Frankfurt University of Applied Sciences

Entwicklung einer Anwendung zur Aufnahme hochauflösender Panoramen von astronomischen Objekten

Bachelorarbeit

Kevin Fath Matrikel-Nummer 1013012

ErstprüferHerr Prof. Dr. BaunZweitprüferHerr Müller-Bady

Inhaltsverzeichnis

Ał	obilc	lungsverzeichnis	IV
Sy	mbc	olverzeichnis	V
1	Eint 1.1 1.2 1.3	führung Problem	2 2 3 3
2	Gru 2.1 2.2 2.3	Astronomische Grundlagen	5 6 10 12 13
3	Star 3.1 3.2 3.3	nd der Technik EQMOD/EQMosaic Erklärung und Installation Vor- Und Nachteile	18 18 18 23
4	Ent [*] 4.1 4.2	wicklung und Vorstellung der Anwendung Anforderungsanalyse und Use-Case 4.1.1 Anforderungsanalyse 4.1.2 Use-Case Interessante Segmente der Entwicklung	25 26 26 27 28
	4.3	GUI-Erklärung	37

Inhaltsverzeichnis

	4.4	Vor- Und Nachteile	39
	4.5	Mögliche Erweiterungsgebiete	39
_			4.4
5	Aus	swertung	41
	5.1	Vergleich der Anwendung	41
	5.2	Endresultat der Bilder	42
	_		
6	Faz	it	43
Ti	torat	urvorzoichnic	15
ы	ieral		40

Abbildungsverzeichnis

1.1	Andromeda-Galaxie, aufgenommen vom Hubble-Weltraumteleskop	3
2.1	Horizontsystem	6
2.2	Äquatorsystem	8
2.3	Bahn der Sonne	9
2.4	Mond	13
2.5	Oberer Teil der Nachführung ohne Gegengewichte und Teleskop	14
2.6	Azimutschrauben	15
2.7	Altitudschrauben	16
2.8	Fertig montierte Nachführung	17
3.1	EQASCOM Toolbox	19
3.2	ASCOM Setup	20
3.3	Stellarium	21
3.4	EQMOSAIC	22
4.1	Use-Case für die Anwendung des Programms	27
4.2	Hauptfenster der Anwendung	28
4.3	Farbliche Hervorhebung der Unterschiedlichen Bereiche	38

Symbolverzeichnis

Allgemeine Symbole

Symbol	Bedeutung
0	das Grad °
1	die Bogenminute /
//	die Bogensekunde //
Α	der Azimut A
h	die Höhe <i>h</i>
Z	die Zenitdistanz z
δ	die Deklination δ
t	der Stundenwinkel t
γ	der Frühlingspunkt γ
α	die Rektaszension α
α	der Bildwinkel α

Danksagung

An dieser Stelle möchte ich all jenen danken, die mich im Rahmen dieser Bachelorarbeit begleitet haben.

Ganz besonders möchte ich mich auch bei meinen Eltern bedanken, die mir durch ihre Unterstützung mein Studium ermöglicht haben.

Ebenso möchte ich Herrn Prof. Dr. Baun danken, der meine Arbeit und Themenfindung durch seine fachliche und persönliche Unterstützung begleitet hat.

1 Einführung

Kapitel 1 beschreibt welches konkrete Ziel diese Ausarbeitung hat, woher die entsprechende Motivation stammt, sowie die erdachte Lösung.

1.1 Problem

Die Nachtfotografie - weitestgehend auch als Astrofotografie bekannt - lässt sich in drei größere Bereiche eingrenzen. Objekte innerhalb des Sonnensystems und außerhalb des Sonnensystems, werden als Deep-Sky-Objekte bezeichnet. Der letzte Bereich beschränkt sich auf den Nachthimmel. Die Milchstraße gehört dabei zu den beliebtesten Objekten. Diese erstreckt sich über den gesamten Nachthimmel und ist besonders in den Sommermonaten gut sichtbar, teilweise bereits mit bloßem Auge. Die Milchstraße zu fotografieren ist an und für sich keine problematische Aufgabe, jedoch gibt es unterschiedliche Herangehensweisen.

Entweder wird ein sogenanntes Weitwinkelobjektiv verwendet und ein einzelnes Foto angefertigt. Eine weitere Option wäre der Einsatz eines Objektives mit größerer Brennweite, wobei jedoch immer nur ein kleinerer Teil des Himmels fotografiert werden. Durch längere Belichtungszeiten kann es zu weiteren Problemen kommen, dazu mehr in Abschnitt 2.3. Da mit jedem Bild nur ein Teil des Himmels fotografiert wird, müssen diese dann später wieder zusammengesetzt werden. Damit dies reibungslos funktioniert, müssen die entsprechenden Bilder möglichst nah aneinander sein, um diese gut verarbeiten zu können. Dies ist auch das Ziel dieser Ausarbeitung und der dabei entwickelten Anwendung.

1 Einführung

1.2 Motivation

Die Idee für diese Anwendung stammt von persönlichem Interesse an der Astronomie und der Fotografie. Panoramen und das allgemeine Interesse an der Milchstraße waren ebenfalls ausschlaggebend für die Entscheidung, sowie insbesondere ein hochauflösendes Panorama der Andromeda-Galaxy, siehe Abbildung 1.1[1].



Abbildung 1.1: Andromeda-Galaxie, aufgenommen vom Hubble-Weltraumteleskop

Ein weiterer Aspekt war der Wunsch nach einem minimalistischem Programm, welches sich auf einige grundlegende Aufgaben beschränkt. Ebenso sollte der Anwender in der Lage sein, die benötigten Koordinaten vor der Aufnahme zu ermitteln.

Eine geringe Lichtverschmutzung ist von immensem Vorteil, weshalb oftmals dunkle Gegenden bevorzugt sind. Da über einen Zeitraum von mehreren Stunden fotografiert wird, ist ein Stromzugang vonnöten, was jedoch nicht immer gewährleistet werden kann. Wenn bereits vor Aufbruch die entsprechenden Daten auf Papier vorhanden sind und nicht erst per Laptop ermittelt werden müssen, reduziert dies den Bedarf an Strom deutlich.

1.3 Erdachte Lösung

Wie in Abschnitt 1.2 beschrieben, war eine leichtgewichtige Lösung das Ziel. Der Anwender gibt neben der Brennweite des verwendeten Objektives oder Teleskopes die Höhe und die Breite des Sensors an. Die Anwendung errechnet den entsprechenden Bildwinkel in der

1 Einführung

Vertikalen sowie in der Horizontalen. Die jeweiligen Startkoordinaten setzen sich aus der Rektaszension sowie der Deklination zusammen. Die gewünschte Anzahl an horizontalen und vertikalen Bilder ist frei wählbar, ist jedoch auf jeweils 10 limitiert. Die entsprechenden Koordinaten werden in einem Textareal ausgegeben. Da die Resultate exportiert und anschließend ausgedruckt werden können, lässt sich hierbei deutlich das Gewicht und der Stromverbrauch der Ausrüstung reduzieren. Als Programmiersprache kommt Java zum Einsatz, da dadurch die Unabhängigkeit vom Betriebssystem gewährleistet ist und das Programm später ohne größere Probleme erweitert werden kann.

Anwendungen, welche ein ähnliches Ziel wie die Eigenentwicklung verfolgen existieren bereits, so zum Beispiel der in Kapitel 3 beschriebene EQMOD Treiber. Allerdings benötigt EQMOD im speziellen einen Computer sowie zusätzliche Anwendungen, wohingegen die Eigenentwicklung die entsprechenden Daten im Voraus produzieren kann. Für diese Ausarbeitung wurden sonstige Alternativen nicht in Betracht gezogen, obwohl es sicherlich eine Großzahl an ähnlichen Anwendungen gibt.

Kapitel 2 deckt die benötigten Grundlagen ab. Hierzu zählen zum Beispiel astronomische Koordinatensysteme sowie die Berechnung des Bildwinkels. Diese Themen wurden jedoch nur oberflächlich angeschnitten, da eine detaillierte Darstellung den Rahmen dieser Arbeit deutlich sprengen würde. Kapitel 3 befasst sich mit EQMOD, der Installation und Einrichtung. EQMOD erlaubt viele unterschiedliche Einstellungen und Kombinationen mit entsprechenden Programmen, allerdings wurde lediglich mit einer Planetariumssoftware gearbeitet. In Kapitel 4 wird die Eigenentwicklung genauer beschrieben. Interessante Aspekte der Entwicklung werden hervorgehoben, die Oberfläche erklärt und mögliche Erweiterungsgebiete erläutert. Nachdem beide Anwendungen getestet wurden, folgt abschließend Kapitel 5. Beide Anwendungen werden hinsichtlich des Ergebnis, des Aufwands und der Bedienbarkeit bewertet und gegenübergestellt.Kapitel 6 reflektiert zusammenfassend über die Entwicklung und die gesamte Arbeit an der Ausarbeitung.

