

**Dokumentation der Seminararbeit „ISS-  
Überflug“**

**<https://github.com/stereolith/ISSueberflug>**

BM2 | Basissysteme der Informationsverarbeitung 2

Historisch-Kulturwissenschaftliche

Informationsverarbeitung

Universität zu Köln

**Seminararbeit Geographisches  
Informationssystem:  
ISS-Überflug**

von

**Lukas M**

**[https://github.com/stereolith/  
ISSueberflug](https://github.com/stereolith/ISSueberflug)**

## **Inhaltsverzeichnis**

1. Einleitung .....	3
1.1 Ziele .....	3
2. Technische Umsetzung.....	4
2.1 Funktion und Bedienung.....	4
2.1.1 Erforderliche Plugins .....	4
2.1.2 Navigieren und Benutzen .....	4
2.2 Produktionsprozess .....	6
2.2.1 Grundkarte.....	6
2.2.2 Layer „Lichtverschmutzung“ .....	6
2.2.3 Layer „ISS“ & TimeManager .....	7
2.2.4 ISS Ground track.....	8
2.2.5 Layer „UserStandort“ .....	9
3. Fazit .....	10
Quellenverzeichnis .....	12

## **1. Einleitung**

Derzeit umkreisen über 4000 Satelliten die Erde, sowie auch die internationale Forschungsstation ISS. Am Nachthimmel bekommt man davon nicht viel mit, denn die Satelliten (im astronomischen Sinne: ein Objekt im Orbit um ein größeres Objekt) sind von der Erdoberfläche nur schwer zu sehen. Zunächst einmal müssen die Satelliten Licht reflektieren, um auf der Erde gesehen zu werden. Reflektiertes Licht ist am Tag gar nicht, und in der Nacht nur schwer zu sehen. Dazu ist die Sicht auf einen dunklen Nachthimmel durch die steigende Lichtverschmutzung (Lichtquellen, die den Himmel bestrahlen, z.B. Straßenlampen) an immer weniger Orten gegeben. Zudem passen sich die Augen eines Beobachters durch Umgebungslicht nur schlecht an die Dunkelheit an. Selbst bei einem dunklen Nachthimmel lassen sich Satelliten manchmal nicht zwischen helleren Sternen ausmachen.

Die ISS Forschungsstation ist 109 Meter breit und ihre großen Sonnenkollektoren reflektieren viel Sonnenlicht. Fliegt die Station günstig über einen Beobachter, etwa in der Dämmerung, ist sie manchmal sogar als der hellste Punkt am Himmel zu erkennen. Um die Beobachtung der ISS zu erleichtern habe ich eine Vorhersage und Rückschau über ISS-Überflüge visualisiert. Dafür nutze ich ein geographisches Informationssystem, das verschiedene Daten über die ISS und die Lichtverschmutzung erfasst und dem Nutzer hilfreiche Ortungsinformationen zur Sichtung der ISS-Raumstation berechnet.

### **1.1 Ziele**

Das GIS soll dem Nutzer ermöglichen ...

- die Laufbahn der ISS über die Zeit nachzuverfolgen
- den optimalen Ort mit wenig Lichtverschmutzung zu finden
- die ISS bei einem Überflug mit dem Kompass zu orten

## **2. Technische Umsetzung**

Für die technische Umsetzung dieses Projektes nutze ich das freie Open Source Geographische Informationssystem QGIS in der Version 2.18.10 für Windows. Das Programm bietet mit zahlreichen Plug-Ins sowie einer Python Konsole mit Script-Editor viele Erweiterungs- und Anpassungsmöglichkeiten, die auch für dieses Projekt genutzt wurden. Es folgen eine Beschreibung der Funktion sowie Aufzeichnungen zum Erstellungsprozess.

### **2.1 Funktion und Bedienung**

#### **2.1.1 Erforderliche Plugins**

Für die Verwendung des GIS werden die beiden Plugins „TimeManager“ und „ScriptRunner“ benötigt, die manuell installiert werden müssen. Diese müssen vor dem Öffnen des Projektes geladen werden. Die Plugins finden sich in dem Menü „Plugins -> Manage and Install Plugins“ im Reiter „All“, wo man die Plugins „TimeManager“ und „ScriptRunner“ suchen, und mit einem Klick auf „Install plugin“ installieren kann.

