Entwurf und Realisierung: Meteorkamera einer Schülersternwarte

In Zusammenarbeit mit dem Schülerlabor Astronomie des Carl-Fuhlrott-Gymnasiums in Wuppertal

Betreuerin: M.Sc. Jana Späth

Nachgebesserte Version vom 19. November 2025
Ursprüngliche Abgabe am 17. November 2025

Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

INFORMATIONS SYSTEM TECHNIK

INTEGRATED ELECTRONIC SYSTEMS LAB

Carl-Fuhlrott-Gymnasium

Fachbereich Elektrotechnik und Informationstechnik

Inhaltsverzeichnis

Danksagung

Die Durchführung dieses Projekts wäre ohne die Unterstützung zahlreicher Personen nicht möglich gewesen. Ihnen allen möchte ich an dieser Stelle meinen aufrichtigen Dank aussprechen.

Mein besonderer Dank gilt zunächst meiner Betreuerin M.Sc. Jana Späth und dem Leiter des Fachgebiets Integrierte Elektronische Systeme Prof. Dr.-Ing. Klaus Hofmann für die Möglichkeit, dieses Projekt im Rahmen des Fachgebiets durchzuführen.

Die Tatsache, überhaupt eine Meteorkamera auf dem Schuldach bauen zu können, verdanke ich Dipl.-Phys. Michael Winkhaus. Er ist der Leiter des Schülerlabors Astronomie am Carl-Fuhlrott-Gymnasiums in Wuppertal und mein ehemaliger Lehrer. Ohne seine unermüdlichen Initiativen wäre das gesamte Vorhaben nie zustande gekommen. Er stellt nicht nur über Stiftungen die Finanzierung zahlreicher Projekte sicher, sondern hat die Sternwarte als solche ins Leben gerufen und konnte mir schon 2017 das erste Vorhaben zur Digitalisierung der Meteorkamera ermöglichen.

Des Weiteren danke ich Dipl.-Phys. Bernd Koch für seine fachliche Beratung und die damalige Grundausbildung in der Astronomie. Mein Dank gilt ebenso Sophia Haude, die mit dem Bau der analogen Meteorkamera im Jahr 2009 den Grundstein für das vorliegende Projekt legte.

Abschließend bedanke ich mich bei Andreas Hoffmann, einem sehr guten Freund und ehemaligen Kommilitonen, für die wertvolle Unterstützung bei der Entwicklung und die vielen hilfreichen Diskussionen, die das Projekt maßgeblich vorangebracht haben. Ich hoffe, dass ich nie wieder ein altes Ethernetkabel ohne Spezialwerkzeug neu patchen muss.

1 Einführung

Die Analyse von Meteoriten leistet einen essenziellen Beitrag zum Verständnis der Frühzeit des Sonnensystems [1]. Dieses Projekt präsentiert die Konzeption, Realisierung und Nutzung einer automatisierten Meteorkamera. Die resultierenden Daten können in Zusammenarbeit mit anderen Standorten eine genaue Laufbahnberechnung und eine potenzielle Eingrenzung des Einschlaggebietes von Meteoriten ermöglichen [2]. Des Weiteren dient die Kamera der allgemeinen Himmelsbeobachtung und der Dokumentation astronomischer Ereignisse. Insbesondere als Teil der Schülersternwarte des Carl-Fuhlrott-Gymnasiums in Wuppertal ist der didaktische Aspekt von großer Bedeutung.

Die Betriebsweise der hier beschriebenen Meteorkamera vereint dabei Eigenschaften einer modernen AllSky-Kamera und einer analogen Meteorkamera: Sie nimmt zum einen Langzeitbelichtungen auf, welche mehrfach die Sekunde durchbrochen werden, wie in den alten analogen Systemen. Zum anderen wird nicht nur ein Bild pro Nacht aufgenommen, sondern digital ca. eins pro Minute, vergleichbar mit dem digitalen Videosystem einer AllSky-Kamera [3]. Somit lässt sich eine Nacht deutlich schneller durchschauen und das Rauschen ist geringer als in einem Echtzeitvideo; es sind aber im Vergleich zu einer Belichtung feinere Zeitinformationen vorhanden.

Die Meteorkamera auf dem Dach des Carl-Fuhlrott-Gymnasiums hat Historie: Astro-AG-Schülerin Sophia Haude konzipierte und baute 2009 das erste System. Der Betrieb war analog, die Bilder wurden auf Diafilmen belichtet, per Hand musste die Kamera gespannt und per Zeitschaltuhr die Länge der Auslösezeit eingestellt werden. Dies stellte für das von dem deutschen Zentrum für Luft- und Raumfahrt (DLR) betriebene Europäische Feuerkugelnetz den typischen Betrieb da [2].

Die Digitalisierung begann 2017 mit einem Raspberry Pi Einplatinencomputer, der eine Canon EOS 500D Spiegelreflexkamera ansteuerte. Um die Unterbrechung einer Belichtung zu erreichen, wurde eine an einem Gleichstrommotor montierte Blende verwendet. Dieses System gewann die Regionalrunde von Jugend forscht in der Kategorie Technik und war somit in der Landesrunde NRW vertreten. Um die Belichtungsdauer dynamisch anzupassen, wurde das System 2019 mit einem externen Lichtsensor ergänzt, allerdings gab es seitdem Probleme mit der Zuverlässigkeit. Auch ein Zurückkehren zur alten Version wurde erfolglos versucht. Zudem fehlte dem Motor der Blende jeglicher Sensorik, was mindestens zu einer Temperaturabhängigen Schwankung der Geschwindigkeit führte.

Im Rahmen eines Projektseminars am Fachgebiet Integrierte Elektronische Systeme wurde das System neu konzipiert und gebaut um primär die Zuverlässigkeit zu verbessern.

1.1 Astronomische Begriffserklärungen

Zur besseren Verständlichkeit der Arbeit, werden im Folgenden einige Begriffe erläutert, die im Kontext der Meteorbeobachtung von Bedeutung sind.

Die scheinbare Helligkeit in Magnitudo (mag) gibt an, wie hell Himmelskörper einem Beobachter auf der Erde im Vergleich erscheinen. Dieser astronomische Vergleichswert wird anhand einer logarithmischen Skala beschrieben, wobei kleinere oder negative Zahlen für hellere Objekte stehen, während größere positive Zahlen lichtschwächere Objekte darstellen. Das Helligkeitsverhältnis von $5\,\mathrm{mag}$ entspricht $100:1$ und die Skala ist historisch an den hellen Stern Wega gebunden, dem traditionell der Nullpunkt ($0\,\mathrm{mag}$) zugewiesen wird. [4]

Ein *Meteor* ist eine Leuchterscheinung, welche durch den Eintritt eines festen Objekts aus dem Weltraum in eine gasförmige Atmosphäre mit hoher Geschwindigkeit entsteht. Ein Meteor, der $-4\,\mathrm{mag}$ oder heller ist, wird als *Feuerkugel* oder auch *Boliden* bezeichnet und verglüht im Einzelfall nicht immer vollständig in der Atmosphäre [2]. Ein Festkörper natürlichen Ursprungs, der sich im interplanetaren Raum bewegt oder aus diesem kommt, wird allgemein *Meteoroid* genannt, falls dieser eine Größe zwischen ungefähr $30\,\mu\mathrm{m}$ und $1\,\mathrm{m}$ hat. Im Zusammenhang der Meteorbeobachtung kann jedes Objekt, welches einen Meteor erzeugt, Meteoroid genannt werden. Wenn ein natürlicher Festkörper seinen Flug durch die Atmosphäre überlebt und auf den Boden auftrifft, werden die Überreste *Meteorit* genannt. [5]

1.2 Vergleichbare Arbeiten

Das Global Meteor Network (GMN) ist ein weltweites Netzwerk von kostengünstigen, automatisierten Meteorkameras, die auf Raspberry Pi Computern mit 1080p IP Kameras basieren. Das Netzwerk nutzt eine Open-Source-Software und ist so konzipiert, dass es einfach zu installieren und zu warten ist. [6]

Konzeptionell verfolgt das GMN zwar auch die Meteorbeobachtung, hat dabei aber folgende unterschiedliche Schwerpunkte im Vergleich zu der hier beschriebenen Meteorkamera. Das GMN konzentriert sich auf die Massenüberwachung durch viele günstige Systeme, um eine breite Abdeckung zu erreichen und automatisiert Meteorschauer zu analysieren. Zur Einschränkung von Vorhersagemodellen wird der Fluss, die Massenindizes und die Umlaufbahnen der Meteorschauer automatisch berechnet [7]. Gleichzeitig wird sich beim GMN auf die Erfassung von Meteoren in einem breiteren Helligkeitsspektrum konzentriert, während die hier beschriebene Kamera in Bezug auf Meteoriten speziell auf die Erfassung von Feuerkugeln optimiert ist. Das wird durch die Verstärkung und die Wahl des Bildausschnittes erreicht: Das GMN nutzt Objektive, welche ein Sichtfeld von $20 \times 10°$ bis $55 \times 30°$ erlauben [8], während die hier beschriebene Kamera fast den gesamten Nachthimmel abdeckt, was eine allgemeinere Beobachtung auf Kosten der Lichtempfindlichkeit pro scheinbare Fläche ermöglicht. Im GMN Netzwerk werden außerdem keine Farbinformationen des Bildes behalten, sondern es wird monochrom gearbeitet.

Das europäische AllSky7-Feuerball-Netzwerk wurde 2018 in Deutschland gegründet und verwendet verschiedene Versionen der AllSky7 Systeme, die jeweils 7 oder 8 Kameras installiert haben [9]. Da diese Kameras alle Videoaufnahmen machen, wird mit einer sichtbar höheren Verstärkung gearbeitet, um überhaupt ein sichtbares Bild zu erhalten [3].

Die USA haben mit NASA's *All Sky Fireball Network* ein ähnliches Netzwerk aus 17 Kameras, welche jeweils eine nach oben gerichtete Weitwinkelkamera haben [10].

Die Meteorkamera der Schülersternwarte war, wie anfangs beschrieben, Teil des Europäischen Feuerkugelnetzes, das von der Deutsches Zentrum für Luft- und Raumfahrt (DLR) betreut wurde. Die zentralisierte Betreuung wurde im Sommer 2022 eingestellt [2].

