

# **Konstruktion einer Gigapixelkamera und Weiterverarbeitung der erzeugten Bilddaten**

Projektarbeit im Studiengang  
„Bachelor of Science for Print- and Mediatechnologies“  
an der Bergischen Universität Wuppertal

Bearbeitet von

Florian Meier

Matrikelnummer 920719



**BERGISCHE  
UNIVERSITÄT  
WUPPERTAL**







# Danksagung

Ich möchte mich bei meinen Projektbetreuern Prof. Dr.-Ing. Peter Urban und M. Sc. Götz M. Bongartz für die übermäßige Geduld bis zur finalen Abgabe dieser Arbeit bedanken.

Weiterhin danke ich für die Gelegenheit, diese spannende und vielseitige Arbeit durchführen zu können.

Im Weiteren danke ich Ronald Noster für die Fertigung der Metallteile für das Projekt, sowie Roland Bappert und M. Sc. Sascha Kratz für Unterstützung in elektronischen Fragen.

Weiterhin danke ich dem deutschsprachigen Teil des offiziellen Arduino-Forums für die Hilfe bei programmiertechnischen Fragen.

Mein größter Dank gilt meinem Vater. Ohne seine handwerklichen Fertigkeiten wäre dieses Projekt niemals zu dem geworden, was es jetzt ist.

Danke.



# Inhaltsverzeichnis

1. Einführung	1
2. Konstruktion einer Gigapixelkamera	
2.1 Vorüberlegungen	3
2.1.1 Fotografisch	3
2.1.2 Mechanisch	4
2.1.3 Elektronisch/Informationstechnisch	5
2.2 Umsetzung	5
2.2.1 Fotografisch	5
2.2.2 Mechanisch	8
2.2.3 Elektronisch	13
2.2.3.1 Mikrocontroller	13
2.2.3.2 Sonstige Schaltung	13
2.2.4 Informationstechnisch	19
2.2.4.1 Frontend	19
2.2.4.2 Backend	19
3. Weiterverarbeitung der Daten	23
3.1 Grundlegendes	23
3.2 Software/Algorithmen	24
3.2.1 Merkmalsfindung	24
3.2.2 Bildvergleich	25
3.2.3 Bildausrichtung/-verrechnung	26
3.3 Erstellen von Panoramen mit Hugin	28
3.3.1 Einrichtung von Hugin	28
3.3.2 Erstellung eines Panoramas	29
4. Ergebnisse	33
4.1 Probleme/Verbesserungsmöglichkeiten	33
4.2 Fazit	34
5. Quellennachweis und Anhang	35

# 1. Einführung

Ziel dieser Projektarbeit ist die Konstruktion einer so genannten „Gigapixelkamera“ und darüber hinaus die Weiterverarbeitung des so gewonnenen Bildmaterials mittels kostenfrei erhältlicher Software.

So soll gezeigt werden, dass die Erstellung von Gigapixelbildern nicht zwangsläufig mit großem Aufwand und/oder hohen Kosten verbunden ist.

Entgegen den Erwartungen ist mit „Gigapixelkamera“, nicht die Konstruktion eines digitalen Kamerasystems gemeint, dessen Auflösung im Bereich von Gigapixeln liegt. Als Gigapixelkamera wird ein Aufnahmesystem bezeichnet, welches den Benutzer dabei unterstützt, ein so genanntes „Gigapixelbild“ zu erstellen.

Als Gigapixelbild werden Aufnahmen bezeichnet, deren Endgröße deutlich über der mit normalen Kamerasystemen erreichbaren Größe liegt. Wenn gleich der Name eine Endgröße von über 1.000 Megapixeln impliziert, werden in dieser Arbeit auch Aufnahmen mit geringerer Größe als Gigapixelbild bezeichnet.

Aus technischer Sicht handelt es sich bei einem Gigapixelbild meist um ein mehrreihiges Flächenpanorama, das dementsprechend aus vielen Einzelbildern zusammengesetzt ist.

Neben mehrreihigen Panoramen gibt es auch einreihige Panoramen. Bei letzteren existiert nur eine „Linie“ von Bildern, die verrechnet wird, bei ersterem gibt es gleich mehrere Reihen die noch untereinander verrechnet werden.

[Weitere Informationen zu den Projektionsarten finden sich im Abschnitt über die Weiterverarbeitung des Bildmaterials.](#)

Essentiell dafür ist die Drehung der Kamera um einen bestimmten Fixpunkt entlang der X- und der Y-Ebene der Kamera. Nach jeder Änderung der Kameraausrichtung wird eine Aufnahme angefertigt.

Alle Einzelbilder werden schlußendlich durch eine geeignete Software-Lösung zu einem Gesamtbild verrechnet, dessen Größe dann im oben aufgeführten Bereich von hunderten Megapixeln oder aber im Bereich von Gigapixeln liegt.

Für die Erstellung von Gigapixelbildern gibt es mehrere Gründe, wobei vor allem zwei Gründe zum Tragen kommen.

Zum einen können Gigapixelbilder einen weitaus größeren Bildwinkel abbilden, als dies mit Weitwinkelobjektiven möglich wäre. Wenn gleich Fish-eye-Objektive horizontal und vertikal meist einen Bildwinkel von 180° aufweisen, so sind die Verzerrungen, obgleich mittlerweile relativ gut herausrechenbar, störend und trüben die Bildqualität. Ein

Gigapixelbild kann dabei, wie ein Panoramabild, theoretisch einen Bildwinkel von 360° aufnehmen und diesen entsprechend der Projektionsart abbilden.

Der zweite große Vorteil eines Gigapixelbildes liegt in der enormen Größe. Sie ermöglicht es, eine hohe Zahl an Details fotografisch aufzunehmen und bei Bedarf auf diese Daten zurückgreifen zu können. Damit sind Gigapixelbilder zum Beispiel zu Dokumentationszwecken sehr interessant.

Um solche Flächenpanoramen zu erstellen, gibt es für die Aufnahme eine große Anzahl an Hilfsmitteln. Diese unterscheiden sich neben der Fertigungsqualität und der allgemeinen technischen Ausführungen besonders auch im Grad der Automatisierung.

Einfache Systeme, meist unter dem Namen „Panoramaplatte“ gehandelt, erlauben es dem Nutzer auf einfache Art und Weise die Kamera in fest einstellbaren oder auch variablen Schritten um die Hochachse zu rotieren. Diese Panoramaplaten sind bereits ab etwa 50 Euro im Handel erhältlich.

Teure Aufnahmesysteme bieten einen weitaus größeren Funktionsumfang, sind aber auch entsprechend kostspieliger. Aufnahmesysteme im vierstelligen Euro-Bereich bieten oftmals eine vollständige Automatisierung der Aufnahme, was die Bewegung der Kamera um definierte oder er-



rechnet Schrittweilen, aber auch das Auslösen der Kamera beinhaltet.

Zwischen diesen beiden Extremen existiert eine Vielzahl von Systemen, auf die hier nicht näher eingegangen werden soll.

Im Rahmen dieser Arbeit soll ein mechanisches System konstruiert werden, welches unter Zuhilfenahme eines Mikrocontrollers, Motoren und geeigneter Steuerungssoftware eine relativ stark automatisierte Aufnahme der Einzelaufnahmen eines Gigapixelbildes ermöglicht.

Für die Weiterverarbeitung so gewonnener Bilddaten gibt es mittlerweile ebenfalls ein breites Spektrum an Lösungen. Auch hier zeigen sich enorme Unterschiede in Funktionsumfang und Preis.

Es zeigt sich, dass hier ein höherer Preis nicht unbedingt mit besseren Ergebnissen einhergeht. Um dem ursprünglichen Zweck der Arbeit zu genügen, zu zeigen, dass sich mit geringem Kostenaufwand Gigapixelbilder erstellen lassen, wird sich im Rahmen dieser Arbeit auf kostenfrei erhältliche Software beschränkt.

## 2. Konstruktion einer Gigapixelkamera

### 2.1 Vorüberlegungen

Insgesamt soll das Projekt relativ kostengünstig realisiert werden, um einen deutlichen Kontrast zu den kommerziell erhältlichen Systemen mit ähnlichem Automatisierungsgrad darzustellen.

Es dient damit auch als Technologie-Demo, um zu zeigen, dass mit frei erhältlichen Mitteln und vergleichsweise geringem technischen Aufwand entsprechende Systeme gebaut werden können.

Durch eine kostengünstige Realisierung kann diese Projektarbeit auch eine Basis für darauf aufbauende oder ähnlich gelagerte Projekte darstellen.

Einen wichtigen Punkt zur Projektarbeit trägt auch diese Dokumentation bei. Sie soll durch eine ausreichend hohe Detailtiefe in den Schilderungen als grobe Anleitung für den Bau eines entsprechenden Systems funktionieren, ohne sich zu sehr auf ein konkretes Vorgehen festzulegen.

Um die vorgenannten Ansprüche zu erfüllen, wird auf frei erhältliche, gut dokumentierte und kostengünstig zu beschaffende Hard- und Software zurückgegriffen.

Da die genaue Konstruktion meist von der verwendeten Kamera abhängig ist, dienen technische Zeichnungen eher zur schematischen Darstellung einer möglichen Lösung. Sie stellen nicht den Königsweg in der Konstruktion dar, können aber als fundierte Grundlage angesehen werden.

Für ein gutes Verständnis dieser Arbeit sind mindestens grundlegende Kenntnisse in den Bereichen der Elektrotechnik bzw. Elektronik, Informatik, Mechanik und Fotografie vonnöten. Die benötigten Kenntnisse gehen dabei stellenweise über das hinaus, was im Rahmen des Bachelor-Studiums der Druck- und Medientechnologie vermittelt wurde, lassen sich aber mit angemessenem Aufwand aneignen.

#### 2.1.1 Fotografisch

Zur Erstellung von Gigapixelbildern wird eine Vielzahl von Bildern über eine Software automatisch zusammengerechnet. Bei diesem Vorgang werden die Bilder zunächst analysiert, gleiche Bildteile werden als Überlappung der Einzelbilder erkannt und als Basis für die Verrechnung des Bildmaterials genutzt. Damit dieser Vorgang zufriedenstellend ausgeführt werden kann, müssen die Bilder einander natürlich auch ausreichend überlappen.

Näheres zur Funktionsweise der Zusammenrechnung findet sich in Kapitel 3 dieser Arbeit.

Die Überlappung des Bildmaterials ist auch deshalb notwendig, da Objektive keine perfekten optischen Systeme darstellen und die Ränder der Bilder in aller Regel Objektivfehler wie chromatische Aberrationen oder Verzeichnungen aufweisen. Sofern ausreichend Bildinformationen vorhanden sind, können so die schlechten Informationen von der Verarbeitungssoftware verworfen oder ignoriert werden, ohne die Qualität des Endergebnisses zu verschlechtern.

Daraus folgt, dass das Aufnahmesystem nach jedem Bild nicht soweit drehen oder neigen darf, dass keine ausreichende Überlappung mehr sichergestellt ist. Nach Rücksprache mit Nutzern von Panoramasytemen scheinen Werte zwischen 30% und 50% Überlappung pro Bild gute Richtwerte zu sein, um

eine hohe Qualität im zusammengesetzten Bildmaterial zu erhalten.

Ausgehend davon, dass das hier konstruierte System nicht nur für die Erstellung von Panoramabildern genutzt wird, bei denen sich die fotografierte Szenerie sehr weit von der Kamera entfernt befindet, muss auch den Parallaxenfehlern Rechnung getragen werden.

Parallaxenfehler beschreiben einen Fehler der entsteht, wenn man zwei Gegenstände betrachtet, die sich, zum Beispiel hintereinander, auf einer Achse befinden und dann der Beobachtungspunkt rotiert wird. Ein Gegenstand „schiebt“ sich dann hinter dem anderen hervor und das wahrgenommene Bild ändert sich radikal.

Aus der Messtechnik ist dieses Phänomen beim Ablesen von Instrumenten bekannt. Blickt man von seitlich auf ein analoges Messinstrument, so wird man einen falschen Messwert ablesen. Bei Messinstrumenten wird dies durch Spiegelskalen gelöst.

Die Behebung dieses Problems ist auch in der Fotografie relativ einfach lösbar. Die Kamera darf bei der Erstellung der Bilder nicht um einen beliebigen Punkt gedreht werden, sondern muss um den Knotenpunkt des Objektivs, oft besser als „Nodalpunkt“ bekannt, gedreht werden.

Dieser Punkt liegt, von vorne auf die Kamera gesehen, mittig auf X- und Y-Achse des Objektivs. Die Position des Nodalpunktes in Z-Richtung muss für die jeweilige Brennweite experimentell bestimmt werden, ändert sich bei Aufnahmen mit gleicher Brennweite dann allerdings nicht mehr.

## 2.1.2 Mechanisch

Wie bereits erläutert, soll die Kamera automatisiert gedreht werden können, um mehrere Aufnahmen einer Szenerie anzufertigen. Hierzu ist die Konstruktion eines geeigneten mechanischen Systems unabdingbar.

Die Hauptaufgabe dieses Systems ist das motorgestützte Drehen und Neigen der Kamera, um die Aufnahme mehrreihiger Panoramen zu ermöglichen.

Dabei muss sichergestellt sein, dass die Kamera sicher getragen werden kann und eine präzise Ansteuerung zur Positionsveränderung möglich ist.

Durch geeignete Lagerung der zu bewegenden Teile sollen Verschleiß und Ungenauigkeiten minimiert und die Laufruhe erhöht werden.

Durch die Verwendung leichter Materialien, wie beispielsweise Aluminium, kann ein zugleich sehr stabiles aber immer noch leichtes System gefertigt werden. Ein geringeres Endgewicht führt dazu, dass kleinere Antriebe genutzt werden können, was sich vor allem auf die Baugröße aber auch den Energieverbrauch des Systems auswirkt.

### 2.1.3 Elektronisch/Informationstechnisch

Die Automatisierung des Aufnahmesystems setzt eine Verarbeitung eingegebener oder über Sensorik erfasster Daten voraus. Da die Integration von Sensorik jeglicher Art in dieser Arbeit eher wenig bedeutsam ist und nur leichte Steuerungsaufgaben durchgeführt werden, wird kein besonders leistungsfähiger Mikrocontroller bzw. -prozessor benötigt.

Für einen möglichst hohen Grad an Automatisierung ist eine bidirektionale Kommunikation zwischen eingesetzter Kamera und Steuerung nützlich.

Mindestens eine unidirektionale Kommunikation (Mikrocontroller steuert Kamera) muss jedoch gegeben sein, damit ein automatisierter Ablauf funktionieren kann.

Es sei hier bereits vorweggenommen, dass mit den ausgewählten Komponenten eine bidirektionale Kommunikation nicht möglich ist. Diese ist zwar theoretisch möglich, allerdings stehen Kosten und Aufwand nicht mehr im Verhältnis zur eingesparten Arbeit bei der manuellen Steuerung bzw. Programmierung des Aufnahmesystems.

Die Steuerung soll am Ende in der Lage sein, die Motoren des mechanischen Systems anzusteuern

und so das Aufnahmesystem angemessen genau zu bewegen.

Ferner soll die Steuerung aus Eingangsparametern wie der eingestellten Brennweite und dem gewünschten Aufnahmebereich, die jeweils benötigte Bewegung des Aufnahmesystems ermitteln können.

Nach erfolgter Positionierung des Systems gehört die Auslösung der Kamera ebenfalls zu den Aufgaben des Systems..

