**Von wann bis wann waren die Kameras in Betrieb ?**gab es Ausfälle und wie lange -> Kamera 2 war im Sommer 2018 auss3r Betrieb wegen Ethernet

**Konzept der Vorhersage**- raw Bilder 🡪 Aufbereitung 🡪 debayering (BGGR)  
 (siehe: Processing RAW images in Python in : \Bildverarbeitung\HDR images\Processing RAW images in Python.pdf )  
-Segmentierung Wolkendetektion  
- Paper von Miguel Lopez  
- Berechnung der DNI Solarstrahlung -> nach Soumyabrata 🡪 square cropped Luminance  
 Möglichkeit der Interpolation der fehlenden Sonennpositionen erwähnen  
- Schematischen Ablauf «Flussdiagramm» zeichnen zeigen

**Anforderungen an die Kamera**

- HDR Bilder zur Feature detection der Wolken  
- Debecev Algorithmus zur Bildung der HDR Bilder  
- Möglichkeit RAW – Formate zu verwenden wegen der Linearität (10 bit Bildtiefe gegenüber der sonst 8bit bei JPG Bildern)  
- Höhere Auflösung bessere Voraussetzung für die Erkennung der feinen Strukturen in den Wolken

**Kamera Kalibrierung**- Ocamlib von Scaramuzza 🡪 intrisische und extrinsische Kalibrierung  
- Plot der Luminance gegen die Belichtungszeit, iso = const (sind die Zusammenhänge linear?)  
- Charakteristische Kamerafunktion Siehe auch das Paper dazu  
- Paper: «Laying the foundation to use Raspberry Pi 3 V2 camera module for scientific and engineering purposes»  
- Welche Kalibrierungen müssten vorgenommen werden ?

**Entwicklungsumgebung**- Python :  
 - pandas, numpy  
 - Opencv  
 - pysolar, sunpy, pvlib

**Aufbau der Infrastruktur**- Zwei Gebäude, Ethernet.  
- Cloudlösung für Fernwartung der beiden Kameras 🡪 RealVNC  
- Ground Truth -> Pyranometer auf dem Dach des Trackt IV 🡪 Bilder machen !  
- FTP Server auf dem NAS vorläufig wurden dort dir Bilder gesammelt  
- Pyranometer auf dem Trakt IV beschreiben.  
- MySQL Datenbank auf dem NAS Laufwerk

**Aufbau der Hardware**- Suchen und Finden eines geeigneten Gehäuses.  
- Evaluation passender Fischaugenlinsen (180°)  
- Evaluation von geeigneter Hardware -> Raspberry pi und ähnliche Boards  
- Evaluation von geeigneten Sensoren so z.B. MLX29 -> erwähnen das damit Wolken detektiert werden   
 können (**A method to measure total atmospheric long-wave down-welling radiation using a low cost infrared thermometer tilted to the vertical**)  
 siehe auch Protokol 2017 11.10 Wolken detektieren   
- Welche Sensoren sind eingebaut und warum  
- zwei identische Kameras im Einsatz -> man hätte z.B. versuchen können die Wolkenhöhe zu bestimmen

**Erste Probleme mit beschlagener Optik**- Heizung mittels Drahtwiederstand  
- Trocknungsmittel (Microsive) und Ventilation

**Kamera Software**- Problematik mit der Ansteuerung der Hardware **MMAL** -> Hardwarelayers konnten nicht alle   
 Funktionen kontrollieren und es bleibt offen was die SW tatsächlich gemacht hat. Siehe vor allem das   
 Protokoll vom**: 11.03 2017** dort wird auf die Problematik eingegangen. 🡪 adaptives System für Shutter Zeiten **-** HDR – Fotographie   
- 3 unterschiedliche Versionen 🡪 Gründe weshalb drei Versionen: Abstände zu gross  
- später mit automatischer Nachführung der Belichtungszeit Gründe 🡪 wegen Sättigung der Bilder  
 siehe hierzu auch **Protokoll 2018 09.28 Beschreibung der eigenen Implemetation**

**Bildverarbeitung (Software)**- bimodale Farbräume (HSV, YCbCR ! ) siehe **Protokoll 2018 02.02**- Wolkendetektion/Segmentation, Background subtraction siehe **Protokoll 2018 01.19** -> ground truth ?  
- Optischer Flow Farnback siehe **Protokol 2017 11.27** (dense und sparse) siehe auch 2017 12.01  
- Wolkendetektion und Vorhersage von Rampen  
- Postprocessing: siehe **Protokoll 2018 11.16** -> Processing benötigt 1.5Tg Rechenzeit

**Auswertung der Daten**- Luminance Berechnung aus den Bildern resp. HDR, jpg,  
- Problem mit den Fehlenden Pyranometer Daten -> schwierig Bilder mit den Pyranometer Daten von der   
 Allmend in Bezug zu setzen, da sich die Form der Wolken schnell verändern kann.   
- Auszüge aus den Verläufen der Luminace Berechnung

**Validierung der Wolkendetektion**-zuverlässigkeit der Wolkendetektion validieren z.B mittels : <http://vintage.winklerbros.net/swimseg.html>

**Schlussfolgerung (en):**   
das eine Quantitative Aussage aufgrund der Bildqualität nicht möglich ist.   
Gründe könnten sein: Fisheye Objektiv ist von sehr geringer Qualität.  
Kunststoffhaube ist anfällig für Verschmutzung, Kratzer und deshalb für Reflexionen

**Wo lagen die Schwierigkeiten?**- Rechnen der HDR Bilder ist zu rechenintensiv (hätte in c programmiert werden sollen)  
- Witterung, beschlagen der Kamera Heizung brauch zu viel Strom

**Verbesserungsvorschläge:**- Optik, Kuppel,   
- Preprocessing der HDR Bilder muss bereits auf der Kamera stattfinden Algorithmen muss optimiert werden und wahrscheinlich in einer anderen Programmiersprache geschrieben werden z.B. C.  
- Siehe auch Soumyabrata Dev -> Building a Whole Sky camera