Das nachfolgende Kapitel beschreibt die benötigten Grundlagen. Es beschreibt zum einen die astronomischen Koordinatensysteme und differenziert zwischen unterschiedlichen Systemen. Unterabschnitt 2.1.2 befasst sich mit den benötigten Umrechnungen, Abschnitt 2.2 schließt mit den Grundlagen zur Optik ab. Abschnitt 2.3 gibt einen kurzen Ausblick in die Astrofotografie.

2.1 Astronomische Grundlagen

Wie auch auf der Erde werden für astronomische Objekte ein Koordinatensystem benötigt, um entsprechende Objekte zu finden. Allerdings unterscheiden sich diese in einigen Punkten, wobei der Fokus auf den astronomischen Koordinatensystemen liegt, genauer dem Äquatorialen Koordinatensystem. Zuerst einige Grundlagen bezüglich der Astronomischen Koordinatensysteme.

Die Position des gesuchten Objekts wird mittels zweier Winkelkoordinaten beschrieben. Je nach System wird die verwendete Himmelskugel mit einem Grundkreis durchschnitten. Abhängig vom verwendetem System wird an einer beliebigen Stelle ein Nullpunkt festgelegt, von welchem aus eine der beiden Koordinaten gemessen wird. Die zweite Koordinate wird mithilfe der sogenannten Großkreise ermittelt, welche den Grundkreis senkrecht schneiden und in den Polen der Himmelskugel zusammenlaufen.

Die Winkel werden in astronomischen Koordinatensystemen wie folgt angegeben. Als "Haupteinheit" fungiert hierbei Grad, welches mit "°" dargestellt wird. Grad lässt sich in Untereinheiten aufteilen, welche aus Bogenminuten mit Zeichen "/" und Bogensekunden "//" dargestellt werden. Daraus resultiert zum Beispiel die folgende Darstellung:

230°25′ 49″

Deutlich zu sehen ist hier die Übereinstimmung mit GPS-Koordinaten, welche nach dem selben Prinzip aufgebaut sind. Dieses System lässt sich allerdings nicht ohne weiteres auf astronomische Objekte erweitern, da "die Erde komplizierte Bewegungen relativ zu den Sternen ausführt und sich damit die Koordinaten eines Sternes als Funktion der Zeit ändern"([2], S. 17).

2.1.1 Äquatoriales Koordinatensystem

Bevor auf das Äquatoriale Koordinatensystem eingegangen werden kann, muss zuerst das Horizontsystem betrachtet werden. Das Horizontsystem ist eine weitere Möglichkeit, die Position von Himmelsobjekten darzustellen. Hierbei wird, wie der Name schon andeutet, der Horizont als Grundkreis verwendet. Wie in Abbildung 2.1 zu sehen ist, fungiert der Südpunkt als Nullpunkt. Wird durch diesen nun ein weiterer Großkreis gezogen, entsteht dadurch der sogenannte Meridian, welcher die beiden Pole des Systems, Zenit und Nadir, durchschreitet. Eine der verwendeten Koordinaten ist hierbei das Azimut *A*, welches auf dem Horizont entlangwandert, beginnend im Südpunkt über Westen, Norden und Osten.



Abbildung 2.1: Horizontsystem

Da zur genauen Positionsbestimmung zwei Koordinaten vonnöten sind, wird die Höhe h eingeführt, wahlweise auch als Zenitdistanz *z* beschrieben. *z* setzt sich hierbei wie folgt zusammen:

$z = 90^{\circ}$ - h

Durch diese Gleichung lässt sich ohne weiteres bestimmen, ob das gesuchte Objekt unterhalb des Horizonts liegt. Wenn z negativ ist, befindet sich das Objektiv unterhalb des Horizonts. Ein großer Nachteil des Horizontsystems ist allerdings, dass alle Koordinaten vom jeweiligen Beobachtungsort und Zeitpunkt abhängig sind. Bewegt sich der Betrachter, muss die Position des Objektes erneut bestimmt werden, gleiches gilt auch für den Zeitraum. Da die Erde sich in ständiger Rotation befindet, ändert sich auch fortlaufend der Horizont, wodurch Objekte unter diesen fallen und dementsprechend nicht mehr beobachtet werden können. (aus [2], S. 18)

Das Äquatorialsystem nimmt sich der Nachteile des Horizontsystems an und entfernt diese nach und nach. Während der Grundkreis im Horizontsystem eine Variable ist, wird er im Äquatorialen Koordinatensystem als Konstante betrachtet. Der Äquator selbst wird als Grundkreis verwendet. Als Pole kommen hierbei Nord- und Südpol zum Einsatz. Würde man Nordpol und Südpol mit einer Geraden verlängern, erhält man die Erdachse um welche die Erde rotiert. Dadurch, dass diese senkrecht auf dem Äquator steht, rotieren die Sterne nahezu auf Parallelkreisen. Der Winkelabstand zum Äquator ist weitestgehend konstant. Daraus resultiert die erste Koordinate, die sogenannte Deklination, in der Astronomie mit δ bezeichnet. Die Deklination verläuft hierbei parallel zum Äquator. Alle Objekte oberhalb des Äquators, genauer in der Nordhalbkugel, haben einen positiven Wert. Objekte, die unterhalb des Äquators liegen, haben einen negativen Wert. Da, wie auch beim Horizontsystem eine zweite Koordinate vonnöten ist, muss diese auch ermittelt werden. Hierbei gibt es zwei Möglichkeiten. (aus [2], S. 18 f.)

Festes Äquatorsystem

Der Name stammt daher, da der Nullpunkt fest mit dem Beobachtungsort verbunden ist. Der Schnittpunkt des Meridians mit dem Äquator ergibt in diesem System den Nullpunkt. Von diesem aus wird die zweite Koordinate ermittelt, der sogenannte Stundenwinkel *t*. Wie in Abbildung 2.2 zu sehen ist, wird dieser von Süden aus in Richtung Westen, Norden und Osten gezählt.



Abbildung 2.2: Äquatorsystem

Hierbei sind jedoch einige Punkte zu beachten. Zum einen ist der Stundenwinkel von der Position des Beobachters abhängig, da dieser mit dem Meridian zusammenhängt und jeden Tag einen vollen Umlauf durchläuft. Des Weiteren wird er nicht im Bogenmaß gemessen, sondern im Zeitmaß, welches sich wie folgt umrechnen lässt:

$$1^h = 15^\circ \mid 1^m = 15' \mid 1^s = 15''$$

Wenn also ein Objekt die folgende Eigenschaft $t = 0^h$ besitzt, befindet es sich genau auf dem Meridian, da dieser, wie bekannt, genau den Süden schneidet. Mithilfe der Koordinaten δ und t ist es nun möglich, ein Objekt zu einem gegebenen Zeitpunkt in Relation zum Äquator und Meridian zu finden. Allerdings ist diese Methode zur Katalogisierung aufgrund der Abhängigkeit von Ort und Zeit ebenso unbrauchbar wie das zuvor beschriebene Azimut.

Bewegliches Äquatorsystem

Für die Beobachtung von astronomischen Objekten ist ein Punkt, welcher unabhängig von Ort und Zeit des Beobachters ist, eindeutig die bessere Option. Dieser Punkt wird als sogenannter Frühlingspunkt (Zeichen: Y) bezeichnet. Dieser "läuft" mit den Sternen, wodurch sich von hier aus ermittelte Koordinaten nicht mehr ändern. Zweimal im Jahr überschreitet die Sonne den Äquator. Da der Wechsel von Süden auf Norden im Frühjahr vollzogen wird, wird der Punkt auch als "Frühlingspunkt" bezeichnet. Die nachfolgende Grafik verdeutlicht dies:



Abbildung 2.3: Bahn der Sonne

Der Frühlingspunkt ist der Startpunkt für die zweite Koordinate, die sogenannte Rektaszension α . Mit Blick auf Abbildung 2.2 wird ersichtlich, dass α gegensätzlich zum Stundenwinkel verläuft und in einem scheinbar zufälligen Punkt beginnt, wobei es sich jedoch um den zuvor beschriebenen Frühlingspunkt handelt, siehe hierzu auch Abbildung 2.3. α wird ebenfalls in Stunden, Minuten und Sekunden angegeben und greift auf die gleiche Umrechnungsformel zurück.

Abschließend lässt sich nun sagen, dass alle benötigten Koordinaten vorhanden sind, um die Position eines Objektes zu beschreiben. Die Koordinaten des entsprechenden Objektes werden also mithilfe der Rektaszension α und der Deklination δ beschrieben.