#### **2.1.2 Navigieren und Benutzen**

Das Projekt öffnet sich mit einem Doppelklick auf die Projektdatei „ueberflug.qgs“. Der „TimeManager“ sollte sichtbar und aktiviert sein (Abb. 1).


Nun sieht man verschiedene Layer präsentiert (Abb. 2):





- die Position der ISS im Punktelayer „ISS“
- die Laufbahn der ISS als gelbe Linie im Layer „ISS Ground track“
- den Standort eines Beobachters im Vektorlayer „UserStandort“
- die Grundkarte „Stamen Toner (retina)“
- die Information zur Lichtverschmutzung in der Gruppe „Lichtverschmutzung“


Die Grundkarte bietet dem Benutzer Orientierung auf diesem GIS. Die prominente Layergruppe „Lichtverschmutzung“ bietet einen schnellen Überblick über Lichtquellen an einem bestimmten Ort. Die Daten sind nach der Helligkeit an einer Pseudofarbskala von Blau über Gelb bis Rot dargestellt, um die Intensitäten der verschiedenen Lichtquellen genauer unterscheiden zu können (Blau: wenig Licht, Rot: sehr viel Licht). Die gelbe Linie stellt den „Ground track“ der ISS dar – der Pfad auf der Erdoberfläche direkt unter einem Satelliten.

Ist der TimeManager aktiviert, sieht man auf dem Ground track einen roten Punkt mit der Beschriftung „ISS“ (Abb. 3). Dies ist die Position der ISS zu dem Zeitpunkt, der am unteren rechten Rand gezeigt wird.

Die Beschriftungen an dem grünen Punkt im UserStandort-Layer zeigt die Informationen zur Ortung der ISS im Nachthimmel (Abb. 4). Der Azimuth zeigt die Horizontale Differenz zum Nordpol, zum Orten der Richtung zur ISS: Ist der Azimuth beispielsweise  $90^\circ$ , so befindet sich die Raumstation im Osten des Beobachters. Die Elevation ist die Vertikale Höhe der Station zum Horizont. Ist die Elevation negativ, ist die Station nicht zu sehen. Zudem wird hier die Entfernung zur Station angezeigt.

Mit dem „TimeManager“-Plugin, welches sich am unteren Rand befinden sollte, lässt sich die Position der ISS zu verschiedenen Zeiten ansehen. Die aktuelle Zeit ist am unteren rechten Rand des GIS zu sehen. Der Slider im „TimeManager“-Plugin ermöglicht das Anwählen einer bestimmten Zeit. Zudem lässt sich mit dem Play-Button eine Animation der Bewegung der ISS abspielen. Ist der „TimeManager“ ausgeschaltet, sieht man auf der Karte alle Datenpunkte der ISS-Position (Abb. 5). Für die im Folgenden beschreibende Funktion muss der „TimeManager“ *eingeschaltet*  sein.

Im Layer „UserStandort“ kann der User Standort(e) einer Beobachterposition eingeben, um Informationen zur Ortung der ISS zu erhalten. Der Standort des Users lässt sich verändern, indem der Vektorlayer „UserStandort“ bearbeitet wird. Mit einem Klick auf das Bearbeiten-Symbol  startet der Bearbeitungsmodus. Mit  kann ein Punkt hinzugefügt werden und mit  lässt sich ein Punkt verschieben. Beim Erstellen kann eine beliebige numerische ID vergeben werden. Nachdem die gewünschten Punkte gesetzt wurden, muss der Bearbeitungsmodus mit  wieder verlassen werden. Die Label-Daten Distanz, Azimuth und Elevation werden jetzt *noch nicht* angezeigt oder sind noch nicht aktualisiert.

Um die Daten zu berechnen und anzuzeigen, muss ein Python-Skript mit dem Plugin ScriptRunner ausgeführt werden. Dieses findet sich unter „Plugins -> ScriptRunner -> Scriptrunner“. Beim ersten Ausführen muss das Skript „satellit.py“ mit einem Klick auf den +-Button aus dem Projektordner geladen werden. Der Eintrag „satellit.py“ muss ausgewählt sein, um das Skript mit dem Button  auszuführen (Abb. 6). Nun sind die Informationen zur Distanz, zum Azimuth und der Elevation in den Beschriftungen der Punkte zu sehen (Abb. 7). Die Daten werden immer in Bezug zu der aktuell im TimeManager gewählten ISS-Position berechnet.

