Univeristy of Tartu

Faculty of Science and Technology

Institute of Technology

Rain Eric Haamer

Imaging Simulator and Geometric Image Stitching for a Low Earth Orbit Satellite with High Spin Rates

Masters thesis (30 ECTP)
Computer Engineering Curriculum

Supervisors:

Indrek Sünter
Prof. Gholamreza Anbarjafari

Tartu 2019

Abstract

Pildistamise simulaator ja geomeetriline pilditöötlus maa-lähedase orbiidi ja suure pöördesagedusega satelliidil

ESTCube-2 on maa-lähedase orbiidiga satelliit mille põhieesmärgiks on elektrilise päikesepurje tehnoloogia traadi välja kerimine ja katsetamine. Kõrvaline missioon on pinnase taimestiku pildistamine. Traadi välja kerimine ja alal hoidmine vajab äärmiselt suurt pöördesagedust, mis esitab mitmeid väljakutseid olemasolevale pildistamise alamsüsteemile. Selle probleemi lahendamiseks on välja toodud uus pildimoondus ja -õmblus algoritm, mis kasutab geomeetriat piltide kaardistamiseks. Estitatud meetodit katsetati spetsiaalselt loodud simulatsioonikeskonnas, mis jäljendab kaamera parameetreid. Sõltumata suurest pöördesagedusest tingitud moonutustustest, töötas pilditöötlus algoritm soovitud vahemikus.

CERCS: P520 Astronoomia, kosmoseuuringud, kosmosekeemia; P150 Geomeetria, algebraline topoloogia; T111 Pilditehnika; T320 Kosmosetehnoloogia;

Märksõnad: geomeetria, transformatsioon, satelliit, kiire suunamine , kaamera, pilt, simulatsioon, moonutus, säriaeg, sulgur, õmblus

Imaging Simulator and Geometric Image Stitching for a Low Earth Orbit Satellite with High Spin Rates

ESTCube-2 is a low-earth orbit satellite with the main mission of deploying and testing an electrically charged tether and a secondary mission of photographing ground vegetation. The tether deployment and maintaining its separation from the satellite requires a very high spin rate, which poses too many challenges to the imaging system for it to be viable in its current state. In an attempt to resolve this issue, a novel image morphing and stitching algorithm was developed that uses geometric mapping for reconstruction. The proposed method was tested on a specifically made simulation environment that mimics predefined camera parameters. The stitching algorithm was verified to perform within acceptable margins even when expected high spin distortion models were applied.

CERCS: P520 Astronomy, space research, cosmic chemistry; P150 Geometry, algebraic topology; T111 Imaging, image processing; T320 Space technology;

Keywords: geometry, transformation, satellite, ray-casting, camera, image, simulation, distortion, exposure, shutter, stitching

Contents

List of Figures

Acronyms

ESEO European Student Earth Orbiter. 11

FOV Field of View. 10, 11, 18, 24

LEO Low Earth Orbit. 9

NDVI Normalized difference vegetation index. 11

NIR Near-Infrared. 11

SIFT Scale-invariant feature transform. 7, 27

SURF Speeded-Up Robust Features. 7, 27

TLE Two-line element set. 13

1 Introduction

Satellite based optical surface mapping produces a considerable amount of raw image data which needs to be mapped together. There are numerous well known methods of image stitching already in use that perform well and complete the task as intended. Some of the most well known methods use feature detection and feature matching techniques either based on or similar to Scale-invariant feature transform (SIFT) or Speeded-Up Robust Features (SURF) [41]. These methods work by detecting specific points of interest or landmarks in an image, such as gradients, color transitions or angled corners, which need to be well defined. The landmarks are then matched against points of interest detected on the second target image and a transform can be calculated to map the two feature sets so they are overlaid on top of one another.

Methods based on simple feature matching and image transformations are already used for traditional earth monitoring satellite image stitching, but they contain a flaw. Feature matching based stitching cannot be applied when there are not enough matching features or if the transformations are non-linear [11]. In traditional use cases where photos are taken in series and transmitted continuously or as continuous image arrays, it is generally not an issue. However the nature of the ESTCube-2 mission limits this in several different ways. Photos are not only taken completely separately due to on-board camera buffer limitations, but the images themselves can and will exhibit different types of distortions that reduce the quality of feature matching significantly [32, 39].

As a way of combating this shortcoming, this thesis presents a novel method which uses well known parameters of a satellite orbit and image timestamps to approximate the image locations on a obloid surface using ray-casting techniques. These images are further applicable for traditional stitching methods and allow for a full buildup of an image over several orbital passes. As there currently is not enough properly defined and varied data to create the method purely on real data, simulated images had to be generated. Unfortunately there is a lack of commercially available satellite imaging simulation tools that provide high enough resolution images with the types of distortions that will be present in the ESTCube-2 mission. For this purpose a completely new simulation software had to be created. This allowed the distortions to be completely customized during the generation phases and allowed for better performance reviews of the final system.

The thesis is structured in a way that the opening sections go over the technical difficulties and constrains defined by the ESTCube-2 mission and how they would affect the imaging of the mission. Chapters three and four give a detailed overview of the simulation software including its design difficulties as well as the stitching software itself. The final section in this work reviews the results of the proposed method on simulated satellite imagery. In order to gauge the quality of the simulated images, they are compared against pictures taken by ESTCube-1 from orbit.

2 Challenges

The previous ESTCube mission, ESTCube-1, performed image capture by aiming the camera based on approximate calculations and then taking photos. In order to preserve bandwidth, images were filtered by histogram of pixel values before downlinking. These pictures were then manually referenced and stitched based on the satellites' position in orbit. For a low number of images with proper camera aiming, this is generally not an issue. [44,45,53,56]

It was loosely based on the more common method of planning the attitude of a spacecraft so it would be aimed directly at the target location while flying over it [29,42].

The same method will be again applied to the ESTCube-2 mission for the brief duration before tether deployment. After tether deployment, this method can no longer be used as several new factors come into play. In addition to gyroscopic effects that will be experienced during a high spin only a limited tilt angle is allowed to avoid tether touching the walls of the deployment tunnel.

For the on board camera, this means the satellite camera will always be pointing perpendicular to the earths equator with a direction from the north pole to the south. This will severely hamper any method of aiming the on board camera as the angle will be locked in 2 axis. The second major issue will be caused by the spin itself as this can cause several different camera distortions to occur. [32,39]

2.1 Orbital Parameters

The largest hurdle in creating an image stitching system for a satellite are the constraints imposed on the cameras' motion due to orbital mechanics. Best case scenarios for image stitching require either linear rotation or linear motion [25,55]. A combination of the two impose added complexity but with sufficient image overlap and slow motion this can be easily overcome. In the case of a camera in orbit the dynamics are different, the motion is no longer linear and follows a complex trajectory over the surface of a separately rotating body. This coupled with the linear rotation vector of the satellite itself create complex systems that cannot be directly stitched using conventional methods.

2.1.1 Keplerian Coordinate System

A traditional Keplerian orbit can be described using 6 distinct elements [26,57]. This allows for the mapping of distinct trajectories of an orbiting body in motion. A body in orbit can be described with an ellipse with the center of mass of the system in one of its focuses. The shape of the ellipse itself can be described using two parameters, Eccentricity (ε) the parameter describing the elongation of the ellipse, and Semi-major axis (α) the sum of the periapsis and apoapsis distances divided by two.

The second two parameters define the orbital plane of the ellipse. The first of these is the Inclination (i) which defines the tilt of the ellipse relative to the reference plane, which in the case of the earth is the equatorial plane. The second parameter is the Longitude of the ascending node (Ω) and it defines the horizontal orientation of the ellipse relative to the Vernal Point or the First Point of Aries for the Earth [23]. Argument of periapsis (ω) describes the angle between the ascending node and the periapsis, essentially defining the ellipse's orientation.

The final element called True anomaly (ν) is a variable relative to time and it defines the exact position of an orbiting body in an elliptical orbit. There are two more elements which define the same orbital descriptor called Eccentric anomaly (E) and Mean anomaly (M), the latter being preferred in mathematical notation due to its linear variation with time [40]. All of the aforementioned parameters apart from eccentricity, eccentric anomaly, and mean anomaly are also shown in Fig. 2.3, with True anomaly being shown as θ.

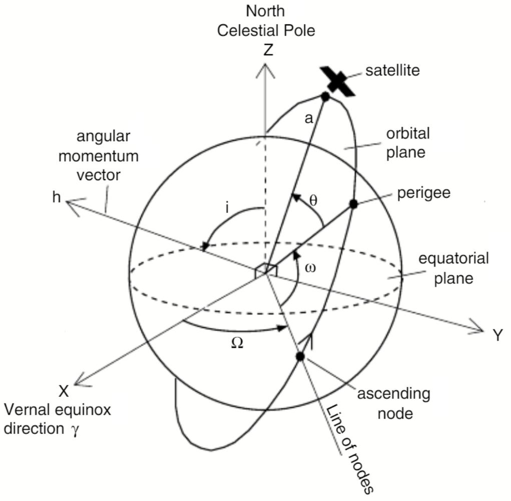

Figure 2.1: Traditional Keplerian orbital parameters [28].

2.1.2 Low Earth Orbit

ESTCube-2 is a Low Earth Orbit (LEO) satellite, meaning the radius of the orbit is going to be below 2000 km, approximately 670 km for a sun-synchronous orbit [24, 32]. This creates a problem of the satellite being relatively close to the surface, which limits the area taken by each image capture. The sparse image captures and the relatively small photographed areas mean that traditional image stitching methods are difficult to implement because of little to no overlap between different images. A way to circumvent this would have been the use of a fish-eye lens, but then the benefits of a telescopic lens with good ground resolution would have been lost.