2.1.2 Umrechnung

Die Rektaszension verfügt über die gleiche Notation wie der Stundenwinkel, ist also auch in Stunden, Minuten und Sekunden aufgelöst. Die Umrechnung erfolgt nach dem gleichen Prinzip, eine Stunde sind 15°, eine Minute sind 15 Bogenminuten und 1 Sekunde sind 15 Bogensekunden. Nachfolgend eine Beispielsrechnung:

Ein unbekanntes astronomisches Objekt hat die folgende Rektaszension und soll in Grad, Bogenminuten und Bogensekunden umgerechnet werden.

23h 59min 59sek

Die Umrechnung erfolgt von rechts nach links, es wird also mit den Sekunden begonnen. Zuerst wird der Gesamtbetrag der Bogensekunden ermittelt:

$$59 \cdot 15 = 885$$

Im nächsten Schritt werden nun die überschüssigen Bogenminuten von der tatsächlichen Anzahl der Bogensekunden getrennt:

$$\frac{885}{60} = 14,75$$

Das Resultat wird in Vorkommazahl und Nachkommastellen aufgeteilt: 14 Bogenminuten und 0,75 Bogenminuten. Nachkommastellen werden nun mit 60 multipliziert, um die tatsächlichen Bogensekunden zu erhalten:

$$0,75 \cdot 60 = 45$$

Der erste Teil der Umrechnung ist damit beendet. Als nächstes werden nun die Bogenminuten ermittelt. Hierbei ist es egal, wann die überschüssigen Bogenminuten addiert werden, das Resultat ist identisch. Der Ablauf der Umrechnung unterscheidet sich gegenüber der Umwandlung der Bogensekunden lediglich darin, dass die überschüssigen Bogenminuten addiert werden müssen. Ebenso entstehen keine überschüssigen Bogenminuten, sondern Grad, welche im letzten Schritt addiert werden.

2 Grundlagen

$$59 \cdot 15 = 885$$

 $\frac{885}{60} = 14,75$
 $0,75 \cdot 60 = 45$
 $45 + 14 = 59$

Wie in der letzten Zeile zu sehen ist, werden dort die überschüssigen Bogenminuten addiert. Die Koordinaten des Objekts sehen nun wie folgt aus:

59' 59"

Umwandlung der Stunden in Grad und Addition der überschüssigen Grad.

 $23 \cdot 15 = 345$ 345 + 14 = 359

Das Endresultat der Umwandlung lautet also:

359°59′ 59″

Dies ist nur ein kleiner Einblick in die unterschiedlichen Koordinatensysteme der Astronomie, lediglich das Äquatoriale ist für diese Ausarbeitung interessant.([2], S. 19 ff.)

Ebenso ist eine Umrechnung von Grad/Bogenminuten/Bogensekunden in Stunden/Minuten/Sekunden möglich. Hierbei wird jedoch rückwärts gerechnet, die Umrechnung beginnt also bei Grad. Als Ausgangspunkt wird 359°59′59″ verwendet. Als Umrechnungskonstante kommt ebenfalls 15 zum Einsatz.

$$\frac{359}{15} = 23,93$$

Das Resultat lässt sich nun in zwei eigene Bereiche aufteilen. 23 beschreibt die Stunden, wohingegen 0,93 die Minuten darstellt. Diese werden mit 60 multipliziert und mit den 59 Bogenminuten addiert. 2 Grundlagen $0,93 \cdot 15 = 14$ $14 \cdot 60 + 59 = 899$

Das Resultat der Rechnung muss ebenfalls durch 15 geteilt werden, um den finalen Wert für die Minuten zu erhalten. Das Resultat lässt sich ebenfalls aufteilen und wird zur Umwandlung der Sekunden benötigt.

$$\frac{899}{15} = 59,93$$
$$0,93 \cdot 15 = 14$$
$$14 \cdot 60 + 59 = 899$$
$$\frac{899}{15} = 59,93$$

Eine weitere Umrechnung ist nicht notwendig, da die Sekunden schon eine ausreichende Präzision aufweisen. Das Endresultat der Umwandlung lautet nun 23h 59min 59sek.

2.2 Optik

Bei der Optik handelt es sich "ursprünglich um die Lehre vom sichtbaren Licht"([6], S. 397), allerdings wird nachfolgend lediglich die Berechnung des Bildwinkels beschrieben. Die allgemeine Formel lautet:

$$\alpha = 2 \cdot \arctan(\frac{L}{2 \cdot f})$$

 α beschreibt hierbei den Bildwinkel, L eine Kantenlänge oder Diagonale des Sensors und f die Brennweite der verwendeten Optik. Da L variabel ist, lässt sich hier die Höhe, Breite oder Diagonale des Sensors verwenden, die Diagonale wird mithilfe des Satz des Pythagoras ermittelt ([7], S. 110 f.).

Angenommen, es kommt eine Kamera mit einer Sensorgröße von 24mmx36mm zum Einsatz, das verwendete Objektiv besitzt eine Brennweite von 35mm. Nun soll der horizontale Bildwinkel berechnet werden:

$$\alpha = 2 \cdot \arctan(\frac{36}{2 \cdot 35})$$

Dementsprechend beträgt α 54,4° in der Horizontalen, die Berechnung der Vertikalen läuft identisch ab. Je nach Kamera oder Sensortyp verändert sich der Bildwinkel bei gleichbleibender Brennweite.

2.3 Astrofotografie

Abschließend noch ein Exkurs in die Astrofotografie. Wie bereits in Abschnitt 1.1 beschrieben, existieren unterschiedliche Bereiche der Astrofotografie. Für Bilder von Planeten, Mond und der Sonne (jedoch nur mit entsprechenden Filtern), ist lediglich ein Stativ, eine Kamera sowie ein Objektiv mit hoher Brennweite oder ein Teleskop vonnöten. Abbildung 2.4 zeigt eine eigene Aufnahme des Mondes.



Abbildung 2.4: Mond

Eine Nachführung wird hier seltener benötigt, lediglich bei der Aufnahme von Planeten ist sie von Vorteil, da hier oftmals Videos aufgenommen werden anstatt einzelne Bilder.

Bei Aufnahmen der Milchstraße ist es abhängig vom Objektiv und dem Ziel der Aufnahme. Ein Weitwinkelobjektiv kommt zum Einsatz, wenn ein möglichst großer Anteil der Milchstraße fotografiert werden soll. Wenn allerdings nur der Kern der Milchstraße oder gewisse Bereiche mit entsprechender Detailfülle aufgenommen werden soll, ist eine Nachführung, wie in Abbildung 2.5 zu sehen ist, unumgänglich.

Aufgrund der höheren Brennweite wird die Erdrotation verstärkt wahrgenommen, was in strichförmigen Sternen resultiert, da lange Belichtungszeiten zum Einsatz kommen. Wenn lange Belichtungszeiten in Verbindung mit hohen Brennweiten zum Einsatz kommen, ist eine Nachführung unumgänglich. Diese führt, wie der Name schon andeutet, nach und verfolgt ein Objekt über den Nachthimmel.



Abbildung 2.5: Oberer Teil der Nachführung ohne Gegengewichte und Teleskop

Bevor diese jedoch eingesetzt werden kann, sind einige Punkte zu beachten. Als erstes muss die Montierung nach Norden ausgerichtet werden, was als sogenanntes "einnorden" bekannt ist. Hierfür wird die Montierung zuerst grob in Richtung Norden ausgerichtet. Die meisten Montierungen besitzen ein sogenanntes Polsucherfernrohr, welches durch die Montierung verläuft und eine präzise Ausrichtung auf den Polarstern erlaubt. Der Polarstern selbst wandert um den wahren Himmelspol. Wie in Abbildung 2.5 bereits erkennbar ist, verfügt die Montierung über Schrauben, welche zur genauen Positionierung verwendet werden. Sobald die Montierung ausgerichtet ist, wird der Polarstern über den Polsucher aufgesucht.



Abbildung 2.6: Azimutschrauben

In Abbildung 2.6 sind die Azimutschrauben markiert, welche die horizontale Ausrichtung der Montierung verändern. Hierbei muss zuerst eine der beiden Schrauben gelockert werden, bevor die andere angezogen wird. In Abbildung 2.7 sind die Altitudschrauben zu erkennen.



Abbildung 2.7: Altitudschrauben

Diese verstellen, wie der Name schon andeutet, die Höhe der Montierung. Um die Montierung nun entsprechend einzunorden, wird der Polarstern mithilfe des Polsucherfernrohrs ermittelt und mithilfe der beschriebenen Schrauben positioniert. Als Unterstützung besitzen die meisten Polsucherfernrohre eine Markierung für den Polarstern sowie umliegende Sternbilder, was das auffinden erleichtert.