Da die durchzuführenden Aufgaben alle nicht sonderlich komplex sind, kann auch der elektronische Teil des Systems relativ einfach gehalten werden. Seine Hauptaufgabe liegt in der Bereitstellung von Energie für die Antriebe, sowie der Darstellung von Informationen bezüglich des Fortschritts einer Aufnahme oder ähnlichem.

## 2.2 Umsetzung

### 2.2.1 Fotografisch

Wie bereits in den Abschnitten 1 und 2.1 erwähnt wurde, soll das Projekt kostengünstig realisiert werden, um Interessierten einen Nachbau zu ermöglichen. Aus diesem Grunde kommt vornehmlich eine digitale Kompaktkamera für die Verwendung in Frage.

Nachteile wie eine geringere Bildqualität aufgrund günstiger Linsenkonstruktionen oder eine nur eingeschränkte Änderungsmöglichkeit der Brennweite müssen zugunsten geringer Kosten hier in Kauf genommen werden.

Theoretisch ist die Verwendung einer beliebigen Kamera möglich, allerdings muss die Kamera vor allem eine Rahmenbedingung erfüllen, auf die im Folgenden eingegangen wird.

Die Kamera muss über einen Fernauslöser zu einem spezifischen Zeitpunkt auszulösen sein. Dieser Zeitpunkt ist nach der Bewegung des Aufnahmesystems an die gewünschte Position.

Während digitale Spiegelreflexkameras in aller Regel über Schnittstellen zum Anschluss von Fernauslösern verfügen oder Infrarot-Empfänger zur Fernauslösung integriert haben, sind solche Funktionalitäten bei digitalen Kompaktkameras nicht,

oder nur bei vergleichsweise hochpreisigen Modellen zu finden.

Für Kompaktkameras des Herstellers Canon gibt es eine Software, welche auch vielen digitalen Kompaktkameras der Einsteigerklasse diese Funktionalitäten zur Verfügung stellt. Es handelt sich dabei um das so genannte CHDK, das „Canon Hacker Development Kit“.

Das CHDK ist eine nicht permanente Erweiterung der Kamerafirmware. Sie wird beim Start von der Speicherkarte der Kamera geladen und erweitert dann den Funktionsumfang der Kamera in beträchtlichem Maße. So stellt das CHDK beispielsweise auch die Möglichkeit zur Fernauslösung über den USB-Anschluss der Kamera zur Verfügung.

Die Funktionsweise und den erweiterten Funktionsumfang des CHDK an dieser Stelle näher zu erläutern, würde den Umfang dieser Arbeit sprengen. Festzuhalten bleibt, dass das CHDK eine faszinierende und unglaublich umfangreiche Software darstellt. Vor dem Hintergrund, dass es sich dabei um Open Source-Software und ein ohne jegliche Unterstützung durch den Hersteller Canon durchgeführtes Projekt handelt, ist die Leistung der Beteiligten äußerst bemerkenswert.

Für weitere Informationen rund um das CHDK sind sowohl die englische Projektseite<sup>[1]</sup>, als auch das deutsche Anwenderforum<sup>[2]</sup> zu empfehlen.

In die engere Wahl für dieses Projekt kommen damit digitale Kompaktkameras der Marke Canon. Nach einiger Recherche fällt die Wahl auf eine Canon Powershot A490<sup>[3]</sup>.



Die verwendete Kamera: Canon Powershot A490

Bei der Canon Powershot A490 handelt es sich um eine digitale Kompaktkamera mit einer Auflösung von 10 Megapixeln und 3,3-fachen optischen Zoom. Die Bilder werden auf SD-Karten gespeichert, die Stromversorgung erfolgt über handelsübliche AA-Batterien bzw. Akkus. Mit einem Gewicht von etwa 175 g (mit Batterien) und Abmaßen von 92 x 62 x 31 mm ist sie leicht und kompakt, was die Konstruktion des mechanischen Systems vereinfacht. Dank ihrer ausreichend hohen Auflösung und dem optischen Zoom eignet sie sich gut für die Erstellung von Gigapixelbildern.

Die Kamera wird leider nicht mehr produziert und ist deshalb nur noch in Restbeständen verfügbar. Damit einher gehen schlechter oder gar nicht mehr vorhandener Support bei Problemen oder Reparaturen. Sollte hierauf Wert gelegt werden, ist ein aktuelles Modell zu bevorzugen.

Nach Installation des CHDK auf einer handelsüblichen 2 GB SD-Speicherkarte ist noch Platz für rund 700 JPEG-Bilder in bester Qualität vorhanden. Dies ist mehr als ausreichend für ein noch so großes Panorama.

Durch das CHDK besteht auch die Möglichkeit Bilder im RAW-Format aufzunehmen. So ist zwar nur noch für knapp 110 Bilder Platz auf der Speicherkarte, theoretisch lässt sich so aber eine weitaus höhere Bildqualität erzielen. Leider schränkt die Verwendung von RAW-Dateien die Weiterverarbeitung der erzeugten Daten etwas ein. Nicht jede Software zur Erstellung von Panoramen kann RAW-Dateien verarbeiten.

Dies sollte aber kein generelles Ausschlusskriterium für die Verwendung der RAW-Funktion des CHDK darstellen, es muss nur vor der Erstellung des Panoramas Zeit für die Entwicklung der RAW-Dateien mit einem entsprechenden Programm eingeplant werden.

Wie in Abschnitt 2.1.1 erwähnt, ist für die Bewegung des Aufnahmesystems der momentane Bild-

winkel der Kamera eine wichtige Kenngröße. Aus der Brennweite des Objektivs und der Diagonalen des Bildsensors lässt sich dieser ermitteln. Die meisten Digitalkameras speichern die Brennweite einer Aufnahme auch in den EXIF-Daten des Bildes.

Da über das CHDK bzw. ohne den Einsatz eines Computers eine bidirektionale Kommunikation mit der Kamera nicht möglich ist, muss die Brennweite normalerweise aus einem bereits aufgenommenen Foto ausgelesen werden. Die Zoom-Einstellung an einer digitalen Kompaktkamera auf einen vorher einmal eingestellten Wert wieder einzustellen gestaltet sich dabei oft als schwierig oder teilweise auch unmöglich.

Bei der verwendeten Kamera stellt dies dankenswerterweise kein großes Problem dar, da sie nur in festen Stufen zoomt und sich eine vorher gewählte Zoomstufe so leicht wiederfinden lässt.

Das CHDK kann auch hier die Umsetzung des Projektes erheblich erleichtern.

Es ist möglich, die aktuelle Brennweite auf dem Display der Kamera anzeigen zu lassen. Dabei besteht die Wahl zwischen der Anzeige des Zoomfaktors der realen Brennweite und der kleinbildäquivalenten Brennweite.

In der konkreten Umsetzung der Arbeit werden

der Bildsensor und die gewählte Brennweite erfasst. Basierend auf dem Crop-Faktor erfolgt so die Umrechnung in das Kleinbild-Format und darauf basierend die Berechnung der Bildwinkel.

Die Brennweite kann nach Auswahl der gewünschten Ausschnittsgröße einfach abgelesen und in die Steuerung des Aufnahmesystems übernommen werden.

Der gesuchte Bildwinkel berechnet sich in Abhängigkeit von der Brennweite und den Maßen des Bildsensors wie folgt<sup>[4]</sup>:

$$\text{Bildwinkel [rad]} = 2 \cdot \arctan \frac{d}{2 \cdot f}$$

$$\text{Bildwinkel [deg]} = 2 \cdot \arctan \frac{d}{2 \cdot f} * \frac{180}{\pi}$$

Dabei können für den Parameter „d“ die Breite, Höhe oder auch Diagonale des Bildsensors eingesetzt werden. „f“ ist jeweils die eingestellte Brennweite, die auf Basis der Sensorgröße umgerechnet wird.

Die gewählte Kamera verfügt über einen CCD-Sensor der Größe 1/2,3", wie man den technischen Daten entnehmen kann<sup>[3]</sup>.

Sensoren dieses Formates sind dabei 6,2 mm breit und 4,3 mm hoch<sup>[5]</sup>.

Aus den fixen Brennweitereinstellungen für die Kamera ergeben sich dann beispielhaft folgende horizontale und vertikale Bildwinkel:

Brennweite [mm]	Horizontal [°]	Vertikal [°]
6,6	50,3	36,1
8,0	42,4	30,1
9,8	35,1	24,7
12,0	29,0	20,3
14,8	23,7	16,5
17,5	20,1	14,0
21,6	16,3	11,4

Um bei der Aufnahme eine 50%ige Überlappung zu erhalten, werden diese Werte jeweils mit dem Faktor 0,5 multipliziert und als entsprechende Schrittweiten beim Verfahren des Systems genutzt.

Mittels der erweiterten Einstellmöglichkeiten des CHDK werden einige Automatik-Funktionen der Kamera übergangen. Die Aufnahme erfordert so von Seiten des Benutzers etwas mehr Aufwand, weil die Belichtungszeit von Hand eingestellt werden muss, sorgt aber für konsistente Bilder.

Würde man die Belichtungsmessung nicht überschreiben, würden die Bilder, abhängig vom Bildinhalt oder der Beleuchtungssituation, unterschiedlich belichtet werden, was bei der Weiterverarbeitung der Daten ein gravierendes Problem darstellen und die Nutzung der Bilder unter Umständen unmöglich machen könnte.

Es empfiehlt sich, Testaufnahmen von verschiedenen Ausschnitten des gewünschten Panoramas aufzunehmen, um so optimale Werte für die Belichtung zu ermitteln. Durch Wahl eines Mittelwertes und Einsatz elektronischer Bildbearbeitung im Anschluss lassen sich geringe Abweichungen von der Vorstellung auch im Nachhinein noch anpassen.

Um auch die Blendeneinstellung der Kamera zu fixieren, kann der „Feuerwerk“-Modus der Kamera genutzt werden. In diesem Modus wird die Blendeneinstellung der Kamera auf einen festen Wert eingestellt, so dass am Ende keine großen Schwankungen mehr in der Helligkeit der Bilder vorhanden sind.

Abhängig vom Einsatzzweck könnte es sich weiterhin anbieten, den Autofokus der Kamera zu übernehmen und von Hand eine Fokusdistanz einzustellen. So ließe sich speziell auf Objekte in einer spezifischen Entfernung fokussieren.

Sofern Objekte in größerer Entfernung aufgenommen werden, bietet sich eventuell auch die Nutzung der Hyperfokaldistanz an. Das CHDK kann die Kamera dann automatisch auf die entsprechende Fokuseinstellung einstellen, die Nutzung der Hyperfokaldistanz ist allerdings umstritten, weil sie per Definition nur „gerade noch [...] akzeptable Unschärfe“<sup>[6]</sup> leistet.

Im praktischen Betrieb zeigte der Autofokus-Modus, ausreichend Licht vorausgesetzt, recht

annehmbare Ergebnisse. Es ist davon auszugehen, dass aufgrund des kleinen Sensors der Kamera sowieso recht schnell die Fokussierung auf „Unendlich“ erreicht ist und somit nicht weiter darüber nachgedacht werden muss.

Darüber hinaus wurde über das CHDK eine Fernauslösung der Kamera ermöglicht. Die entsprechende Einstellung findet sich im CHDK-Menü unter „CHDK-Einstellungen“ → „USB-Fernbedienung-Parameter“.

Bei der konkreten Umsetzung wurde als Funktionsmodus „OnePush“ gewählt, damit wird fokussiert und sofort ausgelöst, nachdem die Fokussierung abgeschlossen ist. Mit dem Modus „TwoPush“ würde zunächst fokussiert, bei erneutem „Drücken“ innerhalb von 500 ms würde ausgelöst<sup>[7]</sup>.

Wird nun am USB-Anschluss der Kamera eine Spannung von 5 Volt angelegt, löst die Kamera aus. Dieser Prozess kann sehr leicht von der Elektronik des Systems übernommen werden und ermöglicht so eine Auslösung nach Beendigung des Verfahrens des Systems.


## 2.2.2 Mechanisch

Wie bereits erläutert, liegt die Hauptaufgabe des mechanischen Systems darin, die Kamera zu tragen, zu drehen und zu neigen. Um diese Funktion sicherzustellen, muss das System eine gewisse Festigkeit aufweisen, aber dennoch leichtgängig laufen, um von Motoren ohne großen Kraftaufwand bewegt zu werden.

Das mechanische System besteht, wie auch im Folgenden zu sehen ist, aus zwei U-förmigen Konstruktionen.

Dabei ist die innere Konstruktion für die Neigung der Kamera zuständig, die äußere für die Drehung der Kamera. Natürlich können beide Funktionen gleichzeitig durchgeführt werden. In der konkreten Umsetzung erfolgt dies allerdings nacheinander, da es sich informationstechnisch leichter umsetzen lässt. Aufgrund der geringen Verfahrenzeiten der Servos und der Tatsache, dass ein Foto erst dann gemacht wird, wenn das System wieder ruhig steht, stellt dies auch kein Problem dar.

Um besagte Funktionen möglichst leichtgängig ausführen zu können, sind die entsprechenden Drehachsen kugelgelagert.

Bei der Auswahl der Kugellager ist zu bedenken, dass die Bolzen des inneren Teils idealerweise in Rillenkugellagern  gelagert werden sollten.

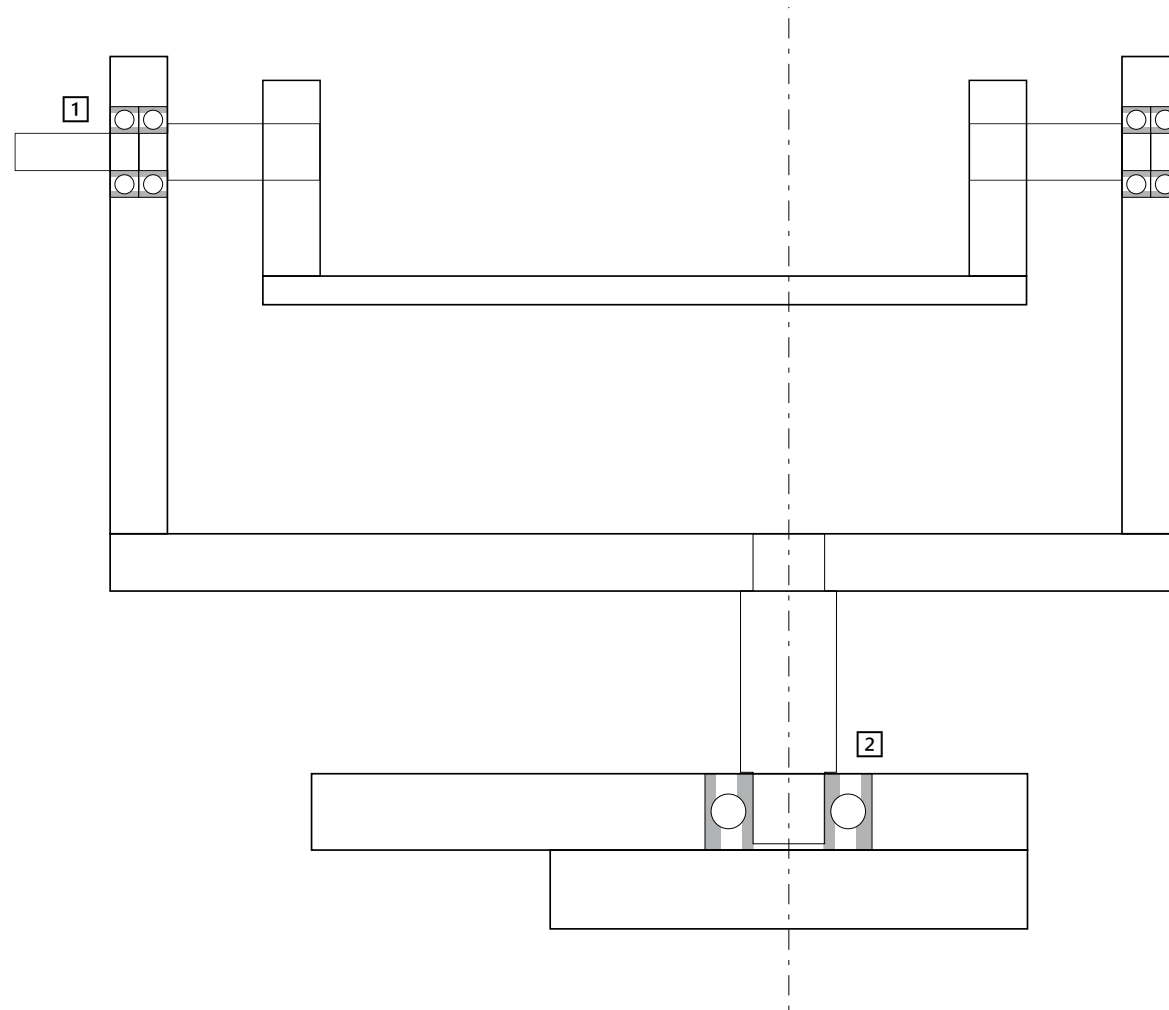
Diese können die Radialkräfte optimal aufnehmen und laufen damit äußerst verschleissarm. Um die Lager zu entlasten und die Bolzen noch stabiler zu führen, sind hier zwei Reihen der gleichen Lager eingesetzt worden.