2.1.3 High Spin Satellite

ESTCube-2's mission statement clarifies that the satellite's main mission is to test the E-sail technology [32]. Deployment of the sail is achieved by spinning the satellite up to near 1 rps

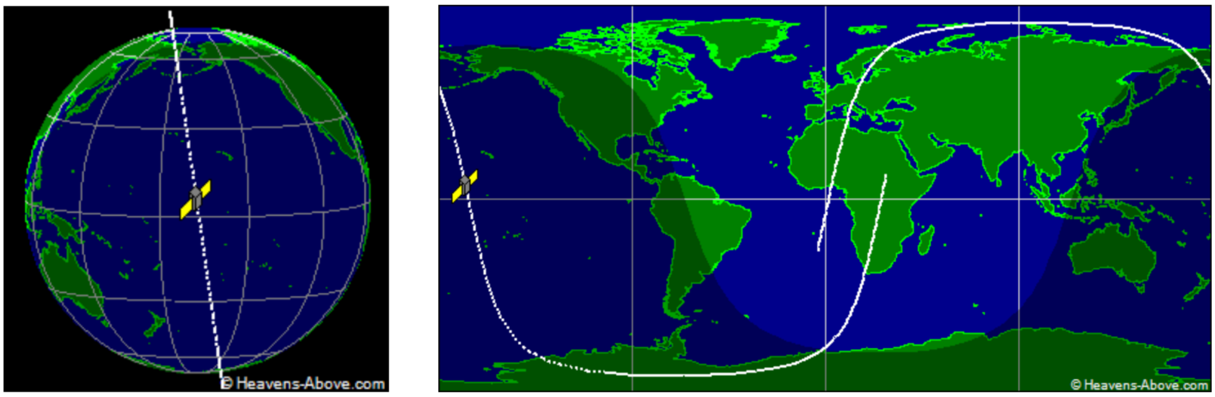

Figure 2.2: Sun-Synchrounous path of ESTCube-1 as of 2019/05/13 from HEAVENS ABOVE [6].

and with the help of a motor wheel, the very fine tether is reeled out. In order to keep the tether taut, the satellite needs to maintain a constant spin with a spin axis perpendicular to Earth's equator. Normally a spinning satellite does not pose a lot of added difficulty to sensors, but in the case of ESTCube-1 and 2, the spin is high enough to affect attitude sensors and visibly influence traditional image capture [31]. This thesis in turn focuses on the problems caused by a spinning optical instrument with a limited image capture rate.

2.1.4 ESTCube Expected Orbit

ESTCube-1 experienced very high controlled spin when the E-sail was being reeled and this is expected to be repeated during ESTCube-2 own tether deployment. The maximum rotational speed recorded during deployment was $841\ ^{\circ}/s$, but during most of the deployment the speed did not exceed $260\ ^{\circ}/s$ [31]. Even though the spin rate for ESTCube-2 should stay near the $260\ ^{\circ}/s$ during spin-up, it is still fast enough to be considered a controlled high spin satellite. [52]

As the satellite for the second mission has not been launched yet, the final orbital parameters are not know. Ideally ESTCube-2 will also be launched into a near-earth sun-synchronous circular orbit just like ESTCube-1 orbit in Fig. 2.2 [24]. The orbit will be at an altitude of about $670\ km$ with a very high inclination of about $98\ ^{\circ}$. This will mean that the satellite is never eclipsed by the earth and will perform a full revolution around the earth every $90 - 100\ min$ [2].

2.2 ESTCube Camera

Based on the nature of the problem, a lot of importance is placed on the actual camera specifications. For raycasting the camera Field of View (FOV) determines the angle of pixel vectors while parameters like shutter speed and exposure time determine the impact of both reversible and irreversible distortions.

2.2.1 ESTCube-1: Camera Specifications

The camera setup used on ESTCube-1 serves as a great baseline for tuning the simulation software. The similarity of the mission statement to that of ESTCube-2 allows for great comparisons of what the satellite might experience in orbit and with what sort of hardware.

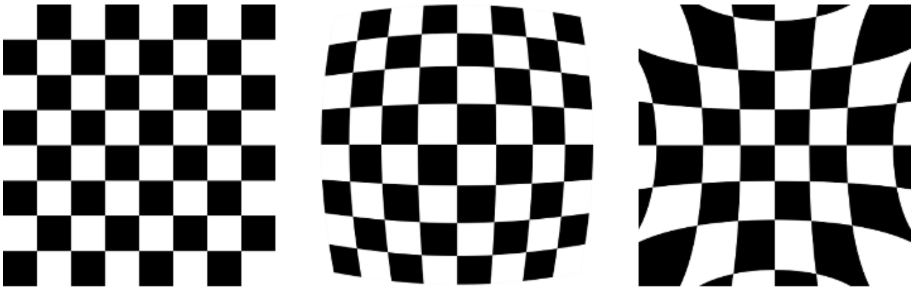

Figure 2.3: Common radial distortion models. From left to right, no distortion, barrel distortion and pincushion distortion [3].

The ESTCube-1 camera used a MT9V011 $1/4$” VGA color CMOS image sensor with a ground resolution of 1 pixel per $1 - 2$ km. At the mentioned resolution the recommended maximum frame-rate is 30 fps at 27 MHz which translates to a rolling shutter of 33.33 ms. The exposure time may be varied and is dependant on outside lighting conditions but for typical CMOS applications, it varies from 0.1 to 20 ms with the highest exposure times ranging up to 500 ms. [1,44]

For the optical setup, the sensor gained an aprture with an Edmund Optics NT57-908 lens that provided a 65.8 ° FOV for a $1/3$” sensor. This meant that the $1/4$” sensor gained a resolution of about 262 per pixel [5, 44]. These parameters were enough in order to create a baseline for simulated images.

2.2.2 ESTCube-2: Camera Specifications

ESTCube-2 will house 2 primary cameras, the cameras will be covered by 2 different wavelength filters, namely red (660 λ / 30 $\Delta\lambda$) and Near-Infrared (NIR) (860 λ / 30 $\Delta\lambda$). These bands enable the calculation of Normalized difference vegetation index (NDVI) which shows the amount of chlorophyll in plants. [36]

As of writing this thesis, the chosen bands are based off of Sentinel-2 camera bands and are not likely to change before launch. The camera specifications are based off of the European Student Earth Orbiter (ESEO) primary camera which employs similar requirements for their imaging optics. [7, 47, 54]

2.2.3 Camera Distortions

Photographs taken with a rolling shutter camera in motion will be governed by 2 distinct distortions, the speed of the shutter and the exposure time. Depending on the camera hardware, each of these parameters could have a profound effect on any motion during capture. Both of these have to be considered as the mission statement defines a high spin satellite. Other camera based distortions such as lens distortion, FOV, background noise, hot pixel correction, aberrations, vignetting, linearity, and spectral responsivity, coupled with physical distortions like atmospheric distortion and atmospheric discoloration to name a few will still be encountered by ESTCube-2 but will not be covered in the scope of this thesis as they have their own existing solutions [35, 38] .

Rolling Shutter Speed

A rolling shutter can cause an effect by the same name on a captured image. A rolling shutter works by sampling each pixel in a row and then moving on to sample the next row. Normally this effect is quite fast and with stationary photographs there is little difference if the first pixel was sampled a fraction of a second before the very last pixel. In the case of high speed motion and a slow shutter, a target may move while sampling pixel rows, resulting in a skewed image. [46]

For the case of ESTCube-2 this can cause the bottom edge pixels to be very heavily rotated resulting in significant but geometrically reversible distortions.

Exposure Time

A bigger problem with a camera in motion, especially in relation to images taken in orbit is the exposure time. If the target, in our case the Earth, is rotating during exposure, then pixels will be exposed to locations that might have been the initial exposure targets of neighbouring pixels. With a long exposure time and a very high spin, the resulting distortion will blur the edges of the captured image in a rotating smudge pattern. This effect cannot be fully restored and might only be circumvented by either lowering the resolution of the images at its edges or by discarding the heavily distorted regions entirely. [21, 48] There are de-blurring methods that do reconstruct images with camera shake but they assume linear or very simple motion paths and cannot function in the above conditions [34].

3 Simulation Software

In order to first evaluate the stitching software, a lot of specific data images were needed. As ESTCube-1 has already finished its mission and ESTCube-2 is yet to be launched, this proposed a very difficult problem. In order to alleviate this, research of available imaging simulation software packages was carried out. The criteria for such a software included in-orbit aimed high resolution image capture with added distortion models. The most notable simulation tools found were NASA Visualization Explorer's Earth From Orbit, Space Engine, and Celestia [4,8,13] but they posed problems with orbital image capture. A closer candidate was VSSGS by TAITUS Software, which provides clear ground segment imaging from a Two-line element set (TLE) sun-synchronous orbit, but the software is not open-source and does not provide image distortion models as needed by this thesis [19].

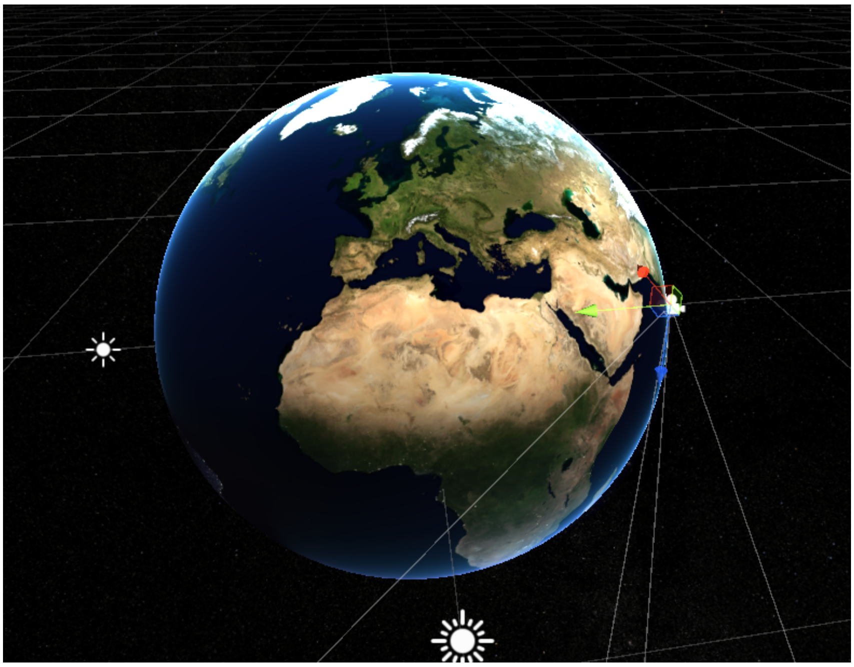

Figure 3.1: View of the simulation environment with a mock ESTCube-2 satellite after spin-up.

Open-source programs which could produce potentially distortable orbital photography resembling the necessary images proved to be generally inadequate for this use-case. Instead of trying to reverse-engineer already existing software, a conclusion was made to produce new simulation software specifically for distorted image capture on a definable orbit. The simulation environment with a mock satellite of ESTCube-2 can be seen in Fig. 3.1. In order to assist future endeavours regarding orbital photography simulation, the proposed program will be made open-source.

