

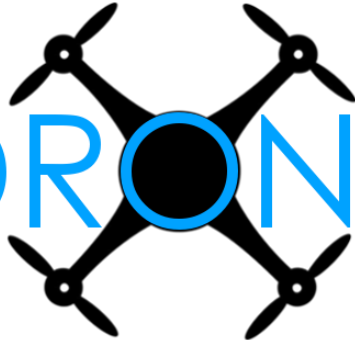


HTL - Perg

Höhere Abteilung für Informatik

Diplomarbeit

AERODRONE

A black silhouette of a quadcopter drone with four propellers, positioned behind the letter 'O' in the word 'AERODRONE'.

Projektteam: Michael Kirchhofer
Nicklas Harringer
Tobias Kapplmüller

Projektbetreuer: Prof. Dipl.-Ing. Christian Aberger

In Zusammenarbeit mit JMK Solar KG
Betreuer Herr Martin Kirchhofer, MSc

Bearbeitungszeitraum: 01.10.2016 – 02.04.2017

Eidesstattliche Erklärung

Hiermit versichern wir, die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der von uns angegebenen Quellen angefertigt zu haben. Alle Stellen, die wörtlich oder sinngemäß aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Perg, _____ Unterschrift _____
(Michael Kirchhofer)

Perg, _____ Unterschrift _____
(Nicklas Harringer)

Perg, _____ Unterschrift _____
(Tobias Kapplmüller)

Gender-Erklärung

Um die Lesbarkeit dieser Diplomarbeit zu verbessern, werden personenbezogene Bezeichnungen, welche sowohl auf Frauen als auch auf Männer zutreffen, prinzipiell nur in der, in der deutschen Sprache üblichen, maskulinen Form verwendet.

Das soll jedoch auf keinen Fall als Diskriminierung des weiblichen Geschlechts oder als Verletzung der Gleichberechtigung von Mann und Frau angesehen werden.

Perg, _____ Unterschrift _____
(Michael Kirchhofer)

Perg, _____ Unterschrift _____
(Nicklas Harringer)

Perg, _____ Unterschrift _____
(Tobias Kapplmüller)

Danksagung

Unser Dank gilt all jenen Personen, die uns im Laufe unserer Diplomarbeit mit Rat und Tat zur Seite gestanden sind.

Vor allem möchten wir uns bei unserem Auftraggeber, der JMK Solar KG und unserer Ansprechperson Martin Kirchhofer, MSc bedanken. Durch ihn wurde die Arbeit möglich gemacht. Er hat uns im Verlauf unserer Entwicklungsarbeit vor allem im Gebiet der theoretischen Grundlagen von Photovoltaikanlagen aufgeklärt und unterstützt.

Ein weiterer, sehr herzlicher Dank gilt Herrn Dipl.-Ing. Christian Aberger, der uns als Betreuungslehrer immer für Rückfragen und Problemhilfe zur Verfügung gestanden ist und auch selbst viel Begeisterung für das Projekt zeigte.

Impressum

Schule

- HTBLA Perg, Höhere Abteilung für Informatik
- Machlandstraße 48
- 4320 Perg

Schuljahr

- 2016/17

Klasse

- 5AHIF

Projekttitlel

- AERODRONE
- Software zur Auswertung von Wärmebildern von Photovoltaikanlagen

Projektteam

- Michael Kirchhofer
- Nicklas Harringer
- Tobias Kapplmüller

Betreuungslehrer

- Prof. Dipl.-Ing. Christian Aberger

Auftraggeber

- JMK Solar KG
- Martin Kirchhofer, MSc

Inhaltsverzeichnis

Eidesstattliche Erklärung	2
Gender-Erklärung.....	3
Danksagung.....	4
Impressum	5
Inhaltsverzeichnis.....	6
1) Kurzbeschreibung	9
2) Abstract	10
3) Projektumfeld	11
3.1) Team	12
3.1.1) Michael Kirchhofer	13
3.1.2) Nicklas Harringer	14
3.1.3) Tobias Kapplmüller.....	15
3.2) Betreuungslehrkraft	16
3.2.1) Professor Dipl.-Ing. Christian Aberger.....	16
3.3) Ausbildungseinrichtung	16
3.3.1) HTL Perg, Höhere Abteilung für Informatik	16
3.4) Auftraggeber.....	17
3.4.1) JMK Solar KG	17
3.4.2) Martin Kirchhofer, MSc	19
4) Einleitung.....	20
4.1) Ausgangssituation.....	20
4.2) Aufgabenstellung.....	21
5) Grundlagen	22
5.1) Theoretische Grundlagen	22
5.1.1) Photovoltaik Allgemein	22
5.1.2) Funktionsweise einer Photovoltaikanlage	24
5.1.2.1) Der Photoeffekt	24
5.1.2.2) Innerer Photoeffekt	24
5.1.2.3) Funktionsprinzip einer Solarzelle	26

5.1.2.4)	Vorgänge innerhalb einer Solarzelle bei Bestrahlung	27
5.1.2.5)	Aufbau einer Solarzelle.....	28
5.1.2.6)	Reihenschaltung von Solarzellen.....	29
5.2)	Entwicklungssysteme	31
5.2.1)	Visual Studio 2015.....	31
5.2.2)	Notepad++.....	32
5.2.3)	Atlassian Bitbucket	33
5.2.4)	SourceTree	33
5.3)	Sonstige verwendete Software	35
5.3.1)	Microsoft Office	35
5.3.1.1)	Microsoft Word	35
5.3.1.2)	Microsoft PowerPoint.....	35
5.3.1.3)	Microsoft Excel	35
5.3.2)	GanttProject	36
5.3.3)	GIMP 2.....	37
5.3.4)	Dropbox.....	37
5.3.5)	Telegram.....	38
5.4)	Verwendete Technologien	39
5.4.1)	.NET	39
5.4.1.1)	.NET Framework	39
5.4.1.2)	.NET Core	39
5.4.1.3)	.NET Standard	39
5.4.2)	C-Sharp	40
5.4.3)	WPF – Windows Presentation Foundation	40
5.4.4)	XAML	41
5.4.5)	EmguCV	42
5.4.5.1)	Cross Plattform	42
5.4.5.2)	Cross Language und Beispielcode	42
5.4.5.3)	OpenCV im Detail	43
6)	Realisierung	44
6.1)	Grundaufbau des Programmes	44

6.1.1)	Liste der importierten Bilder	45
6.1.2)	3D-Darstellung.....	45
6.1.3)	Anzeige der Wärmebildaufnahme	46
6.1.4)	Leistungsdatenanalyse	46
6.2)	Bilder hinzufügen.....	47
6.3)	Bilderkennung & -verarbeitung.....	48
6.3.1)	Erkennen einzelner Module	48
6.3.1.1)	Umwandeln in Schwarz-Weiß-Bild	48
6.3.1.2)	Canny – Algorithmus	49
6.3.1.3)	Erkennung der Rechtecke.....	55
6.3.2)	Positionsberechnung innerhalb der Anlage.....	57
6.3.3)	Überprüfen der Funktionsfähigkeit.....	58
6.3.3.1)	Berechnung der Durchschnittswerte einzelner Module	60
6.3.3.2)	Erkennen des Modulstatus.....	61
6.4)	3D-Darstellung.....	62
7)	Resümee	66
8)	Verteilung der Aufgaben	67
8.1)	Michael Kirchhofer	67
8.2)	Nicklas Harringer	67
8.3)	Tobias Kapplmüller	68
9)	Literaturverzeichnis.....	69
9.1)	Glossar	69
9.2)	Quellen	77
9.3)	Bilder.....	79
10)	Anhang.....	81
10.1)	Datenblatt der verwendeten Solarmodule	81

1) Kurzbeschreibung

Die einzelnen Module einer Photovoltaikanlage sind meistens seriell geschaltet. Aus diesem Grund ist es für Anlagenbetreiber ein großes Problem, wenn eines der PV-Module beschädigt wird, da der Ausfall eines Moduls die gesamte Schaltreihe beeinträchtigt. Handelt es sich bei dem Fehlerauslöser nicht um eine klar erkennbare äußere Beschädigung, so kann die Fehlerquelle nicht mit freiem Auge erkannt werden.

AERODRONE ist eine Software, die es ihrem Anwender ermöglicht, Aufnahmen von Photovoltaikanlagen auszuwerten. Hierbei werden sowohl herkömmliche Fotos als auch Wärmebilder in das Programm geladen und anschließend analysiert.

Zuerst werden die normalen Aufnahmen vom normalen Farbspektrum in Graustufen umgewandelt. Das erlaubt es dem Programm, die rechteckige Form der Module zu erkennen, da sich die Aluminium-Rahmen der Module in einem Graustufenbild als helle Linien hervorheben. Die einzelnen Module werden dann anhand dieser Linien ermittelt und in einer Liste abgespeichert.

Anschließend wird das Wärmebild, das jeweils einer normalen Aufnahme zugeordnet ist, in den HLS-Farbraum umgewandelt, weil hiermit die Auswertung einfacher ist. Gleichzeitig werden die Modulkoordinaten auf das Wärmebild übertragen. Aus dem veränderten Wärmebild wird die Durchschnittstemperatur aller Module berechnet und anschließend wird jedes Modul mit dieser verglichen. Weicht die Wärme eines Moduls stark vom Durchschnittswert ab (entweder sehr viel heißer oder sehr viel kälter), so ist dieses Modul zu hoher Wahrscheinlichkeit beschädigt und somit das Modul, das alle anderen Module der Reihe leistungsmäßig beeinträchtigt.

Alle Module werden nach der Analysephase in einer 3D-Darstellung der Anlage visualisiert. Die PV-Module werden nach Funktionsfähigkeit in verschiedenen Farben gekennzeichnet. Hierbei gibt es Blau für voll funktionsfähige, Orange für beeinträchtigte aber sonst korrekt funktionierende und Rot für beschädigte Module.

2) Abstract

The modules of a photovoltaic power plant are mostly serially connected. Due to this fact, if only one single module is damaged, all the modules which are in the same row produce less power. This can result in a major power loss throughout the whole power plant. If the damaged module has no clearly recognizable physical damage, it is nearly impossible to find out which one is damaged with the naked eye.

AERODRONE is a piece of software that enables its user to analyze pictures of his photovoltaic plants. The program uses regular shots as well as thermal images of all the modules.

The normal photographs are transformed into grayscale pictures because in a black and white version of the pictures, the aluminum frames of the modules are represented as light gray lines, while the module itself is a lot darker. Using those light gray lines, the modules can be detected and are stored in a list.

Every normal picture is paired with a heat scan photo so the program can analyze the thermal image by using the previously detected coordinates of the solar modules. To compute the temperature of modules, the thermal image is converted into a HLS-image because that way it is a lot easier to analyze the thermal image. With the coordinates from the normal photograph, the program computes the average temperature of all the modules in the power plant. Every Module is then compared to this average and those that are either a lot hotter or a lot colder are marked as broken.

After the analysis, the data is used to create a 3D-version of the photographed photovoltaic power plant. In this model, every module is assigned a certain color depending on the status of the module. There is blue for completely working modules that produce full power. There is orange for modules that are technically fine but have reduced output due to the failing ones. The defective Modules are marked red, which means that they are either physically damaged or technically broken.

3) Projektumfeld

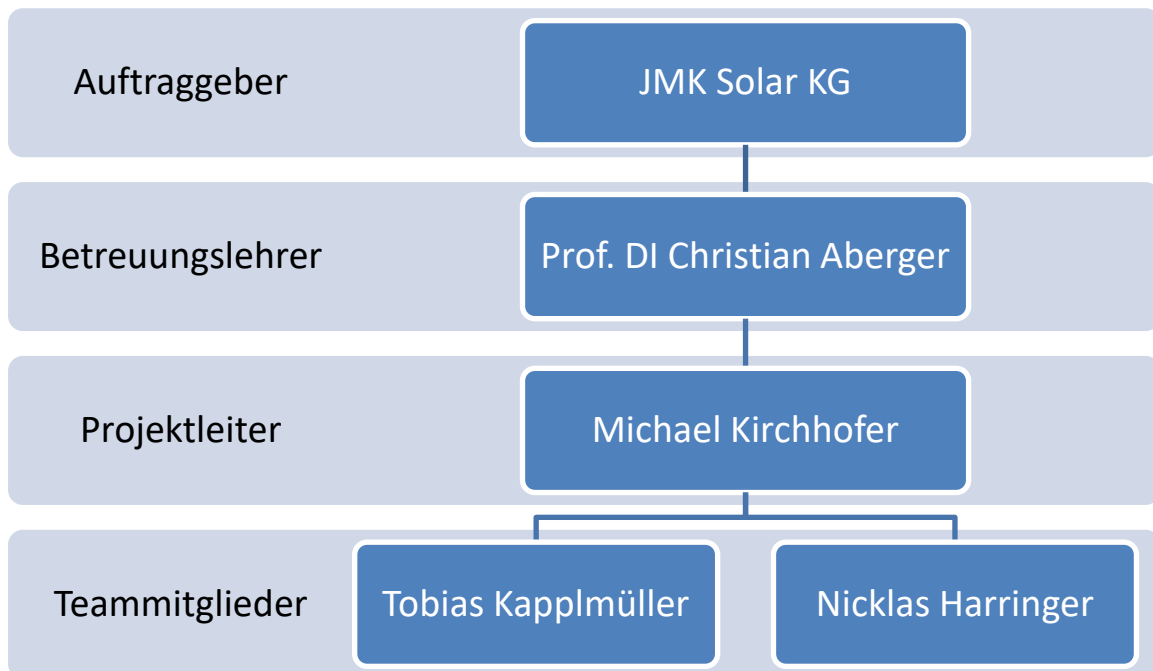


Abbildung 1 Projektumfeld

Bei unserem Auftraggeber bestand aufgrund der in 4.1) beschriebenen Probleme bei der Wartung und Überprüfung von Photovoltaikanlagen bereits seit längerer Zeit der Wunsch nach einem Programm, das diesen Vorgang erleichtert. Da es sich bei dem Komplementär der JMK Solar KG um den Bruder des Projektleiters handelt, wusste das Projektteam von diesem Wunsch und konnte deshalb im Februar 2016 an das Unternehmen herantreten und die genauen Eigenschaften und Funktionen von AERODRONE festlegen.

3.1) Team



Abbildung 2 Projektteam

Die drei Diplomanden sind bereits seit dem Beginn ihrer Ausbildung an der HTL Perg auch außerhalb des Unterrichts sehr gut befreundet und treffen sich regelmäßig in der Freizeit. Da auch in anderen Projekt- bzw. Gruppenarbeiten immer wieder dieselbe Gruppenkonstellation aufzufinden war, hat sich bereits relativ bald abgezeichnet, dass dieses Team auch die Diplomarbeit gemeinsam in Angriff nehmen wird. Natürlich ist es im Projektverlauf mehrere Male zu kleinen oder sogar größeren Meinungsverschiedenheiten gekommen, jedoch konnten diese immer problemlos gelöst werden.

3.1.1) Michael Kirchhofer

Persönliche Daten

Geburtsdatum	28.10.1997
Wohnort	Arbing
Telefon	+43 699 / 17 30 15 49
E-Mail	kirchhofermichael@gmail.com



Abbildung 3 Michael Kirchhofer

Kompetenzen

IT	Java, JavaScript, TypeScript, AngularJS, ExtJS, C#, iOS (Swift), Android, HTML, PHP, CSS, SQL, PL/SQL
Sprachen	Deutsch, Englisch

Schulbildung & Arbeitserfahrungen

2004 - 2008	Volksschule Arbing
2008 - 2012	Europagymnasium Baumgartenberg
Seit 2012	HTL Perg – Höhere Abteilung für Informatik
Sommer 2014	DynaTrace, Linz
Sommer 2015	Energie AG, Linz
Sommer 2016	Energie AG, Linz

3.1.2) Nicklas Harringer

Persönliche Daten

Geburtsdatum	16.06.1998
Wohnort	Schwertberg
Telefon	+43 660 / 299 85 70
E-Mail	harringernicklas@gmail.com



Abbildung 4 Nicklas Harringer

Kompetenzen

IT	Java, JavaScript, TypeScript, AngularJS, ExtJS, C#, iOS (Swift), Android, HTML, PHP, CSS, SQL, PL/SQL
Sprachen	Deutsch, Englisch

Schulbildung & Arbeitserfahrungen

2004 - 2008	Volksschule Schwertberg
2008 - 2012	Hauptschule Schwertberg
Seit 2012	HTL Perg – Höhere Abteilung für Informatik
Sommer 2013	Plasser und Theurer, Linz
Sommer 2014	Plasser und Theurer, Linz
Sommer 2015	Raiffeisen Software GmbH, Linz

3.1.3) Tobias Kapplmüller

Persönliche Daten

Geburtsdatum	15.07.1998
Wohnort	Schwertberg
Telefon	+43 699 / 10 60 46 10
E-Mail	tobi.kapplmueller@gmail.com



Abbildung 5 Tobias Kapplmüller

Kompetenzen

IT	Java, JavaScript, TypeScript, AngularJS, ExtJS, C#, iOS (Swift), Android, HTML, PHP, CSS, SQL, PL/SQL
Sprachen	Deutsch, Englisch

Schulbildung & Arbeitserfahrungen

2004 - 2008	Volksschule Schwertberg
2008 - 2012	Musikhauptschule Saxen
Seit 2012	HTL Perg – Höhere Abteilung für Informatik
Semesterferien 2014	Synthesa Chemie GmbH, Perg
Sommer 2014	SYSco EDV, Schwertberg
Sommer 2015	Schöfer GmbH, Schwertberg
Sommer 2016	Raiffeisen Software GmbH, Linz

3.2) Betreuungslehrkraft

3.2.1) Professor Dipl.-Ing. Christian Aberger



Abbildung 6 Betreuungslehrer
DI Christian Aberger

Herr Professor Dipl.-Ing. Aberger arbeitet bereits seit 2001 an der HTL Perg als Lehrer in den Fächern Netzwerk- und verteilte Systeme (Mobile Computing), Projektentwicklung und Programmieren, wobei er uns in letzterem Fach seit der vierten und in den beiden anderen Fächern seit der fünften Klasse unterrichtet.

Zusätzlich zu seinen Unterrichtsstunden ist er Besitzer und Geschäftsführer der Aberger Software GmbH in Hagenberg. Da er sich in seinem Unternehmen auch vermehrt mit Bilderkennung

und -verarbeitung beschäftigt, zeigte er von Beginn an viel Begeisterung für unsere Diplomarbeit und stand uns bei sämtlichen Fragen stets zur Seite.

3.3) Ausbildungseinrichtung

3.3.1) HTL Perg, Höhere Abteilung für Informatik

Diese Diplomarbeit wurde im Rahmen der standardisierten Reife- und Diplomprüfungen an der HTL Perg im Schuljahr 2016/2017 verfasst. In der Höheren Abteilung für Informatik steht die Ausbildung zum Softwareentwickler im Mittelpunkt. Betriebswirtschaftliche Aspekte werden ebenfalls besonders berücksichtigt.