Der äußere Teil, der die Konstruktion dreht, wird idealerweise in einem Schrägrollenkugellager [2] gelagert. Schrägrollenkugellager können Kräfte in Axialrichtung optimal aufnehmen, die hier überwiegend auftreten. Die radialen Kräfte, die durch die Kraftübertragung auf die Achse auftreten, sind vergleichsweise gering und können vernachlässigt werden.

Bis auf die oben erwähnten Bolzen sind die großen Teile der Konstruktion aus Aluminium gefertigt. Aluminium hat aufgrund seiner Dichte eine vergleichsweise geringe Masse und ist dennoch ausreichend stabil für den Anwendungszweck. Es ist weiterhin leicht bearbeitbar und günstig zu erwerben.

Denkbar wäre allerdings auch eine Konstruktion aus Holz, das natürlich deutlich leichter ist und damit die Verwendung kleinerer Motoren ermöglicht. Bei Auswahl des richtigen Materials ließe sich auch damit eine gute Festigkeit erreichen, ferner ist Holz ohne großen Werkzeugaufwand leicht zu bearbeiten.

Die Einzelteile werden über Schraubverbindungen, in den meisten Fällen vom Typ M4, miteinander



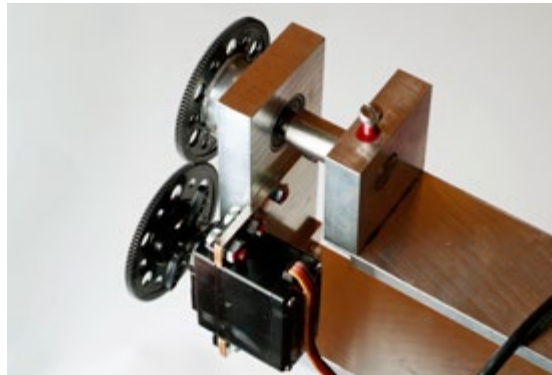
Das mechanische System der konkreten Umsetzung der Arbeit. Zu sehen sind inneres und äußeres „U“ mit Verbindung über Bolzen und die entsprechende Lagerung der Teile.

Aus Gründen der Übersichtlichkeit wurde auf die Einzeichnung von Schraubverbindungen und Bohrungen verzichtet. Ebenso fehlen die Servomotoren und ihre Halterungen.



verbunden. Die Bolzen und Lager werden durch Übergangspassungen mit den jeweiligen Teilen bzw. miteinander verbunden. Um die Übergangspassung aber auch die Schraubverbindungen zu sichern, werden die Bauteile wahlweise mittels eines Metallklebers, mit Schraubensicherungslack oder aber durch Schrauben zur Klemmung der Passung zusätzlich gesichert.

Zum Antrieb der mechanischen Teile werden Servomotoren eingesetzt. Diese bieten eine, im Vergleich zu ihrer Größe, hohe Kraft und sind damit gut für die Bewegung der mechanischen Konstruktion geeignet.



Antrieb der Neigungsachse mittels Servomotor, Kraftübertragung über Zahnräder aus dem Modellbaubereich.

Gut zu sehen ist die Sicherung der Schrauben mit einem dafür geeigneten Lack.

Bei der Durchführung dieses Projektes wurden konkret Servomotoren gewählt, die laut Herstellerangaben, eine Haltekraft von 13 kg auf 1 cm Hebelarm aufweisen. Bedenkt man die Dimensionen der Kamera und des mechanischen Systems, so wird offensichtlich, dass die Motoren etwas überdimensioniert sind.

Ausgewählt wurde dieser Typ Servomotor, weil er innerhalb seines Getriebes Metallzahnäder nutzt und so eine höhere Lebensdauer und Festigkeit zu erwarten ist. Ob dies wirklich von Relevanz ist, lässt sich nicht mit Bestimmtheit sagen, leistungs- und haltbarkeitsmäßig ist man mit dieser Auswahl allerdings auf der sicheren Seite.

Auf einen weiteren Vorteil von Servomotoren, die sehr unkomplizierte Ansteuerung, wird später eingegangen.

Um die Kraft zwischen den Drehachsen und dem entsprechenden Servomotor zu übertragen, werden Zahnräder aus dem Bereich des Helikoptermodellbaus eingesetzt. Mit einer hohen Zahnzahl weisen Sie eine für dieses Projekt ausreichend feine Verzahnung auf. Dies ermöglicht eine ausreichend spielfreie Übertragung der Kraft und damit das genaue Anfahren einer spezifischen Position.

Die Zahnräder werden auf Buchsen geklebt, die sich auf die Drehachsen aufstecken und durch Schrauben sichern lassen. Die Zahnräder auf den Servo-

motoren sind auf Servokränze geschraubt, die zum Lieferumfang der Servomotoren gehören. Dank dieser Konstruktion lässt sich die benötigte Kraft sehr gut übertragen.

Das Übersetzungsverhältnis zwischen den Zahnrädern beträgt sowohl bei der Dreh- als auch bei der Neigungsachse 1 zu 1.

Um die Elektronik unterzubringen, wird ein Kunststoffgehäuse genutzt. In dieses Gehäuse werden der Akku, die Platine, Display und Joystick eingebaut. Darüber hinaus besitzt das Gehäuse Cinch-Buchsen, mittels derer sich das mechanische System und die Steuerung verbinden lassen. Öffnungen für die Cinch-Buchsen, Display, Joystick und Schalter wurden mit einem entsprechenden Handfräsgewerkzeug, besser unter dem Handelsnamen „Dremel“ bekannt, angefertigt.

Die Cinch-Buchsen wurden genutzt, weil sie, aber auch Kabel dafür, besonders kostengünstig erhältlich sind. Online-Händler wie Pollin<sup>[8]</sup> bieten Buchsen mit drei Anschlüssen bereits für 50 Cent an. Mit einer Buchsenleiste mit 3 Buchsen lassen sich theoretisch zwei Servos komplett steuern (gemeinsame Masse, Nutzung von Ader und Schirmung für Stromversorgung und als Datenleitung). Bei der Umsetzung der Arbeit wurden allerdings drei einzelne Buchsenleisten genutzt, um Störungen in den einzelnen Leitungen ausschließen zu können (Schirmung behält ihre Funktion bei).



Gehäuse für die Elektronik

Zu sehen sind links beziehungsweise oben: Joystick und Display zur Dateneingabe und -überprüfung, Kippschalter (links: System ein-/ausschalten, rechts: Displaybeleuchtung ein-/ausschalten)

Dann zunächst ein zweiadriges Kabel für die Auslösung der Kamera, anschließend die Stromversorgungen (rot und weiß) und Steuerleitungen (gelb) für die Servomotoren.

Die Adern der Kabel weisen einen so hohen Querschnitt auf, das selbst deutlich größere Lasten ohne Probleme mit Spannung versorgt werden könnten. Die farbliche Kennzeichnung der einzelnen Stecker erleichtert dem Benutzer darüber hinaus den Anschluss der Kabelpeitsche an die Steuerbox.

Das andere Ende der Kabelpeitsche ist über steckbare Reihenklemmen an das mechanische System anzuschließen. Hier werden die Flachbandkabel der Servomotoren und ein entsprechend gekürztes und abisoliertes Micro-USB-Kabel genutzt, um Steuerung und Auslösung zu übernehmen.



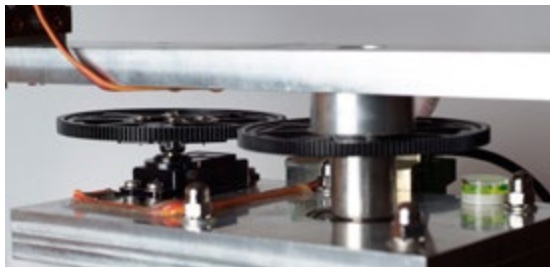
Reihenklemme zum Anschluss der Kabelpeitsche  
Links das Auslösekabel, rechts daneben Stromversorgung und Steuerung der Servomotoren



Gehäuse für die Elektronik von oben gesehen.



Auf die Grundplatte aufgeklebte Libelle zur Ausrichtung.



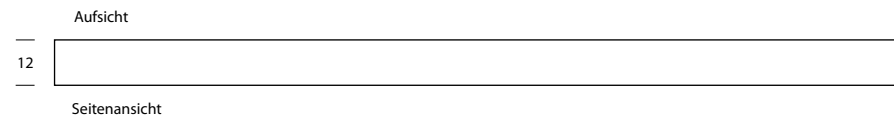
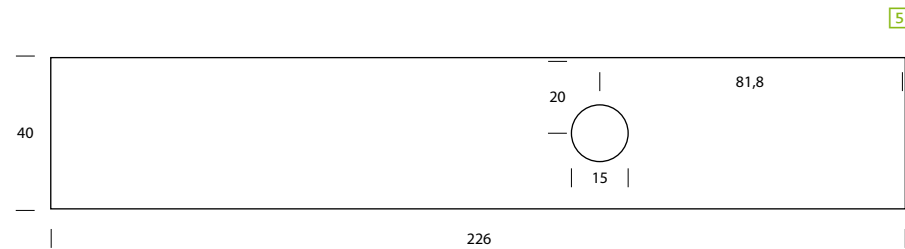
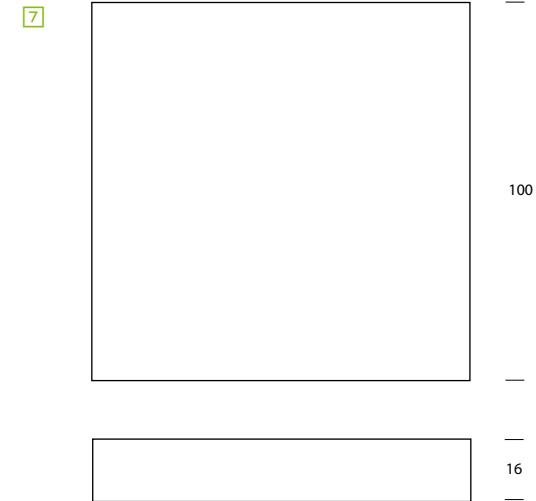
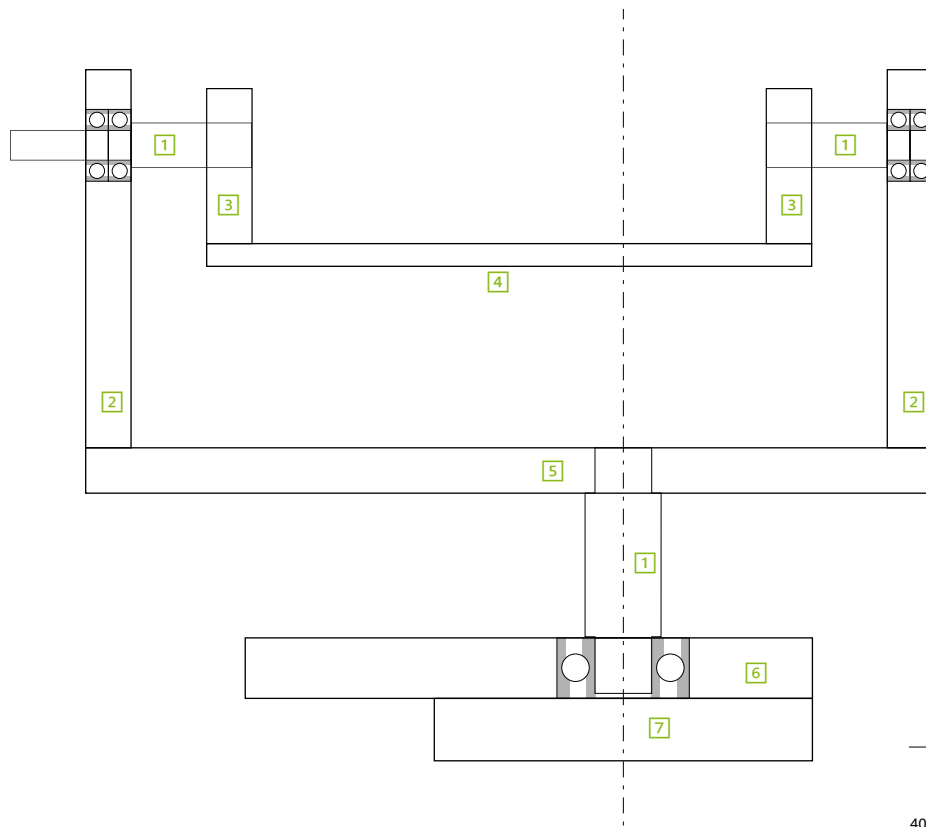
Antrieb der Drehachse mittels Servomotor und Zahnrädern aus dem Modellbaubereich.

Unschärf im Vordergrund ist die Buchse zu erkennen, auf die das Zahnrad aufgeklebt ist. Die Buchse ist auf die Drehachse geschoben und mit einer Schraube gesichert (nicht sichtbar).