3.1 Unity

In order to create the simulation software, the Unity3D Engine was chosen [15]. Not only is the engine well optimized and can produce very high quality data but it is also really easy to work in. The main simulation as well as the data saving components of the simulation had to be hand-made for the purpose, which provided a lot of freedom for distortion and camera based modifications. Because of the optimized engine, all of the 3D rendering, ray-casting and lighting systems themselves did not have to be implemented by hand, it additionally allowed for very fast data generation.

3.1.1 Cartesian Coordinate System

Unity is a general use case 3D Engine, primarily meant for Virtual Reality and Gaming applications. As a result of computer graphics history and how the axis are represented, the coordinate system used in the system is Cartesian, with a Y denoting height [43]. This is important because in astronomical systems the primary coordinate system is Keplerian. A secondary coordinate system sometimes used when converting Keplerian to general physics coordinate systems is a Cartesian system with Z denoting height.

3.1.2 Cartesian to Keplerian Conversion

In order to properly simulate orbital trajectories for image capture, a Keplerian to Cartesian space conversion was needed. This conversion allows for continuous Cartesian location calculation as well as stable orbit propagation in time.

The specifics of this conversion are covered in equations 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, and 4.7.

3.1.3 Simulation Assets

In order to properly create the simulation environment, a couple of Unity assets had to be created. The most notable among these assets was the animated model of the earth, with textures for the Earth albedo and cloud layers created with images from NASA's image catalogue in Visible Earth [17, 18].

Other notable assets included a satellite placeholder that would follow orbital trajectories and provide image capture with configurable camera parameters and finally a true star-map backdrop for image debugging as well as providing data samples for the ESTCube-2 Star Tracker team [10]. Smaller resolution images of the asset textures, that were used to create the simulation, are provided in Fig. 3.2.

3.2 Simulating Distortions

A considerably important part of the simulated data is to keep it as realistic as possible. Because of this, the software includes ways of not only generating ideal images from fixed point locations, it also contains the ability to induce realistic distortions that can occur given the nature of the final application. Only the most significant distortions present with high spin were included in the simulation, namely rolling shutter, exposure, and atmospheric discoloration.

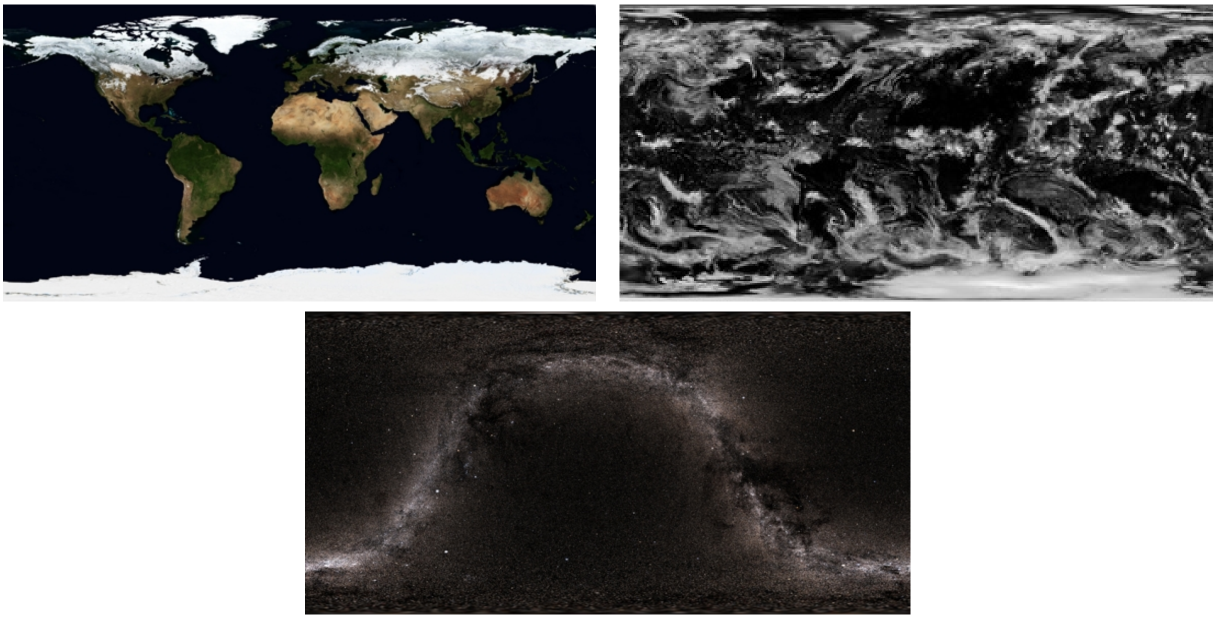

Figure 3.2: Texture maps used to create the planetary surface, cloud layer and background cosmic background for the simulation. The first two images were obtained from NASA: Visible Earth and the skymap from Scientific Visualization Studio [16].

Other common distortion models like salt and pepper-, random background-, fixed pattern- and banding noise to name a few were excluded due to too little of an affect on the stitching algorithm or their irrelevance in the scope of the thesis. Atmospheric refraction effects were excluded due to a combination of the complexity of replicating it as well as how little of an effect it should have on the imaging process.

3.2.1 Camera Based Defects

The distortions could be categorized into two groups based on the nature of the distortion. The first group describes distortions caused by the physical properties of the imaging device and are primarily defined by the specifications of the device itself. This group of defects include radial distortion, exposure time, rolling shutter speed, background-, and salt and pepper noise to name a few. For the current use case the distortion types that could have the largest impact considering the nature of the mission are shutter speed and exposure time. Generated examples of the aforementioned distortions can be seen in Fig. 3.3.

Shutter Speed

A simulated shutter speed is achieved by taking several images over a period of time and combining them as by their separate pixel rows. This method provided an ideal case rolling shutter type distortion. It did not however provide a realistic distortion as there should also be separate delays within each strip that go along the width. This was not included as the effect of such a distortion would only become apparent in very unrealistic conditions. Furthermore the exposure of a frame can and in most cases will have a larger impact than the distortions caused by a rolling shutter. While simulating extreme cases with spins far exceeding $250\ ^{\circ}/s$ and a shutter speed similar to the actual camera, distortions below $25\ ^{\circ}/s$ only became apparent perpendicular to the spin vector. This can be seen in Fig. and 3.4 where the former is taken perpendicular to the spin vector and with a 10 slower shutter rate and the latter near perpendicular. As the camera of

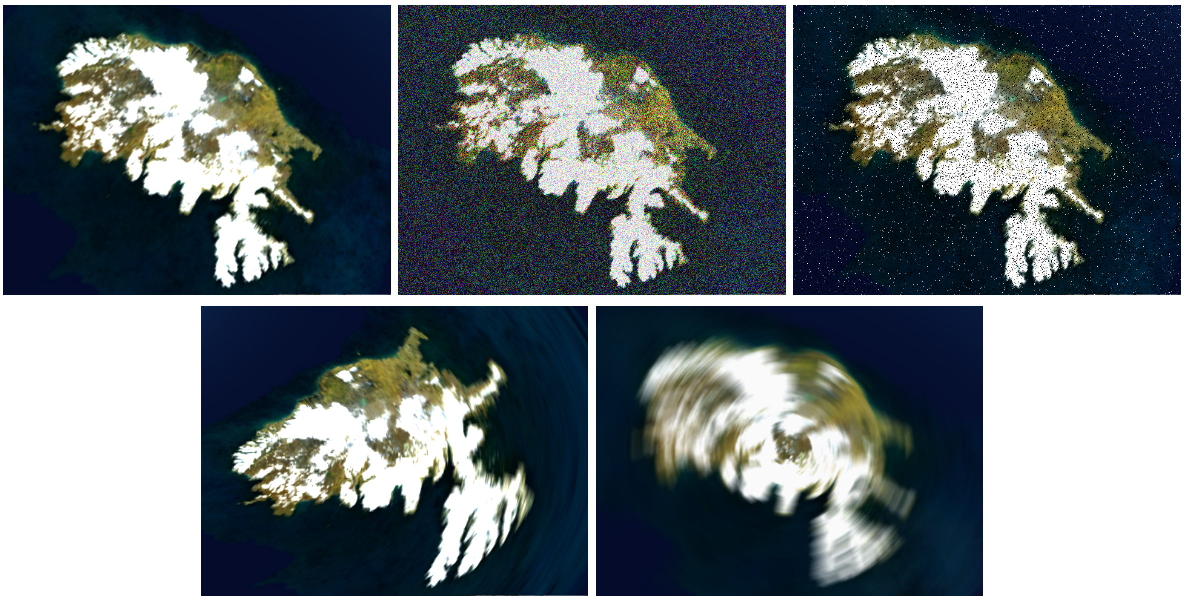

Figure 3.3: Simulated images of Iceland taken from a sun-synchronous orbit. The images on the top row are the original, white- , and salt and pepper noise. The second row shows examples of distortions caused by a slow rolling shutter and long exposure.

ESTCube-2 will be pointing almost parallel to the spin vector, it is apparent that shutter speed will not have any noticeable effect on the imaging system.

Exposure Time

Exposure time was simulated using a similar manner as the rolling shutter example. The difference comes from combining and averaging entire images instead of combining raw sampled pixel rows. The combination of the two distortions was more complicated as each strip needed its own exposure distortion applied to it with the end result being just one of the two distortions. After simulating exposure in Fig. 3.4, it became apparent that lower exposure times provided no visible distortions for the resolution level of the intended camera. Even with higher exposure times, imaging parallel to the spin vector only provided noticeable distortions at exposure times close to $50\ ms$.

3.2.2 Astronomical Defects

The second group of distortions are caused by fundamental physical properties and cannot be improved by the selection of the imaging device nor its aperture. This group contains visual defects caused by the position of the sun, the thickness of the earths atmosphere and combinations of the two. For the simulation software, both a general lighting and reflection based effects, and a color based effect caused by an atmospheric layer were implemented.