## **2.2 Produktionsprozess**

Im Folgenden wird der Produktionsprozess dieses GIS dargelegt. Die Entwicklungs- und Designentscheidungen werden nach den Layern geordnet aufgeführt.

### **2.2.1 Grundkarte**

Als Grundlage für dieses Projekt nutze ich das minimalistische OpenStreetMap-Tile „Positron“, welches ich mit Hilfe des Plugins „QuickMapServices“ geladen habe:

Nachdem das „QuickMapServices“-Plugin über das „Plugins -> Manage and Install Plugins“-Menü installiert ist, lässt sich das Suchfenster mit einem Klick auf „Web -> QuickMapServices -> SearchQMS“ aktivieren. Dort lässt sich das Tile „Stamen Toner (retina)“ finden und mit dem „Add“-Button dem Projekt als Rasterlayer hinzufügen.

Diese Karte eignet sich für dieses Projekt, da nur wesentliche Informationen wie Ländergrenzen und Länder- und Städtenamen codiert sind. Die Karte soll dem Nutzer als Orientierung dienen und mit weiteren Sachinformationen durch andere Layer gefüllt werden.

### **2.2.2 Layer „Lichtverschmutzung“**

Um Standorte mit einem möglich dunklen Himmel zu finden, sind Satellitenbilder hilfreich, die nachts die (vor allem künstlichen) Lichtquellen der Erde zeigen. Der „Visible Infrared Imaging Radiometer Suite“-Sensor auf dem „Suomi NPP“-Satellit der NASA liefert passende Bilder dazu. Die Daten lassen sich auf der Webseite der US-amerikanischen Forschungsagentur „National Centers for Environmental Information“ herunterladen.

Für dieses Projekt wurden die Daten aus März 2017 genutzt. Die Daten sind ein Mittelwert aus Satellitenaufnahmen aus dem ganzen Monat, sowie sind die Bilder von Wolken bereinigt. Da die Aufnahmen in sechs „Tiles“ unterteilt sind, wurden diese in diesem GIS gruppiert.

Die Bilder liegen im „.tif“-Format vor und sind bereits GEO-referenziert. Im GIS hat die Visualisierung der tatsächlichen Daten Schwierigkeiten bereitet. Die Bilder werden nur mit einem Farbband geliefert (also: Graustufen), die Verteilung der Werte (Helligkeitswerte) der einzelnen Datenpunkte (Pixel) ist sehr ungleichmäßig: Ein Blick auf das Histogramm (Abb. 8) zeigt, dass die meisten Pixel im Bereich von 0 – 0,9 liegen,

die anderen Werte erstrecken sich bis zu circa 12000. In diesem Projekt wurden Datenpunkte von 0 – 20 betrachtet, da höhere Werte nur sehr selten vorkommen.

Zur besseren Veranschaulichung der fein unterschiedlichen Helligkeitswerte wurde eine Pseudocolor-Darstellung der Singleband-Werte genutzt (Abb. 9). Die genaue Farbzueweisung des „Blau-Gelb-Grün“-Farbverlaufes wurde geändert, damit vor allem die hohen Werte (>17) fein unterschieden werden können.

### 2.2.3 Layer „ISS“ & TimeManager

Das grundsätzliche Problem bei diesem Layer war es, die Laufbahn der ISS-Raumstation in einen Punktelayer abzubilden, der zusätzlich zur Position auch Attribute wie Zeit oder Höhe bereitstellt. Es gibt ein Datenformat, dass die Bahnelemente von Objekten im Erdorbit codiert. Bahnelemente beschreiben die Bahn und Bewegung eines Objektes im Orbit. Das „Two-line element set“ (TLE)-Format codiert diese Daten in Relation zu einer gegebenen Zeit, dem *epoch*. Mit einem Algorithmus („Simplified perturbations model“) lässt sich aus diesen TLE-Daten die Position des Objektes zu einem gewünschten Zeitpunkt berechnen.

Die TLE-Daten für dieses Projekt wurden aus der Datenbank von „space-track.org“ geladen, welche TLE-Daten zu zahlreichen Objekten wie Forschungs- und Kommunikationssatelliten anbietet. Theoretisch lässt sich mit einem TLE-Datum die Position eines Objektes bis weit in die Zukunft bzw. Vergangenheit berechnen, aber da sich die Laufbahnen von Objekten im Erdorbit ständig ändern (z.B. Kurskorrekturen der ISS), werden TLE-Daten für eine genaue Berechnung täglich mehrmals aktualisiert.