Abbildung 7 HTL Perg

3.4) Auftraggeber

3.4.1) JMK Solar KG



Abbildung 8 Photovoltaikanlage der JMK-Solar KG

Die JMK-Solar KG wurde im November 2014 gegründet und befasst sich seitdem mit der Erzeugung und dem Verkauf von umweltfreundlicher, erneuerbarer Energie. Dies geschieht durch eine Photovoltaikanlage mit einer maximalen Leistung von ca. 50 Kilowatt.

Wie man in Abbildung 8 sehen kann, besteht die Anlage aus 3 Modulreihen mit jeweils über 60 einzelnen Solarmodulen. Bei diesen Zahlen wird schnell klar, dass eine händische Überprüfung aller Module mit einem hohen Arbeitsaufwand verbunden ist.

Jede dieser drei Modulreihen ist mit einem sogenannten Wechselrichter verbunden. Dieser sorgt dafür, dass der, in den Solarmodulen entstehende Gleichstrom in Wechselstrom umgewandelt wird und später ins Stromversorgungsnetz eingespeist werden kann. Dass alle Module der Reihe seriell geschaltet sind und jeweils an denselben Wechselrichter angeschlossen werden, ist der Hauptgrund dafür, dass bei einem defekten Solarmodul die gesamte Reihe weniger Leistung produziert. Es wäre möglich, jedes Modul mit einem eigenen Wechselrichter zu versehen, womit die



Abbildung 9 Wechselrichter der drei Modulreihen



Abbildung 10 Nahaufnahme eines Wechselrichters

Leistung jedes einzelnen Moduls jederzeit überwacht werden könnte. Diese Lösung ist jedoch mit weit mehr Kosten verbunden, weshalb sie sehr selten angewandt wird.

Diese drei Wechselrichter sind mit einem zentralen Verteilerkasten verbunden. In diesem befinden sich u.a. die Schutzschalter und der Einspeisezähler. Letzterer wird benötigt, damit gemessen werden kann, wie viel Strom von der Anlage ins

öffentliche Stromnetz eingespeist wird und dass der Betreiber der Anlage entsprechend vom Netzbetreiber bezahlt wird.

Nachdem sich die Photovoltaikanlage im Freien befindet, ist das Risiko eines Unwetterschadens bzw. Blitzschlages nicht zu vernachlässigen. Vor allem Blitze können Solarmodulen erhebliche Schäden zufügen. Deshalb verfügt jede Modulreihe



Abbildung 11 Verteilerkasten der Photovoltaikanlage

über einen Blitzableiter, der mögliche Einschläge in das Rahmen- bzw. Befestigungsgestänge ableitet und so die Anlage vor Beschädigungen schützt.



Abbildung 12 Blitzableiter der Solarmodule

3.4.2) Martin Kirchhofer, MSc

Kontakt

JMK Solar KG
Hehenberg 31
4342 Baumgartenberg
martin_kirchhofer@gmx.at



*Abbildung 13 Martin Kirchhofer,
MSc*

Unser Ansprechpartner bei der JMK Solar KG ist gleichzeitig auch der Geschäftsführer des Unternehmens. Während seines Studiums der Ökoenergietechnik an der Fachhochschule Wels konnte er sich ein umfangreiches Wissen über die Technik und die Funktionsweise von Photovoltaikanlagen aneignen, welches er uns immer bereitwillig zur Verfügung stellte.

Neben seiner Tätigkeit als Führungsperson ist er als Mitarbeiter der Forschungs- und Entwicklungsabteilung der ZKW Lichtsystem GmbH in Wieselburg tätig.

4) Einleitung

4.1) Ausgangssituation

Der Auftraggeber von AERODRONE, die JMK Solar KG, betreibt eine 50KW Photovoltaikanlage in schwer begehbarer Hanglage. Diese Leistung wird durch ungefähr 180 einzelne Solarmodule erzeugt und in das öffentliche Stromnetz eingespeist. Sollte nun in einem oder mehreren Modulen ein Fehler auftreten, so muss die Anlage geprüft werden. Bei einer herkömmlichen Funktionsprüfung treten allerdings einige Probleme auf.

- Mit freiem Auge können Modulbeschädigungen meist nicht erkannt werden (Sofern es sich nicht um eine offensichtliche physische Beschädigung handelt)
- Jedes Modul muss einzeln auf seine Funktionsfähigkeit geprüft werden
- Bei nassen Witterungsverhältnissen (Schnee bzw. Regen) tritt eine erhöhte Verletzungsgefahr (durch zum Beispiel Ausrutschen) auf.

Vor dem Einsatz von AERODRONE musste jedes Modul der Anlage einzeln geprüft werden. Dadurch entsteht für eine Anlage dieser Größe ein extrem hoher Aufwand. Die Kosten treten hierbei nicht nur durch im Personalbereich, sondern auch bei der Anlage selbst auf. Ist nur ein Modul beschädigt, so produzieren alle Module in dessen Reihe weniger Leistung. Je länger also die Überprüfung und die Reparatur der Anlage dauert, desto höher sind auch die entstehenden Kosten bzw. der Umsatzverlust für unseren Auftraggeber.

4.2) Aufgabenstellung

Wenn nun also die Photovoltaikanlage aus unbekanntem Gründen weniger Leistung erbringt, als es im Normalzustand der Fall ist und mit freiem Auge keine Beschädigung erkennbar ist, bleibt dem Betreiber noch die Auswertung von Wärmebildern. Diese könnten zwar auch ohne AERODRONE durchgeführt werden, jedoch entsteht hier bei großen Anlagen schnell ein sehr hoher Zeitaufwand, der durch eine automatisierte Analysesoftware stark reduziert werden kann. Aus diesen Gründen beauftragte die JMK Solar KG unser Diplomarbeitsteam damit, ein Programm zu entwickeln, mit dem eine Photovoltaikanlage anhand von Wärmebildaufnahmen auf die Funktionsfähigkeit einzelner Module überprüft werden kann. Ursprünglich war auch vorgesehen, dass das Aufnehmen der Wärmebilder im Projektumfang enthalten ist. Nachdem die Anschaffung einer Drohne, die den Kriterien entspricht und einer Wärmebildkamera, die auf dieser Drohne montiert werden kann sowohl für drei Schüler, als auch für den Auftraggeber ein sehr kostspieliges Unterfangen ist wurde dieser Punkt wieder gestrichen. Es ist also nicht von Bedeutung, wie ein Benutzer von AERODRONE diese Aufnahmen gemacht hat, solange sie im richtigen Winkel geschossen wurden.

5) Grundlagen

5.1) Theoretische Grundlagen

5.1.1) Photovoltaik Allgemein

Der Begriff Photovoltaik setzt sich aus den beiden Wörtern „Photo“ und „Volta“ zusammen. „Photo“ stammt aus dem Griechischen und bedeutet Licht und „Volta“ steht für die Einheit der elektrischen Spannung, die vom italienischen Physiker Graf Alessandro Volta abgeleitet wurde. Photovoltaik bedeutet also die direkte Umleitung von Sonnenlicht in elektrische Energie.

Bereits im Jahre 1839 wurde die Grundlage der Photovoltaik-Technologie, der sogenannte Photoeffekt, durch den französischen Physiker Alexandre Becquerel entdeckt.

Der Bau der ersten Photovoltaikzelle wurde jedoch erst über 100 Jahre später durch die Weiterentwicklung der Halbleitertechnik ermöglicht. Diese wurde im Jahre 1954 in den Bell-Laboratories entwickelt und wies einen Wirkungsgrad von nur 5% auf. Der Verwendungszweck dieser ersten Solarzelle war vor allem für die Energiegewinnung bei Weltraumflügen. Die Herstellungskosten erster PV-Zellen waren exorbitant hoch und für die allgemeine Nutzung somit ungeeignet.

Durch den Einsatz von Silizium-Zellen ließ sich der Wirkungsgrad herkömmlicher Module bereits auf 20% steigern. Doch auch mit anderen Werkstoffen wird geforscht, um die Kosten für PV-Anlagen weiter senken zu können. Inzwischen gibt es manche Länder in denen Solarstrom mit herkömmlichen Energiegewinnungsmethoden (zum Beispiel Kohle, Gas oder Erdöl) konkurriert und diese ersetzt. Vor allem durch die Massenproduktion der Module konnten die Herstellungspreise zunehmend gesenkt werden.

Der große Vorteil bei Sonnenkraftwerken liegt in ihrem modularen Aufbau. Von Zellen im Milliwattbereich für Taschenrechner oder Uhren, bis hin zu Kraftwerken, die mittlerweile sogar schon im Gigawattbereich (Huawei Ningxia Kraftwerk in Yanchi) liegen kann Solarstrom vielseitig eingesetzt werden.



Abbildung 14 Ningxia Kraftwerk, China (Huawei, 2017)

Das Haupteinsatzgebiet von Photovoltaikanlagen liegt in der Erzeugung von elektrischer Energie in größeren Mengen, da die Anlagen ohne Eingriff in der Natur auf Hausdächern und Fassaden installiert werden können. Vor allem Deutschland gilt hierbei als Musterbeispiel, da hier bereits ein Drittel des Strombedarfs durch Solarstrom gedeckt werden können. Die Leistung aller auf deutschem Boden montierter Anlagen übertrifft mittlerweile den Ertrag der Kernkraftwerke bei weitem. Damit hat die Photovoltaik eine wichtige Rolle in der Energieversorgung Deutschlands übernommen.

(Quaschnig, 2017)

5.1.2) Funktionsweise einer Photovoltaikanlage

Die Energiegewinnung in einer PV-Anlage funktioniert durch den Einsatz des sogenannten Photoeffektes.

5.1.2.1) Der Photoeffekt

Durch die Energieeinwirkung eines, auf ein Elektron treffendes Photons wird ebendieses Elektron vom Atomkern losgelöst. Die Energie die für eine vollkommene Abspaltung des Elektrons vom Kern nötig ist wird Ionisationsenergie genannt. Diese Energie ist abhängig von der Wellenlänge der jeweiligen Strahlung die auf die Elektronen trifft. Will man zum Beispiel ein Elektron eines Wasserstoffkerns vom Kern lösen, so müsste man das Atom mit einer Strahlung einer Wellenlänge von unter 90nm (Nanometer) beschießen, was der Wellenlänge der Röntgenstrahlung entspricht. Der Vorgang wird auch als äußerer Photoeffekt bezeichnet.

Bei einer Photovoltaik kommen vor allem die sichtbare, die infrarot und die ultraviolette Strahlung zum Einsatz. Diese weisen eine niedrigere Photonenenergie auf. Da für den äußeren Photoeffekt eine hochenergetische Strahlung benötigt wird, kommt bei einem PV-Modul der innere Photoeffekt zum Einsatz.

5.1.2.2) Innerer Photoeffekt

Die Elektronen einzelner Atome können nur bestimmte Energieniveaus einnehmen. Bei Molekülen, bestehend aus mehreren Atomen, liegen die einzelnen Energieniveaus durch interne Wechselwirkungen dicht beieinander. Die einzelnen Energieniveaus eines Festkörpers mit einer bestimmten Anzahl von Atomen sind untrennbar miteinander verbunden. Die Energieniveaus bilden sogenannte Energiebänder, die jedoch nur eine begrenzte Elektronenzahl aufnehmen können.

In der Wissenschaft werden diese Energiebänder anhand des sogenannten „Bändermodells“ dargestellt.

(Quaschnig, 2017)

Bändermodell

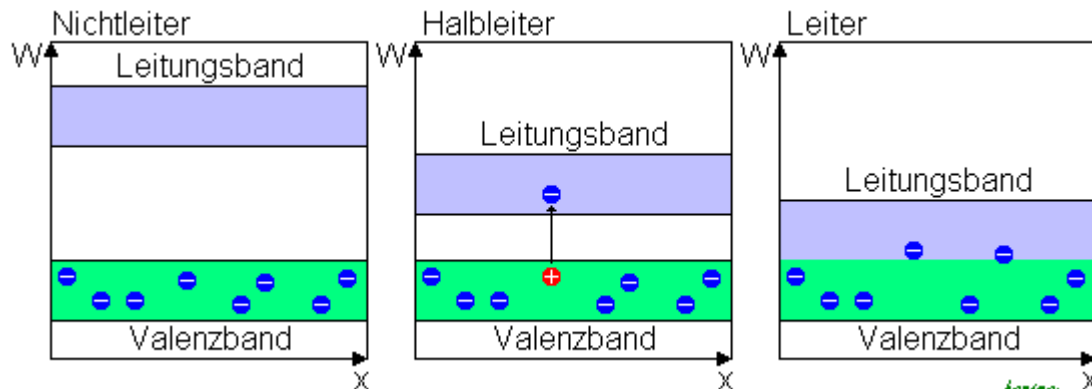


Abbildung 15 Bändermodell (b-landau.de, 2017)

Beim Bändermodell werden alle vorhandenen Bänder der Reihe nach gefüllt. Das letzte vollständig gefüllte Band wird Valenzband (VB) genannt. Das nachfolgende Energieband ist entweder komplett leer oder nur teilweise befüllt und wird als Leitungsband bezeichnet.

Festkörper werden je nach Anordnung und Besetzung ihrer Energiebänder in Nichtleiter/Isolatoren, Halbleiter und Leiter eingeteilt.

Bei Isolatoren ist das Leitungsband unbesetzt und der Bandabstand ist so hoch, dass die Elektronen nur schwer in das Leitungsband angehoben werden können.

Bei Leitern ist das Leitungsband teilweise mit Elektronen gefüllt oder das Leitungs- und das Valenzband überlappen sich. Die Elektronen des Leitungsbandes können sich innerhalb des Festkörpers bewegen und tragen somit zur Elektronenleitung bei. Zu den Leitern zählen vor allem metallische Werkstoffe wie Kupfer oder Eisen.

Bei Halbleitern ist das Leitungsband wie bei den Isolatoren unbesetzt, allerdings ist der Abstand zwischen Leitungs- und Valenzband niedrig genug, dass durch den Einfluss von Strahlung Elektronen in das Leitungsband angehoben werden können. Diese Anhebung durch Strahlung nennt man den **inneren Photoeffekt**.

(Quaschnig, 2017)

5.1.2.3) Funktionsprinzip einer Solarzelle

Eine Photovoltaikzelle besteht aus Halbleitern. In der Herstellung werden meist Halbleiter verwendet, deren Leitungsbänder vier Elektronen enthalten. Zu den Elementarhalbleitern zählen Elemente wie Silizium(Si), Germanium(Ge) oder Zinn(Sn). Verschiedene Element-Verbindungen können ebenfalls Halbleiter Eigenschaften aufweisen.

In der Produktion von PV-Zellen wird meistens das Element Silizium(Si) verwendet. Silizium ist nach Sauerstoff das Element, dass am häufigsten in der Erdkruste vorkommt. Dabei ist es aber in der Regel an andere Elemente gebunden und muss industriell raffiniert werden.



Abbildung 16 Reines Silizium (Topgeo, 2017)

Silizium verfügt über vier Elektronen in der äußeren Schale und bildet daher meist ein Silizium Kristallgitter in dem sich zwei Atome ihre Elektronen „ausborgen“. Im Bändermodell ist das Valenzband voll besetzt und das Leitungsband leer. Durch Licht- oder Wärmestrahlung können allerdings Elektronen dazu bewegt werden, in das Leitungsband überzuwandern. Das Elektron kann sich nun im Kristallgitter frei bewegen.

Die Anzahl der beweglichen Elektronen ist immer von der Temperatur und dem Bandabstand abhängig. Für Silizium gilt: am absoluten Nullpunkt herrscht bei den Elektronen Stillstand, doch mit steigender Temperatur steigt auch die Zahl der frei beweglichen Elektronen. Bei extrem niedrigen Temperaturen nimmt die Leitungsfähigkeit von Silizium, sowie aller Halbleiter, stark ab.

(Quaschning, 2017)

5.1.2.4) Vorgänge innerhalb einer Solarzelle bei Bestrahlung

Wandert ein Elektron vom Valenzband in das Leitungsband, so entsteht im Valenzband ein Defektelektron welches auch „Loch“ genannt wird. Elektronen und Löcher treten dabei meist paarweise auf. Die Anzahl der Elektronen wird mit der Elektronendichte „n“ und die Anzahl der Löcher mit der Löcherdichte „p“ ausgedrückt.

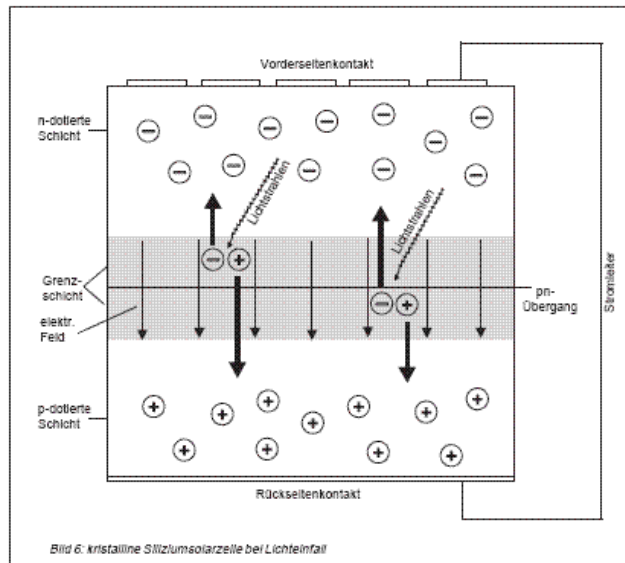


Abbildung 17 Vorgänge innerhalb einer Solarzelle bei Bestrahlung (SFV, 2017)

Grundsätzlich gibt es zwei Arten von Halbleitern. Zum einen die n- Halbleiter, die einen Überschuss an Elektronen aufweisen und zum anderen die p- Halbleiter, die einen Überschuss an Löchern aufweisen. Bringt man nun beide Arten in Kontakt so entsteht ein sogenannter pn-Übergang. Innerhalb dessen wandern die Elektronen des n-Bereichs in das p-Gebiet und die Löcher des p-Gebiets wandern in den n-Bereich.

Durch die Elektronenwanderung entsteht ein elektrisches Feld zwischen den beiden Gebieten. Innerhalb dieses Feldes entsteht eine gewisse Spannung, die sogenannte Diffusionsspannung. Über einen elektrischen Verbraucher lässt sich der Stromkreis schließen und bei solarer Bestrahlung diese „kostenlose“ Form der Energie nutzen.

