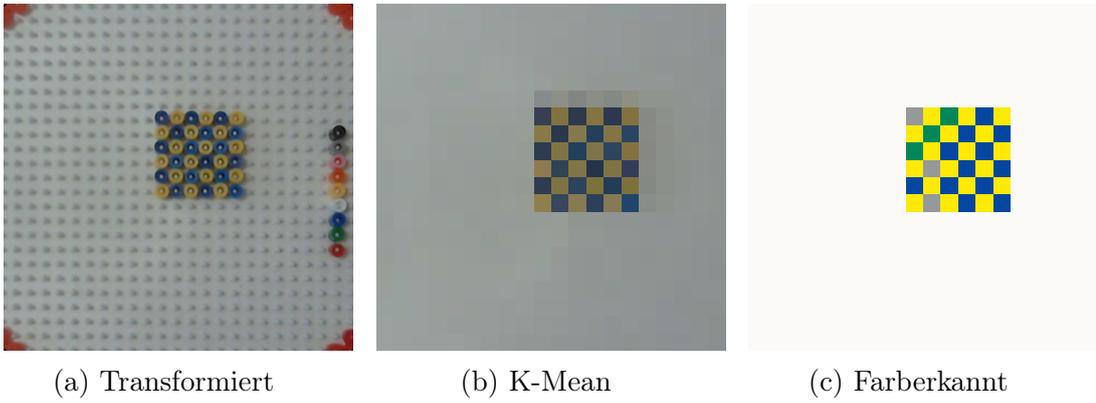
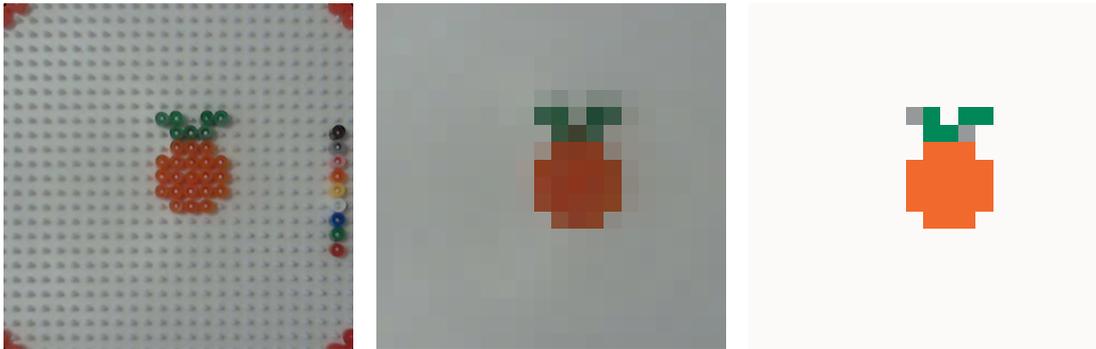