2 Implementation

Abbildung 2.1 gibt einen Überblick des Gesamtsystems. Im ersten Teil des Kapitels wird auf die Hardware und die allgemeine Serverkonfiguration eingegangen, im zweiten Teil auf die Software, welche in zwei Hauptkomponenten unterteilt ist: ein Backend zur Kamerasteuerung und Bildverarbeitung, geschrieben in C++, und ein Frontend zur Steuerung des Backends, geschrieben in Python.

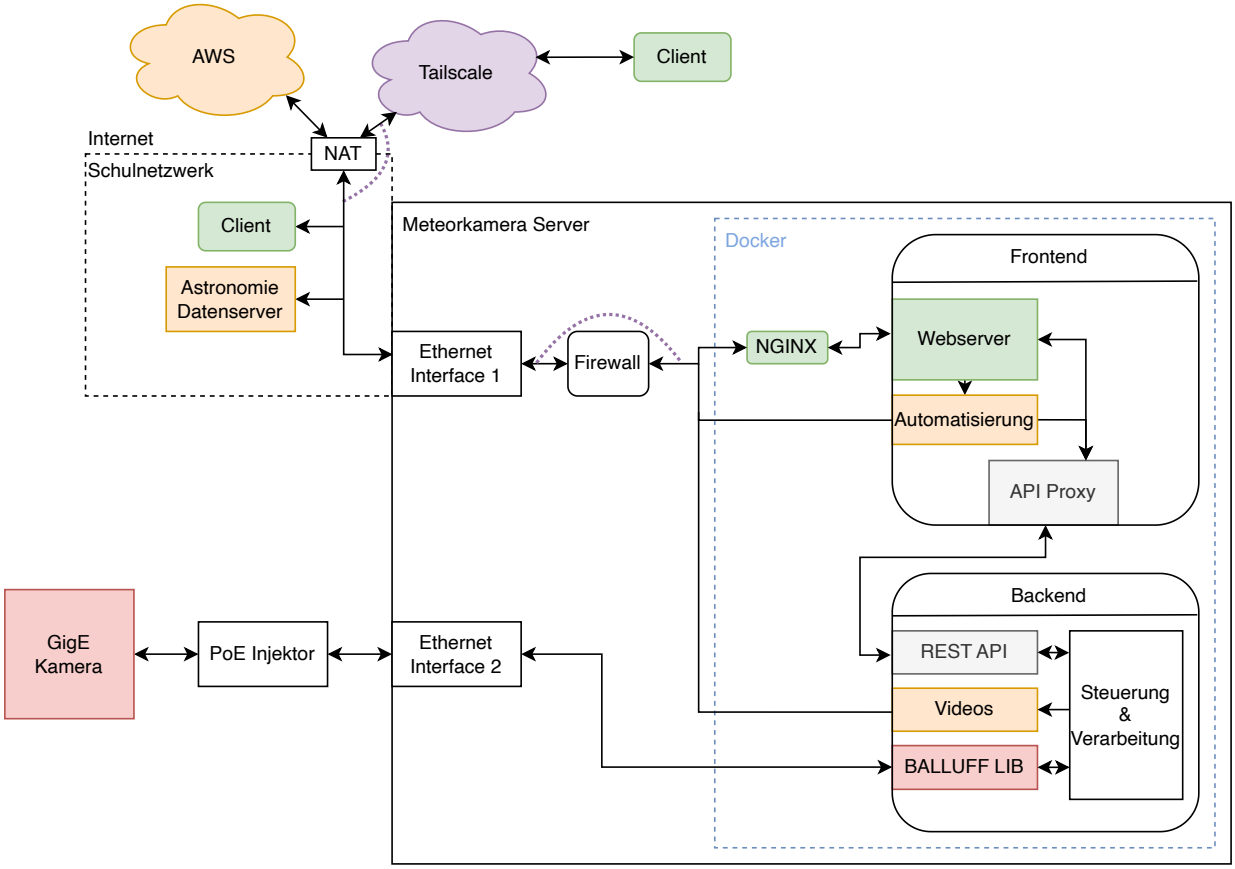


Abbildung 2.1: Übersicht des Gesamtsystems.

2.1 Hardware

2.1.1 Kamera

Der Aufbau der Kamera selbst lässt gut sich aus Abbildung 2.2 entnehmen: Unten ist ein parabolförmiger Spiegel zu sehen, der den Nachthimmel auf die Kamera reflektiert. Die Kamera selbst ist in einem wetterfesten Gehäuse nach unten gerichtet montiert.

Abbildung 2.2: Die Meteorkamera auf dem Dach des Carl-Fuhlrott-Gymnasiums in Wuppertal.

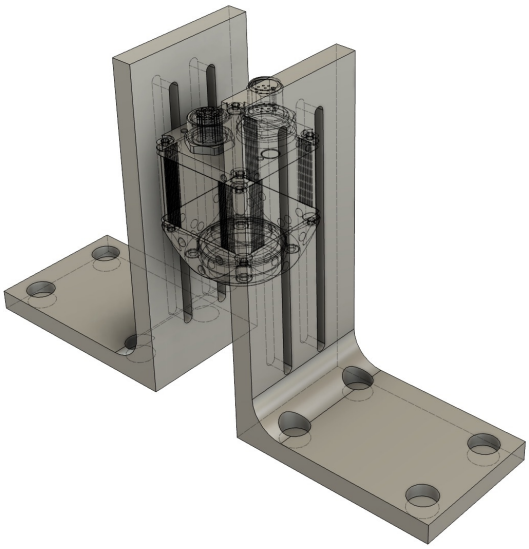
Der beheizte Spiegel ist eine Besonderheit des Systems. Er ermöglicht bessere Aufnahmen bei schlechtem Wetter, da er Kondensation, Nässe und Schnee schnell verdampfen lässt. Solch ein aufwändiger Aufbau wird bei Neuentwicklungen zugunsten von Weitwinkelobjektiven vermieden; hier wurde der Aufbau noch aus der Zeit der analogen Kamera beibehalten.

Die Kamera ist eine *Balluff BVS CA-GX0* Industriekamera der Ausstattung 0081AC-111C40-001, welche IP67 geschützt ist, Power over Ethernet (PoE)-Fähig ist und ein Sony IMX546 verbaut hat. Der Sensor ist in der Farbversion, hat eine Auflösung von 2856 x 2848 Pixeln, eine Größe von $2/3''$ und ist von Sonys Pregius S Serie, welche für eine hohe Empfindlichkeit und ein geringes Rauschen optimiert ist [11]. Die Ansteuerung erfolgt über GigE Vision, einem auf Gigabit Ethernet (GigE) basierendem Standardprotokoll [12].

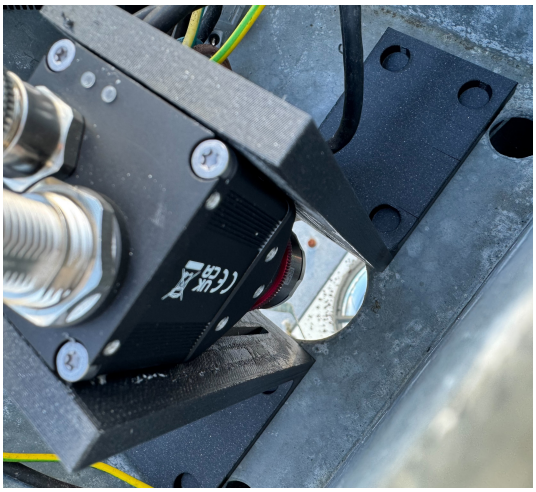
Es wurde sich für diesen Sensor entschieden, da er eine gute Balance zwischen Auflösung und Pixelgröße bietet. Die Pixelgröße von $2{,}74\,\mu\mathrm{m}$ ist ausreichend, um auch lichtschwächere Erscheinungen zu detektieren, während die Gesamtauflösung genug Details liefert. Das Seitenverhältnis beträgt $1:1$, bis auf 8 Pixel Differenz und nutzt somit den runden Spiegel optimal aus. Als Sensor der Pregius Serie verhindert zudem der Global Shutter Verzerrungen bei schnellen Bewegungen, was bei der Aufnahme von schnellen Himmelserscheinungen vorteilhaft ist.

Montiert ist ein Kowa LM25JCM-V C-Mount Objektiv mit einer Brennweite von $25\,\mathrm{mm}$ und einer großen Blendenöffnung von $f/1.4$, was eine gute Lichtempfindlichkeit für Nachtaufnahmen gewährleistet.

Die Kamera ist im Gehäuse über eine dafür entwickelte 3D-gedruckte Halterung fixiert. Diese Halterung besitzt Langlöcher zur Justage der Kamerahöhe, um den Bildausschnitt auf den Spiegel einzustellen, da das Objektiv eine Festbrennweite hat. Die Halterung wird mittels Magneten am Stahlgehäuse befestigt, was das Bohren des feuerverzinkten Stahls vermeidet, wenngleich es keine hochpräzise Justage ermöglicht.



(a) 3D-Modell der Halterung



(b) Montage im Gehäuse (ohne Magneten)

Abbildung 2.3: Modellierung (a) und Montage (b) der Kamerahalterung

Neben Gehäuse und Sensor fiel die Wahl auf ein Balluff Kamerasystem, da der Hersteller auch eine akzeptabel dokumentierte C++ Application Programming Interface (API) zur Ansteuerung der Kamera bereitstellt und die Kamera eine mehrfache Belichtung pro Frame unterstützt. Die recherchierten Konkurrenzprodukte bieten solch eine Funktion nicht, was die RAUSCHER GmbH telefonisch bezüglich Basler Kameras und die Allied Vision Technologies GmbH telefonisch für ihre eigenen Kameras bestätigte.

Dieses Verfahren ermöglicht, dass eine Belichtung alle $1/30$ Sekunde mit einer flexiblen Belichtungszeit gestartet wird, jedoch die aufeinanderfolgenden Belichtungen auf einen Frame summiert werden, wie in Abbildung 2.4 skizziert. Dabei wird noch die Anzahl an Belichtungen pro Frame festgelegt. Der aktuelle Wert von 1500 bedeutet, dass ein Bild $50\,\mathrm{s}$ abbildet. Ohne das Feature müsste die Software 30 Bilder pro Sekunde verarbeiten, was die Komplexität deutlich erhöht hätte und wodurch eine komplexere Kamera ausgewählt werden müsste, die diese Bildrate auch unterstützt. Da so nur alle $50\,\mathrm{s}$ ein Frame zur Verarbeitung anfällt, fallen gewisse Echtzeitkriterien weg und die GigE Geschwindigkeit reicht aus.