(Quaschnig, 2017)

5.1.2.5) Aufbau einer Solarzelle

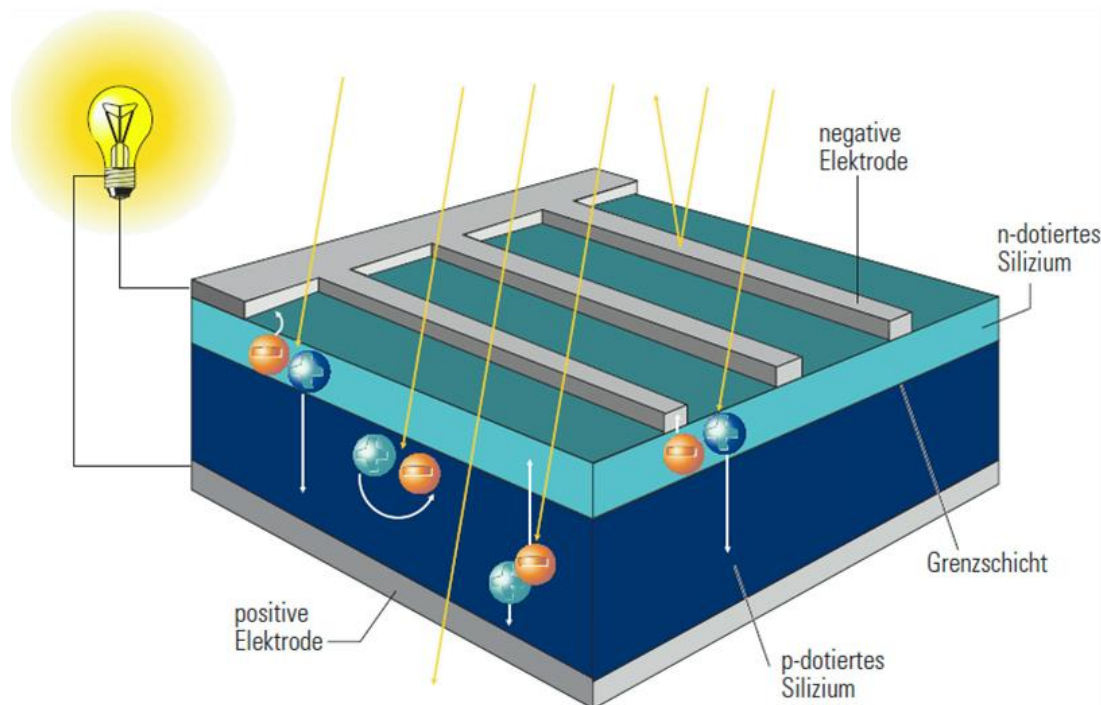


Abbildung 18 Aufbau einer Solarzelle (work-crew.de, 2017)

Mithilfe eines Siebedruckverfahrens wird das Silizium auf die Front- bzw. Rückkontakte gedruckt. Diese bestehen häufig aus Aluminium oder Silber. Dabei wird die Rückseite ganzflächig beschichtet und auf der Vorderseite werden nur dünne Kontaktfinger aufgebracht, um die Solarzellen-Fläche so groß wie möglich ausführen zu können.

Da das metallische Silizium das Sonnenlicht stark reflektieren würde, bringt man eine Antireflexionsschicht auf der Solarzelle auf. Damit kann die Reflektion der Module drastisch gesenkt werden und es entsteht die typisch dunkelblaue Färbung der Solarzellen.

Mittlerweile ist es auch möglich, die Reflexionsschicht in anderen Farben herzustellen, um eine leichtere optische Integration in Gebäude zu ermöglichen.

Man kann die Leistung der PV-Anlage auf verschiedene Art und Weise zusätzlich verbessern. Überzieht man zum Beispiel die Oberfläche der Module (Deckglas) mit mikroskopischen Pyramiden, so kann man die Absorptionsfläche für Sonnenstrahlen vergrößern. Durch das Fräsen kleiner Rillen in die Zelloberfläche lässt sich unter anderem die auftretende Reflexion um einiges verringern.

Durch technische Verbesserungen lassen sich bei neuen PV-Anlagen Wirkungsgrade von zirka 20% erreichen.

(Quaschnig, 2017)

5.1.2.6) Reihenschaltung von Solarzellen

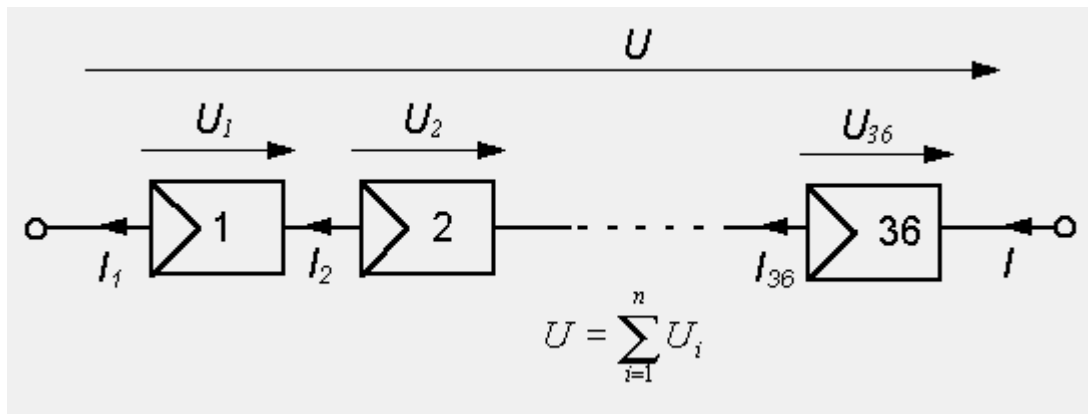


Abbildung 19 Reihenschaltung von Solarzellen (herzo-agenda21.de, 2017)

Aufgrund der niedrigen Spannungen werden Solarzellen grundsätzlich in einem Modul in Reihe geschaltet, anstatt jede Zelle einzeln zu betreiben. Diese Module werden wiederum mit anderen Modulen in Reihe geschaltet.

Durch die Reihenschaltung ist der Strom, der durch alle Module läuft, identisch. Die Summe der Zellspannungen ergibt somit die Modulspannung. Herrschen für alle Module identische Bestrahlungsstärken und Temperaturen so ergibt sich aus der Summe der Modulspannungen die Gesamtspannung der Anlage.

In der Realität sind die Umgebungsbedingungen der Module oftmals unterschiedlich. Durch externe Verschmutzungen, wie Schnee oder Blätter, kann es zu einer Schattenbildung auf den Zellen kommen. Durch Beschädigungen der Zellen treten ähnliche Symptome wie durch eine Abschattung von Modulen auf.

Durch eine Beschädigung oder Abschattung von nur zwei Prozent der Modulfläche lässt sich bereits ein Leistungsverlust von bis zu 70% vermerken. Bei Beschädigungen größerer Art lassen sich exponentiell höherer Verluste erkennen. Steigt der Verlust eines Moduls in eine

gewisse Höhe, so kann es zu einer starken Erwärmung des Moduls und somit zu einer Zerstörung des betroffenen Moduls kommen. Durch die extreme Hitze beginnen die Zellen zu schmelzen oder es kann zu Schäden an den Modulkapseln kommen.

Um einzelne Solarzellen vor thermischer Zerstörung zu schützen, werden parallel zu den Zellen sogenannte „Bypassdioden“ installiert. Bei einer Abschattung oder einer Beschädigung fließt nun der Strom über die Dioden, um die Spannung und somit die Temperaturentwicklung, zu verringern.

Typische Solarmodule bestehen heutzutage aus ca. 60 Solarzellen. Dabei sind entweder alle oder jeweils 20 Solarzellen in Serie geschaltet. Diese beispielsweise 3 mal 20 Solarzellen werden dabei jeweils von einer Bypassdiode vor Beschädigungen durch Überhitzung geschützt. Produzieren 2 der 3 Solarzellenreihen ungehindert Strom und tritt in der dritten Reihe ein Fehler auf, fließt der Strom der anderen beiden Reihen über die Bypassdiode der dritten Reihe. Folglich würde der Bereich am Modul, der die dritte Solarzellenreihe aufnimmt auf einem Infrarot-Bild kälter erscheinen, weil die thermische Wärmeabgabe von den Zellen der dritten Reihe auf deren Bypassdiode verschoben wird.

Bei Solarmodulen ohne Bypassdioden tritt ein anderer Fall auf. Ist beispielsweise nur eine der 60 Solarzellen komplett verschattet oder beschädigt, wird sie vom Stromlieferanten zum Stromverbraucher. Folglich wird die von den übrigen Zellen produzierte elektrische Energie in der beschädigten Zelle in Wärme umgewandelt und erscheint auf einem Infrarot Bild als ein sogenannter Temperatur „Hot Spot“.

(Quaschnig, 2017)

5.2) Entwicklungssysteme

5.2.1) Visual Studio 2015

Microsoft Visual Studio 2015 ist eine kostenpflichtige Entwicklungsumgebung, welche am 20. Juli 2015 auf den Markt kam. Sie wurde von Microsoft für das Betriebssystem



Abbildung 20 Visual Studio Logo

Windows entwickelt und unterstützt unter anderem folgende Programmiersprachen: „Visual Basic“, „.NET“, „C“, „C++“, „C#“, „F#“, „TypeScript“ u.v.m. (Visual Studio, 2017)

Visual Studio wurde für die Realisierung unserer Diplomarbeit verwendet, dabei verwendeten wir die Programmiersprache „C#“.

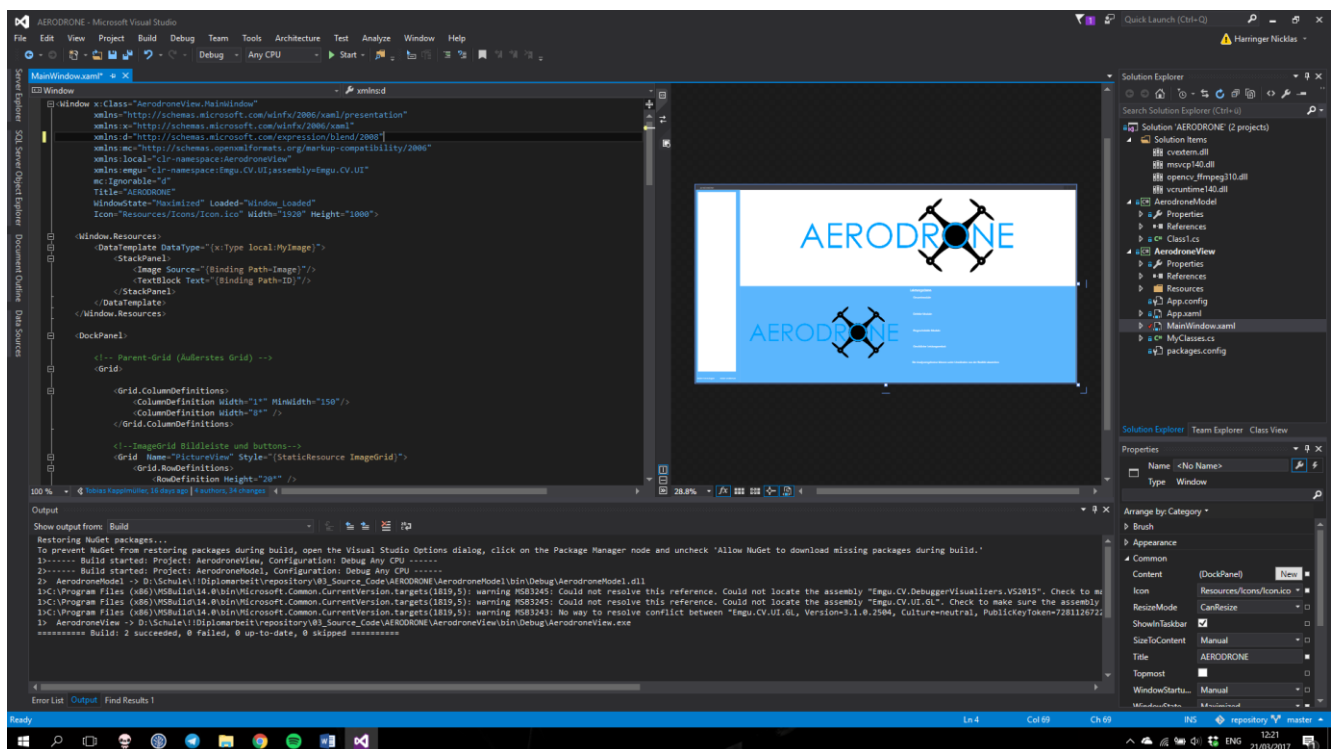


Abbildung 21 Visual Studio 2015

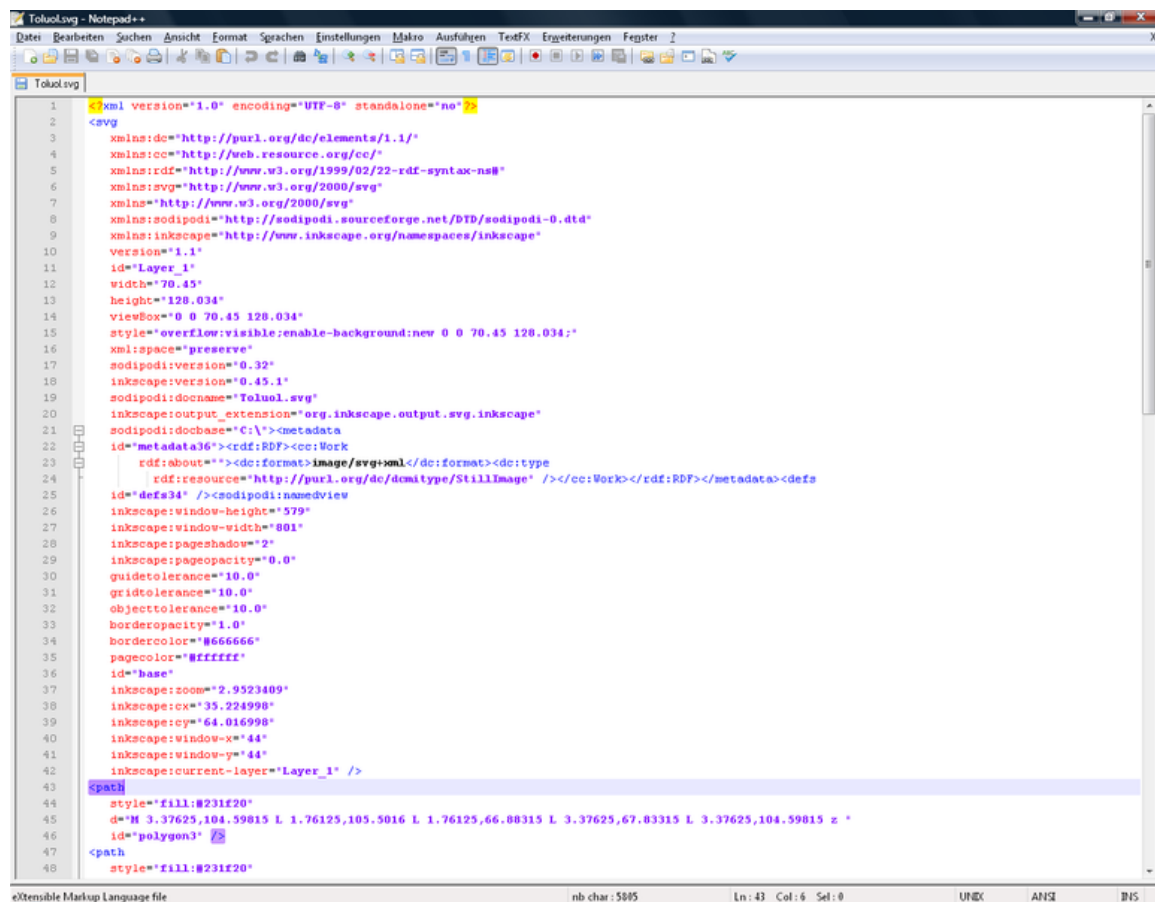
5.2.2) Notepad++

Bei Notepad++ handelt es sich um einen kostenlosen Code-Editor, der auch zum Verfassen einfacher Texte verwendet werden kann. Die Stärke von Notepad++ besteht darin, dass es die meisten Programmiersprachen erkennt und dessen Syntax anzeigt. Durch eine farbliche Kennung der Ausdrücke wird dem Programmierer das Leben zusätzlich erleichtert. (Wikipedia, 2017)



Abbildung 22 Notepad++ Logo

Im Rahmen der Diplomarbeit wurde Notepad++ zur Anzeige von Codebeispielen verwendet, um diese lesbar darzustellen.



```

1  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2  <svg
3      xmlns:dc="http://purl.org/dc/elements/1.1/"
4      xmlns:cc="http://web.resource.org/cc/"
5      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
6      xmlns:svg="http://www.w3.org/2000/svg"
7      xmlns="http://www.w3.org/2000/svg"
8      xmlns:sodipodi="http://sodipodi.sourceforge.net/DTD/sodipodi-0.dtd"
9      xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape"
10     version="1.1"
11     id="Layer_1"
12     width="70.45"
13     height="128.034"
14     viewBox="0 0 70.45 128.034"
15     style="overflow:visible;enable-background:new 0 0 70.45 128.034;"
16     xml:space="preserve"
17     sodipodi:version="0.32"
18     inkscape:version="0.45.1"
19     sodipodi:docname="Toluol.svg"
20     inkscape:output_extension="org.inkscape.output.svg.inkscape"
21     sodipodi:docbase="C:\\"><metadata
22     id="metadata36"><rdf:RDF><cc:Work
23         rdf:about=""><dc:format>image/svg+xml</dc:format><dc:type
24             rdf:resource="http://purl.org/dc/dcmitype/StillImage" /></cc:Work></rdf:RDF></metadata><defs
25     id="defs34" /><sodipodi:navview
26     inkscape:window-height="579"
27     inkscape:window-width="801"
28     inkscape:page-shadow="2"
29     inkscape:pageopacity="0.0"
30     guidetolerance="10.0"
31     gridtolerance="10.0"
32     objecttolerance="10.0"
33     borderopacity="1.0"
34     bordercolor="#666666"
35     pagecolor="#ffffff"
36     id="base"
37     inkscape:zoom="2.9523409"
38     inkscape:cx="35.224998"
39     inkscape:cy="64.016998"
40     inkscape:window-x="44"
41     inkscape:window-y="44"
42     inkscape:current-layer="Layer_1" />
43 <path
44     style="fill:#231f20"
45     d="M 3.37625,104.59815 L 1.76125,105.5016 L 1.76125,66.80315 L 3.37625,67.83315 L 3.37625,104.59815 z "
46     id="polygon3" />
47 </path
48     style="fill:#231f20
  
```

Abbildung 23 Notepad++

5.2.3) Atlassian Bitbucket

Atlassian Bitbucket ist ein frei verfügbarer Dienst für Softwareentwicklungsprojekte, der die Versionsverwaltungssysteme Git und Mercurial unterstützt. Bitbucket wurde von Atlassian entwickelt und erstmalig 2008 herausgegeben. (Wikipedia, 2017)



Abbildung 24 Atlassian Bitbucket Logo

Im Zuge der Diplomarbeit wurde Bitbucket zur Datenspeicherung beziehungsweise zur Versionsverwaltung mittels Git verwendet.