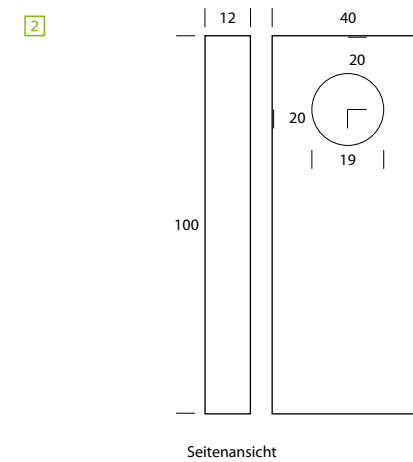
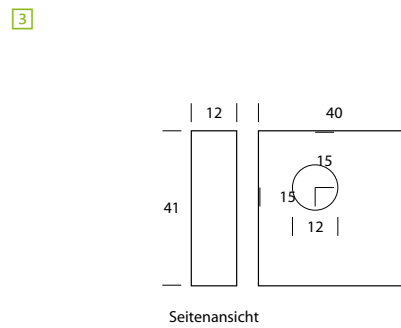
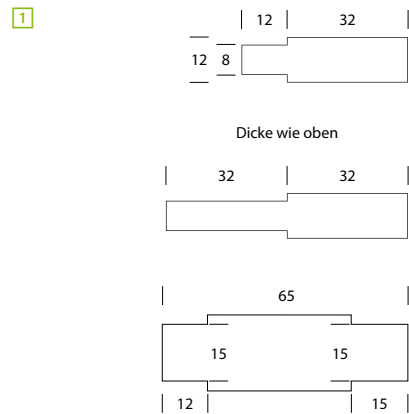
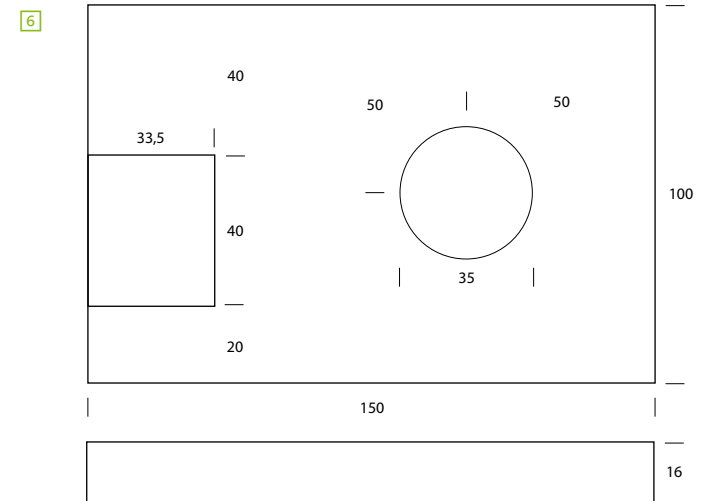
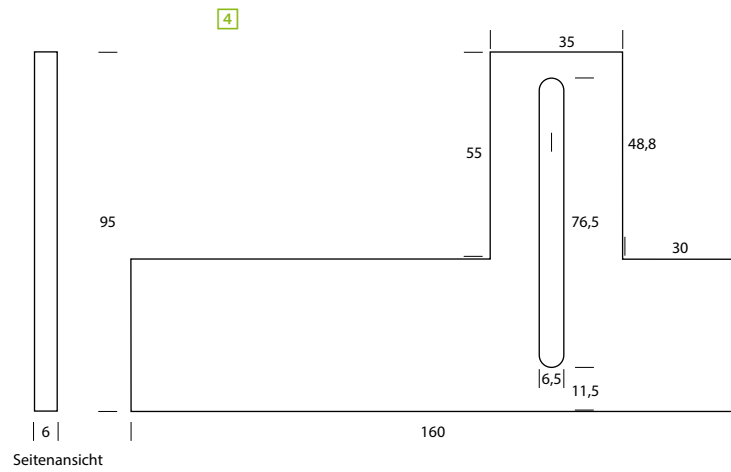
Rechts im Bild ist eine Libelle zu sehen. Mittels dieser lässt sich die waagerechte Ausrichtung des Aufnahmesystems prüfen.



Aufnahmesystem im Einsatz.



Gesamte Konstruktion und deren Einzelteile  
(50% Originalgröße)



## 2.2.3 Elektronisch

Zur Steuerung des Gesamtsystems kommen verschiedene elektrische bzw. elektronische Bauteile zur Anwendung. Im Folgenden soll kurz auf die elektrischen bzw. elektronischen Schaltungen und Bauteile eingegangen werden, die zum Gelingen des Projektes beitragen.

### 2.2.3.1 Mikrocontroller

Zur Steuerung des mechanischen Systems mittels eines Mikrocontrollers wird ein Arduino ausgewählt. Bei Arduino handelt es sich um eine Open Source-(Hardware-)Plattform auf Basis verschiedener ATMEGA-Mikrocontroller. Das Arduino-Projekt erweitert die ATMEGA-Controller um eine Platine mit USB-Anschluss zur einfachen Programmierung und Einbindung der Controller, stellt weiterhin aber auch eine eigene Programmiersprache, die „Arduino programming language“ (im Folgenden „APL“), sowie eine Softwareentwicklungsumgebung („Arduino IDE“) zur Verfügung. Mehr Informationen finden sich auf der offiziellen Internetseite des Projektes<sup>[9]</sup>.

Bereits vorhandene Entwicklungsumgebungen wie zum Beispiel „Eclipse“<sup>[10]</sup> können allerdings auch zur Entwicklung der Software genutzt werden.

Neben der APL kann auch normales C zur Programmierung verwendet werden. Intern wird die APL ebenfalls in C umgewandelt.

Durch seine einfache Sprache und die gute Hardwareplattform bietet Arduino einen leichten Einstieg in die Programmierung und die Arbeit mit Mikrocontrollern.

Eine große Online-Community steht bei Fragen zur Verfügung und viele Elektronik-Händler bieten Zusatzhardware, so genannte „Shields“, an, mit denen sich der Funktionsumfang einfach erweitern lässt.

Im Folgenden wird der Begriff „Arduino“ der Einfachheit halber als Sammelbegriff für den Mikrocontroller, das Programmierboard und die gesamte Entwicklungsumgebung genutzt. Dies entspricht der gängigen Praxis, die genaue Bedeutung ergibt sich in aller Regel aus dem Kontext.

Passend zu den Shields existieren innerhalb des Arduino-Kosmos weiterhin viele fertige Bibliotheken, mithilfe derer sich Aufgaben sehr einfach erledigen lassen. Neben einigen Standard-Bibliotheken aus der Programmiersprache C stehen auch Bibliotheken für die verwendeten Servomotoren oder das Display zur Verfügung.

### 2.2.3.2 Sonstige Schaltung

Die Energieversorgung des Systems erfolgt über einen Lithium-Polymer-Akku aus dem Bereich des Modellsports. So kann der Einsatz des Systems auch fernab vom Stromnetz erfolgen. Der Akku setzt

allerdings ein besonders Ladegerät voraus, das dem fertigen Modell dieser Arbeit mitgegeben wird.

Der genutzte Akku erzeugt aus zwei Akkuzellen mit je 3,7 Volt bei Reihenschaltung eine Gesamtnennspannung von 7,4 Volt. Die Leerlaufspannung liegt entsprechend etwas höher. Die Nennspannung ist für den verwendeten Mikrocontroller und das verwendete Display zu hoch.

Laut Herstellerspezifikationen sollten die Servomotoren mit einer Spannung zwischen 4,8 Volt und 7,2 Volt betrieben werden. Wie Messungen ergaben, liegt die Stromaufnahme eines Servomotors bei 5 Volt Versorgungsspannung im Leerlauf bei etwa 600 mA, unter großer Last steigt sie bis auf das Doppelte an.

Dieser Fakt schließt die Verwendung normaler Akkus oder Batterien leider aus. Eigene Versuche haben gezeigt, dass selbst hochwertige Akkus kaum in der Lage sind, die benötigten Ströme entsprechend schnell bereitzustellen. Als Konsequenz der Akku-„Trägheit“ brach in der Praxis die gesamte Spannungsversorgung zusammen, was zu einem Ausfall beziehungsweise Neustart des Arduino führte.

Der Modellsportakku hat laut Hersteller eine Spitzenstromabgabe von 30 A. Beim normalen Einsatz solcher Akkus stellen solche Ströme keine Seltenheit dar, in diesem Projekt treten entspre-

chende Ströme nicht auf. Dem Autor gelang es bei Verwendung dieses Akkus nicht, die vorher beschriebenen Zusammenbrüche des Systems herbeizuführen, was für die Verwendung des Akkus in diesem Anwendungszweck spricht.

Um die benötigte Leistung für die Servomotoren zur Verfügung zu stellen, werden Festspannungsregler vom Typ L 78S05 eingesetzt. Diese erzeugen bei einer Eingangsspannung ab 6 Volt eine Ausgangsspannung von 5 Volt, bei einem maximalen Strom von 2 Ampere.

Idealerweise sollten Spannungswandler entsprechender Leistung gekühlt werden, da unter Umständen, bei großer Differenz zwischen Ein- und Ausgangsspannung, eine erhebliche Abwärme anfällt. Da im gewählten Gehäuse nur begrenzter Platz zur Verfügung steht, kann dies nicht im idealen Rahmen geschehen.

Unter Berücksichtigung einiger Randfaktoren scheint eine fehlende Kühlung allerdings kein großes Problem darzustellen: Die Spannungsregler sind nur vergleichsweise kurz unter Last in Benutzung und produzieren nur dann entsprechende Abwärme, wenn das mechanische System ein Stück gedreht oder geneigt wird. Dieser Zeitraum dürfte etwa 1 Sekunde pro Bewegung betragen.

Danach wird zunächst eine Aufnahme durchgeführt, erst dann wird das System wieder bewegt.

Dieser Vorgang bietet mehrere Sekunden Zeit, in denen die Spannungsregler wieder abkühlen können.

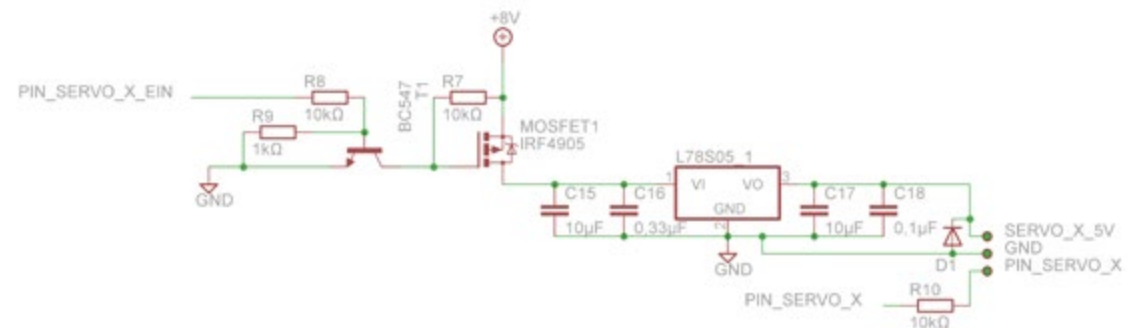
Darüber hinaus ist die Differenz zwischen Ein- und Ausgangsspannung nicht so dramatisch hoch, das hier besonders viel Abwärme zur Kompensation dieser Differenz zu erwarten wäre.

Um die Belastung jedes einzelnen Spannungsreglers noch weiter zu reduzieren, wird jedem Servomotor ein eigener Spannungsregler zugewiesen.

Bei einem Preis von etwa 90 Cent pro Spannungsregler trägt dies nicht merklich zu den Gesamtkosten des Projektes bei. Die Kosten für weitere Bauteile zur Beschaltung des Spannungsreglers

(Stützkondensatoren und Kondensatoren gegen Hochfrequenz-Schwingungen des Bauteils) liegen ebenfalls im Bereich weniger Cent.

Wie man sieht, sind in dem Schaltplan auch ein Transistor, ein MOSFET und verschiedene Widerstände aufgeführt. Diese Beschaltung erlaubt es, über einen der Ausgänge des Arduino die Spannungswandler ein- und auszuschalten. So lässt sich die Spannungsversorgung der Servomotoren abschalten, um Energie zu sparen. In der konkreten Umsetzung wurde bisher darauf verzichtet, sollte festgestellt werden, dass die dauerhafte Versorgung der Servos mit Spannung ein Problem darstellt (Signalrauschen, „Zittern“, sonstige Störungen), lässt sich dies nachträglich erledigen.



Beschaltung der Spannungswandler L 78S05 für die Servomotoren mit Möglichkeit zur Abschaltung dank MOSFET.

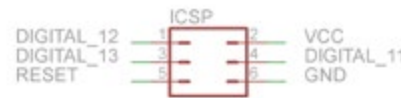
Normalerweise stellt das Arduino-Programmierboard eigene Spannungswandler für 3,3 Volt und 5 Volt zur Verfügung. So kann bei einem breiten Eingangsspannungsbereich ab 5 Volt eine Vielzahl an Komponenten, natürlich inklusive des ATMEGA-Mikrocontrollers, betrieben werden.

Der ATMEGA-Chip in dieser Arbeit allerdings vom Arduino-Programmierboard entnommen und in einen IC-Sockel gesetzt. So bleibt das Programmierboard uneingebunden und die gesamte Schaltung kann kompakter ausgeführt werden.

Mit dem Standalone-Aufbau des ATMEGA-Chips gehen natürlich die Spannungswandler des Boards verloren. Für den ATMEGA-Chip mit seiner Schaltung und das Display müssen damit weitere Spannungswandler zum Einsatz kommen.

Der ATMEGA-Chip benötigt eine Spannung von 5 Volt bei vergleichsweise geringer Leistungsaufnahme. Hier kann ein L 7805 eingesetzt werden. Er ähnelt der bereits genutzten Version L 78S05, stellt jedoch „nur“ bis zu 1 Ampere Strom bei 5 Volt Ausgangsspannung zur Verfügung. Die Anforderungen des ATMEGA und sonstiger, mit 5 Volt betriebener Bauteile, werden damit immer noch bei weitem übertroffen. Die Beschaltung erfolgt hier gleich wie bei den L 78S05 mit Stützkondensatoren und zusätzlichen Kondensatoren ein- und ausgangsseitig gegen Hochfrequenz-Schwingungen des Bauteils.

Um den ATMEGA-Chip während des Programmiervorgangs nicht jedes Mal aus seiner IC-Halterung entfernen zu müssen, wurden Buchsenstecker, so genannte ICSP-Header, installiert. Mit ihnen kann der Controller im eingebauten Zustand über ein Programmiergerät neu programmiert werden.



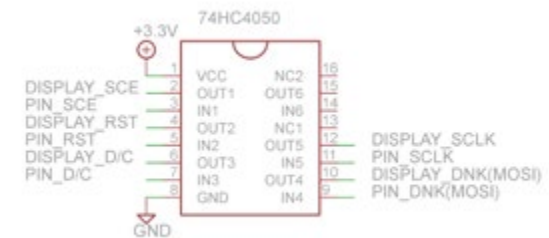
Beschaltung des ICSP-Headers zur Programmierung des Chips im eingebauten Zustand.

Zur Anzeige der Daten wird ein kleines Flüssigkristalldisplay eingesetzt. Hierbei handelt es sich um ein Display das auch in älteren Nokia-Mobiltelefonen vom Typ „5110“ verbaut ist. Dieser Displaytyp ist gut dokumentiert und zur einfachen Verwendung bereits auf einer kleinen Platine mit samt ausgeführten Anschlüssen verbaut, es handelt sich dabei um eines der bereits erwähnten „Shields“, die den Aufbau auch komplexerer Schaltungen mit einem Arduino vereinfachen.

Das Display benötigt für seinen Betrieb und seine Hintergrundbeleuchtung eine Spannung von 3,3 Volt. Diese wird durch einen Spannungswandler vom Typ „LD 1117 V 33“ bereit gestellt. Bei einer Ausgangsspannung von 3,3 Volt und einem maxi-

malen Strom von 800 mA eignet er sich gut für den Betrieb des Displays. Die Beschaltung dieses Spannungswandlers ähnelt der der L 78(S)05, wobei hier Kondensatoren mit anderen Kapazitäten zum Einsatz kommen.