Die Maximalbelichtung pro Frame muss klein genug gewählt werden, damit Lücken in Lichterscheinungen sichtbar sind, gleichzeitig lang genug sein, damit auch lichtschwächere Erscheinungen erkannt werden können. Aktuell scheint eine Maximalbelichtungszeit von $28\,571\,\mu\mathrm{s}$ ein guter Kompromiss zu sein.

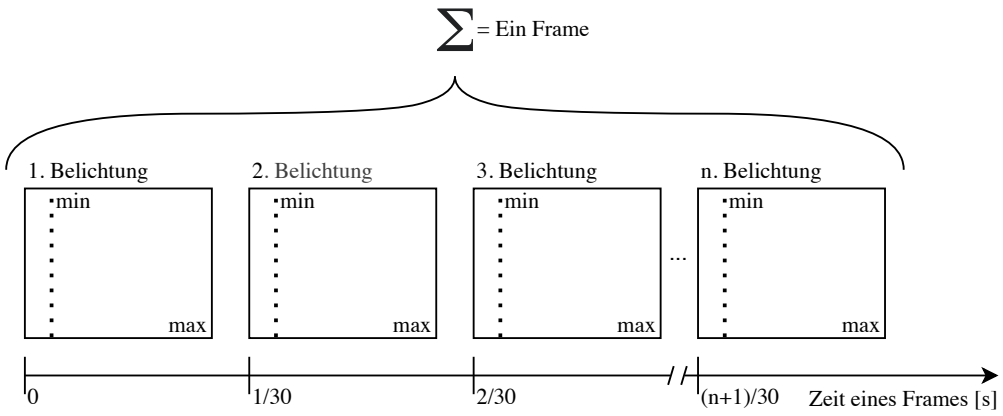


Abbildung 2.4: Veranschaulichung der Mehrfachbelichtung

Mittels eines M12 zu RJ45 Adapter wird die Kamera über ein Standard CAT.6e Netzwerkkabel mit einem PoE Injector verbunden, welcher die Kamera mit Strom versorgt. Am anderen Ende ist der Server angeschlossen.

2.1.2 Server

Die Anforderungen an den Server ähneln denen eines Desktop-PCs: Gute Rechenleistung, jedoch ohne Fokus auf viele Kerne, da keine stark parallelisierbaren Aufgaben anfallen. Es existieren auch keine besonderen Anforderungen an den Massenspeicher, da das Archivieren auf einem gesonderten Server stattfindet. Somit reicht eine SSD mit moderater Kapazität aus. Die einzige wichtige Anforderung ist das Vorhandensein zweier Netzwerkinterfaces: eins für die Kamera, das andere für das Schulnetzwerk. Die Maße des Servers beschränken sich auf 1U Rackhöhe, damit er in das vorhandene Rack passt. Somit fiel die Wahl auf einen *yakkaroo 19"Mini Server 1HE* mit einem aktuellem Intel Core i5-14400 Prozessor, $32\,\mathrm{GB}$ RAM und einer $500\,\mathrm{GB}$ NVMe SSD.

Auf dem Server ist ein *Debian 13.1* Betriebssystem installiert, welches eine stabile und weit verbreitete Linux-Distribution für Serveranwendungen ist. Neben der später beleuchteten Meteorkamera-Software laufen auf dem Server unter anderem auch noch folgende nennenswerte Dienste:

- **NTP-Client:** zur Synchronisation der Systemzeit, was für das genaue Stempeln der Zeit in den Aufnahmen essenziell ist
- **nftables:** für die Firewall-Konfiguration, um den Zugriff auf den Server einzuschränken
- **Docker:** zur Containerisierung und damit einfachen Verwaltung der Software
- **Tailscale:** ermöglicht, zuverlässig auf den Server zuzugreifen, auch wenn dieser hinter der Schul-Firewall und Network Address Translation (NAT) sitzt, vergleichbar mit einem VPN
- **nginx:** Proxy für das Frontend, um den Zugriff auf die Weboberfläche mittels Passwortauthentifizierung zu ermöglichen

2.2 Backend

Das Backend ist die Kernkomponente der Meteorkamera-Software. Es ist direkt für die Ansteuerung der Kamera, die Aufnahme von Bildern und die Verarbeitung der Bilddaten zu Videodateien zuständig.

2.2.1 Technologie-Stack

Das Backend ist in C++ entwickelt, was eine hohe Performance und eine systemnahe Programmierung ermöglicht. Zudem ist die API der Kamera primär in C++ verfügbar, was die Wahl dieser Sprache nahelegt.

- **Bibliotheken:**
  - `Crow` als Webserver für die REST-API
  - `nlohmann/json` für die JSON-Verarbeitung, um Daten zwischen Front- und Backend auszutauschen und um Konfigurationsdateien zu lesen und zu schreiben
  - `mvIMPACT_Acquire` vom Kamerahersteller zur Ansteuerung der Kamera
  - `OpenCV` für die Bildverarbeitung
- **Tests:** Vollautomatisierte Tests wurden nicht implementiert. Lediglich das Kompilieren wird mittels GitHub Actions bei jedem Push getestet. Das manuelle funktionale Testen hat sich als ausreichend erwiesen.
- **Deployment:** Die Anwendung wird mithilfe von Docker containerisiert, was gerade wegen der Abhängigkeit zu der Kamera Bibliothek, die nicht als vorkompiliertes Paket verfügbar ist, eine einfache und reproduzierbare Bereitstellung ermöglicht.

2.2.2 Architektur und Komponenten

Das Programm lässt sich in folgende Hauptkomponenten unterteilen:

- Die **Hauptanwendung** dient als Einstiegspunkt der Anwendung. Umgebungsvariablen für Einstellungen, die nicht über die API gesetzt werden sollen, werden ausgelesen und die benötigten Module werden initialisiert.
- Die **Kamerasteuerung** ist das Modul, das für die Kommunikation mit der Kamera Hardware verantwortlich ist. Es konfiguriert die Kamera, startet und stoppt die Aufnahme und gibt die aufgenommenen Bilder weiter. Für Debugging-Zwecke kann hier auch ein anderes Modul verwendet werden, das Testbilder anstelle von echten Kamerabildern liefert.

- Die **Videoerstellungen** empfangen die Bilder von der Kamera. Hier findet das Zusammenführen der Bilder zu Videos statt, das Einbrennen von Zeitstempeln und die Speicherung der resultierenden Videodateien.
- Der **Webserver** stellt eine REST-API Schnittstelle bereit, über die das Frontend Daten abrufen und Steuerbefehle senden kann.
- Die **Einstellungen** dienen als Modul zur Verwaltung der Konfigurationsdateien, die über die API angepasst werden können.
- Der **Logger** ist zum zentralisiertem Protokollieren von Ereignissen und Fehler konzipiert, um die Überwachung und Fehlersuche zu erleichtern.

Die Beziehungen zwischen den Klassen und Interfaces sind in Abbildung 2.5 dargestellt. Die Hauptanwendung initialisiert dabei `CameraHandler`, `VideoHandler` und `Settings` und gibt dessen Referenzen beim erstellen des `BackendServer` mit. Der `VideoHandler` erstellt dabei intern die Instanzen zu den benötigten `VideoBuilder`. Der benutze `CameraHandler` bekommt einen `FrameSink` als Referenz, explizit den `VideoHandler`, welcher wiederum dann die aufgenommenen Bilder an die `VideoBuilder` weitergibt. Diese verarbeiten das Bild zu Videos. Der Ablauf wird im Abschnitt 2.2.3 beschrieben. Die `Settings` Klasse wird genutzt, um die aktuellen Einstellungen zu lesen und sie erlaubt dem `BackendServer` Einstellungen zu ändern.

Einstellungen werden zwar dadurch zentral verwaltet, jedoch werden Wertkopien von den Komponenten genutzt, sodass Änderungen an den Einstellungen nur explizit übernommen werden. Das sorgt für deterministisches Verhalten und vermeidet Seiteneffekte, falls Einstellungen während der Laufzeit geändert werden.