5.2.4) SourceTree

SourceTree ist eine kostenlose Software zur Verwaltung von Git und Mercurial Repositories. Zusätzlich können alle Git-Operationen durchgeführt werden, wovon die wichtigsten anschließend näher beschrieben werden:



Abbildung 25 Atlassian SourceTree Logo

- Commit
 - Aktuelle Änderungen werden lokal gespeichert
- Clone
 - Ermöglicht das „Klonen“ eines Git-Repository von einem Git-Server, wie zum Bsp. Bitbucket
- Push
 - Gespeicherte Änderungen werden auf den Server hochgeladen
- Pull
 - Änderungen von Teammitgliedern werden vom Server heruntergeladen
- Fetch
 - Abfrage, ob neue Änderungen vorhanden sind

- Branch
 - Ein Branch (zu Deutsch Verzweigung) ist eine Abzweigung vom eigentlichen Produkt
 - Mitarbeiter erstellen für ihre Aufgaben eigene Branches, um unabhängig vom Hauptprodukt zu arbeiten. Wenn diese Aufgabe nun abgeschlossen ist, wird ein Merge durchgeführt
- Merge
 - Wie der Name schon verrät wird bei einem Merge etwas fusioniert
 - In diesem Fall wird ein Branch (Verzweigung) mit dem Hauptprodukt verbunden, d.h. die Änderungen des Branches werden mit dem Hauptstrang vereint.

Bei unserer Diplomarbeit wurde SourceTree zur Verwaltung unseres Atlassian Bitbucket Git Servers verwendet. In der Folgenden Abbildung ist der Arbeitsfluss von Git klar ersichtlich.

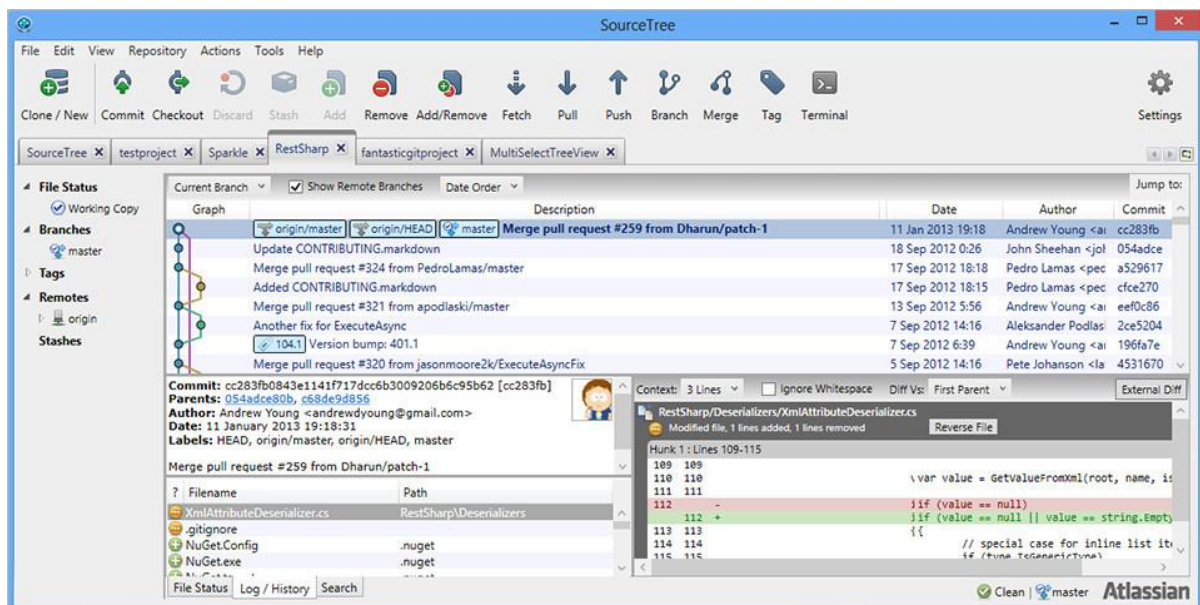


Abbildung 26 SourceTree

5.3) Sonstige verwendete Software

5.3.1) Microsoft Office

Microsoft Office stellt eine Palette verschiedenster Softwareprodukte zur Verfügung, die den Büroalltag vereinfachen. Die folgenden Office Produkte wurden im Rahmen der Diplomarbeit verwendet:



Abbildung 27 Microsoft Office Logo

5.3.1.1) Microsoft Word

- Dokumentation
- Protokolle, Anträge



Abbildung 28 Microsoft Word Logo

5.3.1.2) Microsoft PowerPoint

- Projektpräsentationen
- P@bs Project Award
- Kick-Off



Abbildung 29 Microsoft PowerPoint Logo

5.3.1.3) Microsoft Excel

- Risikomanagement
- Tabellenkalkulation
- SCRUM-Prozess



Abbildung 30 Microsoft Excel Logo

5.3.2) GanttProject

GanttProject ist eine Open-Source Anwendung für die Projektplanung, die wurde in der Programmiersprache Java geschrieben wurde. Die Hauptfunktionalitäten beinhalten unter anderem Gantt-Diagramme, Ressourcenmanagement, Erstellung von Berichten, sowie den Export/Import von HTML und PDF. (Wikipedia, 2017)



Abbildung 31 GanttProject Logo

GanttProject wurde im Zuge der Diplomarbeit zur Terminplanung des SCRUM Prozesses verwendet.

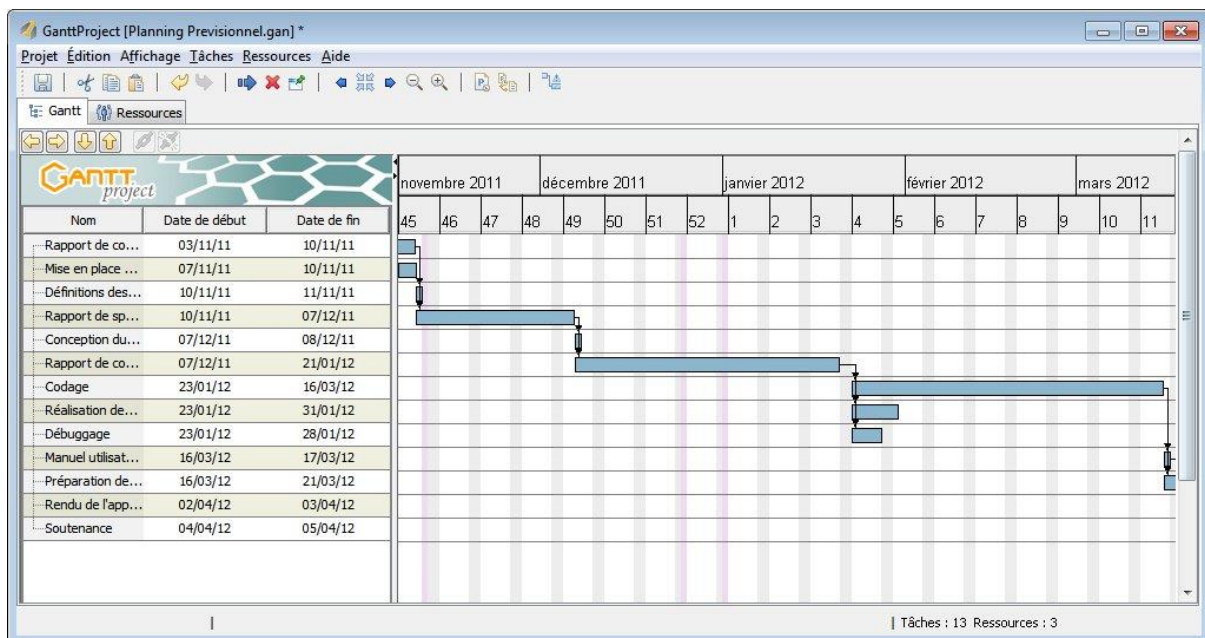


Abbildung 32 GanttProject

5.3.3) GIMP 2

GIMP 2 ist eine kostenlose Bildbearbeitungssoftware. Dabei sind sämtliche Tools und Funktionen enthalten, die zur Bildbearbeitung benötigt werden.

Im Rahmen der Diplomarbeit wurde GIMP 2 zum Erstellen und bearbeiten des Logos und anderen Bilddateien verwendet. Da dem Projektteam in der Anfangsphase des Projekts aus Kostengründen noch keine Wärmebildaufnahmen zur Verfügung standen, wurden zu Testzwecken „gefälschte“ Wärmebilder erstellt



Abbildung 33 GIMP 2 Logo

5.3.4) Dropbox

Dropbox speichert Daten in einer sogenannten Cloud, also online. Das bedeutet, dass alle Dateien, die auf Dropbox hochgeladen werden, auf jedem internetfähigen Gerät verfügbar sind. Auf diese Dateien wird entweder über den Browser zugegriffen oder mithilfe von Programmen oder Apps. (Wikipedia, 2017)

Im Zuge der Diplomarbeit wurde Dropbox verwendet, damit auch der Betreuungslehrer Zugriff auf alle Projektdokumente hat.



Abbildung 34 Dropbox Logo

5.3.5) Telegram

Telegram ist ein frei verwendbarer Instant-Messenger, sozusagen ein Chat-Programm für Smartphones, Tablets und PCs.

Benutzer können Textnachrichten, verschlüsselte Nachrichten, Fotos, Videos und Dokumente austauschen. Die Software wurde von Telegram Messenger LLP entwickelt und erschien im Jahr 2013 (Telegram, 2017)



Abbildung 35 Telegram Logo

Im Rahmen der Diplomarbeit wurde Telegram zum Austausch von Dokumenten, Nachrichten und Fotos verwendet.

5.4) Verwendete Technologien

5.4.1) .NET

.NET ist eine Plattform von Microsoft zur Entwicklung und Ausführung von Programmen. Die .Net Plattform wurde erstmalig 2002 als Teil von Visual Studio veröffentlicht. (Wikipedia, 2017)



Abbildung 36 Microsoft .NET Logo

5.4.1.1) .NET Framework

Das .NET Framework ist ein Teil der Microsoft Plattform .NET und besteht aus einer Laufzeitumgebung, einer Sammlung von Klassenbibliotheken, Programmierschnittstellen und Dienstprogrammen, wie etwa Services. Eine dieser Klassenbibliotheken steckt hinter dem Konzept der „Portable Class Libraries“, die es dem User erlauben .NET Code auf Silverlight, Windows Phone oder Xbox360 zu portieren. Wie es für Microsoft Produkte üblich ist, ist das .NET Framework nur unter dem Betriebssystem Windows verfügbar. (Wikipedia, 2017)

5.4.1.2) .NET Core

.NET Core ist eine freie und quelloffene Software-Plattform innerhalb der .NET-Plattform die ebenfalls zur Entwicklung und Ausführung von Anwendungsprogrammen dient. Gegensätzlich zum .NET Framework ist .NET Core plattformunabhängig, das heißt es ist nicht nur unter Windows, sondern auch unter Linux und OS X verfügbar. (Wikipedia, 2017)

5.4.1.3) .NET Standard

Der .NET Standard definiert die Spezifikation von .NET-APIs. Dies führt zu einem umfassenden Einheitlichen .NET-Ökosystem. (Microsoft, 2017)

5.4.2) C-Sharp

C-Sharp (C#) ist eine objektorientierte Programmiersprache für die .NET Plattform. Sie schlägt ihre Wurzeln in der C Familie und wurde mit dem .NET Framework eingeführt. Sie ist ähnlich zu C++ und Java.

C-Sharp vereinfacht die Portierung von Programmcodes auf die verschiedensten Betriebssysteme, zusätzlich können in C# erstellte Anwendungen mit Programmen verknüpft werden, die auf anderen Programmiersprachen basieren. Zusammenfassend unterstützt C# die folgenden Punkte:

- Skalierung von Anwendungen
- Datentypsicherheit
- Einfache Typendeklaration
- Automatische Speicheroptimierung



Abbildung 37 C# Logo

Die Programmiersprache C-Sharp wird in dieser Diplomarbeit verwendet um die Analyse der Wärmebilder durchzuführen, dies geschieht mittels Frameworks, so wie WPF, (siehe 5.4.3), und EmguCV, (siehe 5.4.5).

5.4.3) WPF – Windows Presentation Foundation

Das Framework Windows Presentation Foundation, kurz WPF, ermöglicht es den Programmierern, Desktop-Clientanwendungen mit Benutzeroberflächen zu kreieren. Es vereint die Vorteile von Adobe Flash, HTML, CSS und Windows Forms. Als Bestandteil des .NET Frameworks von Microsoft stellt WPF ein umfangreiches Modell für den Programmierer bereit. Durch die Verwendung der Auszeichnungssprache XAML (siehe 5.4.4) werden automatisch die Präsentation und die Geschäftslogik getrennt. (Windows Presentation Foundation, 2017)

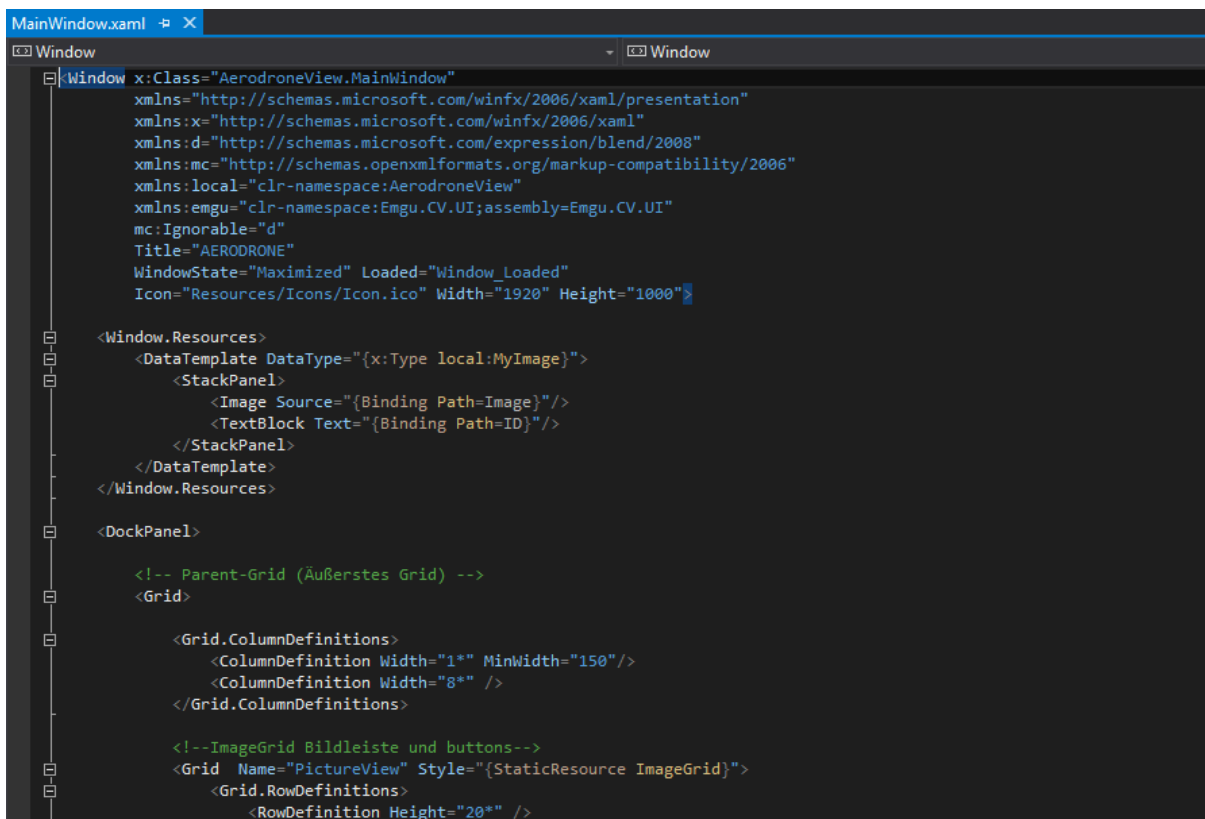
Im Rahmen der Diplomarbeit wurde das WPF Framework zur Visualisierung unserer Anwendung und der 3D-Darstellung der Anlagen verwendet.

5.4.4) XAML

XAML oder „Extensible Application Markup Language“ wurde von Microsoft entwickelt. Es handelt sich hierbei um eine sogenannte Beschreibungssprache zur Gestaltung von Grafischen Oberflächen. XAML ist ebenfalls im .NET Framework enthalten und wurde zum ersten Mal in der .NET Version 3.0 veröffentlicht.

XAML wird verwendet, um grafische Elemente, Benutzeroberflächen, Verhaltensweisen, Animationen, Transformationen, Darstellung von Farbverläufen, Abspielen von Mediadateien und um vieles mehr zu definieren. (Wikipedia, 2017)

Im Zuge der Diplomarbeit wurde XAML zur grafischen Gestaltung der Benutzeroberfläche verwendet.



```
MainWindow.xaml
Window
Window x:Class="AerodroneView.MainWindow"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:AerodroneView"
xmlns:emgu="clr-namespace:Emgu.CV.UI;assembly=Emgu.CV.UI"
mc:Ignorable="d"
Title="AERODRONE"
WindowState="Maximized" Loaded="Window_Loaded"
Icon="Resources/Icons/Icon.ico" Width="1920" Height="1000"

<Window.Resources>
  <DataTemplate DataType="{x:Type local:MyImage}">
    <StackPanel>
      <Image Source="{Binding Path=Image}"/>
      <TextBlock Text="{Binding Path=ID}"/>
    </StackPanel>
  </DataTemplate>
</Window.Resources>

<DockPanel>

  <!-- Parent-Grid (Äußerstes Grid) -->
  <Grid>

    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="1*" MinWidth="150"/>
      <ColumnDefinition Width="8*" />
    </Grid.ColumnDefinitions>

    <!-- ImageGrid Bildleiste und buttons-->
    <Grid Name="PictureView" Style="{StaticResource ImageGrid}">
      <Grid.RowDefinitions>
        <RowDefinition Height="20*" />
      </Grid.RowDefinitions>
    </Grid>
  </Grid>
</DockPanel>
```

Abbildung 38 XAML

5.4.5) EmguCV

EmguCV ist ermöglicht es dem Programmierer, die Bibliothek OpenCV in Programmiersprachen wie zum Bsp. C#, VB oder VC++ zu verwenden. Es ist handelt sich hierbei lediglich um einen Wrapper der die Funktionen von OpenCV auf allen anderen Plattformen bereitstellt. EmguCV wurde bei unserer Diplomarbeit verwendet, da AERODRONE in C# geschrieben ist, aber diese Sprache von OpenCV nicht unterstützt wird. Darüber hinaus bringt EmguCV auch noch einige andere Vorteile mit sich:

5.4.5.1) Cross Plattform

EmguCV wurde komplett in C# geschrieben. Das bedeutet es kann unter allen Plattformen, die den C# Compiler unterstützen, kompiliert werden. Beispiele dafür sind:

- Windows
- Linux
- MacOS X
- iOS
- Android
- Windows Phone

5.4.5.2) Cross Language und Beispielcode

EmguCV kann für verschiedenste Programmiersprachen verwendet werden, wie zum Beispiel:

- C#
- VB.NET
- C++
- IronPython

Die Wiki Seite von EmguCV stellt für all diese Sprachen Beispielcodes zur Verfügung (<http://www.emgu.com/wiki/index.php/Tutorial#Examples>) (EmguCV, 2017)

5.4.5.3) OpenCV im Detail

OpenCV (CV steht für „Computer Vision“) ist eine frei verwendbare Programmiersbibliothek die Algorithmen für Bildverarbeitung und maschinelles Sehen zur Verfügung stellt. Sie ist wie schon erwähnt nur für C und C++ geschrieben.