Da das Display mit Eingangspegeln von 3,3 V arbeitet, der ATMEGA-Chip aber 5 V Ausgangspegel hat, muss hier ein Spannungswandel der Pegel durchgeführt werden, um eine Beschädigung des Displays zu verhindern. Der Pegelwandler-IC „74HC4050“ erledigt diese Aufgabe. Er wandelt einen Eingangspegel größer des Ausgangspegels in einen beliebigen, im konkreten Fall 3,3 Volt, Ausgangspegel um. Dies ließe sich auch mittels Widerständen über Spannungsteilerschaltungen erledigen, Bauteil- und Aufbauaufwand wären allerdings deutlich höher als beim Einsatz des IC.



Der IC 74HC4050 wandelt die Eingangspegel IN1–IN5 von 5 Volt herunter auf 3,3 Volt Ausgangspegel OUT1–OUT5.

So wird eine Beschädigung des angeschlossenen Displays vermieden.

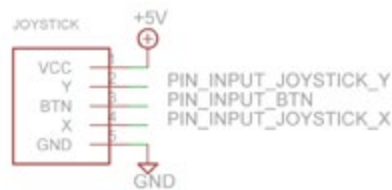


Wie bereits in Abschnitt 2.1.3 erwähnt wurde, ist eine bidirektionale Kommunikation mit den gewählten Komponenten nicht möglich.

Die Eingabe von Daten erfolgt deshalb über einen kleinen Joystick. Bei dem gewählten Joystick handelt es sich um einen 4-Achsen-Joystick mit integriertem Taster in seiner Mittelstellung. Der Joystick verfügt innerhalb der Achsen über Potentiometer. Basierend auf der Referenzspannung und der tatsächlich anliegenden Spannung kann der Arduino die Stellung des Joysticks erkennen und entsprechend reagieren.

Die Achsen dienen hierbei zur Einstellung oder Auswahl von Werten, der Taster in Mittelstellung zur Bestätigung von Eingaben.

Auch der Joystick ist als Shield ausgeführt und damit sehr einfach in das Projekt integrierbar.



Das verwendete Joystick-Shield nutzt einen Joystick, der in Spielkonsolen vom Typ „Playstation 2“ ebenfalls zum Einsatz kam.

Die Auslösung der Kamera erfolgt derart, dass am USB-Anschluss der Kamera eine Spannung von 5 Volt angelegt wird. Hierzu werden ein Transistor und die Spannungsversorgung vom ATMEGA-Chip genutzt.

Die Montage sämtlicher Bauteile findet auf einer Lochrasterplatine statt. Die Fertigung einer speziell gelayouteten Platine wäre eine deutlich professionellere Lösung, ist aber ohne entsprechende Ausrüstung und Kenntnisse nicht machbar. Anstelle von fertigen Leiterbahnen kommen dementsprechend Draht- oder Lötzinnbrücken zur Anwendung.

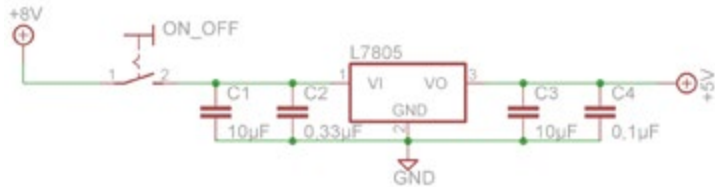
Allgemein wurde versucht, auf viele Feinheiten und Sicherheitsvorkehrungen beim Entwurf der Schaltung zu achten, auch wenn es sich um ein relativ berechenbares System handelt.

Beispielhaft wurden Schutzwiderstände zwischen Microcontroller-Ausgängen und anderen Bauteilen eingefügt. Ebenso wurden Freilaufdioden für die Servos eingesetzt, um die Spannungswandler und die vorgelagerte Schaltung vor der Induktionsspannung der Servos bei Abschaltung des Systems zu schützen.

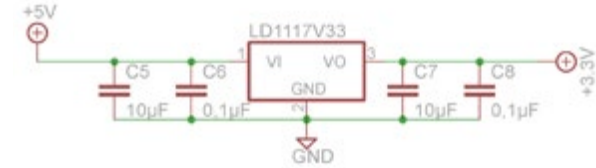
Für die Konstruktion des konkreten Systems wurden rechts aufgeführte Bauteile verwendet.

Anzahl	Beschreibung
1	Spannungswandler L 7805
2	Spannungswandler L 78S05
1	Spannungswandler LD1117V33
1	Mikrocontroller ATMEGA 328P-PU
1	Pegelwandler 74HC4050
2	P-Kanal MOSFET IRF 4905
1	IC-Sockel 16-polig
1	IC-Sockel 28-polig
2	Diode 1N4002 oder ähnlich
1	Quarz 16 MHz
3	NPN-Transistor BC547, BD237, oder ähnlich
8	Kondensator 10 µF
3	Kondensator 0,33 µF
5	Kondensator 0,1 µF
2	Kondensator 22 pF
8	Widerstand 10 kΩ
2	Widerstand 1 kΩ
2	Kippschalter
1	Joystick-Shield <sup>[11]</sup>
1	Display-Shield <sup>[12]</sup>
Diverse	Kabel, Stiftleiste, Buchsenleiste, Schrumpfschlauch, Gehäuse, Lüsterklemme, Cinch-Buchsen, Cinch-Kabel, USB-Kabel, Reihenklemme

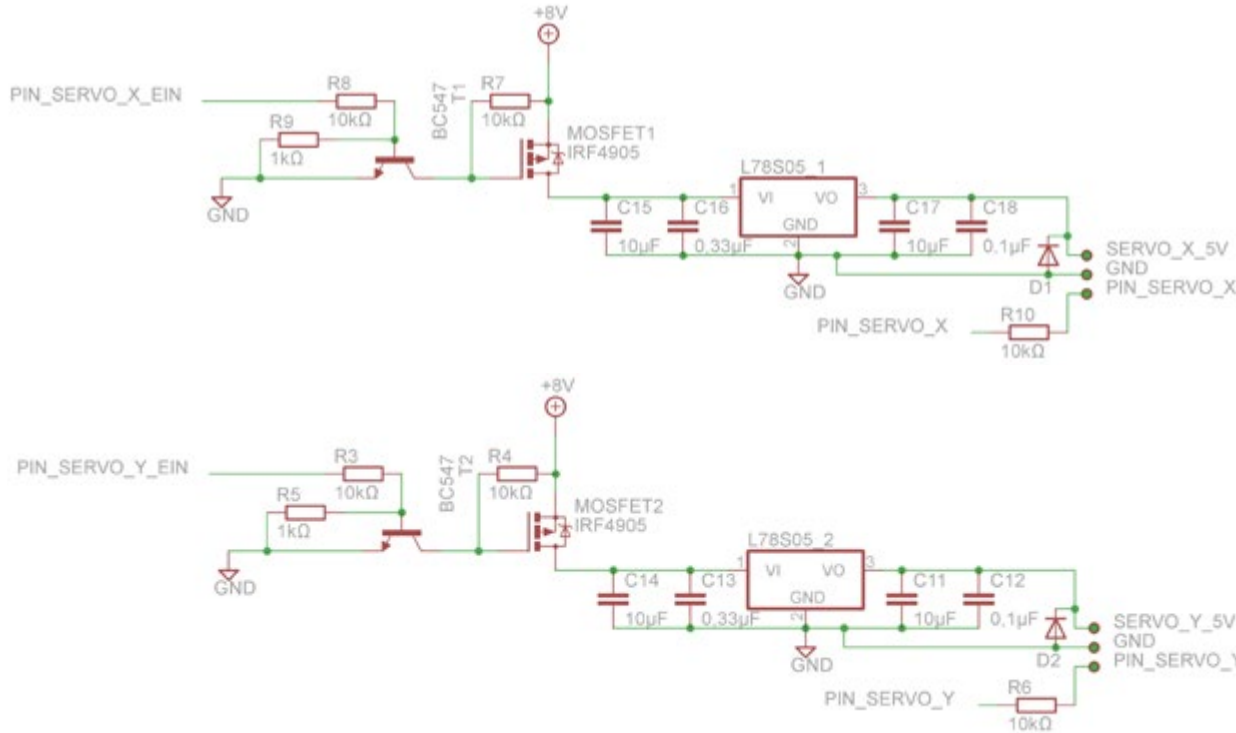
5 Volt Spannungsversorgung für Arduino und Auslösung



3,3 Volt Spannungsversorgung für Display



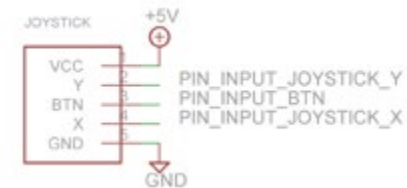
5 Volt Spannungsversorgungen (digital abschaltbar) für Servos



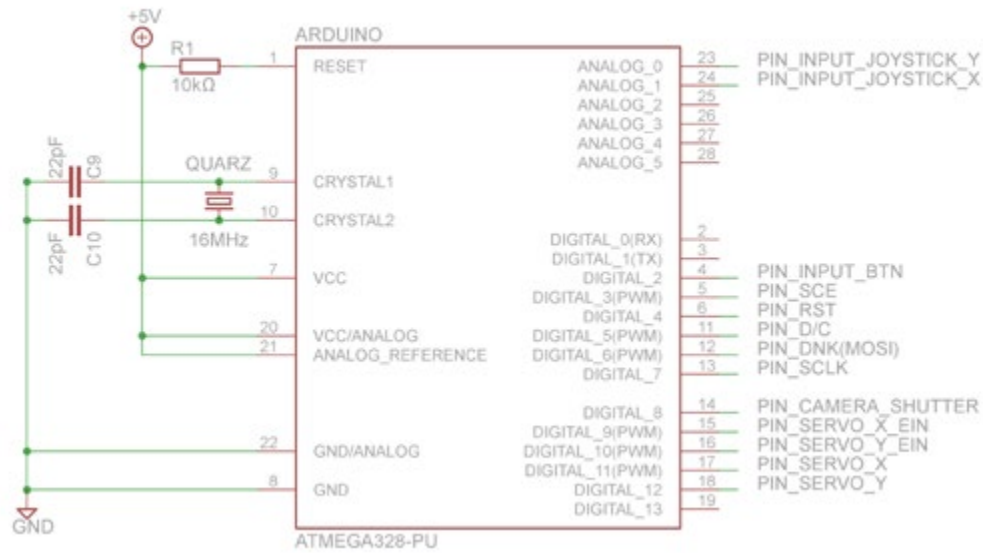
ICSP-Header (In-circuit serial programming) zur Programmierung des ATMEGA im eingebauten Zustand



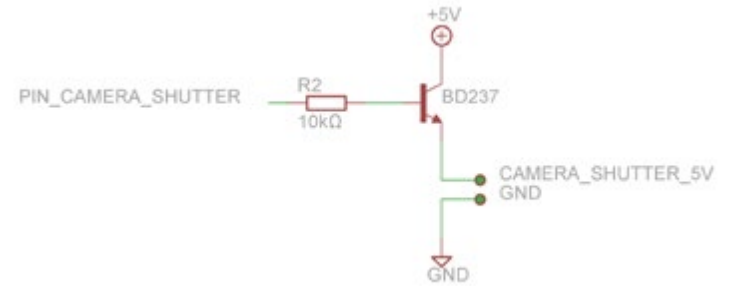
Joystick zur Steuerung des Aufnahmesystems



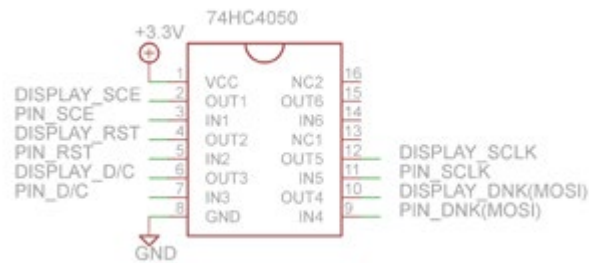
ATMEGA 328P „Arduino“



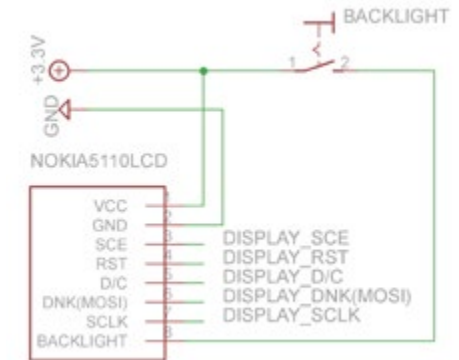
Schaltung zur Auslösung der Kamera



Pegelwandler (5 Volt auf 3,3 Volt) zum Schutz des Displays



Display zur Anzeige verschiedener Daten



## 2.2.4 Informationstechnisch

Die Programmierung erfolgt bei diesem Projekt stark prozedural. Dieser eher schlechte Stil ist aufgrund des sehr geradlinigen Ablaufs des gesamten Prozesses hier allerdings kein wirkliches Problem.

Um auf die Aufgaben der Software einzugehen, wird im Folgenden die Benutzersicht, das "Frontend" und die Hintergrundaktivität, das "Backend" unterschieden.

### 2.2.4.1 Frontend

Im Programmablauf werden zunächst die obere linke Ecke des Bildes und die untere rechte Ecke des Bildes vom Benutzer mithilfe des Joysticks angefahren und die Positionen durch Druck auf den Button des Joysticks bestätigt.

Als nächsten Schritt muss der Benutzer die Sensorgröße der Kamera und die eingestellte Brennweite eingeben. Auf Basis dieser Größen erfolgt eine Umrechnung der eingestellten Brennweite in das Kleinbild-Format und darauf basierend die Berechnung der Bildwinkel.

Durch Bewegungen des Joysticks kann in den konkreten Umsetzung zwischen einigen festgelegten Werten (Sensorgröße) gewählt oder eine freie Eingabe getätigt werden. Auch hier wird die Eingabe durch Druck auf den Joystick bestätigt.

Als vierten Schritt muss der Nutzer noch eine "Auslösezeit" einstellen. Dieser Wert legt fest, wie lange das System für das Auslösen der Kamera pausiert wird. Dieser Wert sollte in der Praxis etwas größer als die Belichtungszeit gewählt werden, um neben der reinen Belichtungszeit noch die Auslöseverzögerung und die Speicherung des Bildes zu berücksichtigen.

Mit diesen Schritten sind alle erforderlichen Daten erfasst worden. Im Folgenden informiert das System den Benutzer noch über die Anzahl der Bilder, die aufgenommen werden. Nach Bestätigung dieser Informationen beginnt das System seine Arbeit.

Ohne weiteres Zutun des Benutzers fährt das System nun reihenweise die errechneten Positionen an, nimmt ein Foto auf und fährt zur nächsten Position. Beginnend von der oberen linken Ecke wird so bis zur unteren rechten Ecke vorgegangen.

Nach erfolgreicher Abarbeitung fährt sich das System in seine Ausgangsposition zurück und kann abgeschaltet werden.

### 2.2.4.2 Backend

Direkt nach dem Start des Systems werden die benötigten Ein- Ausgänge des Arduino als solche definiert, um sie im weiteren Verlauf nutzen zu können. Darüber hinaus werden, wie üblich, Variablen deklariert und initialisiert.

Die Software erzeugt nach der Initialisierung des Displays jetzt die erste Ausgabe und informiert über den Start des Systems.

Die Spannungswandler für die Servomotoren werden eingeschaltet, die in Sekundenbruchteilen dann Spannung für die Servos bereit stellen. Jene werden dann in die Ausgangsposition des Systems gefahren und stehen damit für alles weitere bereit.