Nachdem dies erledigt ist, wird das Teleskop montiert und austariert. Abschließend muss die Montierung korrekt ausgerichtet werden, was über 1-3 Sterne erfolgt. Hierbei wird aus einer vorgegebenen Liste ein beliebiger Stern ausgewählt und von der Montierung selbstständig angefahren. Normalerweise zeigt das Teleskop meist in eine komplett andere Richtung, weshalb mit Hilfe der Handsteuerung der Stern in die Mitte des Teleskops positioniert werden muss. Sollte eine Ausrichtung mit 2 oder 3 Sternen ausgewählt worden sein, wird danach ein weiterer Stern ausgewählt und ebenfalls angefahren. Wenn die vorherigen Schritte korrekt durchgeführt wurden, befindet sich der gesuchte Stern meist recht nah in der Mitte des Teleskops, jedoch muss der Stern wieder mittig platziert werden, um die Genauigkeit zu erhöhen. Nachdem diese Schritte erledigt sind, kann mit der Langzeitbelichtung begonnen werden. Je öfter

eine Ausrichtung durchgeführt wird, desto präziser ist diese am Ende. Die Nachführung erlaubt es dem Anwender, eine Großzahl unterschiedlicher astronomischer Objekte aus einer Datenbank auszuwählen und mit der Nachführung direkt anzusteuern. Abbildung 2.8 zeigt die Nachführung mit montierten Gegengewichten sowie Teleskop.



Abbildung 2.8: Fertig montierte Nachführung

Dieses Kapitel beschreibt eine bereits vorhandene und ausgereifte Alternative zur Eigenentwicklung. Abschnitt 3.2 befasst sich mit der Installation und Funktion von EQMOD, in Abschnitt 3.3 werden die entsprechenden Vor- und Nachteile aufgelistet.

3.1 EQMOD/EQMosaic

Nachfolgend wird die Installation und Anwendung von EQMOD sowie EQMosaic beschrieben. Im nachfolgenden Abschnitt wird auf die entsprechenden Vor- und Nachteile eingegangen.

3.2 Erklärung und Installation

EQMOD, auch EQASCOM genannt, erlaubt die direkte Kontrolle einer Nachführung per Computer. Die Unterstützung ist allerdings auf einige spezielle Marken limitiert. Des Weiteren wird eine zusätzliche Modifikation der Handsteuerung oder ein zusätzliches Modul benötigt. EQMOD ist kein eigenständiges Programm, sondern benötigt eine weitere Anwendung um korrekt zu arbeiten. EQMOD selbst fungiert nur als Treiber, welcher in Zusammenarbeit mit ASCOM funktioniert.

Der Anwender kann zum Beispiel Stellarium oder Cartes du Ceil verwenden. Hierbei handelt es sich um Astronomieprogramme, welche ein Planetarium simulieren. Sie zeigen präzise unterschiedliche astronomische Objekte auf und erlauben deren Beobachtung am eigenen Computer. Zudem lassen sich zukünftige Positionen sowie zusätzliche Informationen über das anvisierte Objekt anzeigen. Nachdem die Handsteuerung mit dem Computer verbunden wurde, entweder per USB oder RS232, wird ASCOM installiert, gefolgt von EQMOD selbst. Sobald alles korrekt installiert wurde, wird die EQASCOM Toolbox gestartet, siehe Abbildung 3.1.[8]

	- 10 - 10 - 10 - 10 - 10 - 10 - 10 - 10	Province and the second
eqmod.exe 👱	Deregister	Register
Setup		8 E
eqmod.exe 🔄 💌	Driver Setup	ASCOM Connect
	Comms Statistics	
Configuration files		P
eqmod.exe 📃 👱	Backup	Delete
General 🔄	Restore	View/Edit
rom	То	
	1	Copy Configuration

Abbildung 3.1: EQASCOM Toolbox

Ein Klick auf "Driver Setup" öffnet Abbildung 3.2. Im Bereich "EQMOD Port Details" werden folgende Werte eingegeben:

> Timeout: 2000 Retry: 2 Baud: 9600

ASCOM Setup		×
	SCOM SETUP	
Mount Options Auto Detect	Site Information	ASCOM Options Strict Conformance Synchronous Park Allow Site Writes VIssue Exceptions VPulseguide Exceptions SideOfPier Pointing (ASCOM)
	- Guiding ASCOM PulseGuiding	Gamepad Configuration
Allow Auto Meridian Flip Windows Process Priority Normal Language	Slew Preset Rates: 1:1 2:8 3:64 4:800 Set	Show Advanced Options Update Notifications Disabled Friendly Name
	ок	

Abbildung 3.2: ASCOM Setup

Abschließend muss noch Breitengrad, Längengrad, die Höhe des Beobachtungsort und die Nord- oder Sübhalbkugel eingestellt werden. Nach erfolgreichem Verbindungsaufbau kann die Nachführung nun mit dem Computer verbunden und in Stellarium angezeigt werden. Im Falle von Stellarium wird allerdings "StellariumScope" benötigt, da Stellarium von Haus aus nur wenige Montierungen unterstützt. Der Start erfolgt über StellariumScope, ebenso wie der Verbindungsaufbau mit der Nachführung selbst. Sobald die Verbindung erfolgreich aufgebaut ist, wird Stellarium gestartet und zeigt nun die genaue Position des Teleskopes an, wie in Abbildung 3.3 zu sehen ist.



Abbildung 3.3: Stellarium

EQMOD bietet zudem auch die Möglichkeit, einen sogenannten periodischen Fehler auszugleichen. Hierbei handelt es sich um Fehler, welche durch minderwertig produzierte Zahnräder in der Nachführung hervorgerufen werden. Der Fehler macht sich in einer Art wackeln bemerkbar und lässt sich mechanisch reduzieren, allerdings nicht komplett beheben.

Nachdem EQMOD eingerichtet wurde, kann EQMOSAIC installiert werden. Der Installati-



onsprozess ist selbsterklärend und muss nicht weiter beschrieben werden.

Abbildung 3.4: EQMOSAIC

Zuerst muss die Nachführung mit EQMOSAIC verbunden werden, was per Klick auf "Choose Scope" geschieht. Es ist ebenso möglich, in diesem Schritt die Planetariumssoftware zu starten, was die Zielauswahl später deutlich erleichtert. Nachdem die Nachführung ausgewählt wurde, muss diese nun verbunden werden. Wenn die Verbindung erfolgreich hergestellt wurde, ist der Mittelpunkt des Mosaiks identisch mit dem des Teleskopes oder Objektives. Die Bewegung der Nachführung erfolgt nun entweder über die Planetariumssoftware oder per Doppelklick in eines der Felder in EQMOSAIC. Über den Bereich "Frame Settings" lassen sich zudem weitere Einstellungen vornehmen, wie zum Beispiel die Größe des Bildes. Veränderungen hierbei sind nur für die Anwendung interessant, da diese die entsprechende "Distanz" zurücklegen muss, um nah beieinander liegende Bilder zu erstellen. "Frame Overlap" hingegen gibt an, wie groß die Überschneidung der Bilder ist.

3.3 Vor- Und Nachteile

Vorteile

- EQMOD bietet durch den Einsatz einer Planetariumssoftware eine grafische Oberfläche im Vergleich zu den textbasierten Handboxen der Nachführungen. Ebenso hat der Anwender dadurch Zugriff auf nahezu endlose Auswahl- und Einstellungsmöglichkeiten. Da die meisten Programme visuelle Beispiele direkt mitliefern, erhält der Anwender ebenfalls eine Vorschau auf das zu beobachtende Objekt. Dies erleichtert das Beobachten und Fotografieren ungemein.
- Speichern von benötigten Daten, zum Beispiel Längengrad, Breitengrad und so weiter, welche sonst von Hand immer neu eingegeben werden müssen. Des Weiteren lassen sich bis zu 10 unterschiedliche Beobachtungsorte speichern.
- Erweitertes Ausrichten: Falls nur die Handbox der Nachführung verwendet wird, ist die Anzahl der auswählbaren Sterne sehr eingeschränkt, zudem lässt sich maximal eine 3-Sterne-Ausrichtung durchführen. Mit EQMOD ist der Anwender hingegen in der Lage, so viele beliebige Sterne zu verwenden, wie er wünscht. Dies resultiert, wie bereits im ersten Punkt beschrieben, in einer deutlich erhöhten Genauigkeit.
- Verbessertes GoTo: Unter GoTo versteht man das automatische Ausrichten auf ein bestimmtes Objekt. EQMOD bietet hier ebenfalls verbesserte Genauigkeit und iterative Ausrichtung.
- Ein weiterer Vorteil ist der Zugriff auf eine immens große Datenbank. Die Handboxen der Nachführungen verfügen über recht große Datenbanken, sind jedoch im Vergleich zu EQMOD stark limitiert.