Die Stärke von OpenCV liegt in ihrer Geschwindigkeit und ihrer großen Ansammlung an Algorithmen.

Die Bibliothek umfasst Algorithmen für die Gesichtserkennung, 3D – Funktionalitäten, Haar-Klassifikation, verschiedene Filter (wie z.B. Canny) und Funktionen für die Kamerakalibrierung. (Wikipedia, 2017)



Abbildung 39 OpenCV Logo

Im Rahmen der Diplomarbeit wurde EmguCV zur Bildverarbeitung und zur Analyse beziehungsweise zur Analyse der Wärmebilder verwendet.

6) Realisierung

6.1) Grundaufbau des Programmes



Abbildung 40 Grundaufbau des AERODRONE-Fensters

Die grafische Oberfläche von AERODRONE lässt sich in 4 Bereiche unterteilen:

- Liste der importierten Bilder
- 3D-Darstellung
- Anzeige der Wärmebildaufnahme
- Leistungsdatenanalyse

6.1.1) Liste der importierten Bilder

Wie in diesem Dokument bereits einige Male beschrieben worden ist, werden in AERODRONE immer eine normale Aufnahme der Anlage und eine Wärmebildaufnahme, die zum gleichen Zeitpunkt vom selben Punkt aus fotografiert wurden miteinander verbunden und gemeinsam gespeichert.

Am linken Rand des Programmes wird eine Liste aller importierten Bilder angezeigt, dafür wird jeweils das normale Foto der Anlage verwendet. Wenn ein Bild aus der Liste ausgewählt wird, wird mit den beiden Aufnahmen die Analyse durchgeführt und die Ergebnisse werden angezeigt. Am unteren Rand dieser Liste befinden sich noch zwei Buttons, mit denen Bilder hinzugefügt oder entfernt werden können.



Abbildung 41 Liste der ausgewählten Bilder

6.1.2) 3D-Darstellung

Der Hauptteil der grafischen Oberfläche wird in zwei Hälften unterteilt, wovon in dem oberen dieser zwei Teile die dreidimensionale Darstellung der Photovoltaikanlage angezeigt wird. Dazu wird jedes erkannte

Solarmodul als Quadrat dargestellt, wobei die Farbe des Quadrats vom Status des jeweiligen

Moduls abhängig ist. Defekte

Module werden rot und

funktionsfähige blau

angezeigt. Alle Module, die

zwar technisch einwandfrei

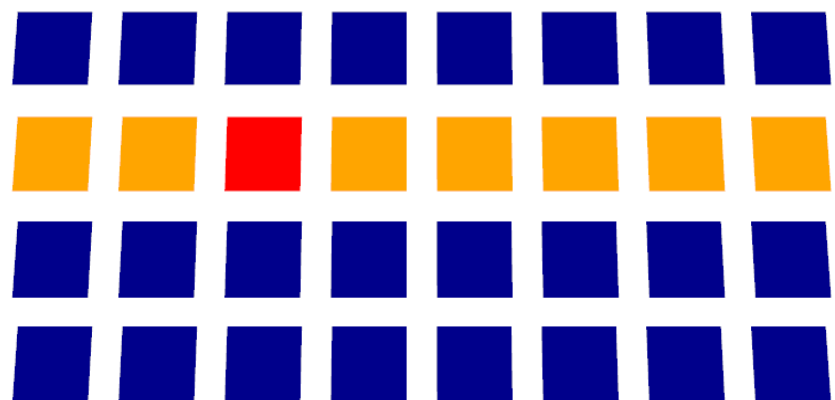
funktionieren, aber durch

ein defektes Modul

innerhalb der Reihe

beschränkt sind werden

orange eingefärbt.



Zurücksetzen

Abbildung 42 3D-Darstellung der Anlage

6.1.3) Anzeige der Wärmebildaufnahme

Die untere Hälfte des Fensters wird wieder in zwei gleich große Teile aufgeteilt. In der linken Hälfte wird die Wärmebildaufnahme der Anlage angezeigt. Dadurch kann sich der Benutzer des Programmes auch in dem seltenen Falle eines Fehlers bei der Bildauswertung einen Überblick darüber verschaffen, welche Module beschädigt sind und welche nicht.

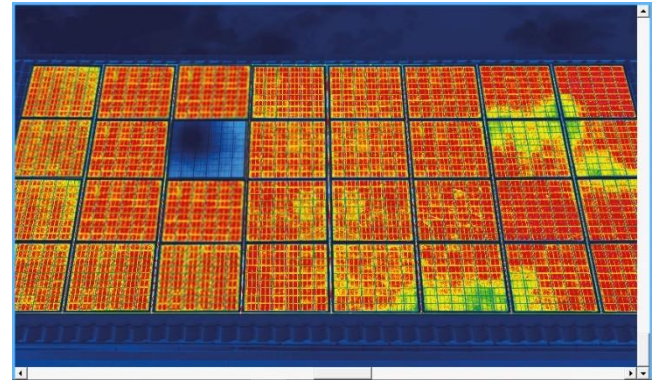


Abbildung 43 Wärmebildaufnahme der Solaranlage

6.1.4) Leistungsdatenanalyse

Im letzten, noch nicht beschriebenen Teil der Benutzeroberfläche werden nach der Auswertung der Bilder noch einige statistische Daten über die Anlage erhoben und angezeigt. So erhält der Anwender hier in schriftlicher Form Informationen darüber, wie viele Module die Anlage hat, wie viele davon beschädigt sind und auch wie viele funktionsfähige Module durch ein beschädigtes eingeschränkt werden. Zusätzlich wird noch der geschätzte Leistungsverlust in Prozent angezeigt.

Leistungsdaten	
Gesamtmodule:	32 Module
Defekte Module:	1 Modul
Eingeschränkte Module:	7 Module
Geschätzter Leistungsverlust:	25 %

Die Analyseergebnisse können unter Umständen von der Realität abweichen

Abbildung 44 Anzeige zusätzlicher Daten

6.2) Bilder hinzufügen

Für den Analysevorgang ist es nötig, dass immer ein Wärmebild gemeinsam mit einer normalen Aufnahme der Anlage zum gleichen Zeitpunkt und vom selben Standort aus fotografiert wird. So kann mit dem herkömmlichen Foto die Modulerkennung durchgeführt werden und mit der Wärmebildaufnahme werden die Module anschließend überprüft.

Damit in AERORDONE immer die richtigen Aufnahmen miteinander gekoppelt werden können, müssen diese entsprechend benannt werden. Der Name des

Wärmebildes muss deshalb stets mit „w_“ beginnen und der des normalen Fotos mit „n_“. Der nachfolgende Teil muss natürlich bei beiden Fotos gleich sein. So heißen die Bilder zum Beispiel „w_0001.jpg“ und „n_0001.jpg“.

Alle auszuwertenden Bilddateien befinden sich in einem Ordner, der bei Programmstart in einem Fenster ausgewählt werden kann. Es werden anschließend alle Dateien aus diesem Ordner ausgelesen und hinzugefügt. Falls zu einem späteren Zeitpunkt noch mehr Bilder hinzukommen, können diese auch nachträglich ausgewählt werden.

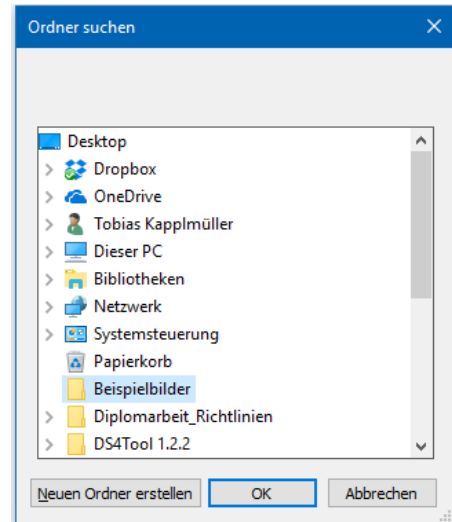


Abbildung 45 Bildauswahlfenster

6.3) Bilderkennung & -verarbeitung

In AERODRONE werden zur Bilderverarbeitung immer eine normale Aufnahme und einer Wärmebildaufnahme, die zum gleichen Zeitpunkt und von der gleichen Position aus aufgenommen wurden, miteinander verbunden. Anhand des normalen Bildes werden dann die einzelnen Module erkannt und deren Koordinaten in einer Liste abgespeichert. Mithilfe dieser Koordinaten kann dann bei dem Wärmebild die durchschnittliche Helligkeit jedes Modules berechnet und mit dem Gesamtdurchschnitt verglichen werden. Wenn hierbei eine gewisse Abweichung gegeben ist, wird das Modul als defekt erkannt.

6.3.1) Erkennen einzelner Module

Zum Erkennen der einzelnen Solarmodule werden vier Arbeitsschritte benötigt. Zuerst wird die Aufnahme in ein Schwarz-Weiß-Bild umgewandelt, da hier aufgrund der hellen Ränder das Solarmodul als Rechteck erkannt werden kann. Als nächstes wird mit dem sogenannten Canny-Algorithmus die Kantenerkennung durchgeführt. Danach werden mit einer von EmguCV zur Verfügung gestellten Funktionen alle Formen in diesem Bild erkannt und aus diesen erkannten Formen die Rechtecke herausgefiltert. Die Koordinaten dieser Rechtecke werden anschließend in einer Liste abgespeichert, zu der im späteren Verlauf der Bilderverarbeitung noch der Status des Modules (Funktionsfähig, Defekt oder Eingeschränkt), die Durchschnittshelligkeit und die Reihe bzw. Spalte innerhalb der gesamten Anlage hinzugefügt werden können.

6.3.1.1) Umwandeln in Schwarz-Weiß-Bild

Zum Umwandeln von Bildern in einen anderen Farbraum stellt EmguCV bereits eine passende Funktion zur Verfügung:

```
CvInvoke.CvtColor(imgOriginal, imgGray, ColorConversion.Bgr2Gray);
```

Hierbei müssen lediglich die passenden Parameterwerte übergeben werden. `imgOriginal` steht hierbei für das Originalbild, `imgGray` ist die Variable, in die das Schwarz-Weiß-Bild anschließend gespeichert werden soll und `ColorConversion.Bgr2Gray` gibt an, in welchen

Farbraum das Bild konvertiert werden soll. In diesem Fall also vom BGR-Farbraum in ein Bild mit Graustufen. (EmguCV Dokumentation, 2017)

Damit das Ergebnis der anschließenden Kantenerkennung besser ausfällt, wird das Schwarz-Weiß-Bild noch mit einem Gaußschen Weichzeichner weichgezeichnet. Dieser basiert auf der Gaußschen Normalverteilung, die wie der Name bereits vermuten lässt von Johann Carl Friedrich Gauß entwickelt wurde. (Wikipedia, 2017)

In AERODRONE wird er mit der folgenden EmguCV-Funktion angewendet:

```
CvInvoke.GaussianBlur(imgGray, imgBlurred, new System.Drawing.Size(5, 5), 0.4);
```

imgGray ist hierbei das zuvor konvertierte Schwarz-Weiß-Bild und imgBlurred das Ergebnis der GaussianBlur-Operation.

6.3.1.2) Canny – Algorithmus

Der Canny-Algorithmus ist eine Methode zur Erkennung von Kanten in Bildern. Er wurde von John Francis Canny entwickelt und ist auch nach ihm benannt. Auch für diese Operation bietet EmguCV eine passende Methode, welche in AERODRONE wie folgt verwendet wird:

```
CvInvoke.Canny(imgBlurred, imgCanny, 80, 250);
```

Wie bei den vorherigen Funktionen gibt es wieder zwei Parameter, welche das Ausgangsbild (imgBlurred – das weichgezeichnete Schwarz-Weiß-Bild) bzw. das Ergebnis der Operation (imgCanny) festlegen. Die übrigen zwei Parameter bestimmen die beiden Schwellwerte für die Kantenerkennung. Deren Bedeutung wird später noch genauer erklärt werden (EmguCV Dokumentation, 2017)

Bei der Entwicklung seines Algorithmus wollte John Canny einen Kantenerkennungsalgorithmus entwickeln der hinsichtlich dreier Kriterien optimal ist. Diese drei Kriterien sind:

- Erkennung: Es sollen alle wirklichen Kanten erkannt werden, falsche Kanten jedoch nicht
- Lokalisierung: Im Bild soll der Abstand zwischen dem Punkt an dem eine Kante ist und dem, an dem sie erkannt wurde so gering wie möglich sein
- Ansprechverhalten: Der Algorithmus darf keine Kanten mehrfach erkennen. Aus diesem Grund ist die Breite einer Kante nicht breiter als ein Pixel

Einer der größten Unterschiede zu anderen Algorithmen zur Kantendetektion ist, dass hier Binärbilder entstehen und keine Graustufenbilder. Somit kann ein Pixel entweder eine Kante (1, weiß) oder keine Kante (0, schwarz) sein. Besonders ist auch, dass aus dem Bild Information über die Richtung der Kante berechnet werden kann.

Im nächsten Abschnitt wird der Ablauf anhand von folgendem Beispielbild erklärt:



Abbildung 46 Canny Algorithmus - Ausgangsbild

Das Verfahren wird in 4 Schritte unterteilt:

1. Glättung – Rauschen im Bild wird unterdrückt
2. Kantendetektion – In jedem Pixel wird die Kante und dessen Richtung geschätzt
3. Unterdrückung von Nicht-Maxima – In Bezug auf die Kantenstärke werden nur lokale Maximalwerte als mögliche Kanten angesehen
4. Hysterese – Nichtrelevante Kanten werden entfernt

Glättung und Kantendetektion

Am Anfang wird das Bild durch die Anwendung eines Gaußfilters geglättet, um das Bildrauschen zu beseitigen. Danach werden mit dem sogenannten Sobel-Operator die Richtung und die Stärke der Kante in jedem einzelnen Pixel berechnet. Daraus resultieren zwei Bilder - ein Graustufenbild, welches alle Kanten und deren Stärke anzeigt:



Abbildung 47 Bild mit erkannten Kanten

Und dasselbe Kantenbild, in dem die Kanten anhand ihrer Richtung verschieden eingefärbt sind. Hierbei sind folgende Farben verwendet worden:

- Rot: horizontale Kanten
- Blau: vertikale Kanten
- Grün: diagonale Kanten, die von rechts oben nach links unten verlaufen
- Türkis: diagonale Kanten, die von links oben nach rechts unten verlaufen

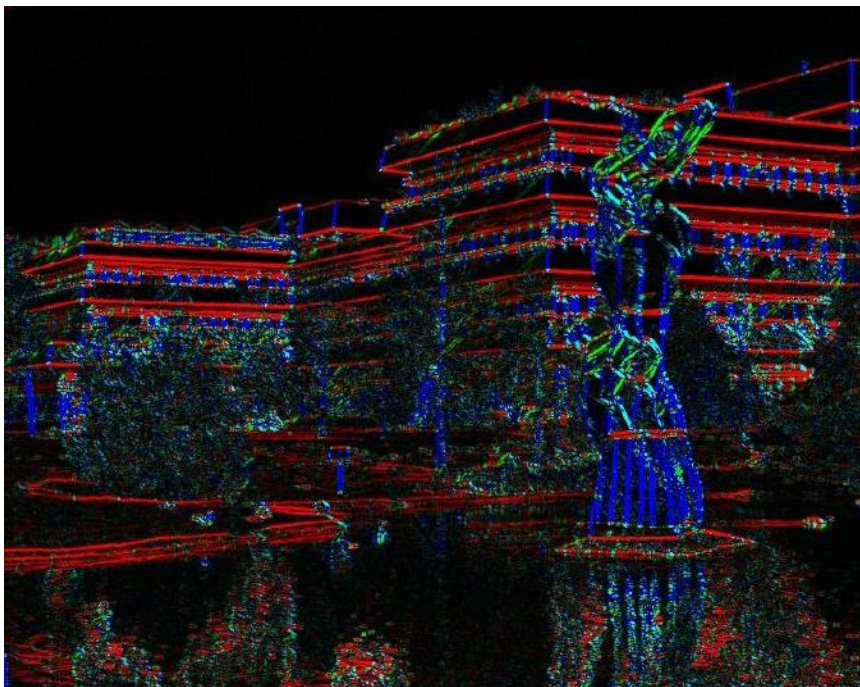


Abbildung 48 Bild mit Kanten und deren Richtungen

Unterdrückung von Nicht – Maxima

In diesem Schritt werden alle bereits erkannten Kanten auf eine Breite von genau einem Pixel reduziert. Dies trägt dazu bei, dass man die Kanten so genau wie möglich fehlerfrei lokalisieren kann. Um für diese Fehlerfreiheit garantieren zu können ist es wichtig, zu wissen, wo diese ein Pixel breite Kante positioniert ist.