Day/Night Cycle

A day/night cycle as well as different material reflections were relatively easy to implement given Unity3Ds' extensive lighting engine. The reflections were thus created using different reflection and normal maps layered on top of the albedo color map, which described the Earth's surface. Materials with very strong reflections like the sea, had larger amounts of specular shading, thus causing very bright areas where the Sun's rays were directly reflected. However,

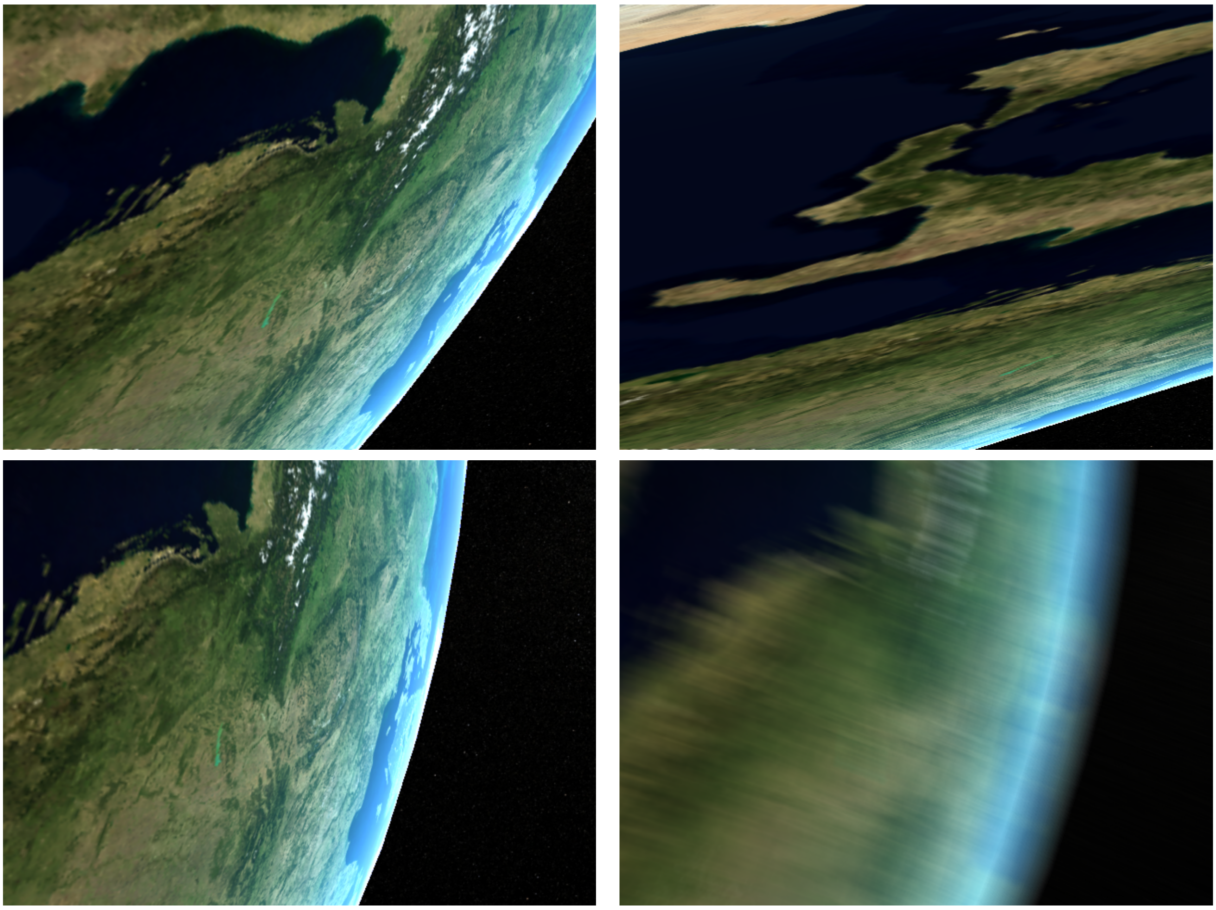

Figure 3.4: Simulated distortions of a sun-synchronous orbit with a $250\ {}^\circ/s$ rotation. Top images have a rolling shutter distortion of $69.4375\ \mu s/row$ and $694.375\ \mu s/row$. The bottom images have an exposure distortion of $1\ ms$ and $20\ ms$ respectively.

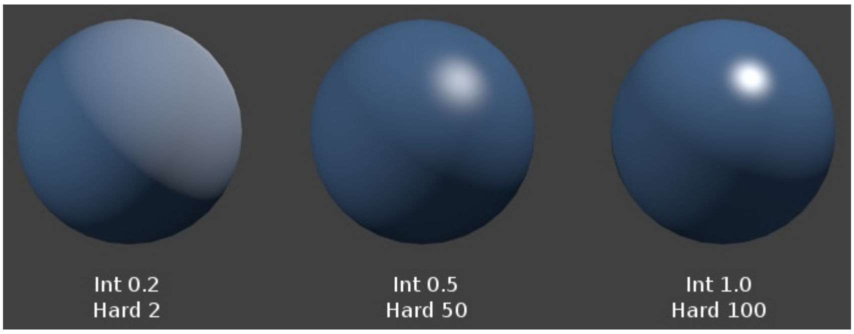

Figure 3.5: Example of Cook-Torrance shading with different light intensity- and material hardness levels [27].

landmass scatters more light so the shading used was a more diffuse variant of Cook-Torrance shading. An example of Cook-Torrance shading can be seen in Fig. 3.5 [27].

Atmosphere

For the first implemented versions of the simulation, the atmosphere was defined to be a very basic gradient extending outward from the surface of the planet. The gradient only effects the color gradient in images taken with a high angle relative to the surface normal but does not recreate any atmospheric refraction distortions [51]. These were not added due to their minuscule relevance compared to the expected camera distortions.

3.3 Optical Navigation

The simulation software was created with the intention of being general purpose and very robust, so it could be later expanded upon and implemented elsewhere. As of writing this thesis, the program is already being employed for other notable developments both for ESTCube-2 and ESTCube-3. The high attention to detail and photo-realism of the simulation allows for near-life image capture in a very controlled and configurable environment. The added camera defect models and the adjustable orbital parameters also make theoretical edge case testing very simple. Two of the most notable projects which employ this software are the ESTCube-2 star-tracker and ESTCube-3 inter-planetary optical navigation system.

3.3.1 Data for Star-Tracker

The created simulation software was most notably used to generate high quality testing images as can be seen in Fig. 3.6 for the star-tracker team. As these images were captured via camera simulation, the camera type could be configured to follow the same FOV and capture parameters as the actual star-tracker. Not only that but the direction of the camera in the star map could be converted and returned as either Euler rotation vectors or as a full quaternion with any definable reference point.

As the backdrop texture in the simulation was $16384 8192$, which is relatively very high quality, the simulated star-tracker camera could perform without any aliasing defects posed by a lower resolution image target.

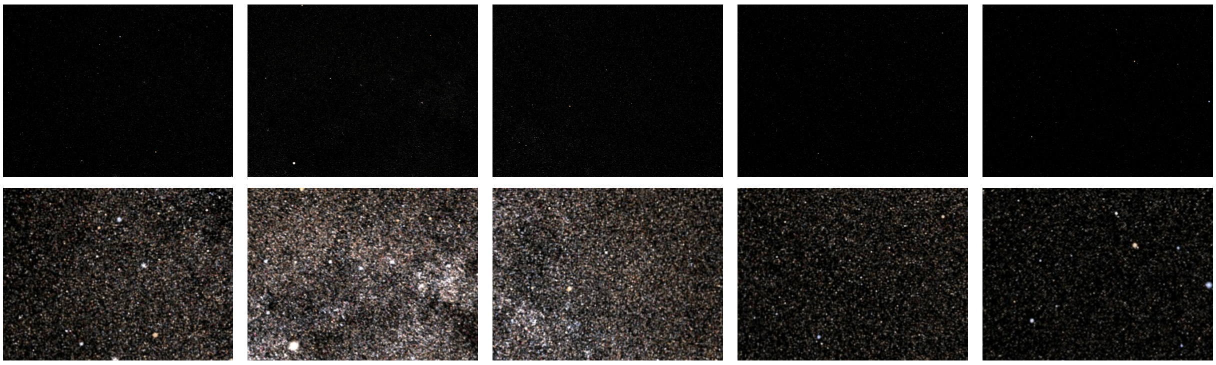

Figure 3.6: Cosmic background image samples for the Star-tracker development. The first row contains the original high resolution images sent to the star-tracker team and the bottom row is of the images highlighted

3.3.2 Planetary Course Correction

Similarly to the star-tracker use case, the simulation will also be further expanded for planetary optical navigation. A proposed method, currently in development as of releasing this thesis, will use expected planetary locations in order to extrapolate and error-correct the position of an interplanetary satellite.

Due to the stability of planetary orbits in a Keplerian system, the same Keplerian to Cartesian conversion will be used to simulate planetary motion in our solar system. This can then be recorded with the high resolution cosmic background in order to provide data for a method of course correction that will be developed for ESTCube-3.

4 Image Stitching Software

The image stitching is the main application part of the work. This chapter defines how it was created and what methods were used to do so. The chapter itself is divided into 3 main sections. The first deals with mathematical coordinate conversions between Keplerian and Cartesian coordinate systems. The next section explains how 3D ray-casting techniques are used to transform the images into the correct space. The final section shows how conventional techniques are applied to the transformed images so they could be properly stitched together.

4.1 Keplerian to Cartesian Conversion

All of the orbital parameters for a satellite are generally defined using a Keplerian coordinate system. This system defines a particular orbit and the time variable defines the exact position an orbiting body is in space. The Keplerian coordinate system is very useful for very accurately approximating the location of an orbiting body after a period of time. For computer systems however, this coordinate system cannot be directly used to display objects in 3D space. Because of this the coordinates must be converted into Cartesian space which gives a rough estimate of a quantifiable location.

4.1.1 Orbital Position

The orbital position algorithm is comprised of the final set of equations that use inclination (i), argument of periapsis (ω), true anomaly (ν), longitude of ascending node (Ω) and semi-major axis (α) along with specific radius (r) to generate the actual (x, y, z) Cartesian vector.

$$OP(t) : \begin{cases} x = r \cdot (cos(\Omega) \cdot cos(\omega + \nu) - sin(\Omega) \cdot sin(\omega + \nu) \cdot cos(i)) \\ z = r \cdot (sin(\Omega) \cdot cos(\omega + \nu) + cos(\Omega) \cdot sin(\omega + \nu) \cdot cos(i)) \\ y = r \cdot (sin(i) \cdot sin(\omega + \nu)) \end{cases} \tag{4.1}$$

4.1.2 Mean Anomaly

Mean anomaly or the fraction from period of an elliptical orbit is calculated by multiplying the gravitational parameter (n) by the elapsed time. The elapsed time is adjusted against epoch time which aligns the orbit to a general reference point.

$$M(t) = n \cdot (t - epoch) \tag{4.2}$$

4.1.3 Eccentric Anomaly

Eccentric anomaly is an angular parameter, it defines the position of a body on an elliptic Kepler orbit and it can be used to convert to and from the true anomaly [12]. In the case of Keplerian to Cartesian conversions, the eccentric anomaly is conversely used to convert a mean anomaly into a true anomaly.