Um aus TLE-Daten nun einen Vektorlayer mit Positionspunkten zu erstellen, wurde das Linux/Mac-Tool „Ground-Track-Generator“ genutzt. Es wendet einen „Simplified perturbations models“-Algorithmus auf ein TLE-Datum an und berechnet ein GIS-kompatibles shapefile mit Punkten und Attributen. Die Punkte sind mit mindestens mit dem Zeit-Attribut angegeben, zusätzlich lassen sich noch Daten wie Höhe und Geschwindigkeit abspeichern. Das Intervall und die Zeitspanne der Positions-Punkte lässt sich frei wählen. Ein Beispiel: Der Befehl

```
gtg --input ISS.tle --output ISS_groundtrack --start epoch --end  
epoch+90m --interval 30s
```

erstellt das shapefile „ISS\_groundtrack“ aus dem TLE-Datum „ISS.tle“ mit Positionspunkten im 30s-Intervall vom epoch bis 90 Minuten nach dem epoch. Das hier verwendete TLE-Datum stammt vom 19.09.2017:

ISS 1

```
1 25544U 98067A 17261.22934060 +.00002325 +00000-0 +42231-4 0 9991
2 25544 051.6426 294.4989 0003733 287.6879 189.9670 15.54210685076228
```


Für dieses Projekt wurden die einzelnen Punkte mit der zusätzlichen Option `--attributes standard` mit allen berechneten Informationen (inkl. Höhe und Geschwindigkeit) in der Attributtabelle für die weitere Verarbeitung im „ISS“-Label und Layer „UserStandort“ gespeichert. Ein mit diesem Tool erstelltes shapefile lässt sich in das GIS importieren.

Das Plugin „TimeManager“ nutzt die in der Attributtabelle des „ISS“-Layers gespeicherten Zeiten, um die einzelnen Features (hier: Punkte) an einem Zeitstrahl auswählen zu können. Eine Animation an der Zeitachse entlang ist auch möglich. Die Zeit des aktuell ausgewählten Punktes wird am unteren rechten Rand in der Karte angezeigt.

Da die Zeitangaben aus dem generierten shapefile im ursprünglichen Feld „time“ in UTC im Format „2017-08-14 11:55:19.201248 UTC“ vorliegen, mussten die Werte zunächst mit dem Field Calculator in ein Format gebracht werden, welches der „TimeManager“ nutzen kann. Dies ist mit der Funktion „to\_datetime(„time“)“ möglich. Zudem musste eine entsprechende Rechenoperation angewendet werden, die die Zeit zu MEZ mit Sommerzeit (UTC+2 Stunden) umrechnet:

```
to_datetime("time") + to_interval('2 hours')
```

Das Ergebnis dieser Operation wird im Feld „time\_mez“ gespeichert und im TimeManager als „Start time“ ausgewählt (Settings -> Add Layer).

Ist der TimeManager aktiviert (oben links: ) , wird nur der Punkt auf der Karte angezeigt, der am Zeitstrahl ausgewählt ist.

## 2.2.4 ISS Ground track

Die Linie im Layer „ISS Ground track“ zeigt eine Annäherung der durchgehenden Laufbahn des ISS Ground tracks. Die Line wurde mit dem Plugin „Points2One“ aus den Punkten im Vektorlayer „ISS“ interpoliert. Als Ergebnis wird ein shapefile ausgegeben, welches in das GIS importiert werden kann.



### 2.2.5 Layer „UserStandort“

Dieser Layer stellt mit Punkten eine oder mehrere Beobachterpositionen dar. Für jeden Punkt werden die Daten Distanz, Azimuth und Elevation in Bezug zur aktuell im „TimeManager“ ausgewählten ISS-Position berechnet und in der Attributtabelle unter dem jeweiligen Feld gespeichert. Diese Daten werden in einem Label unter dem Punkt angezeigt.

Zur Berechnung der Werte wird das Python-Skript „satellit.py“ verwendet. Die Python-Konsole in QGIS bietet einen einfachen Editor. Zum Ausführen über ein GUI wird das Plugin „ScriptRunner“ genutzt, damit das Skript nicht händisch über die Kommandozeile ausgeführt werden muss.