Wenn also eine Kante eine Breite von mehr als nur einem Pixel hat, kann man davon ausgehen, dass sich die Kante dort befindet, wo der Betrag der erkannten Kante am höchsten ist, d.h. dort, wo die weiß eingezeichnete Kante auf dem ersten Ergebnisbild (Abbildung 47) am hellsten ist. Alle Pixel, die keine solche Maximalwerte sind werden auf 0 gesetzt (schwarz eingefärbt). Das Ergebnis sieht dann folgendermaßen aus:



Abbildung 49 Bild ohne Nicht-Maxima

Hysterese

Als Ergebnis des vorherigen Schrittes haben nun alle Kanten nur noch eine Breite von einem Pixel. Die Helligkeit der Kanten ist jedoch nach wie vor unterschiedlich und es befinden sich noch falsche Kanten. Falsche Kanten sind Kanten, deren Grauwerte unter einem gewissen Schwellenwert liegen. Dass jedoch in unserem Programmcode zwei solche Schwellenwerte angegeben wurden liegt daran, dass der Canny-Algorithmus ein Verfahren verwendet, welches zwei Schwellenwerte benötigt: die sogenannte Hysterese. Um die Erklärung zu vereinfachen werden diese beiden Werte S_1 und S_2 genannt, wobei $S_1 \leq S_2$ gilt. Nun wird der Grauwert jedes Pixels mit S_1 und S_2 verglichen. Ist dieser Wert kleiner als S_1 wird das Pixel als Nicht-Kante markiert und auf Schwarz gesetzt, ist er größer als S_2 passiert das Gegenteil, der Bildpunkt wird als Kante markiert und auf Weiß gesetzt. Bei den Pixeln, deren Wert zwischen den beiden Schwellenwerten liegt, wird überprüft, ob die umliegenden Pixel Kantenpixel sind oder nicht. Wenn ein solcher Bildpunkt der einzige Kantenpixel in der Umgebung wäre, wird er als Nicht-Kante markiert, andernfalls als Kante. Somit ist die Kantenerkennung abgeschlossen. (Wagner, 2017)



Abbildung 50 Ergebnis der Kantenerkennung

Nachdem der Canny-Algorithmus auf die Aufnahme angewendet wurde sieht das Ergebnis folgendermaßen aus:

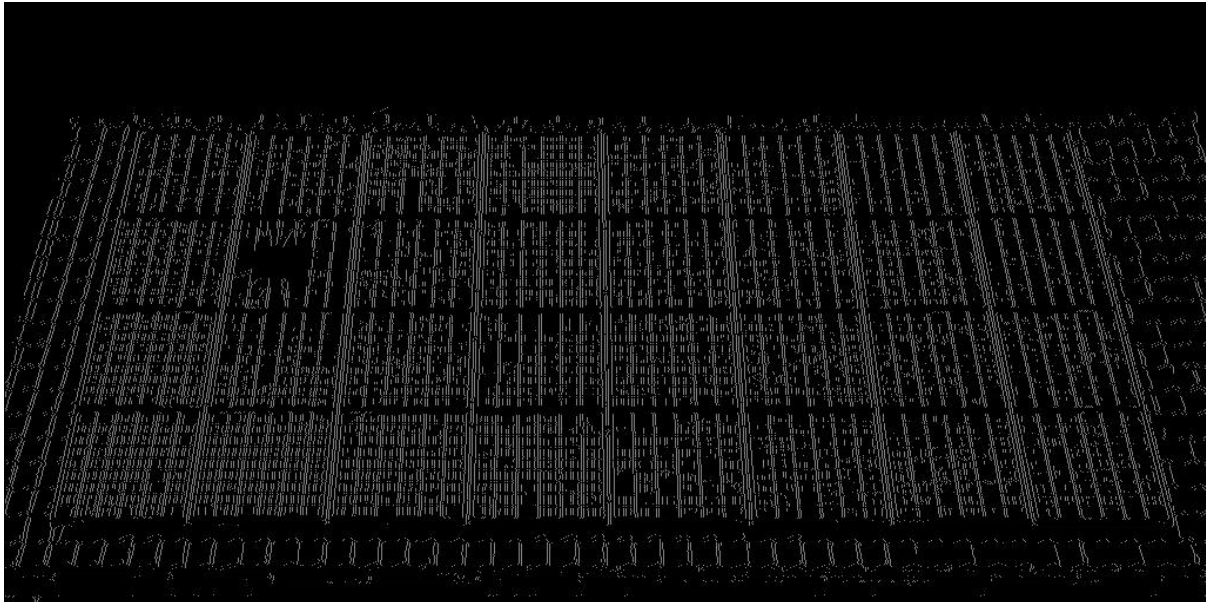


Abbildung 51 Bild einer Photovoltaikanlage nach Anwendung des Canny-Algorithmus

6.3.1.3) Erkennung der Rechtecke

Damit die einzelnen Solarmodule auf einem Bild erkannt werden können, werden anhand der Kanten einzelne Formen erkannt. Auch hierfür wird die entsprechende Funktionalität von EmguCV zur Verfügung gestellt:

```
CvInvoke.FindContours(imgCanny, contours, null, RetrType.List,  
ChainApproxMethod.ChainApproxSimple);
```

Dieser Befehl erkennt in dem angegebenen Bild (imgCanny) alle Formen und speichert diese als Liste (RetrType.List) in der Variable contours ab.

ChainApproxMethod.ChainApproxSimple steht hierbei für die Annäherungsmethode, mit der die Formen erkannt werden. (EmguCV Dokumentation, 2017)

Als nächstes werden alle Einträge in dieser Liste nacheinander durchlaufen und anhand gewisser Kriterien gefiltert, damit am Ende nur die entsprechenden Rechtecke übrigbleiben.

Die folgenden Arbeitsschritte werden also mit jeder einzelnen erkannten Form durchgeführt:

Als erstes werden mit der folgenden Abfrage alle Formen mit einer Fläche unter 2000 Pixel aussortiert, um zu vermeiden, dass zu kleine Objekte, wie zum Beispiel Dachziegel etc. als Solarmodul erkannt werden. Die variable `approxContour` ist in diesem Fall die Form, die gerade überprüft wird.

```
if (CvInvoke.ContourArea(approxContour, false) > 2000)
```

Danach wird überprüft, wie viele Seiten eine Form besitzt. Da es sich bei den Modulen immer um Rechtecke handelt, werden nur mehr die Formen mit exakt 4 Seiten übernommen:

```
if (approxContour.Size == 4)
```

Im letzten Schritt werden nun alle Winkel des jeweiligen Vierecks überprüft:

```
if (angle < 70 || angle > 110)
{
    isRect = false;
    break;
}
```

Wenn sich alle Winkel in einem Bereich zwischen 70° und 110° befinden, wird die Form als Rechteck erkannt und die Koordinaten werden in einer Liste von Objekten der `SolarModule`-Klasse gespeichert. Ein `SolarModule` enthält die X- und Y- Koordinaten innerhalb des Bildes, die Reihe und Spalte innerhalb der Anlage, die durchschnittliche Helligkeit des Moduls und den Status (Funktionsfähig, Defekt oder Eingeschränkt), wobei hier vorerst nur die Koordinaten gespeichert werden und die restlichen Daten erst später hinzugefügt werden.

Anschließend können die erkannten Rechtecke noch auf einem Bild eingezeichnet werden. Nachdem in AERODRONE nie ein Bild angezeigt wird, auf dem die Rechtecke eingezeichnet sind, wurde diese Funktion nicht implementiert, man kann es jedoch trotzdem auf der folgenden Abbildung sehen:

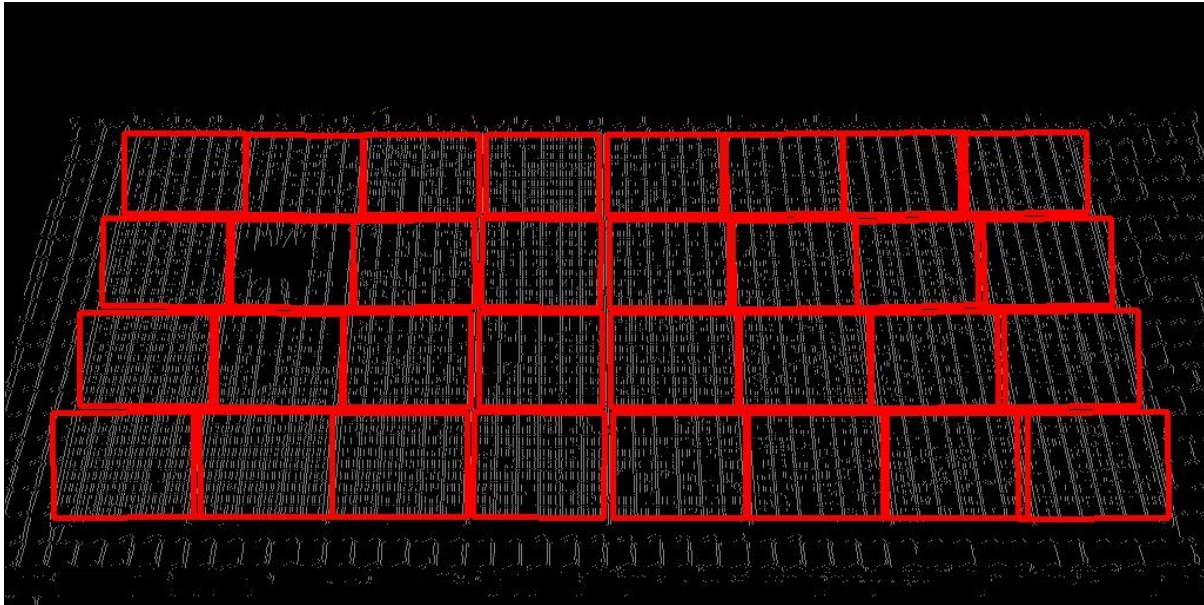


Abbildung 52 Photovoltaikanlage nach Anwendung des Canny-Algorithmus, erkannte Rechtecke wurden eingezeichnet

6.3.2) Positionsberechnung innerhalb der Anlage

Um später die dreidimensionale Darstellung der Photovoltaikanlage korrekt zu ermöglichen, ist es wichtig, dass die genaue Position jedes Modules innerhalb der Anlage bekannt ist. Um diese zu berechnen, werden die Pixel-Koordinaten der Module verwendet. Als Erstes wird die Modulliste aufsteigend nach der Y-Koordinate sortiert:

```
this.modules = this.modules.OrderBy(m => m.centerY).ToList();
```

Das oberste Modul auf dem Bild steht nun also an erster Stelle der Liste. Es werden nun alle Einträge der Reihe nach durchlaufen und der Y-Wert mit dem des vorherigen Modules verglichen. Wenn der Unterschied einen gewissen Wert überschreitet, d.h. wenn ein Modul um einen gewissen Wert niedriger positioniert ist als das vorhergehende, wird angenommen, dass sich dieses in einer neuen Reihe befindet. Nach dieser Vorgehensweise wird jedes Modul überprüft, bis für alle Einträge der Liste die Reihe bekannt ist.

Um als nächstes die Spalte jedes Moduls zu bestimmen, wird die Liste der Solarmodule aufsteigend nach der zuvor ermittelten Reihe sortiert. Nachdem jedoch mehrere Module in derselben Reihe sind, werden die Solarmodule derselben Reihe nach der X-Koordinate sortiert. Dies geschieht mit einem einzelnen Befehl:

```
this.modules = this.modules.OrderBy(m => m.row).ThenBy(m => m.centerX).ToList();
```

Nach diesem Vorgang befinden sich die Module der obersten Reihe ganz am Anfang der Liste, wobei das Modul ganz links der erste Listeneintrag ist. Nach dem Modul, welches sich in der obersten Reihe ganz rechts befindet steht das am linken Rand befindliche Modul der zweiten Reihe usw. Jetzt müssen nur noch für jede Modulreihe in aufsteigender Folge von rechts nach links die passenden Spaltenzahlen eingetragen werden und die Berechnung der Positionen ist abgeschlossen.

6.3.3) Überprüfen der Funktionsfähigkeit

Jetzt wurden bereits die einzelnen Module der Anlage erkannt und deren Position bestimmt. Als nächstes muss bestimmt werden, ob eines oder mehrere dieser Solarmodule beschädigt sind und wenn ja, um welches Modul es sich handelt. In EmguCV werden Bilder standardmäßig im BGR-Farbraum abgespeichert. Dieser ähnelt dem RGB-Raum sehr stark, bei beiden Modellen werden für jeden Pixel die Rot-, Grün- und Blauwerte abgespeichert.

Um Wärmebilder miteinander vergleichen zu können ist dieses Farbmodell jedoch eher schlecht geeignet, da es für diesen Zweck besser wäre, wenn man die Bildpunkte statt den Farben, aus denen sie sich zusammensetzen, anhand ihrer Merkmale, wie zum Beispiel Helligkeit oder Intensität identifizieren könnte. Aus diesem Grund verwendet AERODRONE für die Wärmebildauswertung den HLS-Farbraum. Bei diesem werden statt Rot, Grün und Blau die drei Werte für den Farbton (engl. Hue), die Helligkeit (engl. Lightness) und der Sättigung (engl. Saturation) gespeichert. (Wikipedia, 2017)

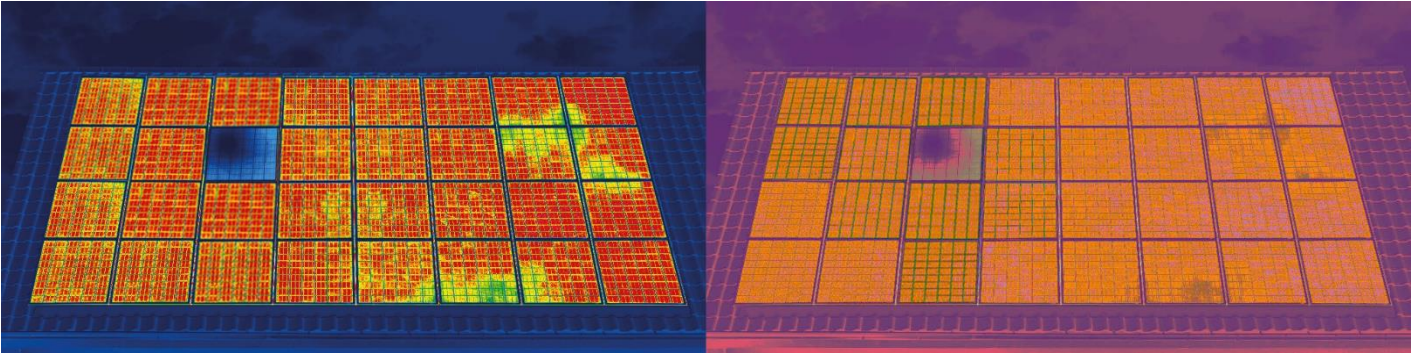


Abbildung 53 Wärmebild im BGR-Farbraum (links) und HLS-Farbraum (rechts)

Dafür wird mit fast demselben Befehl, der bereits zum Umwandeln des normalen Bildes in ein Schwarz-Weiß-Bild verwendet wurde, das Wärmebild in ein HLS-Bild konvertiert:

```
CvInvoke.CvtColor(imgHeatscan, hlsHeatscan, ColorConversion.Bgr2Hls);
```

Auch hier stehen die Parameter wieder für das Eingangsbild (`imgHeatscan`), das Ausgabebild (`hlsHeatscan`) und die Art der Konvertierung (`Bgr2Hls` – also vom BGR- in den HLS-Farbraum). (EmguCV Dokumentation, 2017)

Die eigentliche Überprüfung der Funktionsfähigkeit lässt sich in folgende Schritte aufteilen: Zuerst werden von jedem Modul die Helligkeitswerte von einer gewissen Anzahl an Pixeln ermittelt und daraus ein Durchschnitt berechnet. Mit diesen Moduldurchschnitten wird anschließend ein Gesamtdurchschnittswert für die ganze Anlage berechnet. Am Schluss wird der Durchschnittswert jedes Moduls mit dem Gesamtdurchschnitt verglichen. Wenn hier eine gewisse Abweichung überschritten wird, wird das Modul als defekt erkannt.

6.3.3.1) Berechnung der Durchschnittswerte einzelner Module

Im ersten Schritt zwei Listen mit Gleitkommazahlen angelegt:

```
List<double> pixelLightnesses = new List<double>();  
List<double> avgList = new List<double>();
```

Die Variable `pixelLightnesses` enthält die Helligkeitswerte jedes Pixels, das für die Überprüfung eines Moduls kontrolliert wird und wird daher vor jedem Modul wieder geleert. Aus diesen Werten wird anschließend die durchschnittliche Helligkeit des aktuellen Moduls berechnet und in die zweite Liste, `avgList`, eingetragen. Aus den Durchschnittswerten jedes Moduls kann am Schluss der Gesamtdurchschnitt der Anlage berechnet werden.

Von der zuvor erfolgten Erkennung aller Solarmodule sind für jedes Modul die X- und Y-Koordinaten des Mittelpunktes bekannt. Mit diesen Koordinaten werden alle Helligkeitswerte eines 40*40 Pixel großen Quadrats in der Mitte jedes Moduls ermittelt und in die oben erwähnte Liste gespeichert. Dazu wird für jeden Pixel innerhalb dieses Quadrats ein Objekt des Typs `Hls` angelegt und daraus die Helligkeit (engl. `Lightness`) ermittelt:

```
Hls pixel = heatscan[y, x];  
pixelLightnesses.Add(pixel.Lightness);
```

Danach wird von diesen Werten ein arithmetisches Mittel berechnet und in die Liste der Moduldurchschnitte hinzugefügt:

```
avgList.Add(pixelLightnesses.Average());
```

Abschließend wird die erste Liste wieder zurückgesetzt und derselbe Vorgang wird mit dem nächsten Solarmodul durchgeführt, bis die ganze Modulliste durchlaufen wurde:

```
pixelLightnesses.Clear();
```

6.3.3.2) Erkennen des Modulstatus

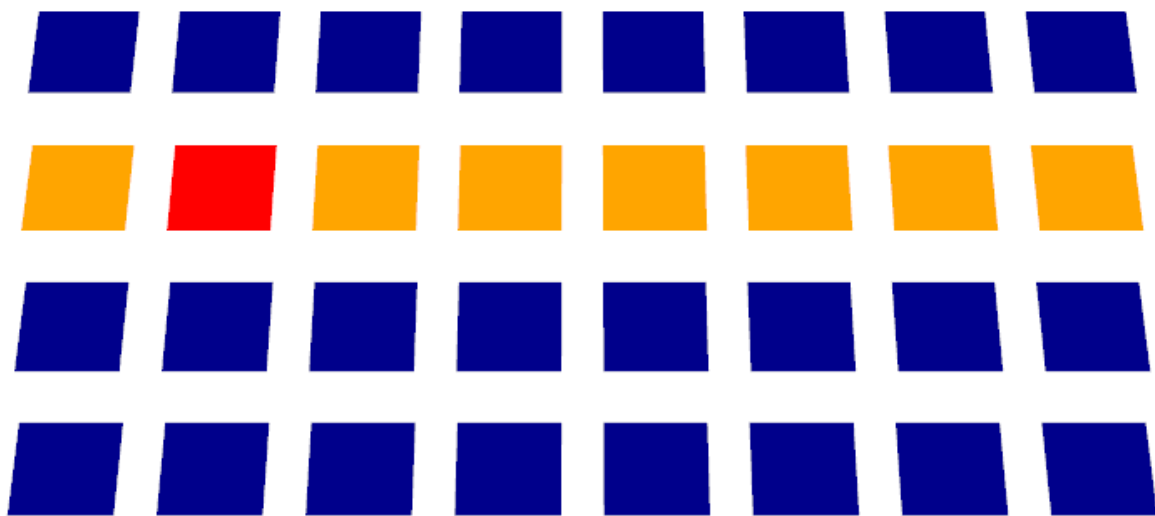
Als Ergebnis des vorherigen Arbeitsschrittes existieren nun eine Liste, in der die durchschnittliche Helligkeit jedes Solarmoduls enthalten ist, und ein berechneter Gesamtdurchschnitt aller dieser Helligkeitswerte. Da ein fehlerhaftes Modul entweder sehr viel wärmer oder sehr viel kälter ist als ein funktionsfähiges, muss im nächsten Schritt nur jeder Eintrag dieser Liste mit dem Gesamtdurchschnitt verglichen werden.