For the following equations eccentricity will be designated as ε. As the calculation for an eccentric anomaly from a mean anomaly is more akin to an optimization problem, the calculation follows the pattern described in 4.1. The variable initialization for the eccentric anomaly is defined as,

$$\delta_E = \begin{cases} M, & \text{if } \varepsilon < 0.8 \\ \pi, & \text{else} \end{cases} \tag{4.3}$$

As the optimization requires the use of a cycle to improve the accuracy of the calculation, an error coefficient F is used to determine the validity of the result.

$$F = \delta_E - \varepsilon * sin(M) - M \tag{4.4}$$

The value dp is the optimization coefficient which determines the number of decimal places, for my case I have set it to 10 in order to keep an acceptable accuracy while not significantly reducing calculation time.

Finally the actual eccentric anomaly can be approximated as,

$$E(t) = \frac{round(\frac{E*180}{\pi} \cdot 10^{dp})}{10^{dp}} \tag{4.5}$$

4.1.4 True Anomaly

True anomaly is the angular parameter that defines the position of an orbiting body in a Keplerian orbit. The true anomaly can be directly calculated from the eccentric anomaly through a simple relation and it can be later used to calculate the Cartesian coordinates for an orbiting body.

$$\delta_\nu = \frac{\arctan_2(\sqrt{(1.0 - \varepsilon^2)} \cdot sin(\frac{\pi}{180} \cdot E), cos(\frac{\pi}{180} \cdot E) - \varepsilon) \cdot 180}{\pi} \tag{4.6}$$

The final true anomaly still contains an approximation step which uses the same optimization coefficient dp that specifies the number of decimal places.

$$\nu(t) = \frac{round(\delta_\nu \cdot 10^{dp})}{10^{dp}} \tag{4.7}$$

4.2 Coordinate Image to Spheroid Mapping

After the location and view angle of a satellite have been mapped, can an image be projected in 3D space. There are methods like LSD-SLAM that can then combine the images to create a 3D

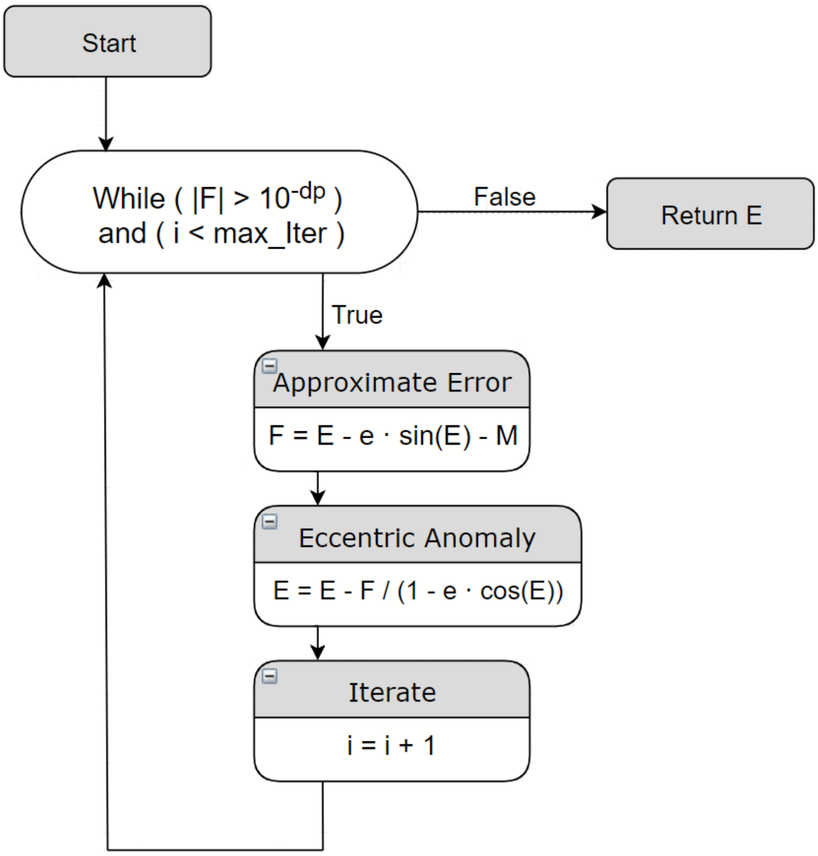

Figure 4.1: Loop for approximating the eccentric anomaly.

point cloud of the resulting space, but the point-cloud generated will have some drawbacks [33]. Namely because of the nature of the data and the intent of capturing high definition images and preserving all of the captured data, the images cannot be converted into a single large point-cloud. Only a couple of images can be used at a time and this would make surface estimations with existing systems either very inaccurate or impossible. [22, 30]

In order to solve this issue, the proposed method uses 3D ray-casting to map individual pixel locations onto an obloid surface as seen in Fig. 4.2. The mapping can be done individually for each image, meaning $O(M)$ memory usage, instead of conventional methods which require $O(N \cdot M)$ coupled with large enough image groups so there would be guaranteed image overlap. As a bonus, the manipulation considers each pixel whether it is of the planetary surface or of the cosmic background, reducing the likelihood of stitching occurring based on the background stars. This also enables individual manipulation of the images in case there are small distortions. Because the resulting images are still complete images with only planetary surface data, they can be stitched together with more traditional methods.

4.2.1 Orbital Rotation

In order to account for the orientational motion of an orbiting body, a simple set of equations can be used to determine the orientation in time. The orientation is defined via pitch, yaw and roll but a single quaternion can also be used. For simplicity, the rotation vector is defined with

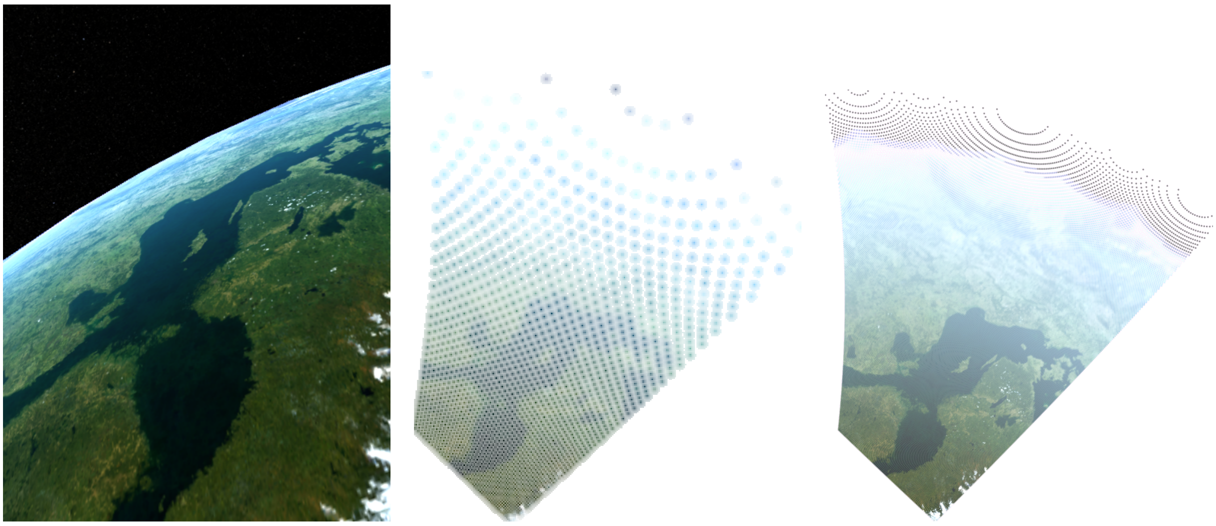

Figure 4.2: Results of image mapping to an obloid surface. The image on the left is the original raw satellite image while the two following it are the $\frac{1}{10}$ resolution and original resolution mapping.

Euler angles as,

$$R(t) : \begin{cases} pitch = \frac{\pi}{180} \cdot mod_{360}(pitch_{speed} \cdot t + \Delta pitch) \\ yaw = \frac{\pi}{180} \cdot mod_{360}(yaw_{speed} \cdot t + \Delta yaw) \\ roll = \frac{\pi}{180} \cdot mod_{360}(roll_{speed} \cdot t + \Delta roll) \end{cases} \tag{4.8}$$

4.2.2 Transformation Matrix

A series of quaternion multiplications is not particularly efficient for calculating the new angular vectors for each pixel, this is why a 3D transformation matrix must be created. A separate matrix Φ_x can be constructed for each axis of rotation,

$$\Phi_{yaw} = \begin{bmatrix} cos(yaw) & -sin(yaw) & 0 \\ sin(yaw) & cos(yaw) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

$$\Phi_{pitch} = \begin{bmatrix} cos(pitch) & 0 & sin(pitch) \\ 0 & 1 & 0 \\ -sin(pitch) & 0 & cos(pitch) \end{bmatrix} \tag{4.10}$$

$$\Phi_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(roll) & -sin(roll) \\ 0 & sin(roll) & cos(roll) \end{bmatrix} \tag{4.11}$$

and then matrix multiplied to form the final rotation matrix Φ.

$$\Phi = \Phi_{pitch} \times \Phi_{roll} \times \Phi_{yaw} \tag{4.12}$$

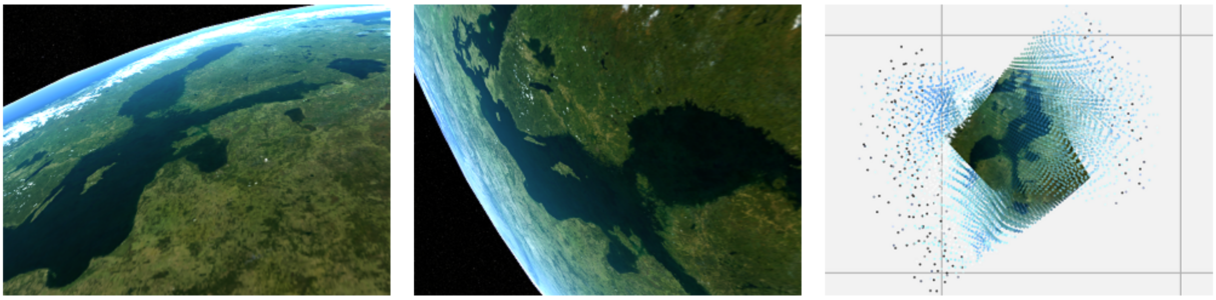

Figure 4.3: Two example images from an image series ray-casted onto an obloid surface as a point cloud. The captured images are of the Baltic region.