Das Skript „satellit.py“ besteht hauptsächlich aus Rechenoperationen. Die geometrischen Rechenoperationen, die unter anderem auch Vektorrechnung beinhalten, wurden aus den unten genannten Quellen bezogen. Diese sind zusätzlich am Ende der jeweiligen Funktionsbeschreibung in Klammern genannt. Für die Einarbeitung in die Python-Sprache wurden die Tutorials von „tutorialspoint.com“ genutzt (Tutorialspoint). Für mathematische Funktionen wie die Berechnung der Quadratwurzel oder des Kosinus wurde die Python-Library „math“ importiert.

Eine Übersicht der Funktionen:

#### **run\_script(iface):**

Die Funktion „run\_script(iface)“ wird vom ScriptRunner-Plugin ausgeführt, wenn ein geladenes Skript mit „Run Script“ gestartet wird. Der Parameter „iface“ muss angenommen werden, benötigt aber keine weitere Verarbeitung in der Funktion.

#### **calculate():**

Diese Funktion ist die Hauptfunktion, die alle Lese-/Speicher und Berechnungsoperationen zusammenführt. Im ersten Teil werden die Positionsdaten des einen Satelliten (hier: die ISS Raumstation) geladen. Im zweiten Teil werden für jeden Punkt im Layer „UserStandort“ folgende Daten mit den unten genannten Funktionen berechnet: Die Distanz, der Azimuth und die Elevation zum oben vorher geladenen Punkt (hier: ISS). Für die Distanz und Elevation werden die „Latitude/Longitude“-Koordinaten in ein kartesisches Koordinatensystem (x|y|z) umgewandelt. Diese berechneten Werte werden danach für den jeweiligen Punkt in die Attributtabelle des Layers „UserStandort“ gespeichert. Diese Werte werden in Labeln im Layer „UserStandort“ angezeigt.

**cAzimuth(a, b):**

„cAzimuth“ berechnet den Azimuth von Punkt a zu Punkt b. Beide Parameter beinhalten Positionsdaten im Latitude/ Longitude-Format. (jeromer)

**cElevation(ap, bp):**

„cElevation“ berechnet die Elevation von Punkt ap zu punkt bp. Beide Parameter beinhalten Positionsdaten im kartesisches Koordinatensystem mit den Variablen x/y/z. (whuber)

**distancePoints(ap, bp):**

„distancePoints(ap, bp)“ berechnet die Distanz zwischen Punkt ap und bp. Beide Parameter beinhalten Positionsdaten, die ebenfalls als x/y/z-Koordinate codiert sind. (Cross)

**locationToPoint(c):**

Diese Funktion konvertiert eine Latitude/ Longitude-Koordinate in eine x/y/z-Koordinate. Für diese Umrechnungen der Koordinaten in ein kartesisches Koordinatensystem müssen die ellipsoide Form der Erde sowie das Umrechnen des geodätischen Breitengrades in einen geozentrischen Breitengrad berücksichtigt werden. Dafür werden die beiden Funktionen „earthRadiusInMeters(latitude)“ respektive „geocentricLatitude(lat)“ genutzt. Weitere Informationen zu diesen Rechnungen findet sich im Quelltext der Implementation von Don Cross. (Cross)

Genauere Beschreibungen finden sich in den Kommentaren im Skript „satellit.py“.

**3. Fazit**

Dieses Projekt lässt sich folgendermaßen erweitern: Zunächst ist es möglich, weitere Objekte aus dem Erdorbit in diesem GIS abzubilden. Mit dem „Ground Track Generator“ lassen sich Vektorlayer für alle Flugbahnen erstellen, für die man ein TLE-Datum als Grundlage vorliegen hat. Die o.g. Plattform „space-track.org“ bietet TLE Daten für 42.000 Objekte. Dies beinhaltet auch Objekte, die sich nicht mehr im Erdorbit befinden. Neben der Planung von Sichtungen ist also auch eine historische Betrachtung denkbar. Die Gruppe der Iridium-Kommunikationssatelliten eignen sich für den Aspekt der

Satellitensichtung besonders, denn deren Antennen reflektieren das Sonnenlicht in der Dämmerung besonders gut auf die Erde (sichtbar als die sogenannten „Iridium Flares“). Es befinden sich 66 Iridium-Satelliten im Orbit. Diese kann man alle in separaten Layern im GIS abbilden. Für die Berechnung der Sehwinkel wäre eine Anpassung nötig, da diese in diesem Projekt nur in Bezug auf *ein* Objekt berechnet werden (hier: ISS).