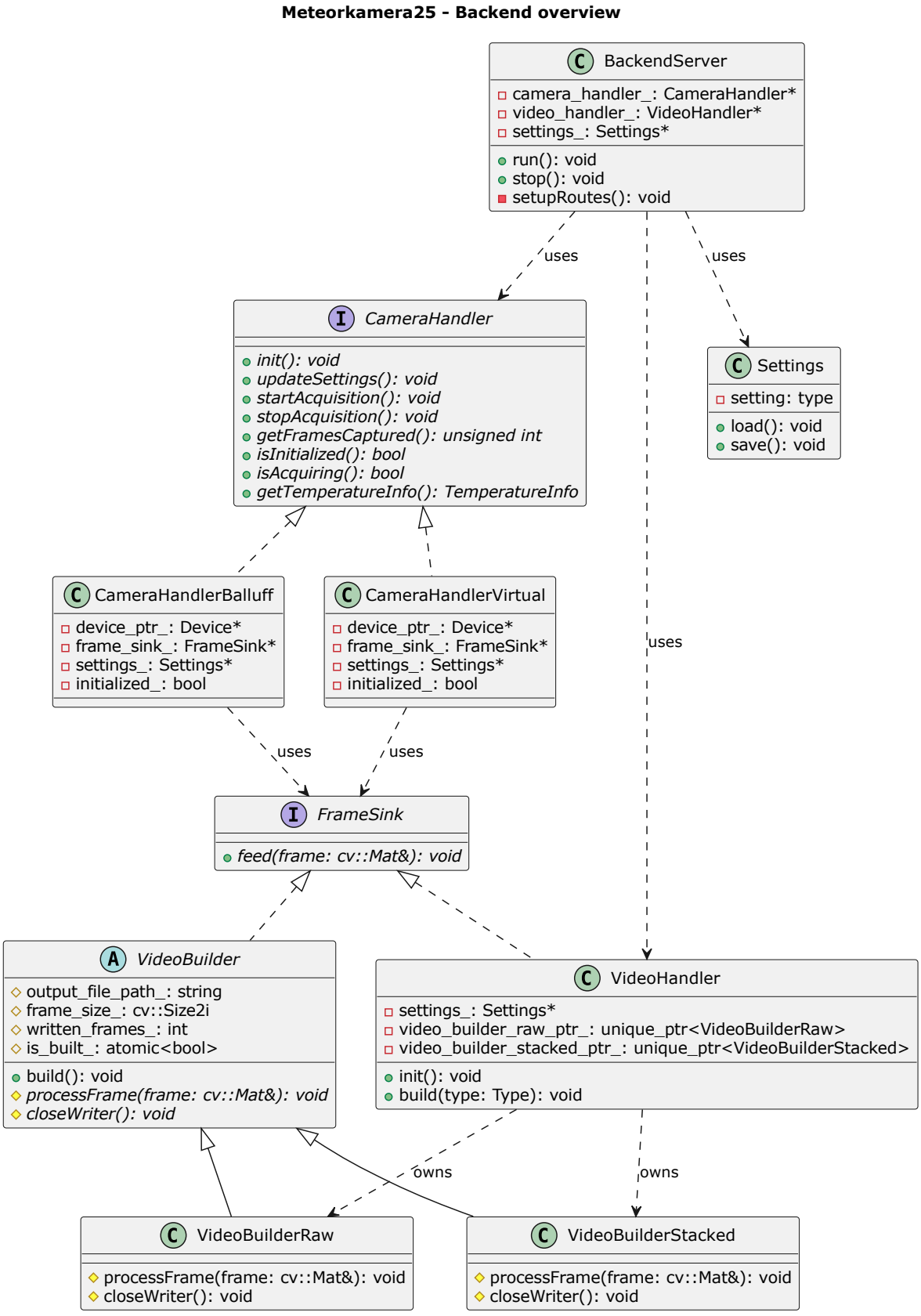


Abbildung 2.5: Vereinfachtes UML-Diagramm der Backend-Architektur.

2.2.3 Bildverarbeitung

Wie die Bildverarbeitung abläuft, ist in Abbildung 2.6 dargestellt. Nachdem ein Frame aufgenommen wird, spiegelt die Kamera diesen senkrecht, da frontal auf einen Spiegel gefilmt wird. Die Kamera übernimmt auch das Debayern, sodass direkt das gespiegelte Farbbild im `BayerRG12Packed` Format vorliegt, also als Farbbild mithilfe eines Bayer RGRB Sensors und mit $12\,\mathrm{bit}$ pro Kanal pro Pixel.

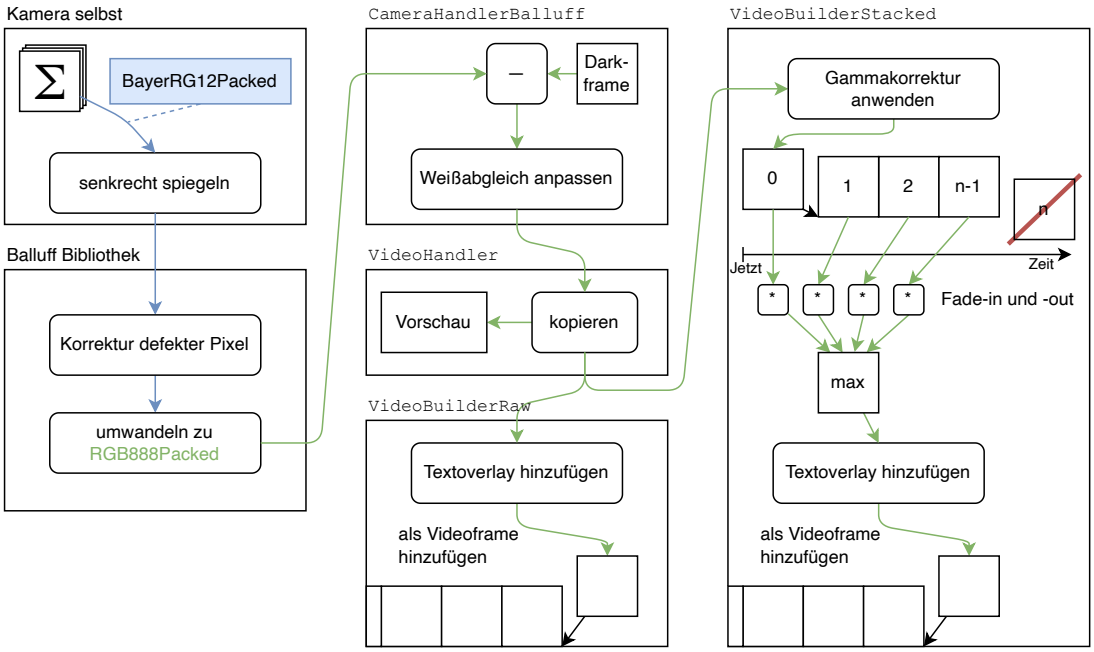


Abbildung 2.6: Übersicht der Arbeitsschritte eines Frames. Die Reihenfolge der Schritte in der Balluff Bilbiothek sind aus [13]. Die Liste an Arbeitsschritten hier ist nicht vollständig.

Die Balluff Bibliothek korrigiert defekte Pixel, indem der Mittelwert der umliegenden Pixel eingesetzt wird. Die Positionen der defekten Pixel sind vom Hersteller in einer statischen Liste hinterlegt, welche in der Kamera gespeichert sind. Zudem ist die Bibliothek so konfiguriert, dass die Bilddaten, als `RGB888Packed` Format vorliegen, also als Farbbild mit $8\,\mathrm{bit}$ pro Kanal pro Pixel in der Reihenfolge Rot, Grün, Blau. Somit kann das Bild in der asynchronen Abrufroutine direkt in ein OpenCV `Mat` Objekt umgewandelt werden, welches intern die Daten im `CV_8UC3` Format speichert, also ebenfalls als Farbbild mit $8\,\mathrm{bit}$ pro Kanal pro Pixel in derselben Reihenfolge.

Ebenfalls in der Abrufroutine wird das Bild noch zweier Korrekturen unterzogen: Zuerst wird ein Dunkelbild subtrahiert, um das Dunkelrauschen zu minimieren. Solches Bildrauschen entsteht durch thermisches Rauschen im Sensor, wobei einzelne Pixel sehr unterschiedlich stark betroffen sind. Das Rauschen und somit die Korrektur sind zwar temperaturabhängig [14], jedoch ist ein

statisches Dunkelbild für die gegebenen Bedingungen vorerst ausreichend. Um das Dunkelbild zu erstellen, muss der Kamerasensor händisch abgedeckt werden und ein Bild mit denselben Parametern aufgenommen werden, wie auch im Betrieb. Hierfür wird die Maximalbelichtungszeit angenommen.

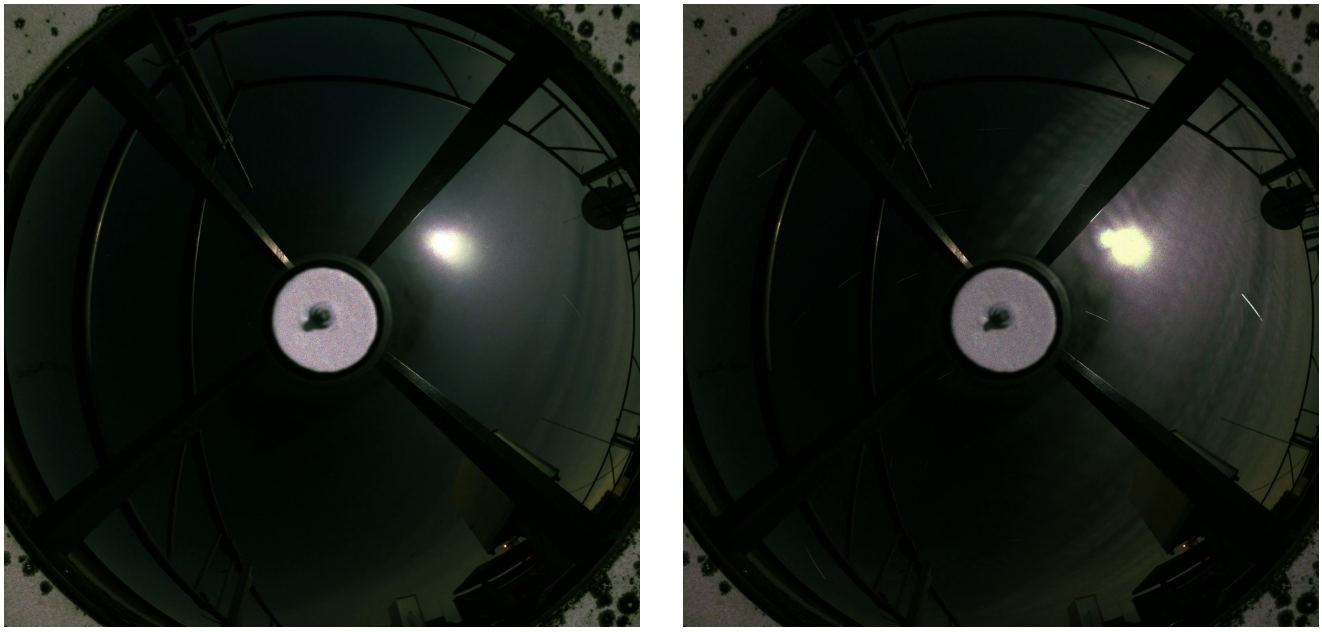
Anschließend wird der Weißabgleich durchgeführt. Da der ganze Nachthimmel aufgenommen wird, ist eine Kalibrierung nicht trivial, da entschieden werden muss, was im Bild als Weiß gilt. Es wurden verschiedene Verfahren zum Weißabgleich getestet u.A. der Grauweltalgorithmus, der annimmt, dass die durchschnittliche Farbe eines Bildes Grau ist, und ein statisches Skalieren der Farbkanäle. Letztendlich wurde die mittige Spiegelmontierung der Kamera weiß angemalt und wird als Referenz für einen dynamischen Weißabgleich genutzt, welcher die Farben so skaliert, dass die Referenz grau erscheint.