4.2.3 Image Ray-casting

A vital part of mapping individual pixel values to an obloid surface is to first find the individual vectors for each of those pixels. A series of computations can be formed on each pixel in image $I_{W,H}$ of dimensions (W, H). In order to reduce confusion in the following operations, W and H will denote the width and height of $I_{W,H}$. A single pixel will be denoted as $i_{x,y}$, with its pixel position being i_x and i_y and relative position in screen space being r_x and r_y.

$$r_x = (i_x - \frac{W-1}{2})/(H-1) \cdot 2 \tag{4.13}$$

$$r_y = -(i_y - \frac{H-1}{2})/(H-1) \cdot 2 \tag{4.14}$$

The equations are not symmetric because they are defined relative to a vertical FOV. The FOV of the camera is taken into account when generating the final pixel direction vector, it is a hardware parameter of the on-board camera and will be denoted as θ_{FOV}. FOV is required to calculate the final parameter h of the relative vector of pixels $\vartheta_{x,y}$ and it is not dependant on pixel or camera location.

$$h = \frac{1}{tan(\frac{\pi}{180} \cdot \frac{\theta_{FOV}}{2}))} \tag{4.15}$$

Thus the relative pixel vector for each pixel in the camera space can be defined as,

$$\vartheta_{x,y} = \frac{\begin{bmatrix} r_x & r_y & h \end{bmatrix}}{\left\| \begin{matrix} r_x & r_y & h \end{matrix} \right\|} \tag{4.16}$$

This then needs to be multiplied by the transformation matrix to apply the rotation of the orbiting body in question. This will produce the final pixel vectors that can be applied to the obloid intersection algorithm as seen in Fig. 4.2 and 4.3.

$$\gamma_{x,y} = \Phi \times \vartheta_{x,y} \tag{4.17}$$

4.2.4 Obloid Intersection

For the following equations we define the semi-major axis as α with the the semi-minor axis being β for an obloid shape. The vector (x, y, z) defines the origin of the raycast and $\gamma_{x,y} =$

(u, v, w) is the directed pixel vector itself.

In order to compute the intersection of the parameterized line with an obloid surface and compute the vector magnitude t, the following equation must be equated,

$$(x/\alpha)^2 + (y/\alpha)^2 + (z/\beta)^2 - 1 \tag{4.18}$$

using the zero set of the function as,

$$(x, y, z) + t \cdot (u, v, w) \tag{4.19}$$

This gives a quadratic equation that can be summarized as,

$$t = -\frac{1}{\gamma_1} \cdot (\gamma_2 + \frac{1}{2} \cdot \sqrt{(4 \cdot (\gamma_2)^2 - 4 \cdot (\gamma_1) \cdot (\beta^2 \cdot (-\alpha^2 + x^2 + y^2) + \alpha^2 \cdot z^2))}) \tag{4.20}$$

Where γ_1 and γ_2 are simplifications of,

$$\gamma_1 = \beta^2 \cdot (u^2 + v^2) + (\alpha^2 \cdot w^2) \tag{4.21}$$

and,

$$\gamma_2 = \beta^2 \cdot (u \cdot x + v \cdot y) + \alpha^2 \cdot w \cdot z \tag{4.22}$$

Computing for t gives 2 possible answers, one is the magnitude at which the vector enters the obloid and the second when it exits. In this case, the exiting distance can be discarded as only the first intersect is required for pixel mapping. Plugging in the newly calculated t value into the original line function,

$$P_o \in (X_o, Y_o, Z_o) = (x, y, z) + t \cdot (u, v, w) \tag{4.23}$$

the actual location of the ray-cast intersection can be calculated. [9]

4.2.5 Earth Rotation

In order to account for the apparent rotation of the earth, a rotational motion vector is used to describe the apparent movement of the surface in time. This is crucial for stitching together images over a long time span as the the motion of the orbiting body and the celestial body itself are independent.

T is the rotational period of the earth in seconds, as this needs to be converted into an angular parameter, it is inverted and converted to radians per second. The time parameter t is used to calculate the final rotational angle of the earth $\phi_\oplus$.

$$\phi_\oplus = \pi \cdot \frac{(-t) \cdot 2}{T} \tag{4.24}$$

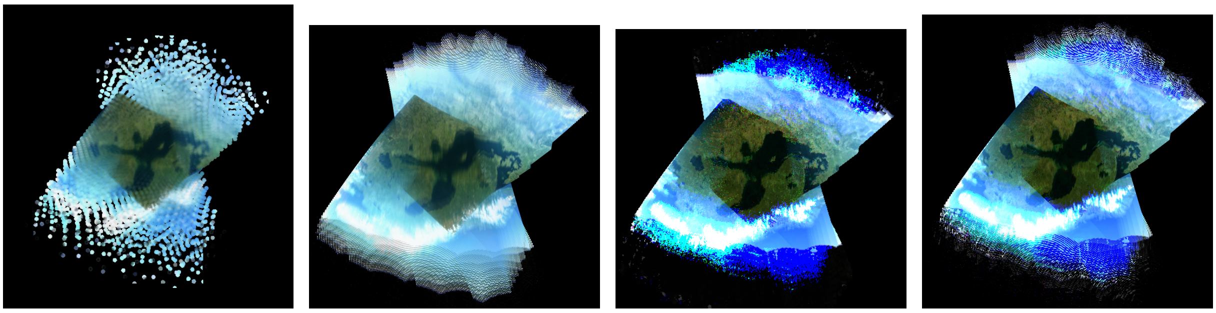

Figure 4.4: Application of 2 different overlapping methods. The first two are the low and high resolution images of mean overlap and the last two are the same for median overlap.

The calculated angle is then applied to the previously calculated obloid intersection points $P_o \in (X_o, Y_o, Z_o)$ in order to convert them to mapped points on the the earths surface $P_\oplus$.

$$P_\oplus \in (X, Y, Z) = \begin{cases} X_o \cdot cos(\phi_\oplus) - Z_o \cdot sin(\phi_\oplus) \\ Y_o \\ Z_o \cdot cos(\phi_\oplus) + X_o \cdot sin(\phi_\oplus) \end{cases} \tag{4.25}$$

4.3 Image Mapping Corrections

The images created are still individual images of the planetary surface. These images can be roughly stitched together using the aforementioned location approximations, in order to create a higher quality image.

When stitching the images, there are 3 main methods that can be used for merging overlapped areas of uneven images: pixel overwriting, -mean and -median. Due to the nature of the data in this work, only the latter two methods are explored. Overlapping using mean values allows for greater control over the color values, but the resulting image becomes more blurred with each misaligned pixel. Using a median retains the image sharpness, but creates single pixel value defects that are not as easy to remove and space requirements expand linearly with each added image. The results of the two methods can be seen in Fig. 4.4 along with a higher resolution versions of the two overlapping methods.

4.3.1 Image Matching with Feature Points

Because of the perspective transform and rough estimates of the planetary surface coordinates generated by the previous steps, traditional stitching can be utilized for specific areas in order correct image misalignment. There are several well known and used methods as well as open-source solutions that perform general image stitching.

Feature point based image matching can be divided into 3 main steps: feature detection, feature matching and image transformation. Feature matching works by dividing an image into very simple regions of interest. These are different types of corners, color gradient directions and sharp transitions. Each image matching technique uses its own feature extraction method and this usually defines their use case. Some of the most common methods for feature detection are

SIFT, SURF, ORB [20,41,50].

The second step in the process is feature matching. This is a well known optimization problem and there are different solutions that give different results depending on the importance of accuracy vs speed. The most common methods here are FLANN and KNN mathing algorithms [37].

The third and final step is the application of a system of equations that generate the transformation matrix. This transformation matrix not only describes the morphological operations needed to translate the feature coordinates from the target to the base image, but they can be used on every single pixel in the target image.

In case of image matching, a base image first has its features detected. These are then compared and matched against the detected features of a target image. These detected and matched features then form a system which can be equated to find the transformation matrix that describes the relation between the two. This transformation matrix can be multiplied with the coordinate of each pixel in the target image in order to transform it into the reference frame of the original image. This transformation will not only consider 2D morphological operations like scale and translation but also 3D operations which enable rotation and shear in three axis.

A downside of this method is the requirement of continuous images with more than $\frac{1}{4}$ overlap with the given resolution, due to these issues the method was explored but the resulting image stitching did not produce any meaningful results. The nature of the mission generates too few images for proper overlap. Even if there are images with enough overlap, the camera distortions cause too many false features, making the feature matching unusable.

5 Experimental Results

5.1 Stitching Simulated Images

The proposed stitching method was evaluated on images generated by the aforementioned simulation program. Throughout the development process, several different iterations of stitching complexities were applied to test the limits of the program before any distortions were applied to the images themselves. The proposed algorithm for stitching performed well in all simulated cases. Proper continuous ground coverage was achieved early on with simpler high altitude orbits as seen in the top images of Fig. 5.1. This was then expanded to low earth orbits and even high spin low earth orbits. For both cases the stitching remained visually accurate. As long as the approximate orbital location and accurate rotational vector could be restored, the images could be stitched without the need of external image alignment methods.

5.2 Photo-realism Comparison with ESTCube-1 Images

As a way of verifying images taken by the simulation, they had to be compared against real life data samples. ESTCube-1 was launched in May 2013 and took photos all the way up to January 2015. During its mission, a lot of high quality images were taken from space that can be replicated in simulation. Unfortunately as the orbital parameters of ESTCube-1 varied throughout the mission, full propagation from the first orbit will not be very accurate over a long time period. Secondly as the orbital period of the satellite was about $98\ min$, and time-stamps were properly provided for only a few images, there were not enough images in order to perform proper stitching. [53]

The actual images from ESTCube-1 can thus only be used for image resolution verification and to confirm that the provided image angles actually exist in the simulated orbit. For direct comparisons, three images were recreated in the simulation based on approximate orbital locations during that time frame. The recreated images were also given a randomized cloud layer and color corrections using the Color Transfer between Images algorithm [14, 49] and can be seen in Fig. 5.2-5.3. As can be seen in the images, the results are very similar especially when considering the orbital positions were not a perfect match and any distortions caused by the lens itself were not corrected for.