Des Weiteren wäre eine Visualisierung des Tag/Nacht-Wechsels möglich. Diese kann in Form eines dunklen und hellen Filters, der über die Karte gelegt wird und sich nach der Zeitachse entsprechend bewegt. Diese Information ist für den Nutzer besonders interessant, da Satelliten vor allem in der Dämmerung zu sehen sind, und mit einer entsprechenden Visualisierung genau für diese Zeitspanne geplant werden kann.

QGIS bietet durch die vielseitige Funktionalität ansprechende Visualisierungen für dieses Projekt. Selbst die einem geographischen Informationssystem fernere Thematik von Satelliten und Objekten im Erdbereich lassen sich mit Hilfsmitteln auf eine kartenbasierte Darstellung anwenden.

Die Erweiterbarkeit durch die Python-Konsole bietet vor allem für Datenverarbeitung in Attributtabelle mannigfaltige Möglichkeiten. Dieses Modul ermöglicht für dieses Projekt eine Implementierung geometrischer Berechnungen.

Nicht nur Visualisierungen, sondern auch Datenverarbeitung – und eine Visualisierung der Ergebnisse – machen geographische Informationssysteme zu einem mächtigen Werkzeug zur Nutzung ortsbasierter Sachdaten.

## Quellenverzeichnis

### Sachdaten

- space-track.org für TLE-Daten der ISS-Raumstation
- Earth Observations Group (EOG) at NOAA/NGDC  
[https://ngdc.noaa.gov/eog/viirs/download\\_dnb\\_composites.html](https://ngdc.noaa.gov/eog/viirs/download_dnb_composites.html) für  
Nachtaufnahmen der Erdoberfläche

### Inhaltliche/ Technische Umsetzung

- anoved. 2013. „Ground-Track-Generator.“ <https://github.com/anoved/Ground-Track-Generator> (aufgerufen am 20.09.2017).
- Cross, Don. „Azimuth/Distance calculator.“ Sourcecode der Webseite.  
<http://cosinekitty.com/compass.html> (aufgerufen am 20.09.2017).
- jeromer. 2012. „compassbearing.py.“ <https://gist.github.com/jeromer/2005586>  
(aufgerufen am 20.09.2017).
- Stare, Jurij. 2017. „Help using lightpollutionmap.info.“ FAQ zu  
lightpollutionmap.org. <https://www.lightpollutionmap.info/help.html> (aufgerufen  
am 20.09.2017).
- Tutorialspoint editors, „Python – Tutorial.“  
<https://www.tutorialspoint.com/python> (aufgerufen am 20.09.2017).
- whuber. 2013. „Antwort zu ‚Calculate View Angle‘.“  
<https://gis.stackexchange.com/a/58926> (aufgerufen am 20.09.2017).
- Wikipedia contributors, "Two-line element set," *Wikipedia, The Free Encyclopedia*,  
[https://en.wikipedia.org/w/index.php?title=Two-  
line\\_element\\_set&oldid=790226837](https://en.wikipedia.org/w/index.php?title=Two-line_element_set&oldid=790226837) (aufgerufen am 20.09.2017).

## Anhang

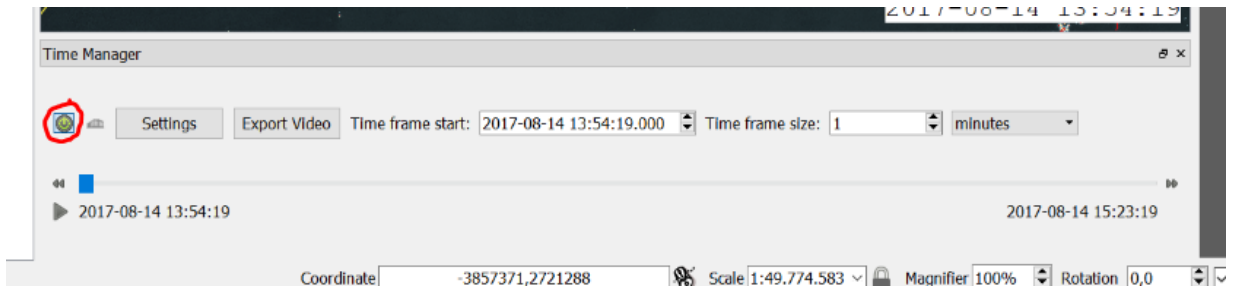


Abb. 1 (TimeManager, aktiviert)