Beide Korrekturen könnten auch mittels der Balluff Bibliothek durchgeführt werden, jedoch besteht keine Möglichkeit, das Dunkelbild zu ändern, ohne dieses mittels der Bibliothek aufzunehmen. Da diese Information aber nach Neustarts verloren geht und der Sensor händisch abgedeckt werden muss, fällt diese Möglichkeit weg. Somit muss die Dunkelbildkorrektur und die Korrekturen, die nach der Dunkelbildkorrektur durchgeführt werden sollen, nach der Bibliotheksverarbeitung erfolgen.

Das resultierende Bild wird dann über das `FrameSink` Interface an die `VideoHandler` Instanz weitergegeben. Diese legt einmal eine Kopie an, welche über die REST-API abrufbar ist, und leitet eine weitere Kopie an die `VideoBuilder` Instanzen weiter. Die `Raw` Variante fügt dabei nur noch einen konfigurierbaren Zeitstempel ein und speichert das Bild in einem `H.264 MP4` Video ab.

Die `Stacked` Variante erleichtert die Durchsicht einer Nacht, indem sie n Einzelbilder zu einem einzigen Bild über eine Maximalfunktion zusammenfügt und somit Ereignisse länger sichtbar sind. Es wurde sich für eine Maximalfunktion statt dem Summieren und Normieren entschieden, damit helle Leuchterscheinungen herausstechen und weniger durch Flächen wie Wolken überdeckt werden, wie auch in Abbildung 2.7 zu sehen ist. Zudem wird das durch eine Gammakorrektur unterstützt, wodurch der Helligkeitskontrast angepasst werden kann. Das zeitliche Verhalten lässt sich mit einem Sliding Window oder Warteschlangen Verfahren, wie in Abbildung 2.6 oben rechts skizziert, vergleichen: immer wenn ein neuer Frame hinzukommt, fällt der älteste heraus, wobei sowohl bei den neusten und ältesten Frames ein Fading stattfindet, sodass das Videobild flüssig erscheint.

Die Anzahl n an Frames bestimmt letztlich, wie lange ein Frame sichtbar ist, was das Hauptproblem an der anderen Variante darstellt: Ein Event ist dort potentiell nur $\frac{1}{25}\,\mathrm{s} = 40\,\mathrm{ms}$ zu sehen und kann schnell übersehen werden. In dieser Variante ist ein Event $n \cdot 40\,\mathrm{ms}$ zu sehen, aktuell mit $n = 60$ also $2{,}6\,\mathrm{s}$, was auch mit Fading ausreicht, um Ereignisse zuverlässig zu erkennen. Am Ende wird auch hier ein Zeitstempel des neusten Frames eingefügt und das resultierende Bild in einem `H.264 MP4` Video gespeichert.



(a) Addition und Normierung

(b) Maximalfunktion

Abbildung 2.7: Vergleich zwischen Stacking durch Addition und Normierung (a) und über die Maximalfunktion (b) bei wolkigen Bedingungen. Dieselben Eingangsbilder wurden verwendet.

2.3 Frontend

Das Frontend ist in drei Komponenten unterteilt:

- **Grafische Benutzeroberfläche (GUI)**: Sie erlaubt als klassisches Frontend den Benutzern die Überwachung und die Konfiguration des Front- und Backends.
- **API-Proxy**: Es fungiert als Vermittler zwischen der GUI und dem Backend, indem es API-Anfragen weiterleitet und die Antworten zurückgibt, da das Frontend als Server auf demselben Host läuft wie das Backend. Eine reine Client-Anwendung könnte je nach Netwerk-Konfiguration nicht direkt auf das Backend zugreifen.
- **Automatisierung**: Das Modul startet die Aufnahme zu jeder Nacht und stoppt sie am Morgen automatisch, basierend auf Sonnenauf- und untergangszeiten. Darauffolgend werden die Videos auf einen SMB Server im Schulnetzwerk kopiert und zudem die täglichen online Videos überschrieben, welche auf einem AWS S3 Bucket gehostet werden und über AWS Cloudfront[1][2], bzw. der Schülerlabor-Webseite[3] erreichbar sind.

2.3.1 Technologie-Stack

Das Frontend ist in Python geschrieben, wobei manche visuellen Eigenschaften der GUI mittels CSS angepasst sind. Python ist eine weit verbreitete Sprache mit etablierten Bibliotheken. Da das Frontend auch nicht so systemkritisch ist, wie das Backend, wurden viele Teile mit GitHub Copilot generiert, was die Entwicklungszeit verkürzte.

- **GUI-Framework:** NiceGUI, das die schnelle Erstellung webbasierter Benutzeroberflächen ermöglicht.
- **Weitere Bibliotheken:**
  - `requests` zur Kommunikation mit der Backend-API
  - `python-dateutil` und `astral` für automatisierte Start- und Endzeitpunkte
  - `boto3` für die AWS-Kommunikation
  - `smbprotocol` für den Upload der Videos auf den Schulserver
- **Tests:** Vollautomatisierte Tests wurden implementiert und werden mittels GitHub Actions bei jedem Push ausgeführt.
- **Deployment:** Auch hier eine Containerisierung mit Docker für eine einfache und reproduzierbare Bereitstellung.

[1] `https://d2p560035h4lwi.cloudfront.net/daily.mp4`
[2] `https://d2p560035h4lwi.cloudfront.net/daily_raw.mp4`
[3] `https://www.schuelerlabor-astronomie.de/meteorkamera/`

2.3.2 Grafische Benutzeroberfläche

In Abbildung 2.8 sind Screenshots der GUI zu sehen. Zuerst lässt sich der aktuelle Modus des Frontends erkennen und umschalten. Im Automatikmodus werden die Aufnahmen vor Sonnenuntergang gestartet und nach Sonnenaufgang gestoppt. Im Manuellmodus kann der Benutzer die Aufnahme starten und stoppen. Darunter ist zu sehen, ob aktuell aufgenommen wird. Unter *System Status* steht unter anderem Temperaturen der Kamera und wie viele Frames schon aufgenommen wurden. Unter *Last Frame* lässt sich das zuletzt aufgenommen Bild betrachten.

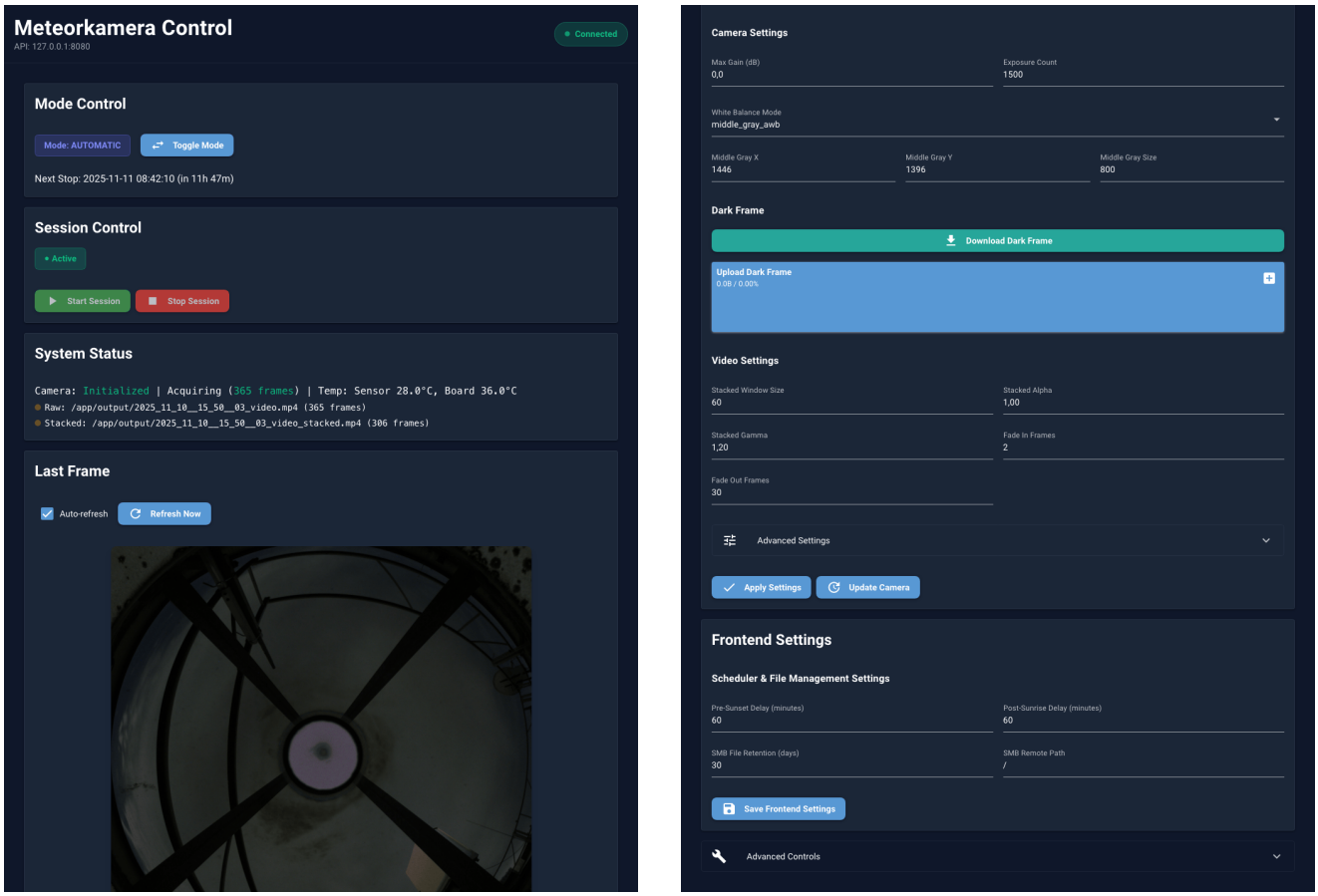


Abbildung 2.8: Screenshots der GUI des Frontends.

Es folgen die Einstellungen der Kamera- und Videoklassen. Am relevantesten sind die Maximalbelichtungszeit, Anzahl der Belichtungen pro Frame, Gammakorrektur und die Anzahl der gestackten Frames. Zudem lässt sich das Dunkelbild aktualisieren. Unter *Frontend Settings* lässt sich bestimmen, wo und wie lange die Videos auf dem Schulserver gespeichert werden sollen und wie viele Minuten vor Sonnenuntergang begonnen und wie viele Minuten nach Sonnenaufgang aufgehört werden soll. Unter dem hier ausgeblendetem *Advanced Controls* lassen sich neben Debuggingoptionen auch die Logs des Backends einsehen.