Dank der in Arduino bereits integrierten Bibliothek für Servomotoren gestaltet sich die Steuerung derselbigen sehr einfach:

---

```
// Neuen Servo erzeugen
Servo Servo_X;

// Servo an einen Pin anbinden
int PIN_Servo_X = 11;
Servo_X.attach(PIN_Servo_X);

// Servo ansteuern
Servo_X.write(90);
```

---

Im weiteren Verlauf muss nur noch über die write(int)-Funktion ein neuer Winkel für den Servo übergeben werden. Dieser wird dann automatisch und sofort angefahren.

Der Nutzer stellt nun obere linke und untere rechte Ecke des Panoramabildes ein. Dabei wertet

der Arduino die Position des Joysticks aus und verfährt das System entsprechend der Stellung.

Ein Druck auf den Button speichert jeweils die Position, genauer den Winkel des Servos auf beiden Achsen.

Auf Basis dieser Daten kann dann ein virtuelles Rechteck aufgespannt werden, welches sowohl in X- als auch in Y-Richtung einen bestimmten Winkel überspannt.

Nach Auswahl des Sensors und Eingabe der Brennweite kann die Software den Bildwinkel der Kamera bestimmen und daraus ableiten, wieviele Aufnahmen gemacht werden müssen, um mit entsprechender Überlappung, wie bereits erwähnt werden 50% gewählt, das aufgespannte Rechteck komplett abzudecken.

Dem Nutzer obliegt es nun noch, die Auslösezeit festzulegen. Die eingestellte Zeit lang wartet das Aufnahmesystem nach Auslösung der Kamera, bevor der Programmablauf fortgesetzt wird. So wird die Belichtungszeit der Kamera nicht durch weitere Aktionen gestört.

Ist auch hier eine Auswahl getroffen worden, zeigt das System noch die errechneten Werte für die Anzahl der Bilder auf beiden Achsen an.

Der Nutzer muss diese Informationen jeweils noch bestätigen, eine letzte Anzeige weist darauf hin, dass

nach einem erneuten Buttondruck das System seine Arbeit beginnt.

Der Autor empfiehlt ein Studium des gut dokumentierten Quelltextes, falls weitergehendes Interesse an den Geschehnissen auf der Backend-Seite besteht. Darüber hinaus ist die Arduino-Sprache auf für Ungeübte relativ gut lesbar und verständlich, so dass hier kein Grund zur Sorge besteht.

Die Software fährt die Kamera dann in die obere linke Ecke und nimmt das erste Foto auf. Abhängig von der gewählten Methode zur Abarbeitung der Bilder wird dann das Aufnahmesystem um einen entsprechenden Winkel gedreht oder geneigt und eine weitere Aufnahme erstellt.

Die Steuerung der Servomotoren erfolgt hierbei über pulswertenmodulierte Signale (kurz PWM), die der ATMEGA-Mikrocontroller erzeugt. Abhängig von der Pulsweite des jeweiligen Signals wird eine entsprechende Position angefahren. Wie bereits früher erwähnt, übernimmt die Servobibliothek des Arduinos die Erzeugung dieses PWM-Signals, so dass auf Programmierseite sehr komfortable mit den bereit gestellten Funktionen gearbeitet werden kann.

Die Auslösung der Kamera erfolgt über das Verbinden von 5 Volt und Masseleitung des USB-Anschlusses. Abhängig von der Einstellung

der Kamera führt dies dazu, dass die Kamera dann sofort, beziehungsweise gegebenenfalls nach erfolgter Fokussierung, ausgelöst wird.

Dieser Prozess wiederholt sich, bis alle Bilder aufgenommen wurden. Am Ende kehrt die Kamera in eine Ausgangsposition zurück und das System kann abgeschaltet oder für eine weitere Aufnahme neu eingestellt werden.

## 3. Weiterverarbeitung der Daten

### 3.1 Grundlegendes

Bereits in der Einleitung dieser Arbeit wurde kurz die Erzeugung üblicher Panorama-/Gigapixelbilder angesprochen.

Dieser Prozess ist im Grunde genommen immer gleich und besteht aus einer übersichtlichen Anzahl an Schritten:

1. Eindeutige Merkmale in den Bildern finden
2. Bilder miteinander vergleichen
3. Bilder mit gleichen Merkmalen entsprechend zueinander ausrichten, bis alle Bilder „verteilt“ sind
4. Bilder miteinander verrechnen und so das Gigapixelbild erzeugen

Ein wichtiger Faktor für den vierten Schritt bei der Erstellung von Gigapixelbildern ist die Projektionsart des Bildmaterials. Sie bestimmt, wie die Einzelbilder von ihrer kugelförmigen Ursprungsanordnung auf ein zweidimensionales Bild übertragen werden.

Gängige Software zur Erstellung von mehrreihigen Panoramen, im Folgenden auch als „Stitchingsoftware“ bezeichnet, bieten oft eine Vielzahl an verschiedenen Projektionsarten an. Abhängig vom

Einsatzzweck und dem Wunsch nach einer mehr oder minder kreativen Bildsprache, was bei Gigapixelbildern eher nicht von Interesse ist, existiert für all diese Projektionsarten durchaus eine Daseinsberechtigung und eine gewisse Bedeutung. Für die Erstellung von Gigapixelbildern ist deshalb aber vor allem eine Projektionsart von gesteigertem Interesse.

Bei der Flächenprojektion werden die Bilder nebeneinander „flach“ angeordnet, was dem Bildeindruck einer Weitwinkelaufnahme entspricht.

Diese Projektionsart hat den Vorteil, dass gerade Linien auch weiterhin gerade abgebildet werden, da hier keine wie auch immer geartete Krümmung der Bilder erzeugt wird.

Mit dieser Projektionsart gehen aber auch die Nebenwirkungen einher, die von Weitwinkelaufnahmen bekannt sind: Randnahe Objekte oder Linien werden verbogen abgebildet.

Diese Probleme treten vor allem dann auf, wenn große Winkel, als Richtwert ist ein Winkel von 100 Grad oft genutzt, durch das Bildmaterial überspannt werden.

Die Überspannung solcher großer Winkel sollte bei „klassischen“ Gigapixelbildern theoretisch nicht vorkommen, passiert in der Praxis jedoch relativ schnell. Entscheidend für den überspannten Winkel

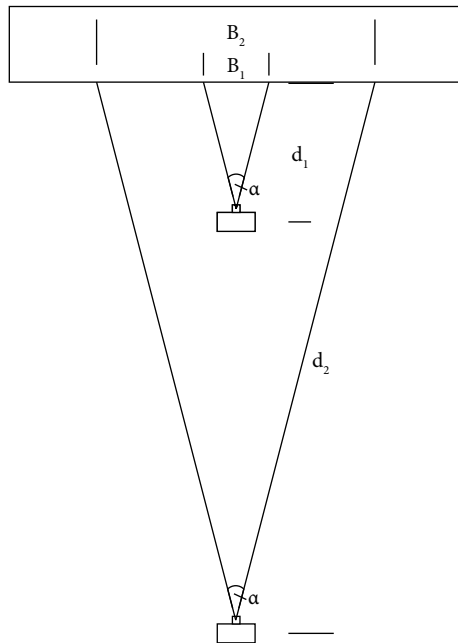
ist neben der fotografierten Szenerie natürlich auch die Entfernung von der Szenerie sowie die Brennweiteneinstellung bei der Aufnahme.

Nimmt man die zu fotografierende Szenerie als fixe Größe wird deutlich, dass mit zunehmender Entfernung zur selbigen der überspannte Winkel geringer wird. Dies kommt der Abbildung, genauer einer verzerrungsfreien Projektion, natürlich entgegen.

Um die Szenerie in der gleichen Größe zu erfassen, ist dann allerdings eine größere Brennweite nötig. Dies wiederum ist ein limitierter Faktor in der konkreten Umsetzung dieser Arbeit. Bei Aufnahmen mit diesem Gigapixelsystem muss also ein Kompromiss zwischen dem Abstand zur Szenerie, der Abbildungsgröße derselbigen und der Verzerrung des finalen Ergebnisses eingegangen werden.

Zugegebenermaßen lassen sich die Verzerrungen mittels elektronischer Bildbearbeitung korrigieren, dafür ist allerdings neben entsprechender Software eine gute Kenntnis derselbigen notwendig.

Weitere Informationen über die Projektionsarten in Hugin finden sich zum Beispiel auf [panoclub.de](http://panoclub.de)<sup>[13]</sup>. Allgemeinere Informationen über die Projektionsarten lassen sich bei einer Suche im Internet aber auch leicht finden<sup>[14]</sup>.



Schematische Darstellung des angesprochenen Sachverhaltes.  
 Abhängig von der Entfernung zum Objekt ( $d_1, d_2$ ) verändert sich die Breite des abgebildeten Bereiches ( $B_1, B_2$ ).  
 Voraussetzung dafür ist eine gleichbleibende Brennweite und damit ein gleichbleibender Bildwinkel.  
 Formelmäßig gilt für diesen Zusammenhang:

$$B = 2 \cdot \sqrt{d^2 + x^2}$$

mit  $x = \frac{d}{\cos(\frac{\alpha}{2})}$

## 3.2 Software/Algorithmen

Vor der Projektion des fertigen Bildmaterials stehen noch drei weitere Schritte, die nicht weniger wichtig für die Erstellung eines Gigapixelbildes sind.

Für die Verarbeitung des Bildmaterials wird in dieser Arbeit die kostenfreie Software „Hugin“ genutzt. Sie dient vornehmlich als Oberfläche für verschiedene kommandozeilenbasierte Softwarelösungen, wie die Kontrollpunktgeneratoren oder die Software zur Überblendung und Verrechnung einzelner Bilder. Jeder beschriebene Schritt ist dementsprechend theoretisch auch „von Hand“ und mit anderen Programmen durchführbar.

### 3.2.1. Merkmalsfindung

Um in den weiteren Schritten die Bilder miteinander zu vergleichen, auszurichten und miteinander zu verrechnen, ist es nötig, dass es eindeutige Punkte gibt, mittels derer sich ein Bild charakterisieren lässt. Hugin nennt diese Punkte „Kontrollpunkte“, englisch Control Points, kurz „cp“.  
 Die Anforderungen an diese Kontrollpunkte leiten sich aus ihrem Einsatzzweck ab. Sie sollen ein Bild, genauer dessen Merkmale, eindeutig definieren. Dabei müssen sie einen gewissen Grad an Einmaligkeit besitzen und robust gegen Verzerrungen, zum Beispiel durch Objektivfehler, sein.

Ein Algorithmus der diese Aufgabe besonders gut erfüllt ist „SIFT“. SIFT steht für „Scale-invariant feature transform“, also „skaleninvariante Merkmalstransformation“.

Dieser Algorithmus arbeitet so, dass zunächst „Oktaven“ der Bilder erzeugt werden. Dabei werden, ausgehend vom Originalbild, immer kleiner werdende, in der Ursprungsimplementation des Algorithmus wurde der Faktor 2 genutzt, Bilder erzeugt. Innerhalb einer Oktave werden die Bilder mit verschiedenen starken Gaußfiltern geglättet, um sie von Bildrauschen zu bereinigen.

Aus benachbarten, geglätteten Bildern einer Oktave werden dann Differenzbilder erzeugt, in denen nur noch markante Bildteile, potentielle Kontrollpunkte, enthalten sind.

In den Bildern wird jetzt nach Minima und Maxima, also Stellen großer Unterschiede im Histogramm, gesucht. Die gefundenen Stellen werden mit den Differenzbildern der anderen Oktaven verglichen. Lassen sich dort die gleichen Stellen finden, handelt es sich für den Algorithmus um einen robusten Punkt, der auch Bildtransformationen stand hält.  
 Ein interessantes Feature des SIFT-Algorithmus ist, dass eine Erkennung von horizontalen und vertikalen, sowie auch diagonalen Linien integriert ist. Basierend auf den Grauwerten und Histogrammen der benachbarten Pixel versucht der Algorithmus zu ermitteln, ob ein Schlüsselpunkt Teil eines Vektors ist und wie sich dieser Vektor fortsetzt.

Weitere Infos über den SIFT-Algorithmus finden sich bei der University of British Columbia<sup>[15]</sup> und in einer Studienarbeit der Uni Koblenz<sup>[16]</sup>.

Der SIFT-Algorithmus wurde 1999 von David Lowe entwickelt und ist patentrechtlich von der Universität von British Columbia (USA) geschützt. Aufgrund dieser Tatsache ist er nicht mehr standardmäßig in Hugin integriert, kann aber leicht nachinstalliert werden.

Als beliebteste Integration ist dafür „autopano-sift“ zu nennen, das von Sebastian Nowozin im Rahmen seiner Zeit an der TU Berlin entwickelt wurde. Auf der Projektseite<sup>[17]</sup> gibt es weitere Informationen über dieses Paket.

Die ursprüngliche Version, die in C# geschrieben wurde, ist mittlerweile durch eine Portierung auf reines C im praktischen Einsatz verdrängt worden. Diese Portierung findet sich als „autopano-sift-c“ im Internet, und auch in Hugin, wieder.

Hugin bringt darüber hinaus seit einiger Zeit den Kontrollpunktdetektor „CPFind“ mit, der keinen patent- oder lizenzrechtlichen Ein- und Beschränkungen unterliegt.

Die Entwicklungen daran laufen seit Anfang 2009<sup>[18]</sup>, obgleich es zum Zeitpunkt der Fertigstellung dieser Arbeit (Januar 2013) wenig Informatio-

nen über den aktuellen Entwicklungsstand und die Zukunft des Verfahrens gibt.

Basis für CPFind ist der „Pan-o-matic“-Kontrollpunktgenerator<sup>[19]</sup>, der auf dem „SURF“-Algorithmus basiert<sup>[20]</sup>.

„SURF“ steht für „Speeded Up Robust Features“, also „Beschleunigte, robuste Merkmale“ und wurde an der Eidgenössischen Technischen Hochschule Zürich von Herbert Bay, Tinne Tuytelaars und Luc Van Gool entwickelt.

Dieser Algorithmus arbeitet ähnlich wie SIFT, verwendet statt Gaußfiltern jedoch Mittelwertfilter. Damit soll er schneller und in der Laufzeit berechenbarer sein als der SIFT-Algorithmus.

Nichtsdestotrotz sind auch die von ihm gefundenen Kontrollpunkte robust gegen Transformationen und erfüllen damit die Ansprüche an Kontrollpunkte für die Erstellung von Gigapixel- oder Panoramabildern.

Die Original-Implementierung des SURF-Algorithmus ist Closed Source, es steht jedoch eine größere Anzahl an Open Source-Implementierungen zur Verfügung, beispielhaft in C++ in der bereits erwähnten Pan-o-matic-Software.

Allen Kontrollpunktgeneratoren ist gemein, dass sie als Ergebnis einen Satz an Kontrollpunkten pro Bild

liefern, der zusammen mit diesem eine Basis für die weitergehenden Arbeiten liefert.



Anzeige erkannter Kontrollpunkte in Hugin.

Auffällig: Starke Verzerrung des Bildmaterials durch ungünstige Aufnahmeposition von weit unten.

### 3.2.2 Bildvergleich

Um Bilder auf Basis der erstellten Kontrollpunkte miteinander zu vergleichen, nutzt Hugin den „RANSAC“-Algorithmus. RANSAC steht für „RANDOM Sample and Consensus“ also „zufällige Proben und Übereinstimmung“ und wurde bereits 1981 von Martin A. Fischler und Robert C. Bolles entwickelt.



Dieser Algorithmus ist sehr beliebt im Bereich der Bildverarbeitung und findet zum Beispiel Verwendung bei autonom fahrenden Fahrzeugen. Diese erzeugen unter Zuhilfenahme des RANSAC-Algorithmus Bilder zur Navigation, genauer zur Bestimmung ihrer eigenen Position auf der Straße.