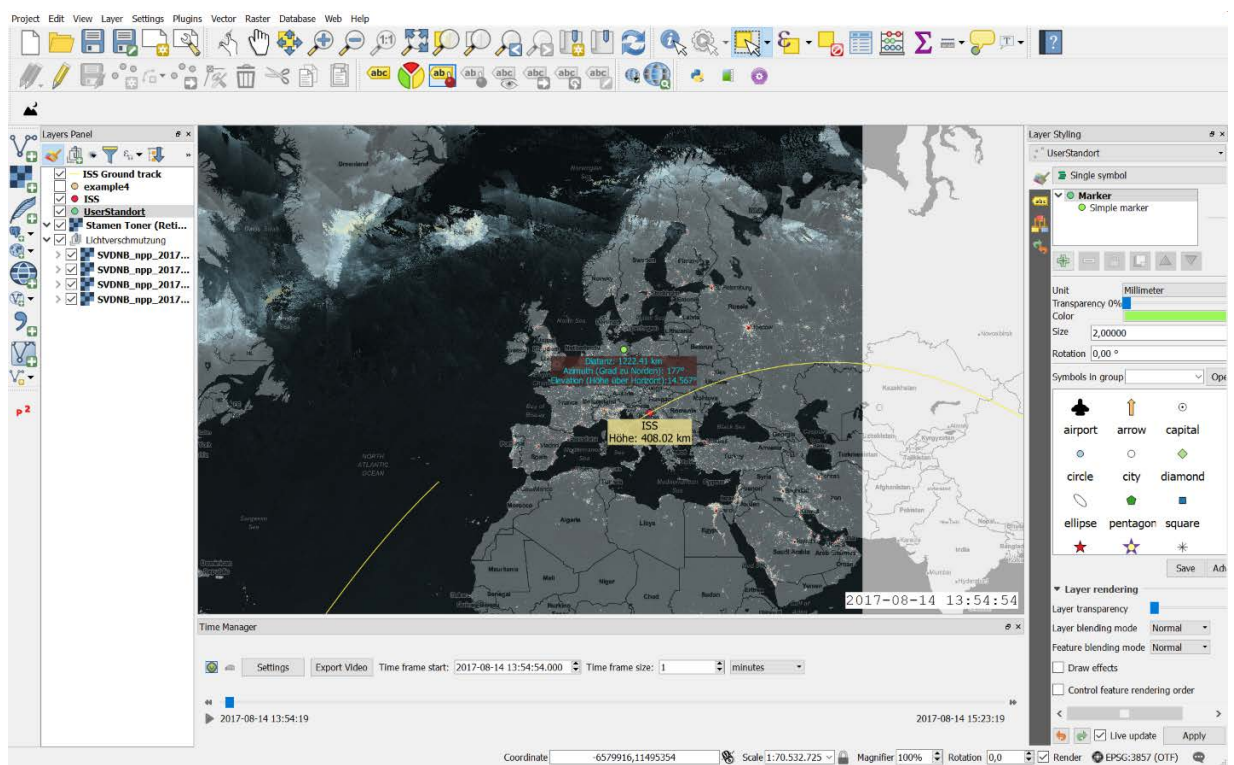


Abb. 2 (geöffnetes Projekt)



Abb. 3 (ausgewählte ISS-Position)



Abb. 4 (Sichtwinkel an Beobachterposition)



Abb. 5 (Karte mit ausgeschaltetem TimeManager)

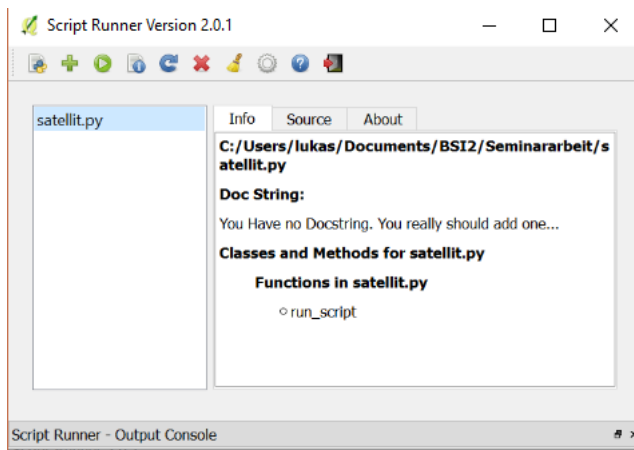


Abb. 6 (ScriptRunner, Skript ausgewählt)