3 Auswertung

Eine wissenschaftliche Auswertung von potenziell aufgenommenen Meteoren steht nicht im Fokus dieser Arbeit und mögliche Vorgehen werden nur oberflächlich behandelt. Für genaue Erläuterungen kann in den entsprechenden Papern nachgelesen werden.

3.1 Laufbahnberechnung

Zum Berechnen der Laufbahn aufgenommener Meteore kommen sowohl klassische Verfahren wie die *Methode der Ebenen* [15] oder *geradlinige kleinste Quadrate* [16] als auch moderne Verfahren wie der *multiparameter Fit* [17] oder die *dynamische Bahnanpassung* [18] infrage. Die klassischen Verfahren nutzen keine Zeitinformationen, sondern nur die Positionen der Meteore am Himmel von verschiedenen Standorten. Moderne Verfahren nutzen diese aber, um eine genauere Berechnung zu ermöglichen. Gerade die dynamische Bahnanpassung scheint hier vielversprechend, da sie auch mit „längeren Feuerbällen von beträchtlicher Masse“ umgehen können, da sie keine geradlinige Flugbahn annimmt [18].

Welche konkreten Verfahren für die Berechnung genutzt werden können, muss in zukünftigen Arbeiten beurteilt werden. Falls sich für die Eingliederung in ein größeres Meteorkamera-Netzwerk entschieden wird, können die Daten auch an ein zentrales System übermittelt werden, welches die Berechnung der Bahnen übernimmt und womit die manuelle Berechnung wegfällt.

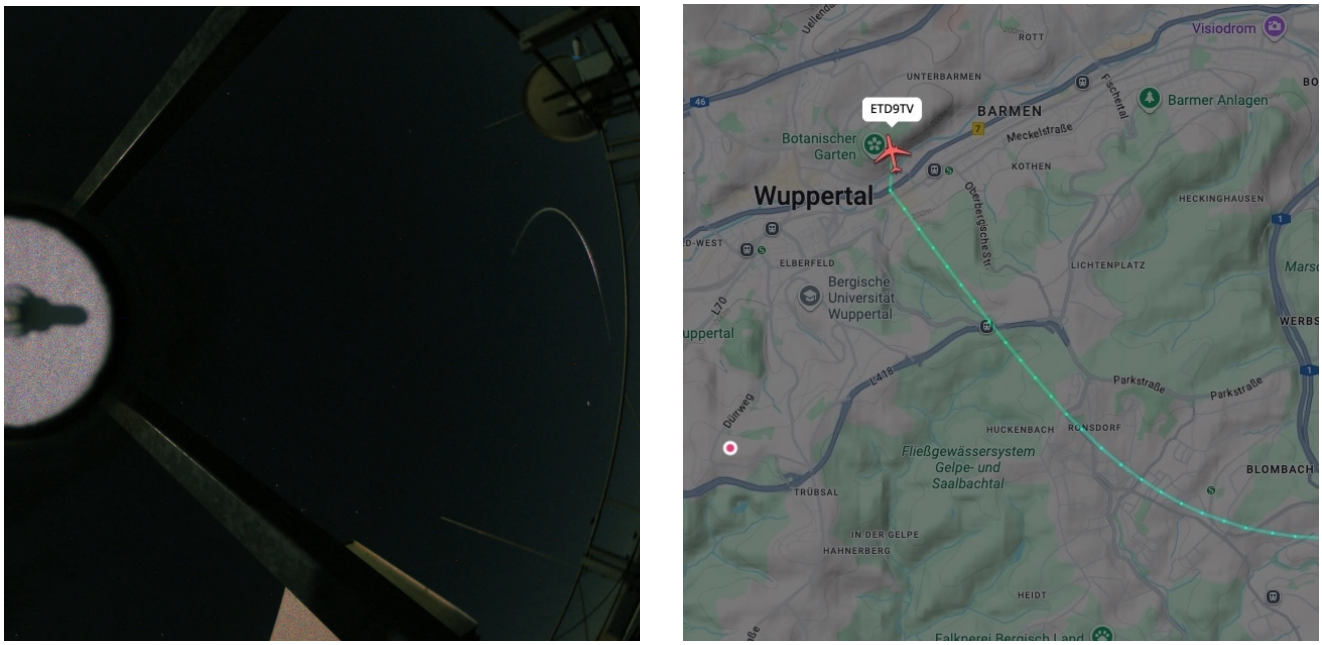
3.2 Beobachtung am 12. November 2025

Am 12. November 2025 wurde zwischen 06:13:59 und 06:14:49 Central European Time (CET) eine Lichterscheinung aufgenommen, die in Abbildung 3.1 zu sehen ist. Ein Unterbrechen der Spur ist nicht eindeutig erkennbar und die Umrisse sind unscharf. Das liegt wahrscheinlich an den Wolken, welche davor waren, was auch die wenigen zu erkennbaren Sterne erklärt. Somit ist eine genaue Analyse nicht möglich. Der Helligkeitsverlauf spricht laut Dipl.-Phys. Bernd Koch, ehemaliger Betreuer des Schülerlabors Astronomie und Lehrbeauftragter für Astronomie und Astrofotografie an der Bergischen Universität Wuppertal [19], für eine Satelliten-Reflexion.

Abbildung 3.1: Ausschnitt vom 11.11.2025 um 06:14:49 CET (Ende der Belichtungen).

3.3 Allgemeine Himmelsbeobachtung

Abbildung 3.2 zeigt als Beispiel für ein nicht astronomisches Ereignis ein Flugzeug, das eine Kurve fliegt. Die Vermutung, dass es sich um ein Flugzeug handelt, folgt aus der Kurve selbst und dass das Licht nicht unterbrochen ist, was für eine geringe Geschwindigkeit der Erscheinung spricht. Durch die Onlineplattform *Flightradar24*, welche Flugbewegungen in Echtzeit, aber auch aus vergangener Zeit anzeigt, konnte die Vermutung bestätigt werden. Demnach handelte es sich um eine Boeing 787-9 der Etihad Airways, welche von Abu Dhabi nach Düsseldorf flog.



(a) Ausschnitt der Aufnahme des Flugzeugs am 07.11.2025 um 05:49:29 CET (Ende der Belichtungen).

(b) Flightradar24-Ansicht des Flugzeugs. Der Punkt auf der linken Seite entspricht der ungefähren Position der Meteorkamera.

Abbildung 3.2: Aufnahme (a) und Flightradar24 Screenshot (b) eines Flugzeugs

Es lässt sich ein deutlicher Mehrwert für die allgemeine Himmelsbeobachtung und die Didaktik im Vergleich zu den vorgestellten anderen Systemen erkennen. Abbildung 3.3 zeigt ein weiteres Standbild, diesmal des gestackten Videomaterials, worauf neben Sternen und dem Mond auch Polarlichter zu erkennen sind, welche auch von anderen Standorten in Deutschland beobachtet wurden [20, 21].

Die visuelle Orientierung ist für Schülerinnen und Schüler besonders einfach, da markante Fixpunkte wie die Brüstung und die Beobachtungsstationen klar erkennbar sind, was vor allem das Nachvollziehen der Himmelsrichtungen erleichtert. Die Bildqualität zeichnet sich durch ein geringes Rauschen bei hoher Auflösung aus. Das eingeblendete Datum und die Uhrzeit in Lokalzeit unterstützen das nachträgliche Nachvollziehen des Beobachtungszeitpunktes.

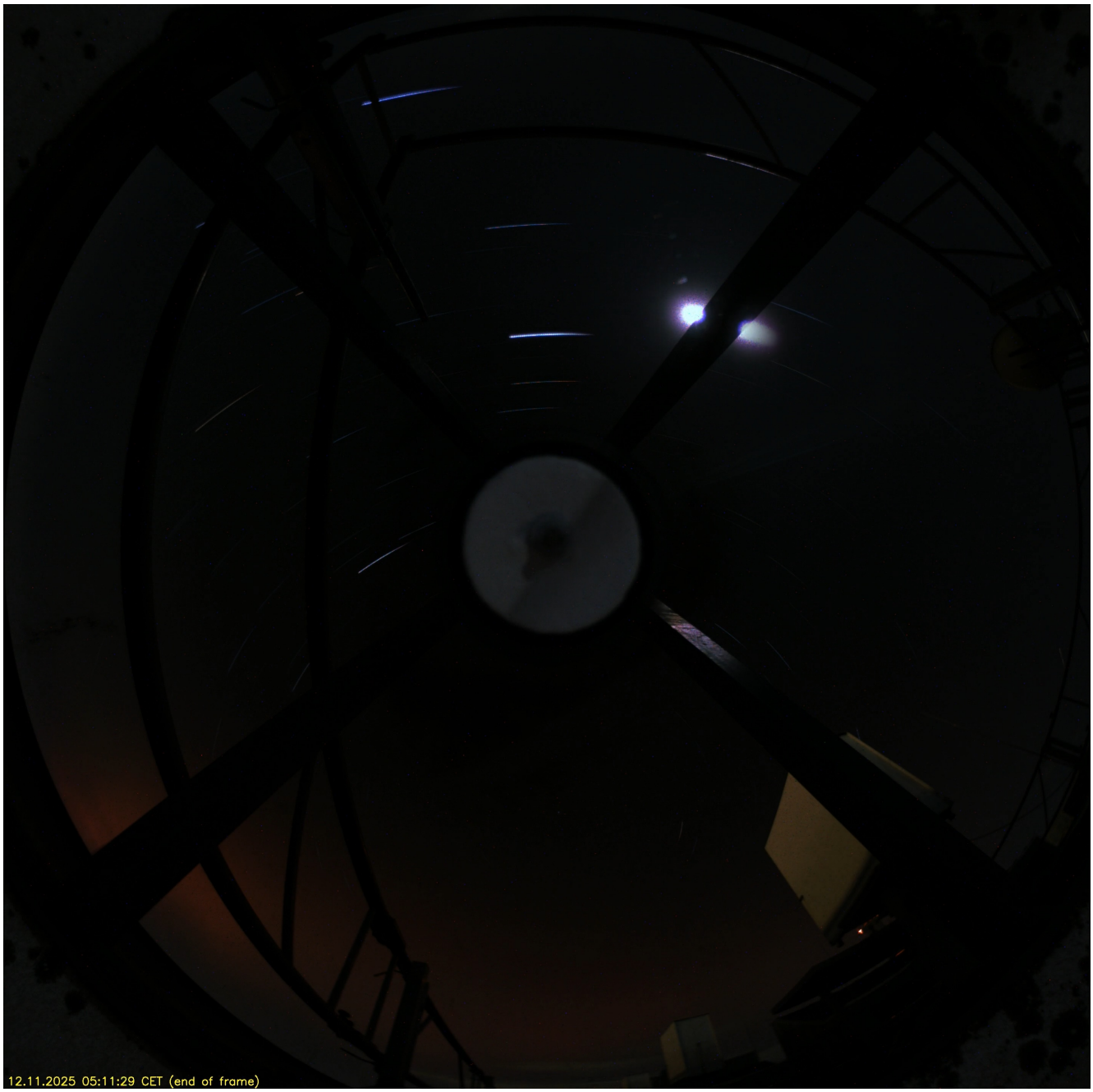
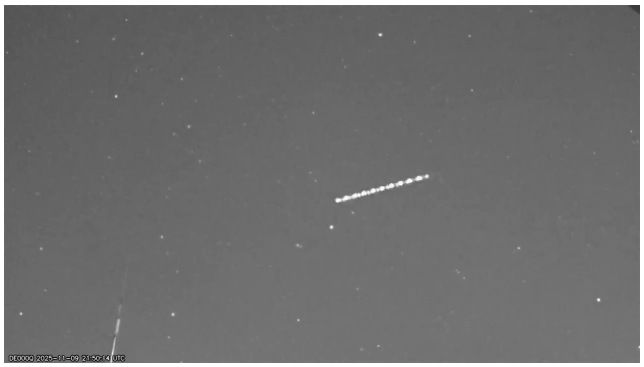
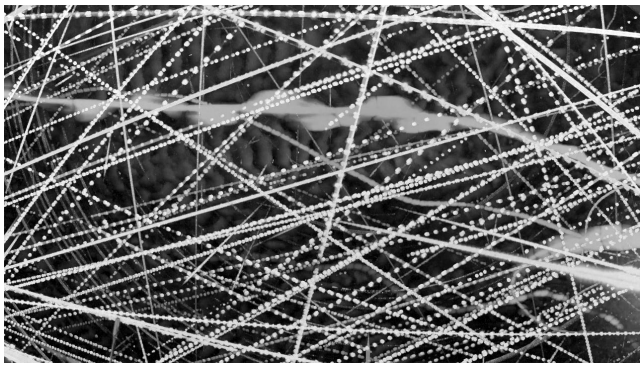