Wie der Name bereits andeutet, handelt es sich um ein Verfahren, welches auf Basis von stochastischen Eingangswerten zu seinen endgültigen Werten gelangt.

Basis dafür ist dementsprechend eine Anzahl an Merkmalen die größer ist, als die letztendlich benötigte Anzahl an Merkmalen.

Aus der Summe der Ursprungsmerkmale werden zufällige Merkmale als Eingangsdaten für den Algorithmus genutzt und die Übereinstimmung mit den Zielwerten geprüft.

Iterativ versucht der Algorithmus mit jedem Durchlauf unwahrscheinliche Merkmale, also stark vom Rest abweichende Werte, zu erkennen und auszusortieren um so immer feiner oder genauer werdende Ergebnisse zu erhalten.

Die Iteration stoppt nach einer definierten Anzahl an Durchläufen oder nachdem ein Schwellwert für die Abweichung unterschritten wird.

Im konkreten Anwendungsfall werden die Kontrollpunkte der einzelnen Bilder als Eingangswerte für den RANSAC-Algorithmus genutzt, das Bild dessen Nachbarn gesucht werden sollen dient als Zielwertvorgabe.

Sofern es sich nicht um Kontrollpunkte aus Bildern handelt, die nebeneinander liegen, werden die Ergebnisse des Algorithmus deutlich über dem festgelegten Schwellwert liegen, womit klar wird, dass es sich nicht um benachbarte Bilder handeln kann. Sind die Bilder benachbart, ergeben sich während Nutzung des Algorithmus gemäß dessen Funktionsweise immer bessere Werte, womit von einem Bildpaar ausgegangen werden kann.

Hier zeigt sich, wieso es von Bedeutung ist, dass die im ersten Schritt erzeugten Kontrollpunkte robust gegen, und damit unabhängig von, Transformationen und Verzerrungen sind. Wären Sie dies nicht, würde es nicht möglich sein, mittels dieses einfachen Verfahrens die Positionierung der Bilder zueinander zu bestimmen.

Weitere Informationen über den RANSAC-Algorithmus lassen sich leicht im Internet finden. Beispielfhaft seien „RANSAC for Dummies“ von Marco Zuliani<sup>[21]</sup> und Vorlesungsunterlagen von Dr. Dmitrij Schlesinger (TU Dresden) zum Thema Bildverarbeitung<sup>[22]</sup> genannt.

### 3.2.3 Bildausrichtung/-verrechnung

Nachdem die Ausrichtung der Bilder durch den Kontrollpunktgenerator in Verbindung mit einem Algorithmus wie RANSAC ermittelt wurde, kann das Panorama/Gigapixelbild aus den Einzelbildern zusammengerechnet werden.

Hugin nutzt dafür standardmäßig die Software „enblend“, die unter wahrnehmungsbasierten Gesichtspunkten die Einzelbilder verrechnet.

Bildteile mit hoher Frequenz, also häufiger Änderung der Tonwerte, üblich für kontrastreiche Bildteile, werden dabei auf sehr kleinem Raum miteinander verrechnet. Bilder mit geringer Frequenz, also wenig Kontrasten, werden mit entsprechend großen Bereichen miteinander überblendet.

Auf diese Weise soll ein möglichst natürlicher Bildeindruck erreicht werden. Harte Übergänge in einem Himmel oder einer gleichfarbigen Fläche würden vom Betrachter sehr schnell wahrgenommen werden, deshalb empfiehlt sich hier ein weicher Übergang. Diese weichen Übergänge würden in einem kontrastreichen Bereich wiederum als störend empfunden werden. Gegenstände oder Kanten wären entsprechend verwischt und unscharf.

Im Vorfeld dieser Überblendung bildet enblend eine so genannte „Übergangslinie“, die festlegt, welches Bild wo das jeweils andere überlagert.

Hierbei versucht die Software die Linie möglichst weit entfernt von potentiell wichtigen, hochfrequenten Bildteilen zu ziehen.

Um weiterarbeiten zu können, konvertiert enblend das Eingangsbildmaterial in TIFF-Bilder. TIFF-Bilder zeichnen sich dadurch aus, dass sie Alpha-Kanäle enthalten können. Mittels dieser lässt sich eine Teiltransparenz von Bildteilen erreichen, was für die Überblendungen nötig ist.

Im folgenden Schritt extrahiert die Software mittels eines Hochpass-Filters die hochfrequentesten Bereiche aus den jeweiligen Bildern und erzeugt daraus ein einzelnes Bild. Die verbliebenen Bildbereiche werden um den Faktor 2 herunterskaliert und dienen wieder als Eingangsmaterial für den gleichen Ablauf. Dieser Vorgang erzeugt eine so genannte „Gaußpyramide“, die zur Weiterverarbeitung dient.

Wenn alle Bilder mittels der Gaußpyramide entsprechend zerlegt worden sind, beginnt die Software damit die nebeneinander liegenden Gaußpyramidenbilder miteinander zu verrechnen. Die kleinsten Bilder enthalten dabei wenig Kontrastinformationen und werden, wie weiter oben erwähnt, mit einer großen Überblendung verrechnet. Aufsteigend werden so aus den Bildern jeder Ebene so genannte „Laplacepyramiden“ erzeugt, wobei der Überblendungsbereich immer kleiner gehalten wird.

Um die Überblendung abzuschließen, werden die gebildeten Laplacepyramiden, die jetzt die kombinierten Kontrastinformationen enthalten, mit den entsprechenden Gaußpyramiden verrechnet, um die ursprünglichen Bildinformationen wiederherzustellen.

Dieser Einsatz der Gauß-Laplace-Pyramide wird als „Mosaicing“ bezeichnet und wird zu verschiedensten Zwecken der digitalen Bildverarbeitung genutzt, wobei die Verrechnung mehrerer Bilder miteinander der häufigste Anwendungsfall ist.

Eine englische Beschreibung des Vorgangs, sowie einige Bilder davon finden sich auf der Projektseite von enblend<sup>[23]</sup>.

Eine deutlich umfangreichere Abhandlung der Thematik wurde von Heung-Yeung Shum und Richard Szeliski mit dem Titel „Panoramic Image Mosaics“ erstellt<sup>[24]</sup>.

Bereits bei dieser recht oberflächlich gehaltenen Betrachtung zeigt sich, welche Komplexität nötig ist, um aus den erzeugten digitalen Fotos ein Panorama, respektive ein Gigapixelbild, zu erzeugen.

Der geneigte Leser findet bereits nach kurzer Suche ausreichend Material um sich intensivst mit der Weiterverarbeitung digitaler Bilder auseinander zu setzen.

Als interessanten Einstieg empfiehlt der Autor dieser Arbeit das Wiki zu den „PanoTools“<sup>[25]</sup>.

Bei den PanoTools handelt es sich um eine Software-Sammlung die zunächst vom Physik- und Mathematik-Professor Helmut Dersch (Hochschule Furtwangen) geschrieben wurde und verschiedenste Aufgaben rund um die Verarbeitung von Bildern erfüllt. Bedingt durch rechtliche Streitigkeiten stellte Dersch die Arbeit an den Programmen ein, diese wurde jedoch von Dritten weitergeführt und ausgebaut.

Das PanoTools-Wiki stellt eine Vielzahl an Informationen zu verschiedensten Programmen zur Verfügung. Neben den hier angesprochen Programmen und Algorithmen finden sich dort auch viele Informationen zu Hugin und der Verarbeitung von Bildern mit dieser spezifischen Software.

Während einige Informationen einer Überarbeitung bedürften, so ist die Grundgesamtheit dennoch recht aktuell und bietet einen guten Überblick über die gesamte Thematik.

## 3.3 Erstellen von Panoramen mit Hugin

### 3.3.1 Einrichten von Hugin

Diese Anleitung geht davon aus, dass Hugin bereits von der offiziellen Seite<sup>[26]</sup> heruntergeladen und installiert, aber noch nicht weiter eingerichtet wurde.

Benutzer des Betriebssystems Windows 7 sollten das Programm mit der Kontextmenüoption „Als Administrator ausführen“ starten, andernfalls kann es zu Problemen bei der Erstellung von Panoramen kommen.

Nach dem ersten Start von Hugin sind einige Einstellungen zu treffen. Das Einstellungsmenü findet sich dafür unter „Datei“ und „Einstellungen“.

Im ersten Reiter „Allgemein“ kann der Wert für die Zwischenspeichergröße heraufgesetzt werden (standardmäßig 256 MB). Bilder werden bei der Bearbeitung dann länger im RAM behalten, was zu einem Plus in der Geschwindigkeit führt.

Im Reiter „Assistent“ bietet es sich an, den Punkt „Bilder nach dem Laden automatisch auszurichten“ zu aktivieren. Dies automatisiert die Erstellung von Panoramen/Gigapixelbildern noch weiter.

Die Option zur Aktivierung von „Celeste“ muss nicht zwangsläufig aktiviert werden, hier sollte der

Benutzer mit konkretem Bildmaterial testen, ob die Ergebnisse sich durch Einsatz dieser Technik verbessern oder nicht. Insgesamt ist zu sagen, dass die Berechnung länger dauert, sofern Celeste aktiviert ist, in den Tests des Autors aber keine Verbesserung der Bildqualität eintrat.

„Celeste“ ist ein Modul im Kontrollpunktgenerator „CPFInd“, welches dafür sorgt, dass Wolken oder wolkenähnliche Bereiche nicht in die Ermittlung von Kontrollpunkten einbezogen werden, da sie keine sicheren Daten liefern.

Im Reiter „Kontrollpunkt-Detektoren“ kann nun der Standard-Kontrollpunktgenerator ausgewählt werden. Darüber hinaus können weitere Generatoren hinzugefügt werden, die vorhandenen können auch bearbeitet werden.

Hier findet sich auch bereits ein Eintrag für den vorher angesprochenen „Autopano-SIFT-C“-Generator, der nach Belieben ebenfalls eingerichtet werden kann.

Nach Markierung eines Generators öffnet ein Klick auf „Bearbeiten“ ein Fenster, in dem der Reiter „Kontrollpunkt-Detektor“ von Interesse ist.

Als Programmname ist hier standardmäßig nur etwas wie „cpfnd.exe“ eingetragen. Es hat sich gezeigt, dass hier mit einem absoluten Pfad gearbeitet werden sollte.

Ein Klick auf „Auswählen“, rechts neben dem Dateinamen, öffnet ein Fenster, mit dessen Hilfe man die Datei suchen kann.

Die meisten Generatoren liegen im „bin“-Verzeichnis des Hugin-Programmordners.

Ist das Programm korrekt lokalisiert worden, kann das Einstellungsfenster für den Kontrollpunktgenerator mit einem Klick auf „OK“ geschlossen werden.

Im Hugin-Einstellungsfenster kann auf dem Reiter „Zusammenfügen“ nun noch das Ausgabeformat umgestellt werden.

Wenngleich TIFF-Dateien höchste Qualität liefern, so sind sie doch auch sehr speicherintensiv.

Die Auswahl von „JPEG“ mit einer Qualität von 90 liefert ebenfalls recht gute Ergebnisse bei deutlich geringerer Dateigröße.

Für den normalen Gebrauch ist die Einrichtung des Programms jetzt mit einem Klick auf „Übernehmen“ und dann auf „OK“ abgeschlossen.

Es finden sich leicht weitere Anleitungen<sup>[27][28]</sup> für die „optimale“ Einrichtung von Hugin. Es obliegt dem Nutzer diese Einstellungen zu testen und den Nutzen abzuschätzen. Es ist allerdings darauf zu achten, dass sich viele dieser Anleitungen auf ältere Programmversionen beziehen.

### 3.3.2 Erstellen eines Panoramas

Nach der grundlegenden Einrichtung des Programms können nun Panoramen/Gigapixelbilder erzeugt werden. Dazu kann in Hugin fortan komfortabel der Assistent genutzt werden.

Auf dem Reiter „Assistent“ sieht man die zunächst sehr eingeschränkt wirkenden Einstellmöglichkeiten, auf die später noch eingegangen wird.

Nach Auswahl der Bilder für das Panorama über den Button „1. Bilder laden“ und bestätigen der Auswahl sollte automatisch die Erzeugung von Kontrollpunkten und das Ausrichten der Bilder beginnen.

Sollte dies nicht erfolgen, sollte die entsprechende Einstellung nochmals geprüft und Hugin gegebenenfalls einmal neu gestartet werden.

Alternativ lässt sich dieser Vorgang auch durch einen Klick auf „Ausrichten“ starten. Infolgedessen öffnet sich ein Fenster, dessen Inhalt meist sehr schnell vorbeiscrollt. Dieses Fenster zeigt die gerade durchgeführten Schritte an.

Wer den Ablauf beobachtet wird erkennen, dass zunächst Kontrollpunkte erzeugt werden. Im Anschluss werden passende Bildpaare gesucht, danach findet eine Ausrichtung statt.

Im Anschluss versucht die Software die ermittelten Werte nochmals zu verbessern, ist auch dieser Prozess abgeschlossen beginnt die vorläufige Verrechnung der Bilder. Darüber hinaus werden die Bilder noch einer Optimierung unterzogen, um Verzerrungen zu entfernen.

Sobald Hugin diese Arbeit beendet hat, öffnet sich die „Schnelle Panoramavorschau“. Sie erlaubt einen ersten Blick auf das errechnete Endergebnis und gestattet weiteren Eingriff für die Weiterverarbeitung.

Der weiße Rahmen zeigt an, welchen Bereich Hugin als Bild exportieren würde. Die Software versucht undefinierte Bildteile, in der Vorschau schwarz angezeigt, nicht mit ins Bild zu nehmen, dementsprechend wird der Rahmen aufgezogen.

Über den Reiter „Beschnitt“ gelangt man in einen Modus, in dem man den Beschnittrahmen noch durch Anfassen der Kanten und dementsprechendes Ziehen verändern kann.

Interessante Informationen hält der Reiter „Layout“ bereit, er zeigt an, wie die Bilder miteinander durch Kontrollpunkte verbunden sind.

Im Reiter „Projektion“ kann nun die Projektionsart gewählt werden. Hugin trifft hier eine Vorauswahl, die aber nicht unbedingt die beste oder gewollte Darstellungsart sein kann. Durch das Dropdown-Menü kann man komfortabel und schnell die

Projektionsarten durchgehen und die gewünschte auswählen.

Ist dies erfolgt kann man im Reiter „Bewegen/Ziehen“ das Panorama noch verschieben. Damit lässt sich beispielsweise die Horizontlinie eines Panoramas neu definieren. Auch hier versucht Hugin aber bereits einen passenden Wert auszuwählen.

Sind alle Einstellungen getroffen, kann die Panoramavorschau geschlossen werden, Hugin übernimmt eventuell getroffene Änderungen dann automatisch.

Unter dem Button „2. Ausrichten“ wird auf der Assistenten-Seite nun angezeigt, wieviele Kontrollpunkte die Bilder verbinden und wie deren Qualität einzuschätzen ist.

Sofern diese nicht „sehr schlecht“ ist, startet ein Klick auf „3. Erstellen des Panoramas“ die Erzeugung des fertigen Bildes.

Dazu muss zunächst eine Projektdatei angelegt werden, diese enthält Informationen über die verwendeten Bilder, die Kontrollpunkte, Projektionsart und vieles mehr.

Danach muss ein Speicherort sowie der Dateiname des Ausgabebildes angegeben werden. Nachdem dort auf „Speichern“ geklickt wurde, startet die Erzeugung des Bildes, die durchaus einige Zeit in Anspruch nehmen kann.