Wenn also bei diesem Vergleich eine gewisse Abweichung festgestellt wird, so wird der Status dieses Moduls auf „1“ (= defektes Modul) gesetzt und alle anderen Module, die sich in derselben Reihe befinden, wie das defekte Modul, werden auf den Status „2“ (= funktionsfähig, aber produziert wegen einem defekten Modul weniger Leistung) gesetzt. Wenn bei einem Modul keine bzw. nur eine geringfügige Abweichung vorhanden ist, wird der Status des Moduls auf „0“ (= funktionsfähiges Modul) gesetzt.

Damit der Auftraggeber bzw. der Benutzer des Programms jederzeit einen kurzen Überblick über die Daten der Photovoltaikanlage hat, werden abschließend noch einige Statistiken ermittelt. Hierbei wird einfach die Gesamtanzahl der Module sowie die Anzahl der defekten und der eingeschränkten Module berechnet und angezeigt. Zusätzlich wird auch der geschätzte Leistungsverlust der Anlage in Prozent berechnet.

6.4) 3D-Darstellung

Auf einem Wärmebild ist es oft nicht auf den ersten Blick eindeutig erkennbar, ob ein Modul beschädigt ist bzw. wenn ja, welches das beschädigte Modul ist. Aus diesem Grund wird in AERODRONE die Photovoltaikanlage nach der Überprüfung auch noch in einer 3D-Ansicht dargestellt. Dabei wird die Farbe, in der ein Modul angezeigt wird je nach dem Modulstatus verändert. Module, die funktionsfähig sind werden dunkelblau dargestellt, defekte Module in Rot und alle Module, die funktionieren würden, jedoch durch ein defektes Modul eingeschränkt sind werden orange eingefärbt.



[Zurücksetzen](#)

Abbildung 54 3D-Darstellung einer Anlage mit einem defekten Solarmodul

Alle Funktionen und Methoden, die für diese dreidimensionale Darstellung nötig sind werden bereits von WPF zur Verfügung gestellt, es ist also keine zusätzliche Software nötig, um diese zu realisieren.

Die Grundkomponente dafür ist ein sogenannter Viewport3D. Dieser ist dafür zuständig, die 3D-Inhalte innerhalb des zweidimensionalen Programmes anzuzeigen. (Viewport3D, 2017)

Damit dies möglich ist benötigt der Viewport3D eine Kamera. Da bieten sich zwei Möglichkeiten an: eine PerspectiveCamera (perspektivische Kamera) oder eine OrthographicCamera (orthografische Kamera).

Der größte Unterschied zwischen diesen beiden Kameraarten ist, dass eine perspektivische Kamera Objekte näher zur Kamera größer darstellt als weiter entfernte Objekte. Bei einer orthografischen Kamera werden alle Objekte mit derselben Größe gleich groß dargestellt, egal wo sich diese befinden. Aus diesem Grund verwendet AERODRONE eine perspektivische Kamera. In dem folgenden Bild lässt sich der Unterschied zwischen den beiden Kameraarten anhand von drei gleich großen Würfeln sehr gut erkennen:

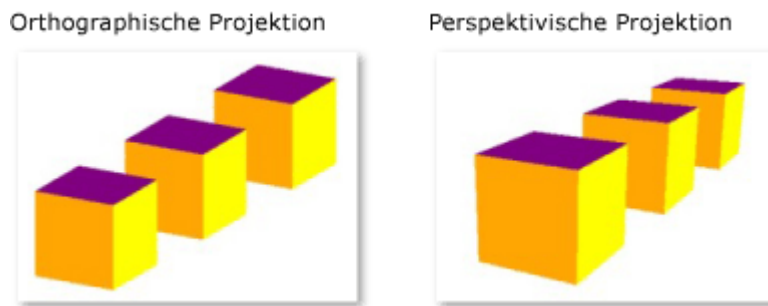


Abbildung 55 Orthografische vs. Perspektivische Projektion (.NET Dokumentation, 2017)

Innerhalb des Viewport3D befindet sich ein ModelVisual3D, welches wiederum eine Model3DGroup beinhaltet. Hier können nun die Objekte hinzugefügt werden, die dargestellt werden sollten. Außerdem wird hier auch die Beleuchtung festgelegt, die nötig ist, damit die darzustellenden Objekte sichtbar sind. Auch hier gibt es mehrere Möglichkeiten:

- AmbientLight: Es werden alle Objekte gleichmäßig von allen Richtungen beleuchtet, egal wo sie sich befinden.
- DirectionalLight: Verhält sich wie eine weit entfernte Lichtquelle. Es wird eine Richtung angegeben, aus der das Licht kommt, jedoch keine konkrete Position.
- PointLight: Verhält sich wie eine naheliegende Lichtquelle. Hier werden Position und Richtung des Lichtes angegeben.
- SpotLight: Vergleichbar mit PointLight, nur dass hier das Licht in einem Kegel abgestrahlt wird.

Bei AERODRONE wird ein AmbientLight verwendet. (Übersicht über 3D-Grafiken, 2017)

Jetzt fehlen nur noch die einzelnen Module, die in der Model3DGroup enthalten sein sollen. Dazu wird jedes Solarmodul als Viereck mit den Abmessungen 5*5 dargestellt, der Abstand zwischen zwei Modulen wird auf 2 festgelegt. Bevor die Koordinaten jedes Moduls berechnet werden können muss zuerst die Gesamthöhe berechnet werden:

$$\text{Reihenanzahl} * 5 + (\text{Reihenanzahl} - 1) * 2$$

Als nächstes muss die Breite jeder Reihe berechnet werden:

$$\text{Modulanzahl} * 5 + (\text{Modulanzahl} - 1) * 2$$

Nun können die vier Eckpunkte jedes Modul ermittelt werden. Dazu werden folgende Formeln verwendet:

```
double yBottom = height / 2 - (mod.row * 5 + (mod.row - 1) * 2);
double yTop = yBottom + 5;
```

```
double xRight = (mod.column * 5 + (mod.column - 1) * 2) - (rowWidth / 2);
double xLeft = xRight - 5;
```

Hierbei wird die halbe Gesamthöhe bzw. die halbe Breite der Reihe deshalb berücksichtigt, weil der Nullpunkt des Koordinatensystems genau in der Mitte der Anlage ist, wie man auf der folgenden Abbildung veranschaulicht wird:

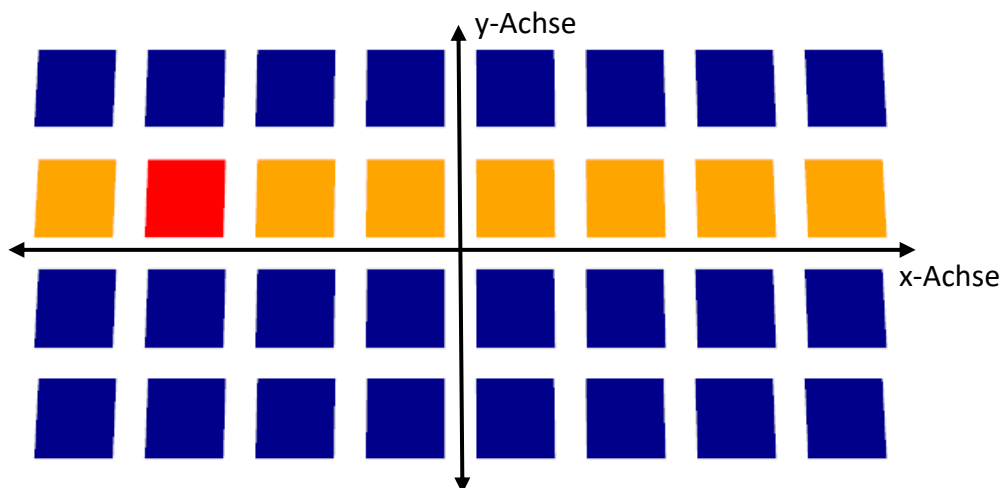


Abbildung 56 3D-Darstellung inkl. Koordinatensystem

Wenn alle vier Eckpunkte eines Moduls bekannt sind, wird anhand des Status die entsprechende Farbe ausgewählt und mit einer eigens implementierten Methode ein Viereck daraus generiert. Diese Methode ist nötig, weil standardmäßig nur dreieckige Formen unterstützt werden. Deshalb werden einfach zwei solche Dreiecke zusammengesetzt, damit das Ergebnis aussieht wie ein Viereck.

Die Ansicht kann natürlich auch mit dem Mauszeiger verschoben und gedreht werden und mit dem Mausekranz ist es möglich zu zoomen. Es ist auch ein Zurücksetzen-Button verfügbar mit dem man die Ansicht nach dem Drehen oder Zoomen wieder zurück in die Ausgangssituation bringen kann.

7) Resümee

Die Entwicklung von AERODRONE und das anschließende Verfassen der Diplomarbeit hat dem gesamten Projektteam geholfen, wichtige Erfahrungen für unser späteres Berufsleben zu sammeln. Durch die selbstständige Arbeit konnten wir in den Bereichen Projektmanagement, Softwareentwickeln und dem Verfassen formeller Dokumente viel dazulernen. Durch den Unterricht an der HTL-Perg sind die ersten beiden dieser Bereiche nichts Neues, jedoch wurden die Arbeiten außerhalb der Schulzeit abgewickelt, weshalb wir hier einen besseren Bezug zum Berufsalltag eines Softwareentwicklers herstellen konnten.

Außerdem mussten wir uns mit neuen Technologien auseinandersetzen, wodurch wir lernten, dass die Entwicklung eines größeren Projektes meist mit viel Aufwand zum Einlesen in unbekannte Technologien oder Vorgänge verbunden ist.

Dadurch, dass die Diplomarbeit zum größten Teil außerhalb der Schule entwickelt wurde, war auch kein Betreuer anwesend, der uns bei auftretenden Problemen sofort helfen konnte. Wir mussten die Probleme in den meisten Fällen eigenständig analysieren und lösen, da der Betreuungslehrer erst in der nächsten Projektentwicklungsstunde zur Verfügung gestanden ist und wir darauf mehrere Tage warten mussten.

Alles in allem war diese Arbeit für uns einer der wichtigsten Bestandteile der Ausbildung. Wir haben viel darüber gelernt, uns selbst zu koordinieren und uns bei der Zusammenarbeit ohne einen direkten Vorgesetzten aufeinander abzustimmen.

8) Verteilung der Aufgaben

8.1) Michael Kirchhofer

- Kurzbeschreibung
- Abstract
- Projektumfeld
 - Team
 - Auftraggeber
- Grundlagen
 - Theoretische Grundlagen
 - Photovoltaik Allgemein
 - Funktionsweise einer Photovoltaikanlage
- Resümee
- Literaturverzeichnis
- Glossar

8.2) Nicklas Harringer

- Kurzbeschreibung
- Projektumfeld
 - Team
- Einleitung
 - Ausgangssituation
- Grundlagen
 - Entwicklungssysteme
 - Sonstige verwendete Software
 - Verwendete Technologien
- Resümee
- Literaturverzeichnis
- Glossar

8.3) Tobias Kapplmüller

- Kurzbeschreibung
- Projektumfeld
 - Team
 - Betreuungslehrkraft
 - Ausbildungseinrichtung
 - Auftraggeber
- Einleitung
 - Ausgangssituation
 - Aufgabenstellung
- Realisierung
 - Grundaufbau des Programmes
 - Bilder hinzufügen
 - Bilderkennung & -verarbeitung
 - 3D-Darstellung
- Resümee
- Literaturverzeichnis
- Glossar

9) Literaturverzeichnis

9.1) Glossar

Absoluter Nullpunkt	Unterer Grenzwert der Temperatur, liegt bei $-273,15^{\circ}$ Celsius. Unter dem absoluten Nullpunkt bewegen sich keine Elektronen mehr und alle Elemente, die diese Temperatur erreichen befinden sich in festem Zustand
Absorptionsfläche	Schicht, die auf Festkörper aufgetragen wird, um deren Reflexionen zu vermeiden.
Algorithmus	Klar definierte Vorgehensweise zur Lösung eines gewissen Problems, welche sich aus mehreren Arbeitsschritten zusammensetzt
Antireflexionsschicht	Beschichtung von elektrischen Bauteilen, die eine auftretende Reflexion so gut wie möglich verhindern soll, je niedriger die Reflexion von Sonnenstrahlen bei Solarzellen, desto besser ist auch der Wirkungsgrad der Module
API	Application Programming Interface = Programmierschnittstelle
Bändermodell	Physikalisches Modell, das die elektrischen Zustände eines Elements beschreibt. Es bildet vor allem die Energieniveaus der Atome ab
Bildrauschen	Verschlechterung eines digitalen Bilds durch gewisse Störungen, die in keinerlei Verbindung mit dem Bildinhalt stehen

Binärbild	Jeder Pixel des Bildes wird mit einem Bit (nur 0 oder 1 möglich) gespeichert, sodass der Wert 0 für Schwarz steht und der Wert 1 für Weiß
Button	Ein Knopf bzw. eine Taste in einem Programm, welche angeklickt werden kann und dadurch eine Aktion im Programmcode auslöst
Bypassdiode	Elektrische Bauteile von modernen Photovoltaikzellen, um diese im Schadensfall vor schädlicher elektrischer Spannung zu schützen. Sie reduzieren die Hitzeentwicklung in beschädigten Zellen und verringern somit die Brandgefahr innerhalb von Modulen
Cloud	Bereitstellen von IT-Infrastruktur wie z.B. Speicherplatz, Rechenleistung oder Anwendungssoftware. Speziell in Bezug auf Speicherplatz kann man es sich so vorstellen, dass die Daten online auf einen Server gespeichert werden
Code Editor	Ähnlich zur Entwicklungsumgebung, nur ist der Code Editor leichter zu handhaben und bietet meist weniger Funktionen
Defektelektron	Auch Loch genannt, nach der Abspaltung eines Elektrons aus dem Valenzband bildet sich in diesem ein Loch, weil innerhalb des Bandes plötzlich ein Elektron fehlt. Diese Löcher stellen keinen wirklichen Bestandteil eines Atoms dar. Sie werden lediglich dazu verwendet um das Fehlen eines Valenzelektrons zu symbolisieren.
Diffusionsspannung	Elektrische Spannung, die der Elektronenwanderung entgegenwirkt. Ist die Spannung innerhalb eines Atoms niedriger als die Diffusionsspannung so können keine Elektronen vom Valenz- in das Leitungsband gelangen

Diode	Elektrisches Bauteil, das Strom in eine Richtung leitet und in die andere Richtung isoliert
Elektrischer Verbraucher	Bauelement oder Gerät, das elektrische Energie in andere Energieformen umwandelt (zum Beispiel ein elektrischer Heizkörper, der Strom in Wärme umwandelt oder eine Lampe, die Energie in Licht konvertiert)
Elektrisches Feld	Physikalisches Feld, das die elektrische Übertragung von Energie ermöglicht. Dabei binden sich Elektronen an einen Atomkern und es werden somit andere Atome beeinflusst und verändert
Elektron	negativ geladenes Teilchen eines Atoms
Elektronenwanderung	Der Übergang eines Elektrons vom Valenzband in das Leitungsband. Tritt meistens durch die Einwirkung eines elektrischen Feldes auf
Energieband	Darstellungsmodell für jeweils ein Energieniveau
Energieniveau	Energiewerte mehrerer Atome, die so nahe beieinanderliegen, dass man diese nicht mehr einzeln betrachten kann
Entwicklungsumgebung	Programm zur Entwicklung von Programmen, es besteht aus verschiedensten Tools die den Programmierer unterstützen
erneuerbarer Strom	Strom, der durch Energiearten gewonnen wurde, die sich im Gegensatz zu fossilen Brennstoffen verhältnismäßig schnell erneuern, Beispiel: Sonnenenergie

Farbraum	Alle Farben, die durch eine bestimmte Farbgebungsart dargestellt werden können, zum Bsp. der RGB (Rot-Grün-Blau) Farbraum kann 16,7Millionen Farben darstellen
Framework	Rahmen bzw. Programmierstruktur, die für jedes Programm individuell anpassbar sind
Funktion	Mehrere Befehle, die öfter verwendet werden, werden zusammengefasst, um nicht jedes Mal alle diese Befehle schreiben zu müssen
Germanium	Halbleitermetall, das unter bestimmten Bedingungen elektrische Leitfähigkeit besitzt. War früher der am meisten verwendete Halbleiterwerkstoff, bevor es von Silizium verdrängt wurde
Halbleiter	Element, welches unter bestimmten Bedingungen (passende Temperatur und Zusammensetzung des jeweiligen Elements) elektrischen Strom leiten kann
Halbleitertechnik	Fachbereich, der sich mit der Entwicklung und der Herstellung von technischen Komponenten mit Halbleiterelementen spezialisiert (zum Beispiel Solarzellen)
implementieren	Entwickeln von Programmcode
Infrarotstrahlung	Elektrische Strahlung außerhalb des für den Menschen sichtbaren Bereichs, kann mit speziellen Kameras eingefangen werden, um zum Beispiel die Wärmebildung eines Objektes festzustellen
Isolator	Elektrisches Bauteil, das eine sehr geringe elektrische Leitfähigkeit besitzt. Wird in der Elektrotechnik häufig für die Ummantelung von elektrischen Leitern verwendet

Kante (Bildererkennung)	Trennung von Flächen in einem Graustufenbild anhand von Konturen
Klasse	Mehrere Variablen und Methoden werden zu einer Klasse zusammengefasst (z.B. die Klasse Person besteht aus den Variablen Vorname, Nachname, etc.)
Klassenbibliotheken	Bereitstellung verschiedenster Funktionen in einer sogenannten Bibliothek
Komplementär (KG)	Ein Gesellschafter der Kommanditgesellschaft, der mit seinem Vermögen unbeschränkt haftet
Kristallgitter	Darstellungsart von Molekülen oder Atomen, um deren Verbindungen und Aufbau grafisch darzustellen
Leiter	Element, das entweder Strom, Wärme, Licht oder Magnetismus zwischen verschiedenen Orten transportieren kann
Leitungsband	Bestandteil des Bändermodells. Beschreibt die Komponente des Modells, in der sich Elektronen bewegen können und somit den Transport von elektrischer Energie ermöglichen
Liste (Programmierung)	Angeordnete Speicherung von Daten bzw. Objekten
Methode	Ähnlich wie Funktion. Methoden gehören jedoch im Gegensatz zu Funktionen immer zu einer Klasse
Molekül	Teilchen, die sich aus mehreren Atomen zusammensetzen
Neutron	neutrales Teilchen eines Atoms