Abb 7 (Berechnete Werte in Beschriftungen der UserStandort-Punkte)

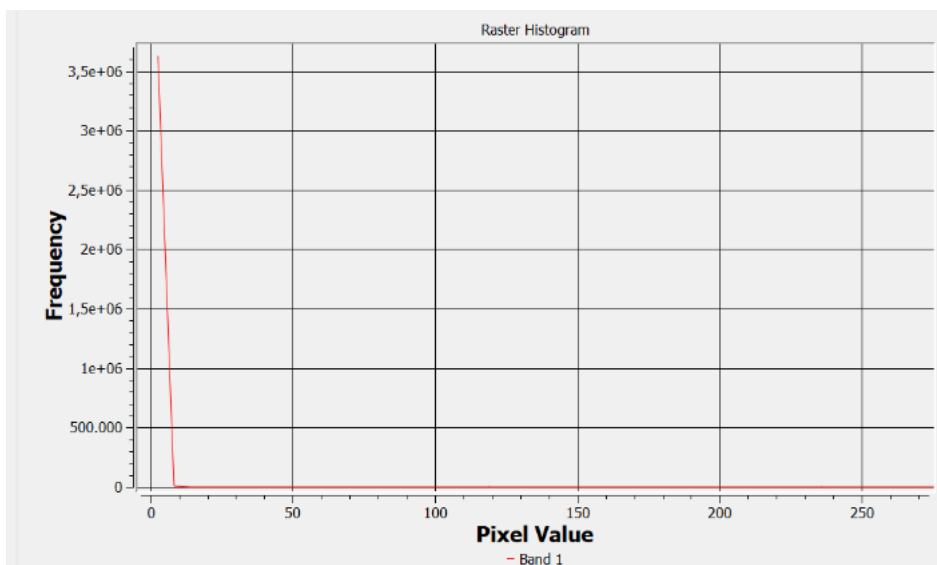


Abb. 8 (Histogramm der Rohdaten des Erdbeobachtungssatelliten)



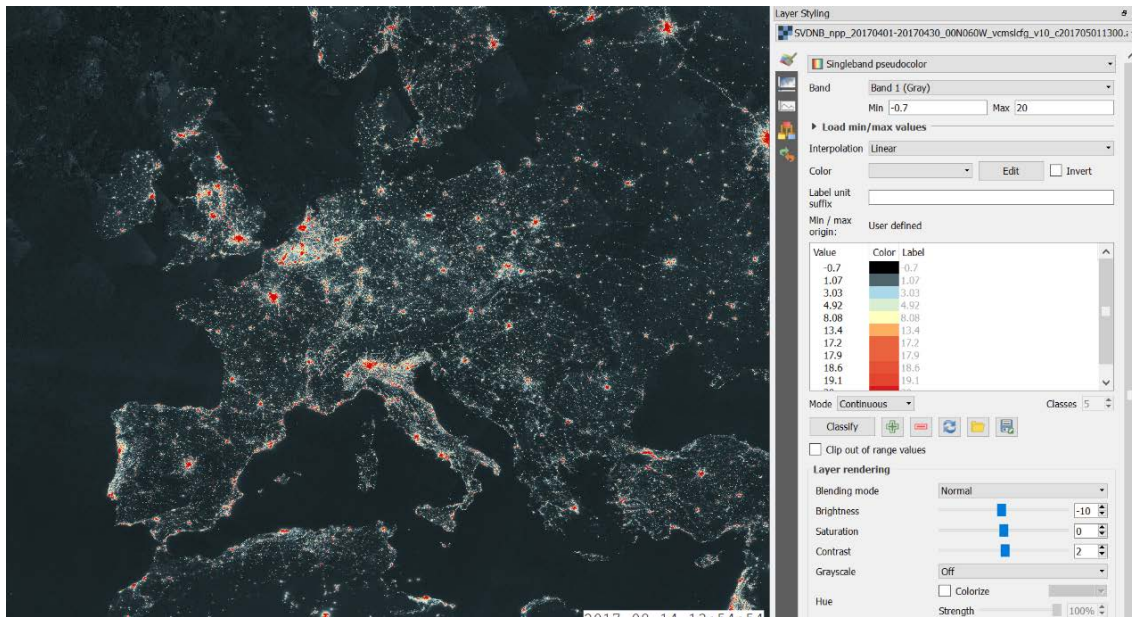


Abb. 9 (Darstellung der Satelliten-Rohdaten in Pseudocolor)