Nachteile

 Der größte Vorteil von EQMOD ist gleichzeitig auch der größte Nachteil. Dadurch, dass die Anwendung nur mithilfe eines Computers verwendet werden kann, muss zusätzliche Ausrüstung zum Beobachtungsort transportiert werden. Wenn die ganze Nacht beobachtet oder fotografiert werden soll, ist zudem eine zusätzliche Stromversorgung erforderlich. Diese Nachteile sind vernachlässigbar, wenn die Nachführung im eigenen Garten oder in der Nähe einer Stromzufuhr aufgestellt wird. Sie können

aber auch recht schnell problematisch werden, wenn abseits von jeglicher Lichtquellen fotografiert wird.

Wie bereits in Abschnitt 1.3 beschrieben, war das Ziel eine leichtgewichtige Lösung zu entwickeln, welche unabhängig von einem spezifischen Betriebssystem ist. Des Weiteren sollten folgende Funktionalitäten vorhanden sein:

- Eingabe von Brennweite, Sensorhöhe, Sensorbreite und Berechnung des Vertikalen und Horizontalen Bildwinkels.
- Umrechnung der Rektaszension aus dem Format Stunden, Minuten, Sekunden in Grad, Bogenminuten und Bogensekunden.
- Eingabe von Rektaszension, Deklination sowie der Anzahl an Bildern, welche in der Horizontalen und Vertikalen aufgenommen werden sollen. Diese Informationen werden zusammen mit dem Bildwinkel in die entsprechenden Koordinaten umgerechnet.
- Exportieren der Ergebnisse als .txt-Datei, um diese zu drucken und zum Beobachtungsort mitzunehmen.

Ursprünglich war geplant, die Kamera selbst anzusteuern. Dies wurde durch den damit verbundenen Aufwand verworfen, da die entsprechende Anwendung sich nicht per Java entwickeln ließe, sondern nur per C#. Eine einfache Integration war hierdurch nicht gewährleistet. Abgesehen davon wurden alle Ziele erreicht und die Anwendung funktioniert wie erdacht. An dieser Stelle muss jedoch erwähnt werden, dass diese nicht perfekt ist und kein perfektes Resultat erwartet werden darf. Ebenso handelt es sich bei der Anwendung um einen Prototyp, welcher unbekannte Fehler enthalten kann, welche unregelmäßig auftreten können.

4.1 Anforderungsanalyse und Use-Case

Um die Entwicklung der Anwendung zu vereinfachen und um einen strukturierten Entwicklungsplan zu erstellen, wurde vor Beginn der Entwicklung eine Anforderungsanalyse durchgeführt und in einem Use-Case-Diagramm zusammengefasst.

4.1.1 Anforderungsanalyse

Die Anwendung muss in der Lage sein

- den Bildwinkel eines Objektives sowohl in der Horizontalen als auch in der Vertikalen zu errechnen, hierfür benötigt sie die Brennweite sowie Höhe und Breite des verwendeten Sensors.
- die Rekatszension von ihrem ursprünglichen Format (Stunden/Minuten/Sekunden) in (Grad/Bogenminuten/Bogensekunden) umzurechnen.
- die gewünschten Koordinaten zu errechnen, welche für ein Panorama des Nachthimmels benötigt werden. Hierfür benötigt die Anwendung den Vertikalen und Horizontalen Bildwinkel, wie in Punkt 1 bereits beschrieben. Ebenso wird die Rektaszension, Deklination und die Anzahl der Vertikalen und Horizontalen Bilder benötigt. Sobald alle benötigten Daten vorhanden sind, kann der Anwender die Koordinaten berechnen lassen, diese werden in einem Textareal angezeigt.
- die Koordinaten in eine Textdatei zu speichern, damit der Anwender diese zwischenspeichern kann.

4.1.2 Use-Case

Abbildung 4.1 verdeutlicht die typische Nutzung der Anwendung anhand eines Use-Cases.



Abbildung 4.1: Use-Case für die Anwendung des Programms

4.2 Interessante Segmente der Entwicklung

▲ Calculator v0.1
Brennweite
Höhe des Sensors
Breite des Sensors
Vertikale Bildwinkel in Grad
Vertikale Bogenminuten
Vertikale Bogensekunden
Horizontale Bildwinkel in Grad
Horizontale Bogenminuten
Horizontale Bogensekunden
Rektaszension 00 h 00 m 00 s
Rektaszension > ° ' "
Deklination 00 ° 00 ' 00 "
Anzahl der Horizontalen Bilder
Anzahl der Vertikalen Bilder
Löschen
Export
Koordinaten berechnen

Abbildung 4.2: Hauptfenster der Anwendung

Am interessantesten sind die Umrechnungen, welche in Abschnitt 2.2 und Unterabschnitt 2.1.1 beschrieben wurden. Der restliche Code ist weitestgehend trivial und bedarf keiner Erklärung.

Abbildung 4.2 zeigt das Hauptfenster der Anwendung. Die Felder "Brennweite", "Höhe des Sensors" sowie "Breite des Sensors" erlauben nur die Eingabe einer Ganzzahl, abweichende Eingaben werden nicht registriert. Dieser Code basiert auf dem *DocumentFilter*, welcher die entsprechende Eingabe filtert und nur das erwünschte Ergebnis in das Textfeld schreibt. Nachdem alle drei Felder befüllt sind, errechnet das Programm automatisch die entsprechenden Werte für den horizontalen und vertikalen Bildwinkel. Der Quelltext wurde nicht selbst entwickelt, sondern teilweise aus nachfolgenden Quelle entnommen [9].

```
Listing 4.1: DocumentFilter.java
```

```
1 DocumentFilter df = new DocumentFilter() {
               @Override
2
              public void insertString(FilterBypass fb, int i,
3
                 String string, AttributeSet as) throws
                 BadLocationException {
4
                   if (isDigit(string)) {
5
                       super.insertString(fb, i, string, as);
6
                       calcAndSetTotal();
7
                   }
8
              }
a
10
               @Override
11
              public void remove(FilterBypass fb, int i, int i1)
12
                 throws BadLocationException {
                   super.remove(fb, i, i1);
13
                   calcAndSetTotal();
14
              }
15
16
               @Override
17
              public void replace (FilterBypass fb, int i, int i1,
18
                   String string, AttributeSet as) throws
                 BadLocationException {
                   if (isDigit(string)) {
19
                       super.replace(fb, i, i1, string, as);
20
                       calcAndSetTotal();
21
22
                   }
23
              }
24
25
              private boolean isDigit(String string) {
26
                   for (int n = 0; n < string.length(); n++) {</pre>
27
                       char c = string.charAt(n);
28
```

Listing 4.2 beschreibt die Erstellung des *DocumentFilter*. Abschließend muss dieser den entsprechenden Feldern hinzugefügt werden. Dies geschieht im Listing 4.2.

Listing 4.2: addDF.java

```
1 ((AbstractDocument) (sensorHeight.getDocument())).
setDocumentFilter(df);
2 ((AbstractDocument) (sensorWidth.getDocument())).
setDocumentFilter(df);
```

Der *DocumentFilter* erlaubt nur noch Ganzzahlen, alle anderen Eingaben werden direkt gelöscht.

Listing 4.3: calcAndSetTotal.java

```
void calcAndSetTotal() {
2
     if (!sensorHeight.getText().isEmpty() && !sensorWidth.
3
        getText().isEmpty()) {
             Calculation.calculateFieldOfViewVertical(Double.
4
                parseDouble(sensorHeight.getText()), Double.
                parseDouble(focalLengthT.getText()));
             Calculation.calculateFieldOfViewHorizontal(Double.
5
                parseDouble(sensorWidth.getText()), Double.
                parseDouble(focalLengthT.getText()));
         }
6
     }
7
```

Die Funktion Listing 4.2 wird aufgerufen, sobald die Höhe und Breite des Sensors in das entsprechende Textfeld eingetragen wurden, wie in Zeile 3 zu sehen ist. Die Funktion selbst ruft lediglich *calculateFieldOfViewVertical* auf und übergibt die Sensorhöhe sowie die Brennweite, gleiches gilt für den horizontalen Bildwinkel, wohingegen hier die Breite übergeben

wird. Da sich die Berechnung lediglich im Übergabewert unterscheiden, wird nur eine der beiden Funktionen näher betrachtet. Zur besseren Übersicht wird die Funktion unterteilt und einzeln betrachtet.