Objekt	Zusammenfassung verschiedener Eigenschaften/Attribute bzw. Methoden, Bsp.: Ein Auto hat folgende Eigenschaften: Farbe, Kraftstoff, Preis, Marke und folgende Methoden: beschleunigen, bremsen, Messung der Geschwindigkeit usw.
Open-Source	Deutsch: Quelloffen, das heißt der Quellcode des Programms ist für jeden kostenlos und ohne Copyright verfügbar und kann auch erweitert werden
Optische Integration	Möglichst unbemerkbare Einbindung von Photovoltaikmodulen in Häuserfassaden oder Landschaften, um deren Idylle nicht zu stören
Parameter	Übergabewert an eine Methode eines Programms
Photon	Teilchen, aus dem sich elektrische Strahlung zusammensetzt
Photovoltaik	Direkte Umwandlung von Sonnenlicht in elektrischen Strom
Photovoltaik-Zelle	Elektrisches Bauteil, das den Photovoltaikeffekt nutzbar macht
Portierung	Umstellung von einem System auf ein anderes, z.B. in der IT die Umstellung von einem Datenbanksystem (z.B. Oracle) auf ein anderes (z.B. Neo4j)
Programmierschnittstellen	Anbindung von einem Programm zu einen Anderen
Proton	positiv geladenes Teilchen eines Atoms
Raffination	Technische Verfahren zur Gewinnung von Rohstoffen oder technischen Produkten (zum Beispiel die Verarbeitung von rohem Erdöl zu Benzin oder Diesel)

Reflexion	Zurückwerfen von Strahlung an der Oberfläche eines Elements (z.B. die die Spiegelung der Sonnenstrahlen in einem Fenster)
Repository	Unterstützung zur Versionsverwaltung, dienst zur Speicherung der geänderten Daten
Siebedruckverfahren	Das zu druckende Element wird durch ein feinmaschiges Gewebe auf das zu bedruckende Metall gedrückt. Stellen, die nicht bedruckt werden sollen, werden einfach durch eine Schablone im Gewebe freigelassen
Silizium	Halbleitermetall, das unter bestimmten Bedingungen elektrische Leitfähigkeit besitzt. Ist das am häufigsten verwendete Element in der Photovoltaikzellenfertigung
Sobel-Operator	Einer von mehreren Kantendetektions-Filtern in der Bildverarbeitung
Stromkreis	Ein System aus stromleitenden Elementen, das in sich geschlossen ist (z.B. ein Netz aus Kupferkabel oder Silizium-Solarzellen)
Syntax	Ordnung bzw. Korrektheit von Wörtern in einer Sprache. Dies ist speziell bei Programmiersprachen wichtig
Thermische Energie	Energie, die von der Wärme eines Stoffes abhängig ist. Gibt an, welche Energie innerhalb der Elektronenbewegungen gespeichert ist
Valenzband	Bestandteil des Bändermodells. Beschreibt die Komponente des Modells, in der Elektronen fest gebunden sind. Um elektrisch Leiten zu können, müssen Elektronen vom Valenzband in das Leitungsband

Variable	Ein bestimmter Wert innerhalb des Programmcodes (z.B. eine Zahl oder ein kurzer Text)
----------	---

Versionsverwaltungssystem	Dieses System erfasst Änderungen in einem Dokument oder einer Datei und speichert diese in einem Archiv mit einem Zeitstempel
---------------------------	---

Wellenlänge	Abstand zwischen zwei Maximalwerten einer Welle
-------------	---

Wirkungsgrad	Physikalische Kennzahl, die besagt wie viel Prozent der zugeführten Energie (zum Beispiel Sonnenenergie in Form von Sonnenstrahlen) tatsächlich in elektrischen Strom umgewandelt wurden
--------------	--

9.2) Quellen

- .NET Dokumentation. (25. März 2017). Von [https://msdn.microsoft.com/de-de/library/ms747437\(v=vs.110\).aspx](https://msdn.microsoft.com/de-de/library/ms747437(v=vs.110).aspx) abgerufen
- .NET Dokumentation. (25. März 2017). Von [https://msdn.microsoft.com/de-de/library/system.windows.controls.viewport3d\(v=vs.110\).aspx](https://msdn.microsoft.com/de-de/library/system.windows.controls.viewport3d(v=vs.110).aspx) abgerufen
- b-landau.de. (24. März 2017). Von <http://b-landau.de/wp-content/uploads/2010/02/Leitungsband.png> abgerufen
- EmguCV. (29. März 2017). Von http://www.emgu.com/wiki/index.php/Main_Page#Advantage_of_Emgu_CV abgerufen
- EmguCV Dokumentation. (23. März 2017). Von <http://www.emgu.com/wiki/files/3.1.0-r16.12/document/html/fc9004a0-ce7f-2b0b-a090-ad67a7f97fea.htm> abgerufen
- EmguCV Dokumentation. (23. März 2017). Von <http://www.emgu.com/wiki/files/3.1.0-r16.12/document/html/80190679-ef98-e1ef-f6b6-4c8b9f3b7f64.htm> abgerufen
- EmguCV Dokumentation. (23. März 2017). Von <http://www.emgu.com/wiki/files/3.1.0-r16.12/document/html/cb24a129-d9ce-57f3-19ad-0eaa27a77317.htm> abgerufen
- herzo-agenda21.de. (29. März 2017). Von http://www.herzo-agenda21.de/_ilse/solar/solar7.html abgerufen
- Huawei. (22. März 2017). Von <http://www.huawei.com/en/all-products/Solar/References/Yanchi> abgerufen
- Microsoft. (27. März 2017). Von <https://docs.microsoft.com/de-de/dotnet/articles/standard/library> abgerufen
- photovoltaik4all.de. (27. März 2017). Von <http://www.photovoltaik4all.de/solarmodule/jinko-solar-jkm260p-60-poly> abgerufen
- Quaschnig, V. (2017). *Regenerative Energiesysteme*. Hanser Verlag München.
- SFV. (26. März 2017). Von <http://www.sfv.de/lokal/mails/phj/solarzel.htm> abgerufen
- Topgeo. (25. März 2017). Von <http://www.topgeo.de/silizium> abgerufen

Wagner, C. (23. März 2017). *Kantenextraktion - Klassische Verfahren*. Von Universität Ulm:

<http://www.mathematik.uni->

[ulm.de/stochastik/lehre/ws05_06/seminar/ausarbeitung_wagner.pdf](http://www.mathematik.uni-ulm.de/stochastik/lehre/ws05_06/seminar/ausarbeitung_wagner.pdf) abgerufen

Wikipedia. (27. März 2017). Von <https://de.wikipedia.org/wiki/.NET> abgerufen

Wikipedia. (27. März 2017). Von https://de.wikipedia.org/wiki/.NET_Core abgerufen

Wikipedia. (28. März 2017). Von

https://de.wikipedia.org/wiki/Extensible_Application_Markup_Language abgerufen

Wikipedia. (27. März 2017). Von https://de.wikipedia.org/wiki/.NET_Framework abgerufen

Wikipedia. (21. März 2017). Von <https://de.wikipedia.org/wiki/Bitbucket> abgerufen

Wikipedia. (26. März 2017). Von <https://de.wikipedia.org/wiki/Dropbox> abgerufen

Wikipedia. (24. März 2017). Von <https://de.wikipedia.org/wiki/Farbraum> abgerufen

Wikipedia. (23. März 2017). Von <https://de.wikipedia.org/wiki/GanttProject> abgerufen

Wikipedia. (26. März 2017). Von <https://de.wikipedia.org/wiki/Notepad%2B%2B> abgerufen

Wikipedia. (29. März 2017). Von <https://de.wikipedia.org/wiki/OpenCV> abgerufen

Wikipedia. (27. März 2017). Von https://de.wikipedia.org/wiki/Telegram_Messenger

abgerufen

Wikipedia. (21. März 2017). Von https://de.wikipedia.org/wiki/Visual_Studio abgerufen

Wikipedia. (23. März 2017). Von <https://de.wikipedia.org/wiki/Weichzeichnen> abgerufen

Wikipedia. (27. März 2017). Von

https://de.wikipedia.org/wiki/Windows_Presentation_Foundation abgerufen

work-crew.de. (28. März 2017). Von <http://www.work-crew.de/photovoltaik/> abgerufen

9.3) Bilder

Abbildung 1 Projektumfeld	11
Abbildung 2 Projektteam	12
Abbildung 3 Michael Kirchhofer	13
Abbildung 4 Nicklas Harringer	14
Abbildung 5 Tobias Kapplmüller	15
Abbildung 6 Betreuungslehrer DI Christian Aberger	16
Abbildung 7 HTL Perg.....	16
Abbildung 8 Photovoltaikanlage der JMK-Solar KG	17
Abbildung 9 Wechselrichter der drei Modulreihen.....	17
Abbildung 10 Nahaufnahme eines Wechselrichters	18
Abbildung 11 Verteilerkasten der Photovoltaikanlage.....	18
Abbildung 12 Blitzableiter der Solarmodule.....	18
Abbildung 13 Martin Kirchhofer, MSc	19
Abbildung 14 Ningxia Kraftwerk, China (Huawei, 2017)	23
Abbildung 15 Bändermodell (b-landau.de, 2017)	25
Abbildung 16 Reines Silizium (Topgeo, 2017).....	26
Abbildung 17 Vorgänge innerhalb einer Solarzelle bei Bestrahlung (SFV, 2017).....	27
Abbildung 18 Aufbau einer Solarzelle (work-crew.de, 2017).....	28
Abbildung 19 Reihenschaltung von Solarzellen (herzo-agenda21.de, 2017).....	29
Abbildung 20 Visual Studio Logo	31
Abbildung 21 Visual Studio 2015	31
Abbildung 22 Notepad++ Logo	32
Abbildung 23 Notepad++	32
Abbildung 24 Atlassian Bitbucket Logo	33
Abbildung 25 Atlassian SourceTree Logo	33
Abbildung 26 SourceTree.....	34
Abbildung 27 Microsoft Office Logo	35
Abbildung 28 Microsoft Word Logo.....	35
Abbildung 29 Microsoft PowerPoint Logo	35

Abbildung 30 Microsoft Excel Logo	35
Abbildung 31 GanttProject Logo.....	36
Abbildung 32 GanttProject	36
Abbildung 33 GIMP 2 Logo	37
Abbildung 34 Dropbox Logo	37
Abbildung 35 Telegram Logo	38
Abbildung 36 Microsoft .NET Logo	39
Abbildung 37 C# Logo	40
Abbildung 38 XAML.....	41
Abbildung 39 OpenCV Logo	43
Abbildung 40 Grundaufbau des AERODRONE-Fensters	44
Abbildung 41 Liste der ausgewählten Bilder	45
Abbildung 42 3D-Darstellung der Anlage	45
Abbildung 43 Wärmebildaufnahme der Solaranlage	46
Abbildung 44 Anzeige zusätzlicher Daten.....	46
Abbildung 45 Bildauswahlfenster	47
Abbildung 46 Canny Algorithmus - Ausgangsbild	50
Abbildung 47 Bild mit erkannten Kanten.....	51
Abbildung 48 Bild mit Kanten und deren Richtungen	52
Abbildung 49 Bild ohne Nicht-Maxima	53
Abbildung 50 Ergebnis der Kantenerkennung	54
Abbildung 51 Bild einer Photovoltaikanlage nach Anwendung des Canny-Algorithmus.....	55
Abbildung 52 Photovoltaikanlage nach Anwendung des Canny-Algorithmus, erkannte Rechtecke wurden eingezeichnet.....	57
Abbildung 53 Wärmebild im BGR-Farbraum (links) und HLS-Farbraum (rechts).....	59
Abbildung 54 3D-Darstellung einer Anlage mit einem defekten Solarmodul	62
Abbildung 55 Orthografische vs. Perspektivische Projektion (.NET Dokumentation, 2017) ..	63
Abbildung 56 3D-Darstellung inkl. Koordinatensystem.....	64
Abbildung 57 JinkoSolar JKM260P-60 Datenblatt Teil 1 (photovoltaik4all.de, 2017).....	81
Abbildung 58 JinkoSolar JKM260P-60 Datenblatt Teil 2 (Photovoltaik4all, 2017)	82

10) Anhang

10.1) Datenblatt der verwendeten Solarmodule

www.jinkosolar.com

JinKO Solar
Building Your Trust in Solar

JKM265P-60

245-265 Watt

POLYKRISTALLINE MODULE

Positive Leistungstoleranz von 0/+3 %






ISO9001:2008 - ISO14001:2004 - OHSAS18001
zertifiziertes Werk

IEC61215 - IEC61730 - IEC61701 - IEC62716
zertifizierte Produkte

(4BB)



KEY FEATURES

- 
Solarzelle mit 4 Busbars:
 Die Solarzelle mit 4 Busbars verbessert durch die neue Technologie erheblich die Effizienz und das Erscheinungsbild der Module, was sie zur perfekten Lösung für Aufdachanlagen macht.
- 
Hocheffizienz:
 Hohe Moduleffizienz (bis zu 16,19%) durch innovative Herstellungstechnologien
- 
Schwachlichtverhalten:
 Dank modernster Bearbeitungstechnologien von Glas- und Zelloberflächen wird ein exzellentes Schwachlichtverhalten ermöglicht.
- 
Hohe Wetterfestigkeit:
 Zertifiziert um Windlast (2400 Pascal) und Schneelast (5400 Pascal) zu widerstehen.
- 
Langlebigkeit bei extremen Umweltbedingungen:
 Hohe Resistenz gegen Ammoniak und in salzhaltigen Atmosphären zertifiziert von TÜV NORD.



Lineare Leistungsgarantie

10 Jahre Produktgarantie • 25 Jahre lineare Leistungsgarantie

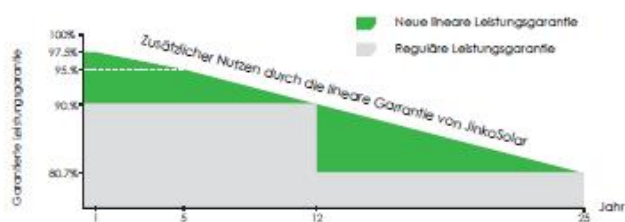
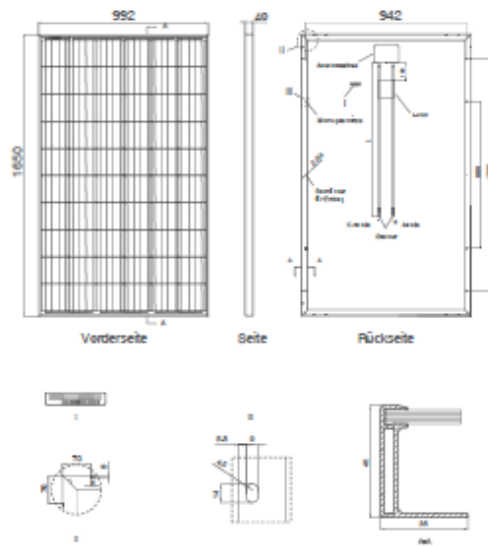


Abbildung 57 JinkoSolar JKM260P-60 Datenblatt Teil 1 (photovoltaik4all.de, 2017)

Technische Zeichnungen

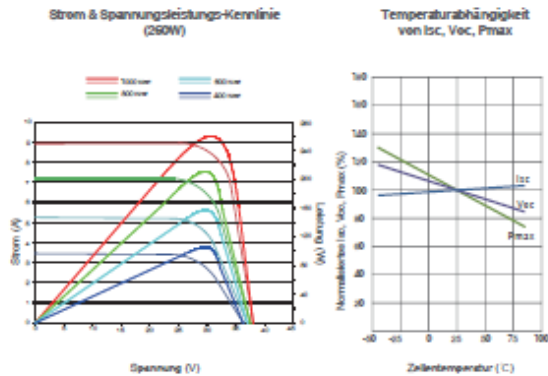


Versandeinheiten

(Zwei Boxen = Eine Palette)

25 Stück/Box, 50 Stück/Palette, 700 Stück/40'HQ Container

Elektrische Leistung & Temperaturabhängigkeit



Mechanische Eigenschaften

Cell Type	Polykristalline 156x156 mm (6 Zoll)
Anzahl der Zellen	60 (6x10)
Abmessungen	1650x992x40 mm (65,00x39,05x1,57 Zoll)
Gewicht	19,0 kg (41,9 lbs.)
Frontglas	3,2 mm, hoher Transmissionsgrad, eisenarmes, temperiertes Glas
Rahmen	Eloxierte Aluminiumlegierung
Anschlussdose	Schutzklasse IP67
Ausgangskabel	TÜV 1x4,0mm ² , Länge 900mm

SPEZIFIKATIONEN

Modell	JKM245P		JKM250P		JKM255P		JKM260P		JKM265P	
	STC	NOCT	STC	NOCT	STC	NOCT	STC	NOCT	STC	NOCT
Maximale Leistung (Pmax)	245Wp	181Wp	250Wp	184Wp	255Wp	189 Wp	260Wp	193Wp	265Wp	197Wp
Optimale Betriebsspannung (Vmp)	30.1V	27.8V	30.5V	28.0V	30.8V	28.5V	31.1V	28.7V	31.4V	29.0V
Optimaler Betriebsstrom (Imp)	8.14A	6.50A	8.20A	6.56A	8.28A	6.63A	8.37A	6.71A	8.44A	6.78A
Leerlaufspannung (Voc)	37.5V	34.8V	37.7V	34.9V	38.0V	35.2V	38.1V	35.2V	38.6V	35.3V
Kurzschlussstrom (Isc)	8.76A	7.16A	8.85A	7.21A	8.92A	7.26A	8.98A	7.31A	9.03A	7.36A
Modulwirkungsgrad (%)	14.97%		15.27%		15.58%		15.89%		16.19%	
Betriebstemperatur(°C)	-40°C - +85°C									
Maximale Systemspannung	1000VDC (IEC)									
Maximaler Bemessungsstrom bei Reihensicherung	15A									
Leistungstoleranz	0 - +3%									
Temperaturkoeffizient von Pmax	-0.41%/°C									
Temperaturkoeffizient von Voc	-0.31%/°C									
Temperaturkoeffizient von Isc	0.06%/°C									
Nenntemperatur bei Zellbetrieb (NOCT)	45±2°C									

STB: Strahlungsintensität 1000W/m² Temperatur der Zelle 25°C Luftmasse = 1,5

NOCT: Strahlungsintensität 800W/m² Temperatur der Umgebung 20°C Luftmasse = 1,5 Windgeschwindigkeit 1m/s

• Messtoleranz: ± 3 %

Das Unternehmen behält sich das Recht auf Änderungen der hier aufgeführten Informationen vor. GE-MKT-265P_v1.0_rev2015

Abbildung 58 JinkoSolar JKM260P-60 Datenblatt Teil 2 (Photovoltaik4all, 2017)