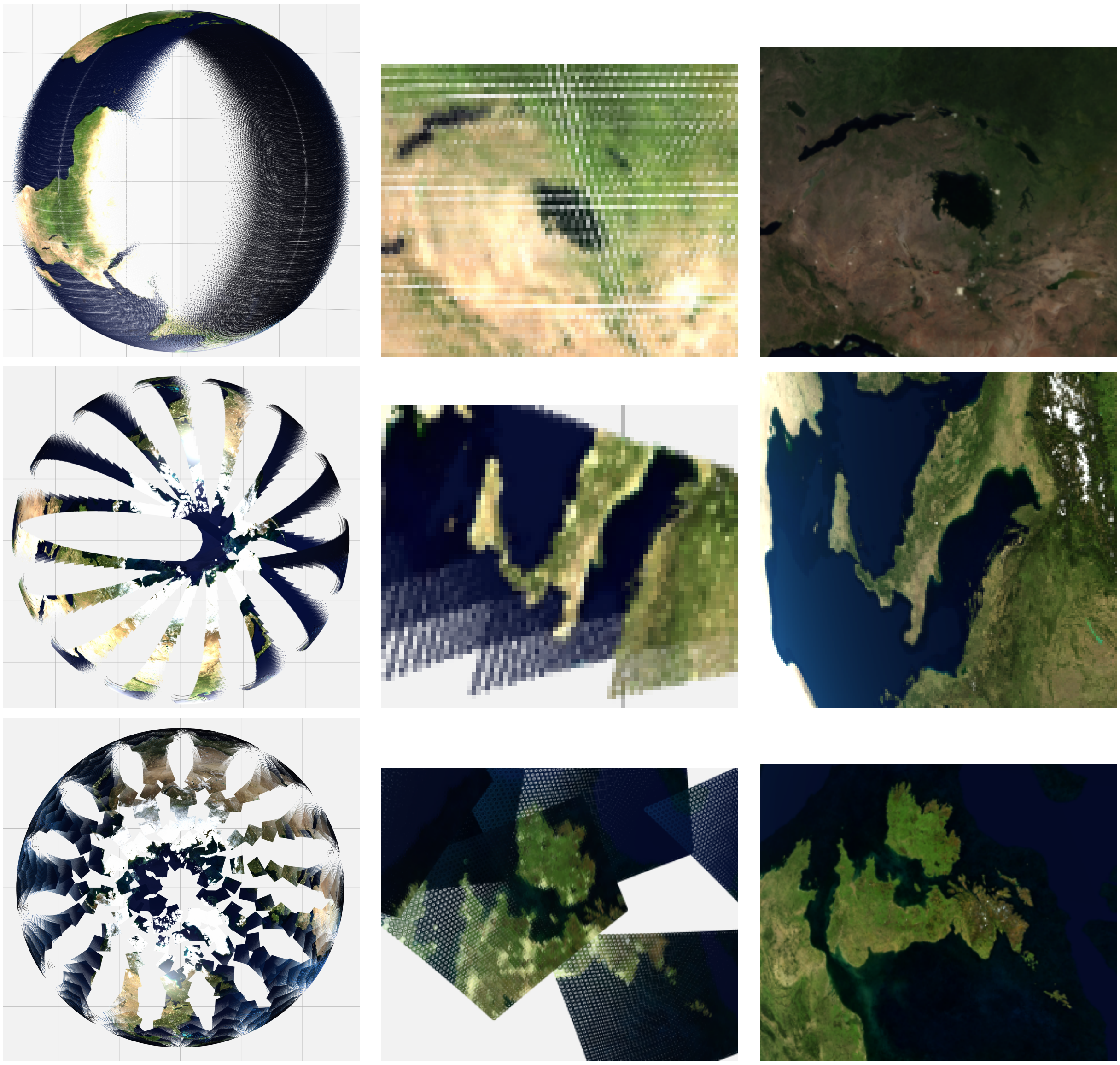

Figure 5.1: Stitching results for 3 different difficulty levels. From top to bottom, fully circular orbit with no inclination, ESTCube-1 orbit with no spin, ESTCube-1 orbit with $250\ {}^{\circ}/s$ spin. The center and right column images are zoomed in and true images of lake Victoria, Italy and the British Isles.

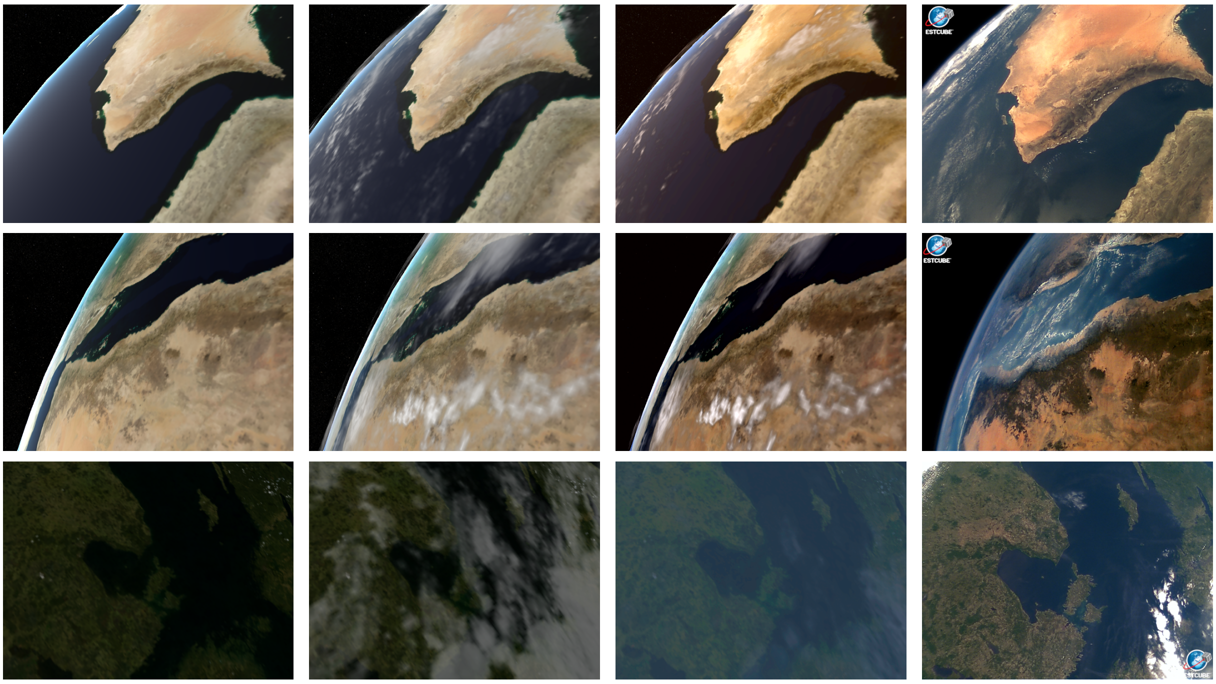

Figure 5.2: Simulated images compared against real ESTCube-1 images. From left to right, simulated image without cloud cover, with clouds, color corrected and a real color corrected photo of the same location. The real photos from top to bottom were taken 2013.10.10 of the Gulf of Oman, 2013.12.08 of the Red Sea, and 2014.04.28 of the Baltic Sea

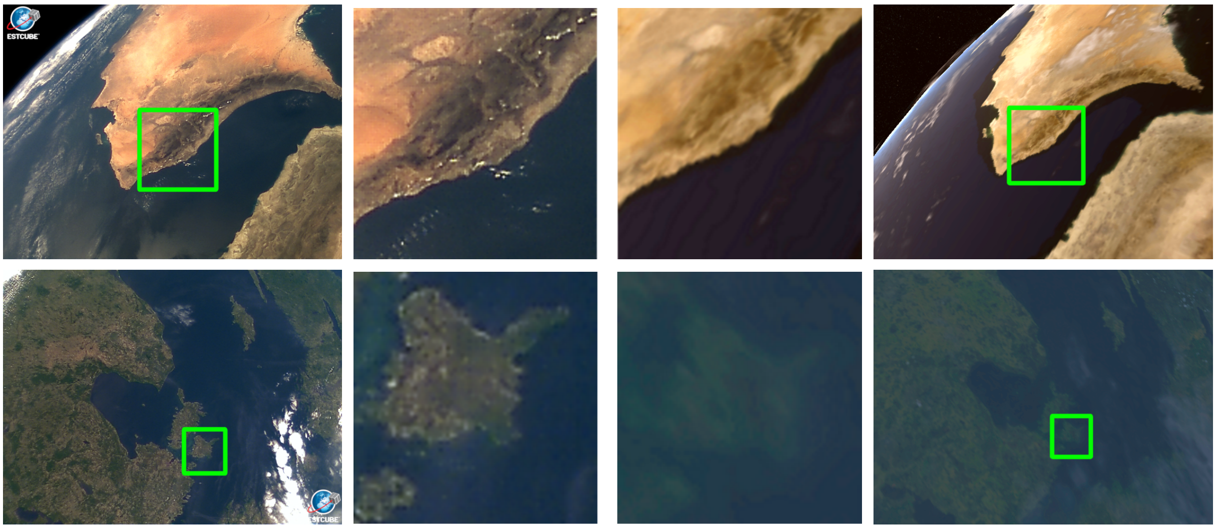

Figure 5.3: Zoomed in sections of actual and simulated images for quality comparison with smaller details. Top image is of Muscat, Oman and the bottom image is of Hiiumaa, Estonia.

6 Conclusion and Future Work

This thesis outlines a geometric image stitching algorithm that relies on attitude logging instead of image overlap in order to produce properly surface-mapped and stitched images. As the proposed stitching algorithm requires adequately logged orbital parameters and precise time-stamps, previous images generated by ESTCube-1 could not be used for mapping verification. Instead a separate photo-realistic method for simulating orbital image capture with properly definable camera and orbital parameters was developed for image data generation.

The first half of the thesis describes the simulation environment, its creation and the types of distortion models considered for image stitching. Due to the use of photo-realistic albedo maps from NASA images together with easily definable camera and orbital parameters to create the simulation, it is very flexible as an application.

The second half goes over the exact algorithms created and used to apply geometric image mapping and later stitching for overlapping images. Similar and competing image stitching methods are analyzed and outlined for the case of ESTCube-2. The proposed image stitching method was verified to work on simulated data and performs within acceptable margins even when the expected high spin distortion models are added.

6.1 Future Work

The simulation program has already found use cases outside the scope of this project, namely for data generation in verifying new optical navigation systems for ESTCube-2 and -3. Current uses being star-tracker algorithm verification and a new planet based optical navigation system.

The image stitching method can be further improved to include more varied and noisy attitude data and to reconstruct images afflicted by atmospheric refraction. It can also be used to validate attitude determination by judging the accuracy of the stitching.