Abbildung 55: Farberkennung Test 2

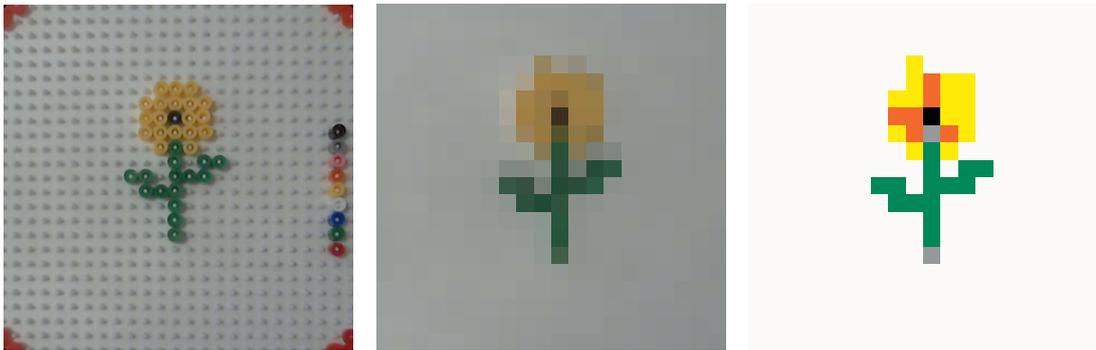


(a) Transformiert

(b) K-Mean

(c) Farberkannt

Abbildung 56: Farberkennung Test 3

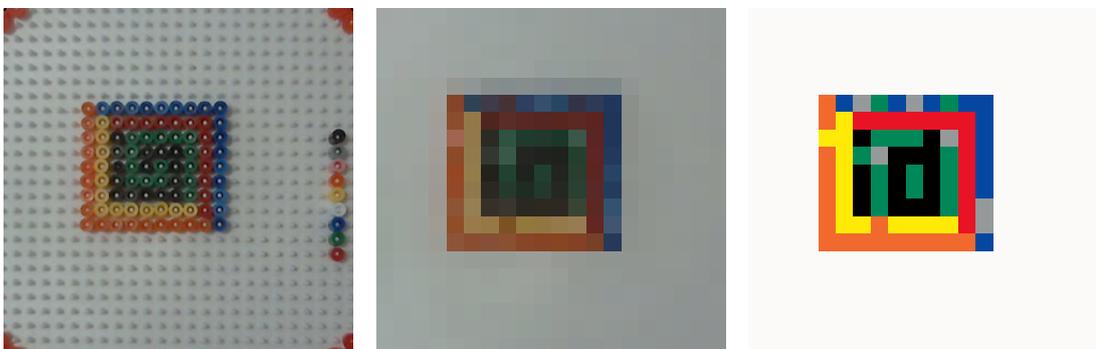


(a) Transformiert

(b) K-Mean

(c) Farberkannt

Abbildung 57: Farberkennung Test 4

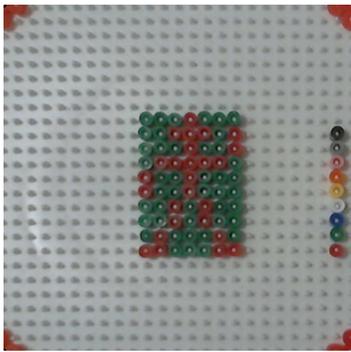


(a) Transformiert

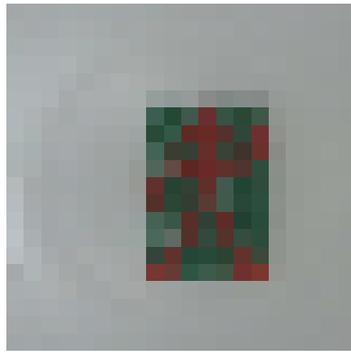
(b) K-Mean

(c) Farberkannt

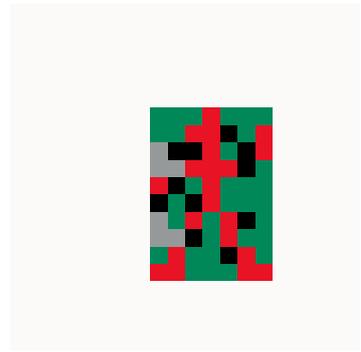
Abbildung 58: Farberkennung Test 5



(a) Transformiert



(b) K-Mean



(c) Farberkannt

Abbildung 59: Farberkennung Test 6

D Changelog der Modelle

Schieber	
V 0.1	<ul style="list-style-type: none">- Schieber entworfen- Neuer Stopper entworfen
V 0.2	<ul style="list-style-type: none">- Stopper: Aussparung für Perlenmagazin vergrößert; neuer Durchmesser: 8,4 mm
V 0.3	<ul style="list-style-type: none">- Am hinteren Ende oben Material entfernt
V 0.4	<ul style="list-style-type: none">- Öffnung auf 5,6 mm verkleinert
V 0.5	<ul style="list-style-type: none">- Öffnung trichterförmig obere Durchmesser: 6 mm unterer Durchmesser: 5,6 mm
V 0.6	<ul style="list-style-type: none">- Zentriereinheit hinzugefügt
V 0.7	<ul style="list-style-type: none">- Öffnungsdurchmesser verringert: 5,4 mm (Wegen Setzrohr)

Abbildung 60: Changelog Schieber

Setzrohr

V 0.1

- Setzrohr nachgebaut
- Halterung zur Befestigung hinzugefügt
- Lichtschrankshalterung hinzugefügt