Listing 4.4: calculateFieldOfViewVertical.java

```
double calc1 = (x / (2* y));
1
         double a = 2 * Math.toDegrees(Math.atan(calc1));
2
         a = a * 100;
3
         a = (int) a;
4
         double b = a/100.0;
5
6
         int verticalDegree = (int) b;
7
         MainWindow.bildwinkelVerticalValue.setText(String.
8
            valueOf(verticalDegree));
```

In Zeile 1 und 2 aus Listing 4.2 kommt die Formel zur Berechnung des Bildwinkels aus Abschnitt 2.2 zum Einsatz. Das Resultat aus dieser Rechnung wird in den Zeilen 3 bis 5 auf zwei Nachkommastellen reduziert und in Zeile 7 in eine Ganzzahl umgewandelt, welche für die spätere Berechnung der Bogenminuten nötig ist. Zeile 8 beschreibt das Befüllen des Textfelds in der grafischen Oberfläche.

Listing 4.5: calculateFieldOfViewVertical.java

```
double c = b - verticalDegree;
c = c*60;
c = c*100;
c = c*100;
c = (int) c;
c = c/100.0;
int verticalArcMinute = (int) c;
MainWindow.bogenMinutenVertikalT.setText(String.valueOf
(verticalArcMinute));
```

Zeile 1 in Listing 4.2 zeigt den Code zur Berechnung der Nachkommastelle. Bei *b* handelt es sich um den Bildwinkel mit Nachkommastelle, bei *verticalDegree* um den zur Ganzzahl umgewandelten Wert. Diese werden voneinander subtrahiert um die entsprechende Nachkommazahl zu erhalten. Wie aus Abschnitt 2.1 bekannt, muss das Resultat mit 60

multipliziert werden, um die Bogenminuten zu erhalten. Abschließend erfolgt die Reduktion auf Zwei Nachkommastellen und das Schreiben in das entsprechende Textfeld. Der Code zur Berechnung der Bogensekunden ist weitestgehend identisch, weshalb dieser hier nicht aufgelistet wird.

Die Umrechnung der Rektaszension vom Format h/min/sek zu Grad/Bogenminuten/Bogensekunden ist ebenfalls interessant. Die Funktion Listing 4.2 benötigt hierbei die entsprechenden Koordinaten aus dem Textfeld. Die genaue Erklärung der Umrechnung wurde in Unterabschnitt 2.1.2 beschrieben und wird deshalb hier nicht weiter erwähnt.

Listing 4.6: convertRektToDAMAS.java

```
double arcs = z*15;
1
          double arcsD = arcs/60;
2
          if((arcsD - (int)arcsD) == 0) \{
3
              arcmSurplus = (int)arcsD;
              MainWindow.rektaszensionS.setText("00");
5
          }
6
          else {
7
              arcmSurplus = (int)arcsD;
8
              double arcsRestValue = arcsD - (int)arcsD;
              arcs = arcsRestValue*60;
10
              arcsI = (int)arcs;
11
              MainWindow.rektaszensionS.setText(String.valueOf(
12
                 arcsI));
          }
13
```

Die Umrechnung ist, wie oben beschrieben, weitestgehend bekannt. *z* beinhaltet die Sekunden der Rektaszension, wird mit 15 multipliziert und durch 60 geteilt, um die Bogenminuten und restlichen Bogensekunden zu erhalten. Dieser Wert wird in der Variable *arcsD* gespeichert. Zeile 3 bis 6 beschreibt eine Abfrage, welche prüft, ob *arcsD* eine Ganzzahl ist. Falls dies der Fall ist, wird eine Umwandlung durchgeführt und der Wert als Überschuss gespeichert. Andernfalls wird in den Zeilen 8 bis 12 die Umrechnung durchgeführt. *arcsD* wird umgewandelt und zwischengespeichert. *arcsRestValue* beinhaltet die Nachkommastellen und wird verwendet, um die Bogensekunden zu errechnen. Dieser

Wert wird abermals umgewandelt und letztendlich in das entsprechende Textfeld geschrieben.

Listing 4.7: convertRektToDAMAS.java

```
double arcm = y*15;
1
          arcm += arcmSurplus;
2
          double arcmD = arcm/60;
3
          if((arcmD - (int)arcmD) == 0) {
4
              archSurplus = (int)arcmD;
5
              MainWindow.rektaszensionM.setText("00");
6
          }
7
          else {
8
              archSurplus = (int)arcmD;
9
              double arcmRestValue = arcmD - (int)arcmD;
10
              arcm = arcmRestValue *60;
11
              arcmI = (int)arcm;
12
              MainWindow.rektaszensionM.setText(String.valueOf(
13
                 arcmI));
          }
14
```

Die Umrechnung von Minuten zu Bogenminuten ist weitestgehend gleich, jedoch muss der Überschuss addiert werden, was in Zeile 2 in Listing 4.2 geschieht. Die restliche Operation verläuft identisch zur Umrechnung der Bogensekunden. Der errechnete Überschuss wird im nachfolgenden Code auf die Grad angerechnet.

Listing 4.8: convertRektToDAMAS.java

```
double arch = x*15;
arch += archSurplus;
archI = (int)arch;
MainWindow.rektaszensionD.setText(String.valueOf(archI));
```

Die im Listing 4.2 beschriebene Umrechnung ist weniger sinnvoll, da sie recht selten benötigt wird. Sollte ein Anwender dennoch das Bedürfnis haben eine solche Umrechnung durchzuführen, ist die entsprechende Möglichkeit vorhanden.

Die Koordinaten werden mithilfe der Funktion *fillTextArea* in das Textareal eingetragen. Bevor die Koordinaten jedoch eingetragen werden können, muss der Bildwinkel in das gleiche Format wie die Rektaszension umgerechnet werden. Die Umrechnung wurde bereits in Listing Unterabschnitt 2.1.2 näher erläutert und Bedarf keiner Wiederholung.

Listing 4.9: fillTextArea.java

Die Anzahl der Vertikalen und Horizontalen Bilder gibt den Grenzbereich für die verschachtelten for-Schleifen an, wie in Listing 4.2 beschrieben. Aufgrund der Limitierung auf jeweils 10 Bilder sind maximal 100 Durchläufe möglich, welche dennoch zügig abgearbeitet werden.

Listing 4.10: fillTextArea.java

```
if (rektM <= 0.0) {
    rektM += 60;
    rektH --;
  }
</pre>
```

Da die Minuten den Wert 60 nicht überschreiten dürfen, muss geprüft werden, ob sich der Wert im erlaubten Bereich befindet. Der zuständige Programmcode ist in Listing 4.2 dargestellt. Sollte sich der Wert im negativen Bereich befinden, wird zum Wert 60 addiert, sowie die Stunden um 1 verringert. An dieser Stelle ist zu erwähnen, dass sich die Berechnung je nach Ausgangspunkt unterscheidet. Angenommen, das Sichtfeld des Beobachters ist ein Quadrat, welches in jeweils 4 Quadrate unterteilt werden kann, wobei jedes Quadrat dem Sichtfeld des Objektives entspricht. Im Falle der Anwendung beginnt die Aufnahme im linken oberen Feld und wandert vom Betrachter aus gesehen nach rechts. Aus diesem Grund muss bei der Berechnung des Endresultats die Bogenminuten des Objektives vom Ausgangspunkt subtrahiert werden. Wenn hingegen "unten rechts" begonnen wird, müssen die Werte addiert werden, selbiges gilt für die Deklination.

Listing 4.11: fillTextArea.java

An und für sich ist die Überprüfung aus Listing 4.2 unnötig, da dies nie passieren sollte. Dennoch wurde die Überprüfung der Vollständigkeit halber eingebaut.

Listing 4.12: fillTextArea.java

```
if (rektH <= 0)
rektH += 24;</pre>
```

Eine Überprüfung für den Stundenwert ist ebenfalls vorhanden, wie in Listing 4.2 ersichtlich ist. Hierbei wird überprüft, ob dieser negativ ist oder nicht. Falls dem so ist, wird 24 addiert.

Listing 4.13: fillTextArea.java

```
if (deklD >= 90) {
    deklD = 90 - (deklD - 90);
  }
```

Die Deklination hat den Wert 0 im Äquator. Wie in Abschnitt 2.1 beschrieben, ist die Deklination oberhalb des Äquators positiv mit einem Maximum von 90°. Unterhalb des Äquators ist die Deklination negativ mit einem Maximum von -90°. Der Pol fungiert als Umkehrpunkt. Aus diesem Grund muss eine Überprüfung für Werte größer 90 eingebaut werden, wie oben zu sehen ist. Um den inkorrekten Wert der Deklination zu berichtigen, muss wie folgt vorgegangen werden: Wenn der vorliegende Wert zum Beispiel "95" ist, wird von diesem "90" subtrahiert, was in "5" resultiert. Das Zwischenresultat wird wiederum von 90 subtrahiert, um den korrekten Endwert zu erhalten. In Listing 4.2 befindet sich der zugehörige Programmcode.