Both the simulation software and the stitching algorithm can still have other optic-related use cases for ESTCube-2, ESTCube-3 and other satellite missions with similar imaging or orbital constraints.

7 Acknowledgments

I would like to thank both of my outstanding supervisors Indrek Sünter for guiding me through the specifics of ESTCube and providing indispensable advice, and Prof. Gholamreza Anbarjafari (Shahab) for guiding and encouraging me throughout the thesis process as well as inadvertently developing my critical academic knowledge.

I would also like to thank all of the members of both iCV and ESTCube who helped me in solving some of the more perplexing problems I encountered.

A special thanks to Kadi Paldre who assisted with ample amounts of editorial advice.

Bibliography

[1] A1/4-inch vga digital image sensor - mt9v011 data sheet. https://www.promelec.ru/pdf/MT9V011.pdf. Accessed: 2019-05-18.

[2] Amateur radio peosat. https://www.pe0sat.vgnet.nl/satellite/cube-nano-picosats/ESTCube-1/. Accessed: 2019-05-12.

[3] Camera calibration and 3d reconstruction. https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html?highlight=calib. Accessed: 2019-05-12.

[4] Celestia. https://celestia.space/. Accessed: 2018-11-03.

[5] Estcube-1 cam optics-lens. https://confluence.tudengisatelliit.ut.ee:8433/display/CAM1/Lens. Accessed: 2019-05-12.

[6] Heavens above, estcube 1 - orbit. https://heavens-above.com/orbit.aspx?satid=39161&lat=0&lng=0&loc=Unspecified&alt=0&tz=UCT. Accessed: 2019-13-05.

[7] Multispectral instrument (msi) overview. https://earth.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument. Accessed: 2019-05-12.

[8] Nasa visualization explorer - earth from orbit. https://svs.gsfc.nasa.gov/11268. Accessed: 2018-10-13.

[9] Point of intersection for a ray and earth's surface. https://gis.stackexchange.com/questions/20780/point-of-intersection-for-a-ray-and-earths-surface. Accessed: 2019-04-16.

[10] Scientific cisualization studio - the tycho catalog skymap - version 2.0. https://svs.gsfc.nasa.gov/cgi-bin/details.cgi?aid=3572. Accessed: 2019-13-05.

[11] Sentinel 2 l1c data quality report. https://sentinel.esa.int/documents/247904/685211/Sentinel-2_L1C_Data_Quality_Report?fbclid=IwAR3KwjtIB6CMQeGqzWuVFtcLOm9YEAGNo7q06JlffjgSPCpaMSXbemoJ2lQ. Accessed: 2018-09-11.

[12] Solving kepler's equation of elliptical motion. http://www.jgiesen.de/kepler/kepler.html. Accessed: 2019-04-16.

[13] Space engine. http://spaceengine.org/. Accessed: 2018-10-13.

[14] Super fast color transfer between images. https://www.pyimagesearch.com/2014/06/30/super-fast-color-transfer-images/. Accessed: 2019-05-02.

[15] Unity, unity3d. https://unity.com/. Accessed: 2019-13-05.

[16] Visible earth - a catalog of nasa images and animations of our home planet. https://visibleearth.nasa.gov/. Accessed: 2019-13-05.

[17] Visible earth - blue marble: Clouds. https://visibleearth.nasa.gov/view.php?id=57747. Accessed: 2019-13-05.

[18] Visible earth - december, blue marble next generation w/ topography. https://visibleearth.nasa.gov/view.php?id=74518. Accessed: 2019-13-05.

[19] Visual simulator for space and ground segment. https://www.taitussoftware.com/products/applications/vssgs-visual-simulator-space-ground-segment/. Accessed: 2018-10-02.

[20] Ebtsam Adel, Mohammed Elmogy, and Hazem Elbakry. Image stitching system based on orb feature-based technique and compensation blending. *International Journal of Advanced Computer Science and Applications*, 6(9), 2015.

[21] Richard L Baer. Ccd requirements for digital photography. In *IS AND TS PICS CONFERENCE*, pages 26–30. SOCIETY FOR IMAGING SCIENCE & TECHNOLOGY, 2000.

[22] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[23] Cesare Barbieri. *Fundamentals of astronomy*. CRC Press, 2017.

[24] Ronald J Boain. Ab-cs of sun-synchronous orbit mission design. 2004.

[25] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[26] Vladimir A Chobotov. *Orbital mechanics*. American Institute of Aeronautics and Astronautics, 2002.

[27] Robert L Cook and Kenneth E Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (TOG)*, 1(1):7–24, 1982.

[28] Olivier L De Weck, Uriel Scialom, and Afreen Siddiqi. Optimal reconfiguration of satellite constellations with the auction algorithm. *Acta Astronautica*, 62(2-3):112–130, 2008.

[29] Julian A Dowdeswell and Toby J Benham. A surge of perseibreen, svalbard, examined using aerial photography and aster high resolution satellite imagery. *Polar research*, 22(2):373–383, 2003.

[30] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[31] Hendrik Ehrpais, Johan Kütt, Indrek Sünter, Erik Kulu, Andris Slavinskis, and Mart Noorma. Nanosatellite spin-up using magnetic actuators: Estcube-1 flight results. *Acta Astronautica*, 128:210–216, 2016.

[32] Hendrik Ehrpais, Indrek Sünter, Erik Ilbis, Janis Dalbins, Iaroslav Iakubivskyi, Erik Kulu, Indrek Ploom, Pekka Janhunen, Joel Kuusk, Jānis Šate, et al. Estcube-2 mission and satellite design. In *The 4S Symposium*, 2016.

[33] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[34] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *ACM transactions on graphics (TOG)*, volume 25, pages 787–794. ACM, 2006.

[35] John G Fryer and Duane C Brown. Lens distortion for close-range photogrammetry. *Photogrammetric engineering and remote sensing*, 52(1):51–58, 1986.

[36] Bo-Cai Gao. Ndwia normalized difference water index for remote sensing of vegetation liquid water from space. *Remote sensing of environment*, 58(3):257–266, 1996.

[37] Kiana Hajebi, Yasin Abbasi-Yadkori, Hossein Shahbazi, and Hong Zhang. Fast approximate nearest-neighbor search with k-nearest neighbor graph. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[38] Robert S Hill, Wayne B Landsman, Don Lindler, and Richard Shaw. Cosmic ray and hot pixel removal from stis ccd images. In *Proc. HST Calibration Workshop*, pages 120–125, 1997.

[39] Iaroslav Iakubivskyi, Hendrik Ehrpais, E Oro, E Kulu, et al. Estcube-2 mission analysis: plasma brake experiment for deorbiting. In *Proceedings of the 67th International Astronautical Congress, Mexico*, 2016.

[40] G Janin and VR Bond. The elliptic anomaly. 1980.

[41] Luo Juan and Gwun Oubong. Surf applied in panorama image stitching. In *2010 2nd international conference on image processing theory, tools and applications*, pages 495–499. IEEE, 2010.

[42] K Kim, KC Jezek, and H Liu. Orthorectified image mosaic of antarctica from 1963 argon satellite photography: image processing and glaciological applications. *International Journal of Remote Sensing*, 28(23):5357–5373, 2007.

[43] N Krishnamurthy. *Introduction to computer graphics*. McGraw-Hill Publishing Company, 2002.

[44] Henri Kuuste. Estcube-1 tether end mass imaging system design and assembly, 2012.

[45] Silver Lätt, Andris Slavinskis, Erik Ilbis, Urmas Kvell, Kaupo Voormansik, Erik Kulu, Mihkel Pajusalu, Henri Kuuste, Indrek Sünter, Tõnis Eenmäe, et al. Estcube-1 nanosatellite for electric solar wind sail in-orbit technology demonstration. *Proceedings of the Estonian Academy of Sciences*, 63(2):200, 2014.

[46] Chia-Kai Liang, Li-Wen Chang, and Homer H Chen. Analysis and compensation of rolling shutter effect. *IEEE Transactions on Image Processing*, 17(8):1323–1330, 2008.

[47] Aimé Meygret, Simon Baillarin, Ferran Gascon, Emmanuel Hillairet, Cécile Dechoz, Sophie Lacherade, Philippe Martimort, François Spoto, Patrice Henry, and Riccardo Duca. Sentinel-2 image quality and level 1 processing. In *Earth Observing Systems XIV*, volume 7452, page 74520D. International Society for Optics and Photonics, 2009.

[48] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM transactions on graphics (TOG)*, 23(3):664–672, 2004.

[49] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[50] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011.

[51] J Saastamoinen. Contributions to the theory of atmospheric refraction. *Bulletin Géodésique (1946-1975)*, 107(1):13–34, 1973.

[52] Andris Slavinskis, Urmas Kvell, Erik Kulu, Indrek Sünter, Henri Kuuste, Silver Lätt, Kaupo Voormansik, and Mart Noorma. High spin rate magnetic controller for nanosatellites. *Acta Astronautica*, 95:218–226, 2014.

[53] Andris Slavinskis, Mihkel Pajusalu, Henri Kuuste, Erik Ilbis, Tõnis Eenmäe, Indrek Sünter, Kaspars Laizans, Hendrik Ehrpais, Paul Liias, Erik Kulu, et al. Estcube-1 in-orbit experience and lessons learned. *IEEE Aerospace and Electronic Systems Magazine*, 30(8):12–22, 2015.

[54] Indrek Sünter, Henri Kuuste, Andris Slavinskis, Ants Agu, Erik Ilbis, Georgi Olentšenko, Iaroslav Iakubivskyi, Jacob Luis López Seco, Johan Kütt, Riho Vendt, et al. Design and testing of a dual-camera payload for eseo. In *67th International Astronautical Congress, IAC 2016, 26 September 2016 through 30 September 2016*. International Astronautical Federation, IAF, 2016.

[55] Richard Szeliski. Image alignment and stitching. In *Handbook of mathematical models in computer vision*, pages 273–292. Springer, 2006.

[56] ESTCube Team. Estcube-1 space system and mission description. 2009.

[57] Erich Vogt. Elementary derivation of keplers laws. *American Journal of Physics*, 64(4):392–396, 1996.

Non-exclusive licence to reproduce thesis and make thesis public

Mina, Rain Eric Haamer,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

   Imaging Simulator and Geometric Image Stitching for a Low Earth Orbit Satellite with High Spin Rates,

   mille juhendajad on Gholamreza Anbarjafari ja Indrek Sünter,

   reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. olen teadlik, et punktis 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Rain Eric Haamer,
Tartu, May 20, 2019