V 0.2

- Löcher in Halterung korrigiert

V 0.3

- Lichtschrankshalterung: Abstand angepasst auf 4,5 mm
- Setzrohr mit Aceton behandelt

V 0.4

- Oberes Ende des Setzrohrs verlängert

V 0.5

- Setzrohr in drei Komponenten aufgeteilt

Setzrohr V 0.6

- Innendurchmesser reduziert: 5,5 mm

Setzrohr V 0.7

- Innendurchmesser erhöht: 5,8 mm

Abbildung 61: Changelog Setzrohr

Unterlegscheibe

V 0.1

- Unterlegscheibe entworfen

V 0.2

- Höhe verringert auf 0,9 mm

V 0.3

- Höhe verringert auf 0,4 mm

V 0.4

- Höhe verringert auf 0,3 mm

Abbildung 62: Changelog Unterlegscheibe

Literatur

- [AB] AB, Munkplast: *Nabbi*. www.nabbibeads.com. – Letzter Zugriff: 15.01.2016
- [A/S] A/S, Malte Haaning P.: *Hama*. <http://www.hama.dk/>. – Letzter Zugriff: 15.01.2016
- [BTVG06] BAY, Herbert ; TUYTELAARS, Tinne ; VAN GOOL, Luc: Surf: Speeded up robust features. In: *European conference on computer vision* Springer, 2006, S. 404–417
- [b.va] B.V., SES N.: *SES*. <http://ses-creative.com>. – Letzter Zugriff: 15.01.2016
- [B.Vb] B.V., Ultimaker: *Ultimaker*. <https://ultimaker.com>. – Letzter Zugriff: 08.04.2016
- [dev] DEVTEAM, OpenCV: *Camera calibration With OpenCV*. http://docs.opencv.org/3.0-beta/doc/tutorials/calib3d/camera_calibration/camera_calibration.html. – Letzter Zugriff: 11.02.2016
- [EH] EH, Adelson: *Checker Shadow Illusion*. http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html. – Letzter Zugriff: 15.04.2016
- [Gar] GARAGE, Willow: *OpenCV*. <http://opencv.org>. – Letzter Zugriff: 15.01.2016
- [ide] IDEASONBOARD: *UVC supported devices*. www.ideasonboard.org/uvc/. – Letzter Zugriff: 12.05.2016
- [Kin] KINTEL, Marius: *OpenSCAD*. <http://www.openscad.org/>. – Letzter Zugriff: 08.04.2016
- [Log] LOGITECH: *Logitech Webcam*. <http://www.logitech.com/en-roeu/product/hd-webcam-c270>. – Letzter Zugriff: 12.05.2016
- [qt2] *Qt Project*. <http://www.qt.io/developers/>. – Letzter Zugriff: 15.01.2016

- [S⁺14] SCHOFIELD, Andrew J. u. a.: Texture Variations Suppress Supra-threshold Brightness and Colour Variations. In: *PloS one* 9 (2014), Nr. 12, S. e114803
- [teaa] TEAM opencv d.: *Affine Transformations*. http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/warp_affine/warp_affine.html. – Letzter Zugriff: 15.02.2016
- [teab] TEAM opencv d.: *Clustering*. <http://docs.opencv.org/2.4/modules/core/doc/clustering.html>. – Letzter Zugriff: 05.05.2016
- [teac] TEAM opencv d.: *Hough Circle Transform*. http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html. – Letzter Zugriff: 17.03.2016
- [tead] TEAM opencv d.: *Hough Line Transform*. http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html?highlight=hough%20line%20transform. – Letzter Zugriff: 17.03.2016
- [teae] TEAM opencv d.: *perspective transformation*. http://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html. – Letzter Zugriff: 17.03.2016
- [teaf] TEAM opencv d.: *Template matching*. http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html. – Letzter Zugriff: 17.03.2016
- [Zie13] ZIEGLER, Henning: BeadBox: Entwicklung eines informatischen Systems zur automatisierten Erstellung von Bügelperlenbildern. (2013)

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichungen, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.

Oldenburg, den 27. Juni 2016

Marius Hacker