Abbildung 3.3: Standbild einer gestackten Aufnahme vom 12.11.2025 von 4:21 bis 5:11 CET. Unten links sind Polarlichter zu erkennen.

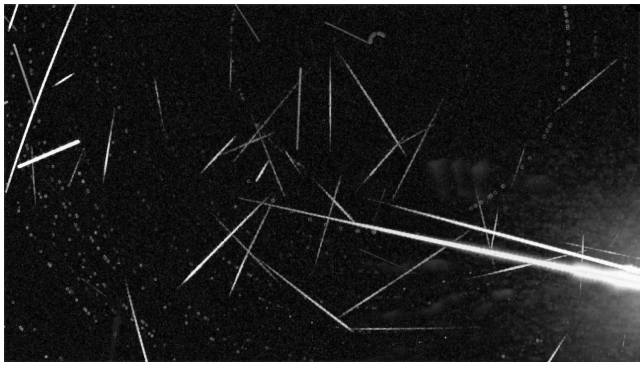
Die Bilder des GMN, wie exemplarisch in Abbildung 3.4, sind monochrom, zeigen nur einen kleinen Ausschnitt des Himmels und das Rauschen und die geringere Aufnahmeauflösung sind deutlich sichtbar. Dies erschwert die didaktische Vermittlung und ein großer Bereich des Himmels vor Ort bleibt unbeobachtet. Der wissenschaftliche Fokus des GMN liegt im Aufnehmen vieler, auch lichtschwächerer, Meteore und Erscheinungen. Zudem findet eine automatische Analyse der Bilder mit dem Netzwerk statt.



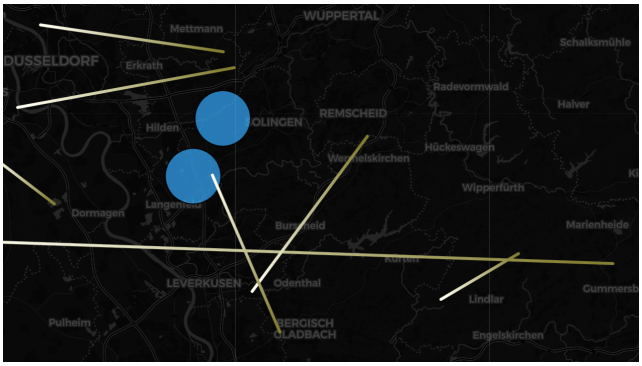
(a) Willkürlicher Ausschnitt der Videoaufnahme.



(b) Gesamte Nacht, gestacked.



(c) Aufgenomme Himmelserscheinungeng, gestacked.



(d) Karten-Webansicht der automatischen Analyse.

Abbildung 3.4: Ausschschnitt von Daten der Nacht vom 09.11.2025 auf den 10.11 der `DE000Q` Station [22] und für (d) auch die umliegenden Stationen [23] des GMN

Anders verhält es sich mit dem europäischen AllSky7 Netzwerk: Die Aufnahmen sind Farbbilder und neue Versionen haben auch eine Weitwinkelkamera. Das Rauschen ist dabei sehr hoch und die Bildauflösung gering, wie in Abbildung 3.5 zu sehen ist. Zudem koexistieren die Aufnahmen der verschiedenen Kameras nur getrennt, sodass der Kontext des restlichen Himmels bei den nicht Weitwinkelaufnahmen fehlt und die didaktische Vermittlung ist schwieriger. Auch hier findet eine automatische Meteorerkennung statt [3].

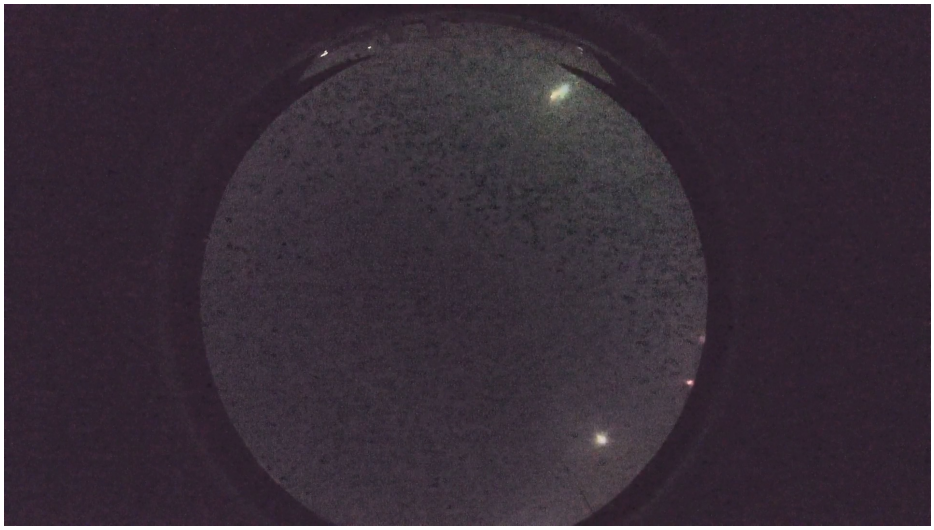


Abbildung 3.5: Feuerkugelaufnahme vom 26.06.2023 der AllSky7 Station `AMS80` aus Garching [9].

3.4 Zuverlässigkeit und Funktionalität

Vom 26. Oktober 2025 bis zum 12. November 2025 konnten keine Ausfälle des Systems festgestellt werden. Die nächtliche Aufnahme startet und stoppt zuverlässig basierend auf den Sonnenauf- und untergangszeiten. Auch die Videoverarbeitung und der Upload auf den Schulserver und AWS S3 funktionieren zuverlässig bis auf einen zweitägigen Ausfall des Schulnetzwerks Anfang November. Die Aufnahmen wurden jedoch lokal auf dem Server zwischengespeichert und konnten nach Wiederherstellung des Netzwerks heruntergeladen werden.

Der Öffentlichkeit stehen täglich das gestackte und das Einzelbildvideo zur Verfügung, Berechtigte der Schule können über den Schulserver auf vergangene Videos zugreifen. Die GUI erlaubt die Überwachung und die Konfiguration des Systems.

Über die Fernwartung und dank der Dockercontainer konnten bisher die Programme zuverlässig aktualisiert und konfiguriert werden.

4 Fazit und Ausblick

Das entwickelte System zur automatisierten Meteorkamera erfüllt die gesetzten Anforderungen an Zuverlässigkeit, Automatisierung und Bildqualität. Die Nutzung einer Industriekamera mit Mehrfachbelichtungsfunktion in Kombination mit einem robusten Server ermöglicht eine stabile und kontinuierliche Beobachtung des Nachthimmels.

Folgende Aspekte bieten Potenzial für zukünftige Verbesserungen und Erweiterungen des Systems.

Die Implementierung einer automatischen Meteor- oder allgemeinen Himmelserscheinungserkennung im Backend wäre ein sinnvoller nächster Schritt. Hierfür könnten Algorithmen zur Bewegungserkennung und Mustererkennung eingesetzt werden, um potenzielle Meteore in den aufgenommenen Videos zu identifizieren und zu markieren. Dies würde die Analyse der Daten erleichtern und die wissenschaftliche Auswertung unterstützen.

Die Integration in ein größeres Netzwerk von Meteorkameras, wie das GMN, könnte die wissenschaftliche Relevanz der Aufnahmen erhöhen. Durch den Austausch von Daten und die Zusammenarbeit mit anderen Stationen könnten genauere Laufbahnberechnungen ermöglicht werden.

Alternativ oder zusätzlich ist das manuelle Hochladen zu übergestellten Netzwerke denkbar, wie der International Meteor Organization (IMO), welche sich auf das Sammeln von Meteorbeobachtungen spezialisiert hat [24], und wird bei der ersten eindeutigen Meteorbeobachtung der Kamera in Erwägung gezogen.