Listing 4.14: fillTextArea.java

```
1 else if(deklD <= -90) {
2 deklD = ((deklD + 90) + 90)*-1;
3 }</pre>
```

Wenn der Wert kleiner als -90 ist, muss eine andere Umrechnung angewendet werden. Angenommen, der Ausgangswert ist "-95". Zuerst wird zum Wert 90 addiert, was in "-5" endet. Dieser Wert abermals mit 90 addiert, was in "85" resultiert. Dieser Wert wird abschließend mit "-1" multipliziert, was als Endresultat den korrekten Wert "-85" produziert. Diese Umrechnung wird im Listing 4.2 beschrieben.

Nachdem nun alle nötigen Funktionen zur Befüllung des Textareals abgearbeitet wurden, geht es nun an den Export, was im Listing 4.2 beschrieben wird. Hierbei kommt Code von folgender Quelle zum Einsatz [10].

Listing 4.15: exportText.java

```
final JFileChooser fc = new JFileChooser();
1
          fc.setApproveButtonText("Save");
2
          int actionDialog = fc.showOpenDialog(this);
3
          if (actionDialog != JFileChooser.APPROVE_OPTION) {
4
              return;
5
          }
6
          File file = fc.getSelectedFile();
8
          if(!file.getName().endsWith(".txt")) {
              file = new File(file.getAbsolutePath() + ".txt");
10
          }
11
12
          BufferedWriter outFile = null;
13
          try {
14
              outFile = new BufferedWriter(new FileWriter(file));
15
              textarea.write(outFile);
16
          } catch(IOException ex) {
17
              ex.printStackTrace();
18
          } finally {
19
              if(outFile != null) {
20
```

```
21 try {
22 outFile.close();
23 } catch (IOException e) {}
24 }
25 }
```

exportText ruft ein Speicherfenster auf, welches es dem Anwender erlaubt, den zu exportierenden Text an einem beliebigen Ort zu speichern. Hierbei ist lediglich ein Klick auf den Button "Export" notwendig um einen geeigneten Speicherort auszuwählen, der Rest erfolgt automatisch. Die erstellte Textdatei enthält den gesamten Inhalt des Textareals.

4.3 GUI-Erklärung

Wie bereits in Kapitel 4 erwähnt, war das Ziel eine leichtgewichtige Lösung. In Abbildung 4.3 ist das Hauptfenster der Anwendung zu sehen. Es ist zeitgleich das einzige Fenster.

- Roter Kasten: In diesen Feldern werden die Brennweite, Höhe und Breite des verwendeten Sensors eingegeben.
- Schwarzer Kasten: Diese Felder lassen sich nicht manuell befüllen, sie werden automatisch mit den entsprechendem Resultat aus der Eingabe aus dem Roten Kasten befüllt.
- Grüner Kasten: Diese Felder sind für die Koordinaten des Startpunkts gedacht. Auf Wunsch kann der Anwender auch eine Umrechnung von Stunden/Minuten/Sekunden zu Grad/Bogenminuten/Bogensekunden durchführen, diese wird in die unteren Felder gespeichert.
- Blauer Kasten: Hier wird die Anzahl der horizontalen und vertikalen Bilder festgelegt. Maximalwert ist jeweils 10, sobald ein größerer Wert eingegeben wird, wird dieser durch den Wert 10 ersetzt.
- Lila Kasten: In diesem Textareal werden die entsprechenden Koordinaten ausgegeben.

	-	
Brennweite Höhe des Sensors Breite des Sensors		
Vertikale Bildwinkel in Grad Vertikale Bogenminuten Vertikale Bogensekunden Horizontale Bildwinkel in Grad Horizontale Bogenminuten		
Horizontale Bogensekunden Rektaszension 00 h 00 s Rektaszension • • • • Deklination 00 • 00 •		
Anzahl der Horizontalen Bilder Anzahl der Vertikalen Bilder Löschen Export Koordinaten berechnen		

Abbildung 4.3: Farbliche Hervorhebung der Unterschiedlichen Bereiche

- Oranger Kasten:
 - Löschen: Setzt alle Felder zur
 ück, bef
 üllt diese gegebenenfalls mit 00 und setzt den Fokus auf das Feld "Brennweite".
 - **Export:** Erlaubt es den Anwender den Inhalt des Textareals in ein Textdokument zu exportieren und auszudrucken.
 - Koordinaten berechnen: Nimmt alle Inhalte aus den benötigten Textfeldern und errechnet die Koordinaten für das nächste Bild.

Die Eingabe der notwendigen Informationen beginnt im Feld *Brennweite*, gefolgt von Höhe und Breite des Sensors. Der Horizontale und Vertikale Bildwinkel wird automatisch errechnet. Die Startkoordinaten in Form der Rektaszension und Deklination werden eingetragen, sowie die Anzahl der Bilder. Mit Druck auf *Koordinaten berechnen* werden die entsprechenden Koordinaten im Textareal ausgegeben. Sollte bei der Eingabe ein Fehler unterlaufen sein, so kann der Anwender alle Felder mit Druck auf *Löschen* zurücksetzen. Sollte die Eingabe korrekt sein, ist der Anwender in der Lage, die Koordinaten in einer Textdatei zu exportieren und auszudrucken.

4.4 Vor- Und Nachteile

Vorteile

- Aufgrund der Möglichkeit, die entsprechenden Koordinaten zu exportieren und auszudrucken, ist es nicht nötig einen Laptop bei der Fotografie zu verwenden, wodurch Gewicht und Strom gespart wird. Alternativ können die Koordinaten auch sofort in die Anwendung eingetragen werden.
- Die Anwendung ist ohne größere Vorkenntnisse und Einarbeitung anwendbar, da sie sich auf die absolut notwendigen Bereiche konzentriert. Dementsprechend muss der Anwender lediglich die Felder befüllen und erhält direkt die entsprechenden Koordinaten.
- Durch die Entwicklung mit Java ist die Anwendung portabel und lässt sich auf nahezu jedem Betriebssystem einsetzen. Eine Installation ist nicht notwendig, ebenso werden keine zusätzlichen Erweiterungen benötigt. Einzige Voraussetzung ist eine aktuelle Version von Java auf dem Zielsystem.
- Um EQMOD vollständig zu nutzen, müssen erst einige Programme installiert und konfiguriert werden. Die einzige Voraussetzung für die Eigenentwicklung ist lediglich eine aktuelle Version von Java.

Nachteile

• Im Vergleich zu EQMOD bietet die Anwendung keinerlei weitere Funktionen, welche das Fotografieren vereinfachen oder unter Umständen Fehler beheben können. Ebenso ist der Anwender auf 10 vertikale und horizontale Bilder beschränkt.

4.5 Mögliche Erweiterungsgebiete

Da die Anwendung mit Java entwickelt wurde, ist eine gewisse Modularität von Anfang an vorhanden. Es wurde bewusst auf Module verzichtet, welche nur in Verbindung mit gewissen Betriebssystemen funktionieren, damit eine uneingeschränkte Nutzbarkeit vorhanden ist. Die nachfolgenden Punkte sind mehr Gedankenspiele als konkrete Pläne, weshalb hierbei auch weniger Nachforschung betrieben wurde.

Da EQMOD als eine Art Bindeglied zwischen Computer und Nachführung fungiert, kann dieser genutzt werden, um über die Anwendung die entsprechenden Koordinaten an die Nachführung zu senden. Eine zusätzliche Erweiterungsmöglichkeit wäre zum Beispiel auch die direkte Kontrolle einer angeschlossenen Kamera. Im Falle der Marke *Canon*, welche auch in dieser Ausarbeitung zum Einsatz kam, gestaltet sich dies allerdings etwas schwierig, da *C*# für die Entwicklung von Applikationen für Canon-Kameras verwendet wird. Dies würde wiederum die Anwendung auf eine Plattform limitierten, was also genau gegen das ursprüngliche Ziel gehen würde. Des Weiteren wäre hierfür zum Beispiel ein Notebook nötig, damit die Anwendung ausgeführt werden kann, gleiches gilt für die Kontrolle der Nachführung.