Prinzipiell wäre damit der gesamte Prozess abgeschlossen, Hugin bietet aber dankenswerterweise viele Eingriffsmöglichkeiten in den Ablauf.

Im folgenden wird davon ausgegangen, dass die Bilder bereits durch den „Ausrichten“-Schritt gelaufen sind.

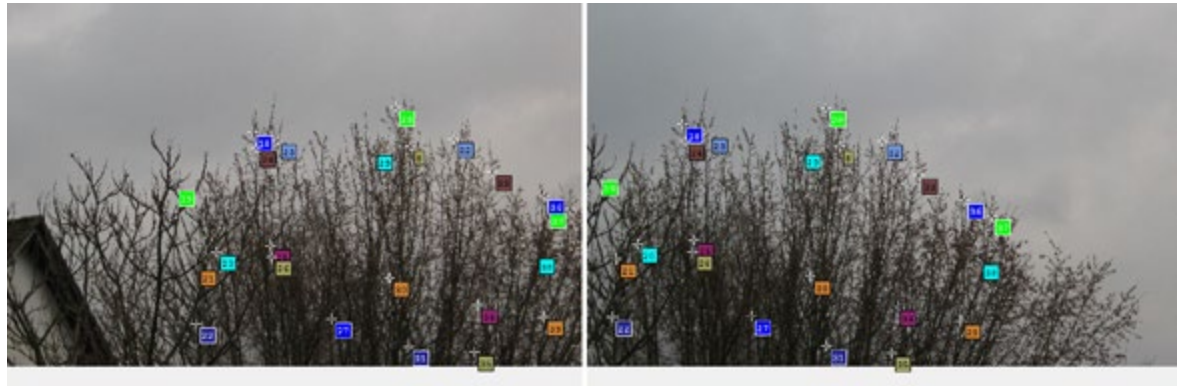
Auf dem Reiter „Bilder“ zeigt Hugin uns nun viele Informationen über die eingelesen, und ausgerichteten Bilder an. Interessant ist hierbei vor allem die Anzahl der Kontrollpunkte, die standardmäßig in der vorletzten Spalte angezeigt wird.

Wie zu erwarten, weisen Bilder mit wenigen Details eine geringere Anzahl an Kontrollpunkten auf, als Bilder mit vielen Details und hohen Kontrasten.

Sofern man mit den erstellten Kontrollpunkten nicht zufrieden ist, kann man im unteren linken Bereich bei „Abgleich von Merkmalen“ einen anderen Kontrollpunktgenerator auswählen und Kontrollpunkte und Ausrichtung der Bilder neu erzeugen lassen.

Ferner besteht die Möglichkeit ein „Ankerbild“ auszuwählen, dass von Hugin als Referenz zum Beispiel für die Belichtung genutzt wird. Die anderen Bilder werden dann diesem Bild angeglichen.

Der Reiter „Kamera und Objektiv“ hält bei Verwendung des konstruierten Kamerasystems keine



Anzeige von Kontrollpunkten in zwei benachbarten Bildern.

Die Kontrollpunkte dienen zur Ausrichtung und Verrechnung der Bilder.

relevanten Informationen bereit. Alle Werte sollten gleich oder nahezu gleich sein, da mit relativ festen Einstellungen fotografiert wurde.

Über die Reiter „Beschnitt“ und „Maske“ lässt sich von Hand in die Eingangsbilder eingreifen, um bestimmte Bildteile auszuschließen oder Bilder noch vor der Verrechnung zu beschneiden. Von dieser Eingriffsmöglichkeit wurde in den Tests abgesehen.

Der Reiter „Kontrollpunkte“ hingegen ist wieder von deutlich größerem Interesse. Er kann neben den Kontrollpunkten eines einzelnen Bildes auch die Kontrollpunkte beispielsweise im Nachbarbild anzeigen. So lässt sich sehr schön anschauen, auf welcher Basis die Bilder zueinander ausgerichtet

und verrechnet werden. Über dieses Fenster können weiterhin Kontrollpunkte von Hand angelegt werden. Dies mag für wenige Bilder praktikabel sein, wird bei größeren Bildmengen aber fast unmachbar. Hier empfiehlt sich der Einsatz der Kontrollpunkt-Generatoren.

Die Reiter „Optimieren“ und „Belichtung“ bieten ebenfalls wieder Eingriffsmöglichkeiten, um sehr fein und gezielt einzelne Bilder zu berücksichtigen und zum Beispiel die Ausrichtung der Bilder nach einzelnen Koordinaten stärker zu gewichten und ähnliches. Auch auf diese Option wurde bei den Tests verzichtet.

Der Reiter „Zusammenfügen“ bietet Optionen um die Ausgabe des Bildes zu beeinflussen. Neben der Projektionsart wird hier auch der errechnete Bildwinkel, die Größe des Panoramas und der gewählte Beschnitt angezeigt.

Darüber hinaus lassen sich Optionen für die Ausgabe als HDR und weiteres festlegen. In den Tests erzielte der Autor mit den voreingestellten Werten annehmbare Ergebnisse.

Nach Prüfung all dieser Werte kann mit einem Klick auf „3. Erstellen des Panoramas“ im Reiter „Assistent“ nun tatsächlich die Ausgabe gestartet werden.

Hugin ist ein unglaublich vielseitiges und mächtiges Werkzeug zur Erstellung von Panorama- und Gigapixelbildern. Aus diesem Grund kann nicht auf jede einzelne Option eingegangen werden.

Der Autor legt dem Leser nahe, eigene Erfahrungen mit der Software zu sammeln. Abhängig vom Einsatzzweck sind dann andere Einstellungen wichtig oder können das Ergebnis verbessern.



Testpanorama

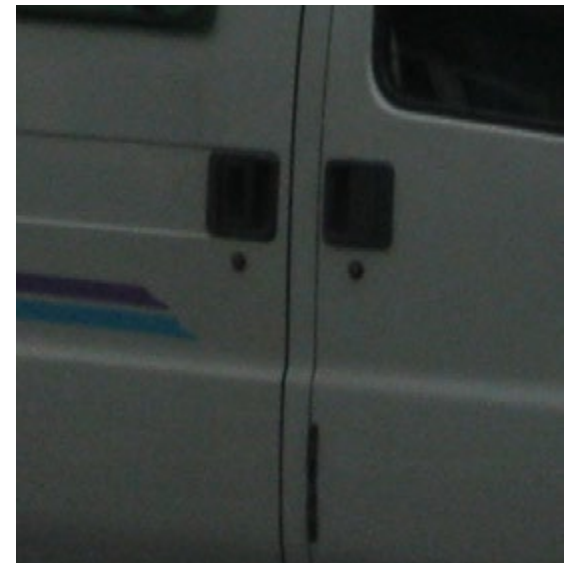
21069 Pixel Breite und 7855 Pixel Höhe, damit 157 Megapixel.

365 cm x 133 cm Druckbreite bei 150 dpi.

Projektionsart: Zylindrisch

Sichtbare Verzerrungen durch großen Aufnahmewinkel und Objektivfehler am oberen Rand des Panoramas.

Rechte Seite: 100% Ausschnitte aus dem Panorama



## 4. Ergebnisse

### 4.1 Probleme/Verbesserungsmöglichkeiten

Wie sich herausgestellt hat, stellt die Kamera das größte Problem oder den größten Schwachpunkt des Systems dar.

Die Auflösung ist ausreichend, allerdings ist die Linsenkonstruktion von relativ geringer Qualität. Dies zeigt sich durch deutliche Verzerrungen in den Randbereichen der Bilder, so dass die gewählte Überlappung der Bilder von 50% gut gewählt wurde.

Weiterhin stellt sich der geringe Brennweitenbereich der Kamera als stark limitierender Faktor dar.

Um wirklich beeindruckende Gigapixelbilder erzeugen zu können, müsste man mit deutlich größerer Brennweite fotografieren können. Dadurch ließen sich mehr Einzelbilder erzeugen und dann miteinander verrechnen.

Das mechanische System funktioniert recht gut, ist jedoch auf einen einzigen Kameratyp beschränkt.

Würde man hier die Flexibilität durch weitere Langlöcher oder ähnliches erhöhen, ließen sich auch andere Kameras nutzen, was sicherlich von Interesse wäre.

Darüber hinaus würden bei einer Neukonstruktion vermutlich keine Zahnräder mehr zur Kraftübertragung genutzt. Auch wenn deren Spiel das leichte Zittern der Servos ausgleicht, so verhindert dieses Spiel eine exakte und wiederholbare Anfahrung eines Punktes. Zahnriemen wären hier rückblickend vielleicht die bessere Wahl gewesen.

Im Weiteren wäre zu überlegen, die Konstruktion nicht mehr aus Aluminium zu fertigen. Das Aufnahmesystem ist schwerer als gedacht und benötigt damit entsprechend groß dimensionierte Motoren.

Mit der zunehmenden Verbreitung von 3D-Druckern wäre es denkbar, die Bauteile des Systems mit einem 3D-Drucker zu fertigen. So wären sehr leicht Anpassungen und die Nachproduktion von defekten Teilen möglich. Darüber hinaus würde die Konstruktion auch deutlich leichter ausfallen, was den verwendeten Motoren zugute käme.

Das elektronische System arbeitet ebenfalls recht zuverlässig, weist allerdings einen kleinen Makel auf.

Sofern die Servomotoren angeschlossen sind und der Akku nicht mehr voll geladen ist, kann es auftreten, dass das Display extrem träge reagiert oder erst nach vergleichsweise langer Zeit überhaupt ein Bild darstellt. Das Problem ist klar in der Stromversorgung des Displays zu suchen und sollte sich noch beheben lassen.

Ein weiterer Negativpunkt ist das Verhalten des Displays, wenn die Hintergrundbeleuchtung abgeschaltet wird. Geschieht dies, dunkelt das Display dramatisch ab, wird fast komplett schwarz, und ist damit sehr schlecht ablesbar.

Nach einiger Recherche im Internet zeigt sich, dass es sich hierbei eventuell um einen Defekt des Displays handelt, mit dem Nokia bereits beim Einsatz in seinen Telefonen zu kämpfen hatte.

Eventuell liegt hier aber auch ein Problem mit der Schaltung vor, dies wäre zu prüfen und gegebenenfalls zu korrigieren.

Die Software für das Aufnahmesystem läuft stabil und verhält sich wie erwartet.

Als Verbesserung wäre hier sicherlich eine Erweiterung des Funktionsumfangs denkbar und wünschenswert. Die Möglichkeit weitere Parameter einzustellen oder Mehrfachbelichtungen für HDR-Aufnahmen anzufertigen wären zum Beispiel denkbar.

Diesen Überlegungen im Wege steht der dann teilweise doch limitierte Zugriff auf die Kamerafunktionen mittels des CHDK.

Kann man etwas auf Automatisierung verzichten, dann wäre hier aber sicherlich noch einiges Potenzial für eine Verbesserung des Systems.



## 4.2 Fazit

Insgesamt ist zu sagen, dass das Projekt äußerst spannend war. Es verband einzelne Elemente des Bachelor-Studiums auf gute Art und Weise und ließ den Autor so das Gelernte rekapitulieren.

Der hohe Praxisbezug, der im Studium oftmals nicht in dieser Form gegeben ist, machte die Arbeit darüber hinaus zu etwas ganz besonderem.

Der Autor freut sich vor allem darauf, die Kamera bei besseren Wetterbedingungen testen zu können und erhofft sich dabei noch deutlich bessere Ergebnisse im Bildmaterial.

Darüber hinaus ist eine Weiterarbeit am bestehenden System geplant, um zumindest die elektronischen Fehler des Systems auszubessern. Eine Überarbeitung der Software ist ebenfalls denkbar, um das System flexibler und interessanter zu gestalten.

Ansonsten plant der Autor die Veröffentlichung der Arbeit im Internet, um anderen Interessierten eine Einstiegsmöglichkeit und Basis zu bieten.

## Verwendete Quellen

- 1 <http://chdk.wikia.com>
- 2 <http://forum.chdk-treff.de>
- 3 [http://www.canon.de/For\\_Home/Product\\_Finder/Cameras/Digital\\_Camera/PowerShot/PowerShot\\_A490/](http://www.canon.de/For_Home/Product_Finder/Cameras/Digital_Camera/PowerShot/PowerShot_A490/)
- 4 [http://www.fotomagazin.de/test\\_technik/technikwissen/detail.php?objectID=5146&class=26](http://www.fotomagazin.de/test_technik/technikwissen/detail.php?objectID=5146&class=26)
- 5 <http://www.henner.info/2mp.htm#Größe>
- 6 <http://timmermann.tv/technik/hyperfokaldistanz.php>
- 7 <http://forum.chdk-treff.de/viewtopic.php?f=13&t=486#p3337>  
(CHDK-Handbuch, Seite 59)
- 8 [www.pollin.de](http://www.pollin.de)
- 9 [www.arduino.cc](http://www.arduino.cc)
- 10 [www.eclipse.org](http://www.eclipse.org)
- 11 [www.komputer.de/zen/index.php?main\\_page=product\\_info?cPath=22&products\\_id=132](http://www.komputer.de/zen/index.php?main_page=product_info?cPath=22&products_id=132)
- 12 [www.komputer.de/zen/index.php?main\\_page=product\\_info?cPath=22&products\\_id=74](http://www.komputer.de/zen/index.php?main_page=product_info?cPath=22&products_id=74)
- 13 [http://www.panoclub.de/hugin\\_tut/spezial.html](http://www.panoclub.de/hugin_tut/spezial.html)
- 14 <http://www.360direct.de/grundlagen/projektionsarten.php>
- 15 <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
- 16 <http://www.uni-koblenz.de/~lb/publications/sift.pdf>
- 17 <http://www.nowozin.net/sebastian/tu-berlin-2006/autopano-sift/index.html>
- 18 <https://groups.google.com/d/msg/hugin-ptx/2cBVjGecrVI/qX98jmT4gB8J>
- 19 <http://aorlinsk2.free.fr/panomatic/>
- 20 <http://www.vision.ee.ethz.ch/~surf/eccv06.pdf>
- 21 <http://vision.ece.ucsb.edu/~zuliani/Research/RANSAC/docs/RANSAC4Dummies.pdf>
- 22 [http://www1.inf.tu-dresden.de/~ds24/lehre/bvme\\_ss\\_2012/bv\\_ransac.pdf](http://www1.inf.tu-dresden.de/~ds24/lehre/bvme_ss_2012/bv_ransac.pdf)
- 23 <http://enblend.sourceforge.net/details.htm>
- 24 <http://www.multires.caltech.edu/teaching/courses/3DP/papers/SchumSzeliski.pdf>
- 25 <http://wiki.panotools.org/>
- 26 <http://hugin.sourceforge.net/>
- 27 [http://hugin.sourceforge.net/docs/manual/Hugin\\_Preferences.html](http://hugin.sourceforge.net/docs/manual/Hugin_Preferences.html)
- 28 [http://www.panoclub.de/hugin\\_tut/index.html](http://www.panoclub.de/hugin_tut/index.html)

## Verwendete Software

Arduino IDE, Version 1.0.1  
Hugin, Version 2012.0.0.a6e4184ad538

## Inhalte der CD

- Projektdokumentation als PDF
- Quelltext für die Software
- Bauteilzeichnungen als EPS-Dateien
- Schaltplan als Eagle-Dokument
- Diverse Datenblätter für Spannungswandler
- Testpanorama samt Einzelbilder