Der Weißabgleich ist nur zufriedenstellend. Aktuell ist oft ein Grünstich zu erkennen, wenn der Mond hell ist und ein Rotstich, wenn die Referenz im Schatten liegt. Die Dämmerungen sehen zudem auch nicht optimal aus, liefern aber auch wegen der Überbelichtung keinen nennenswerten Beitrag. Eine Verbesserung könnte durch die Implementierung eines adaptiven Weißabgleichs erreicht werden, der verschiedene Bereiche des Himmels analysiert und/oder auch mithilfe der Uhrzeit und Mondphase einen besseren Abgleich vornimmt.

Die Darkframes ließen sich eventuell je nach Belichtungszeit anpassen, jedoch scheint es aktuell nicht möglich zu sein, die tatsächliche Belichtungszeit abzurufen. Sonst wäre noch eine Temperaturabhängige Korrektur denkbar, da das Rauschen mit steigender Temperatur zunimmt [14]. Eine Kühlung des Systems ist auch möglich; gerade im Sommer wird das Verhalten nochmal untersucht werden müssen. Alternativ oder zusätzlich lässt sich *Sigma-Clipping* oder *Bias Frames* anwenden, um das Rauschen zu reduzieren, was auch die Empfehlungen von Dipl.-Phys. Bernd Koch sind.

Ob und wie das Bild heller gemacht werden kann, ohne dass ein Rauschen stört, ist noch offen. Ggf. kann hier noch mit der Verstärkung der Kamera, welche aktuell auf das Minimum gestellt ist,

oder der Gammakorrektur experimentiert werden. Die Farben sind auch nicht optimal, hierzu gibt es vom Hersteller zwar Konversionsmatrizen, jedoch scheinen diese keine guten Ergebnisse unter den gegebenen Bedingungen zu liefern.

Das Bildformat der Bilbiothek ist auf `RGB888Packed` eingestellt, was eine gute Balance zwischen Qualität und Datenmenge bietet. Es könnte jedoch untersucht werden, ob ein anderes Format, welches die Rohdaten des Sensors in $12\,\mathrm{bit}$ nutzt, bessere Ergebnisse bei der Bildverarbeitung liefert. Die aktuelle Bibliothek `OpenCV` unterstützt diese Farbtiefe jedoch nicht nativ.

Das Video, das die einzelnen Frames speichert, nutzt den `H.264` Codec, der eine gute Kompression bei akzeptabler Qualität bietet. Es könnte jedoch untersucht werden, ob ein anderer Codec bessere Ergebnisse der Bildqualität liefert, da dieses Video hauptsächlich für die Analyse und nicht für die Durchsicht genutzt wird, wodurch die Dateigröße sekundär ist.

Schließlich konnte der Watchdog Timer des Servers bisher nicht erfolgreich verwendet werden. Trotz Austausch mit dem Hersteller und verschiedener Konfigurationen konnte der Timer nicht durch das Betriebssystem zurückgesetzt werden. Eine weitere Untersuchung und mögliche Zusammenarbeit mit dem Hersteller könnte hier zu einer Lösung führen, um die Ausfallsicherheit des Systems weiter zu erhöhen. Die Wahrscheinlichkeit eines Systemabsturzes auf Betriebssystem- oder Hardwareebene ist aber gering.

Für mich persönlich war die Arbeit an diesem Projekt eine wertvolle Erfahrung, auch wenn einige überraschende Herausforderungen, wie Fehler in der Kamerabibliothek, langsame Kommunikation mit dem Hersteller oder das manuelle Patchen von Netzwerkkabeln bei Wind und Wetter, überwunden werden mussten. Ich freue mich, die Kamera endlich in Betrieb zu sehen, da mein altes Jugend forscht Projekt seit 2019 nicht mehr zuverlässig funktionierte, und hoffe, dass das System wertvolle Beiträge zur Himmelsbeobachtung leisten wird.

Abkürzungsverzeichnis

API Application Programming Interface

CET Central European Time

DLR Deutsches Zentrum für Luft- und Raumfahrt

GigE Gigabit Ethernet

GMN Global Meteor Network

GUI grafische Benutzeroberfläche

IMO International Meteor Organization

NAT Network Address Translation

PoE Power over Ethernet

Literatur

[1] J. F. Kerridge und M. S. Matthews, *Meteorites and the early solar system*. Tucson, AZ, USA: University of Arizona Press, 1988.

[2] D. Heinlein, „Boliden im Blick. Das Deutsche Feuerkugelnetz.," *Sterne und Weltraum*, Jg. 61, Nr. 6, S. 2–7, 2022, Erschienen in der Ausgabe vom Juni 2022. Magazin-Webseite: https://www.spektrum.de/magazin/sterne-und-weltraum/.

[3] M. Hankey, V. Perlerin und D. Meisel, „The all-sky-6 and the Video Meteor Archive system of the AMS Ltd.," *Planetary and Space Science*, Jg. 190, S. 105 005, 2020, ISSN: 0032-0633. DOI: https://doi.org/10.1016/j.pss.2020.105005

[4] IAU Office of Astronomy for Education (OAE), *Glossary term: Apparent Magnitude*, Zugriff am 04. November 2025. Adresse: https://astro4edu.org/resources/glossary/term/15/

[5] I. A. U. (IAU). „Definitions of Terms in Meteor Astronomy (Approved 2017)." IAU Commission F1 on Meteors, Meteorites and Interplanetary Dust. Adresse: https://iauarchive.eso.org/static/science/scientific_bodies/commissions/f1/meteordefinitions_approved.pdf

[6] D. Vida u. a., „The Global Meteor Network - Methodology and first results," *Monthly Notices of the Royal Astronomical Society*, Jg. 506, Nr. 4, S. 5046–5074, Aug. 2021, ISSN: 0035-8711. DOI: 10.1093/mnras/stab2008

[7] D. Vida, R. C. Blaauw Erskine, P. G. Brown, J. Kambulow, M. Campbell-Brown und M. J. Mazur, „Computing optical meteor flux using global meteor network data," *Monthly Notices of the Royal Astronomical Society*, Jg. 515, Nr. 2, S. 2322–2339, Juni 2022, ISSN: 1365-2966. DOI: 10.1093/mnras/stac1766

[8] *Lens Options*, https://globalmeteornetwork.org/wiki/index.php?title=Lens_Options, Zuletzt abgerufen am 9. November 2025, Global Meteor Network, 2023.

[9] *AllSky7 Homepage*, https://allsky7.net, Zuletzt abgerufen am 9. November 2025, AllSky7, 2025.

[10] *NASA All Sky Fireball Network*, https://fireballs.ndc.nasa.gov, Zuletzt abgerufen am 9. November 2025, NASA Meteoroid Environment Office (MEO), 2025.

[11] *PREGIUS Global Shutter Technology*, https://www.sony-semicon.com/en/technology/industry/pregius.html, Zuletzt abgerufen am 9. November 2025, Sony Semiconductor Solutions Corporation, 2025.

[12] *GigE Vision - Die industrielle Bildverarbeitung standardisiert*, https://www.balluff.com/en-de/gige-vision, Zuletzt abgerufen am 9. November 2025, Balluff GmbH, 2025.

[13] Balluff. „Image Processing (in Impact Acquire SDK C++ Reference Manual).“ Generiert am 21. Oktober 2025. Adresse: https://assets.balluff.com/documents/DRF_957352_AA_000/ImageProcessing.html

[14] R. Widenhorn, M. M. Blouke, A. Weber, A. Rest und E. Bodegom, „Temperature dependence of dark current in a CCD,“ in *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications III*, SPIE, Bd. 4669, 2002, S. 193–201.

[15] Z. Ceplecha, „Geometric, dynamic, orbital and photometric data on meteoroids from photographic fireball networks,“ *Astronomical Institutes of Czechoslovakia, Bulletin (ISSN 0004-6248), vol. 38, July 1987*, Jg. 38, S. 222–234, 1987.

[16] J. Borovicka, „The comparison of two methods of determining meteor trajectories from photographs,“ *Astronomical Institutes of Czechoslovakia, Bulletin (ISSN 0004-6248), vol. 41, Dec. 1990*, Jg. 41, S. 391–396, 1990.

[17] P. S. Gural, „A new method of meteor trajectory determination applied to multiple unsynchronized video cameras,“ *Meteoritics & Planetary Science*, Jg. 47, Nr. 9, S. 1405–1418, 2012.

[18] T. Jansen-Sturgeon u. a., „A dynamic trajectory fit to multisensor fireball observations,“ *The Astronomical Journal*, Jg. 160, Nr. 4, S. 190, 2020.

[19] B. Koch, *Profilseite*, https://www.physikdidaktik.uni-wuppertal.de/de/personen/bernd-koch/, Zuletzt abgerufen am 12. November 2025, Bergische Universität Wuppertal, Fachgruppe Physikdidaktik, 2025.

[20] B. Weisheit, *Polarlicht nach starkem Sonnenausbruch von AR 4274*, https://www.pro-bw.de/polarlicht-nach-starkem-sonnenausbruch-von-ar-4274/, Zuletzt abgerufen am 12. November 2025, Sternwarte Huchenfeld, 2025.

[21] tagesschau.de, *Polarlichter*, https://www.tagesschau.de/wissen/polarlichter-232.html, Stand: 12. November 2025, 15:03 Uhr; Zuletzt abgerufen am 12. November 2025, tagesschau.de, ARD-aktuell, 2025.

[22] *Eintrag DE000Q: Beginn 2025-11-09 16:27:13 UTC*, https://globalmeteornetwork.org/weblog/DE/DE000Q/DE000Q_20251109_162713_531240_detected/, Automatisierter Weblog-Eintrag für ein Meteordetektionsereignis; Zuletzt abgerufen am 10. November 2025, Global Meteor Network, 2025.

[23] T. J. Dijkema, *Meteor Map: Meteor trajectory viewer*, https://tammojan.github.io/meteormap/, Visualisierung der GMN- und CAMS-Daten; Zuletzt abgerufen am 10. November 2025, GitHub Pages, 2025.

[24] *About the IMO*, https://www.imo.net/about/imo/, Zuletzt abgerufen am 10. November 2025, International Meteor Organization (IMO), 2025.