Wie bereits durch die Verwendung von EQMOD und Stellarium Scope ersichtlich geworden ist, verwenden diese Anwendungen Rottöne für die Darstellung der Oberfläche. Dies liegt daran, dass das menschliche Auge sich in der Dunkelheit nicht zu sehr an die Helligkeit von rotem Licht gewöhnt und dadurch die Sensitivität verliert, welche für das Fotografieren in Dunkelheit von Vorteil ist. Das farbliche Anpassen der Anwendung in ein ähnliches Farbschema ist durchaus sinnvoll, allerdings bedarf es hierfür vermutlich grundlegende Änderungen, weshalb dies eine recht geringe Priorität hat. Ebenso war die Anwendung nicht dafür gedacht, vor Ort verwendet zu verwenden, sondern vor Abreise zum Zielort. Alles in allem sind die aufgelisteten Erweiterungen

5 Auswertung

Das nachfolgende Kapitel vergleicht die beiden Anwendungen und geht auf die unterschiedlichen Einsatzmöglichkeiten ein.

5.1 Vergleich der Anwendung

Da beide Anwendungen das Ziel haben, den Anwender bei der Astrofotografie zu unterstützen, ist ein Vergleich sinnvoll und einfach durchzuführen. EQMOD bietet eine Vielzahl an Einstellmöglichkeiten und sonstige Funktionen, welche das Endresultat signifikant verbessern können, aber auch entsprechende Einarbeitungszeit benötigt. Wie bereits in Abschnitt 3.1 beschrieben, funktioniert EQMOD nur in Verbindung mit einer Planetariumssoftware. Diese erlaubt es dem Nutzer, die genaue Position seines Objektives oder Teleskopes zu sehen und entsprechend auf Sterne oder Planeten auszurichten. Bevor EQ-MOD allerdings verwendet werden kann, müssen einige zusätzliche Programme installiert werden. Nachdem die Einrichtung abgeschlossen wurde, muss die Nachführung mit dem Computer verbunden werden. Die Verbindung erfolgt hierbei über ein spezielles Kabel, welches auf der Seite der Nachführung einen RJ45-Anschluss hat und auf der Seite des Computers einen regulären USB-Anschluss. Sobald die Verbindung besteht, kann die Planetariumssoftware gestartet werden. Der Aufbau und die Einrichtung benötigt einige Zeit und Recherche, bietet aber auch entsprechende Vorteile.

Die Eigenentwicklung hingegen bedarf lediglich eine installierte Version von Java und kann auf nahezu jedem Betriebssystem ausgeführt werden. Der Anwender gibt die entsprechenden Daten in die Anwendung ein, berechnet die Koordinaten und trägt diese in die Nachführung ein oder exportiert diese. Sonstige Funktionen sind nicht vorhanden, können aber jedoch nachträglich noch hinzugefügt werden.

5 Auswertung

Beide Anwendungen erfüllen letztlich das gleiche Ziel, unterscheiden sich jedoch in der Handhabung deutlich. EQMOD sowie EQMOSAIC bieten eine Vielzahl an Funktionen, die den Anwender verwirren können, aber auch unterstützen. Die Eigenentwicklung verfügt über wenige Funktionen, was für manch einen Anwender zu wenig sein kann. Letztendlich ist es dem Anwender überlassen, welches Programm verwendet wird. Insbesondere für das Erstellen von Panoramen mag EQMOSAIC die komfortablere Variante sein, sofern man lediglich im eigenen Garten fotografiert. Wenn die Fotografie allerdings in einer ländlichen Gegend stattfindet, ist die Eigenentwicklung unter Umständen die bessere Lösung.

5.2 Endresultat der Bilder

Astrofotografie lässt sich nur durchführen, wenn klares Wetter herrscht und der Mond wenig bis kaum sichtbar ist. Da keine der beiden Voraussetzungen erfüllt war, konnte zum Zeitpunkt der Ausarbeitung leider kein Test durchgeführt werden.

6 Fazit

Während des Studiums hatten Projekte immer recht feste Vorgaben. So wurde zum Teil das Thema genau vorgegeben, inklusive der Aufgaben, die das Programm erfüllen müssen. Ebenso war es möglich, ein eigenes Programm zu entwickeln, dass gewisse Rahmenbedingungen erfüllen musste, wie zum Beispiel die Anbindung einer Datenbank. Die Bachelorarbeit markiert hier einen kompletten Gegensatz, da das Thema frei ausgewählt werden konnte. Ebenso war es möglich, das Programm frei nach belieben aufzubauen und mit Funktionen auszustatten. Aufgrund der Freiheiten bei der Entwicklung konnten agile Methoden verwendet werden, um das Programm je nach Fortschritt und Arbeitsaufwand entsprechend anzupassen. Aus diesem Grund wurde zum Beispiel die anfänglich geplante Kameraansteuerung gestrichen, da diese zu viel Zeit und Arbeit in Anspruch nehmen würde.

Aufgrund der klaren Struktur und Auslegung der Anforderungen für die Anwendung, traten keinerlei nennenswerte Probleme während der Entwicklung auf. An dieser Stelle ist allerdings zu erwähnen, dass Kapitel 2 der interessanteste Abschnitt war. Das Auseinandersetzen mit einem fachübergreifendem Themengebiet war ein interessanter Blick über den eigenen Tellerrand, auch wenn das Thema schon bekannt war. Durch das Aufarbeiten der entsprechenden Informationen wurde allerdings ein tieferes Wissen über diesen Themenbereich vermittelt, ebenso wurde das eigene Repertoire bezüglich Astrofotografie um sinnvolle Punkte erweitert.

Das Kennenlernen von EQMOD und insbesondere EQMOSAIC hat hierbei ebenfalls viele neue Möglichkeiten für etwaige Projekte im Bereich der Astrofotografie eröffnet. So lassen sich zum Beispiel mit EQMOSAIC hochauflösende Aufnahmen des Mondes erstellen. Ebenso erleichtert EQMOD das Aufbauen und Einrichten der Nachführung.

6 Fazit

Letztendlich lässt sich sagen, dass diese wissenschaftliche Arbeit einen guten Abschluss des Studiums darstellt. Unterschiedliche Aspekte, welche im Studium gelehrt wurden, wurden zusammen eingesetzt und erlaubten einen reibungslosen Ablauf. Hierbei war zum Beispiel das interdisziplinäre Arbeiten, Programmiertechniken und Kenntnisse aus dem Software Engineering von Vorteil. Auch das abschließende Dokumentieren der Anwendung war ein Bereich, welcher nicht vernachlässigt werden darf. Auch wenn EQMOD beziehungsweise EQMOSAIC letztlich die attraktivere Anwendung ist, so war die selbstständige Entwicklung eines "Konkurrenzprodukts" ein interessantes Unterfangen, welches einen weiteren Ausblick auf zukünftige Aufgaben gegeben hat.

An dieser Stelle muss nochmals erwähnt werden, dass es sich bei der Eigenentwicklung lediglich um einen Prototypen handelt, welcher langfristig kontinuierlich weiterentwickelt wird. Aufgrund des knapp bemessenen Zeitfensters für die Ausarbeitung und Entwicklung war ein ausführlicher Test der Anwendung nicht möglich, weshalb unbekannte Fehler vorhanden sein können.

Literaturverzeichnis

- Villard, Ray: Hubble's High-Definition Panoramic View of the Andromeda Galaxy URL: http://www.nasa.gov/content/goddard/hubble-s-high-definition-panoramicview-of-the-andromeda-galaxy [Stand: 19.4.2016]
- [2] A. Weigert, H.J. Wendker, L. Wisotzki: Astronomie und Astrophysik Ein Grundkurs. Wiley-VCH, 2005
- [3] Wisotzki, L. "Astronomie und Astrophysik" URL: http://bulk.aip.de/public/WeWeWi/index.html, Kapitel 1 Abbildung 1.6
 [Stand: 19.4.2016]
- [4] Wisotzki, L. "Astronomie und Astrophysik" URL: http://bulk.aip.de/public/WeWeWi/index.html, Kapitel 1 Abbildung 1.7
 [Stand: 19.4.2016]
- [5] Wisotzki, L. "Astronomie und Astrophysik" URL: http://bulk.aip.de/public/WeWeWi/index.html, Kapitel 1 Abbildung 1.8 [Stand: 23.4.2016]
- [6] Brockhaus: Der große Brockhaus, 18. Auflage
- [7] Seip, Stefan: Himmelsfotografie. Kosmos, 2014
- [8] Quick Start Guide http://eq-mod.sourceforge.net/tutindex.html [Stand: 15.5.2016]
- [9] http://stackoverflow.com/questions/14174776/how-to-auto-calculate-inputnumeric-values-of-text-field-in-java [Stand: 15.05.2016]
- [10] http://stackoverflow.com/questions/9690686/save-a-the-text-from-a-jtextarea-iesave-as-into-a-new-txt-file [Stand: 22.05.2016]

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Ort, Datum

Unterschrift