

Information extraction and geolocalization of historical aerial imagery

Yushan Liu



Information extraction and geolocalization of historical aerial imagery

Additional Master Thesis report

by

Yushan Liu

Assessment committee:

Supervisors:	F. Dahle	TU Delft
	Dr. R.C. Lindenbergh	TU Delft
Project Duration:	11-2022 - 02-2023	
Student number:	5525829	

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Antarctica plays a crucial role in global climate change research, but current models only extend back to the satellite era from the 1990s to the present, leaving significant gaps in our understanding of the continent's past and covering only limited locations. Historical aerial surveys, which may extend back to the 1930s and cover more locations, offer an opportunity to overcome these limitations. However, creating models from historical imagery poses a significant challenge. This study presents a workflow for automatically extracting camera altitude information from historical imagery and doing preliminary geolocalization by computing the footprints of the imagery based on the extracted altitude information. The workflow was applied to two flight lines in the TMA Single Frames dataset, and the results were evaluated using a scoring system and by comparing the photos with their footprints. This work serves as a preliminary stage in geolocating and georeferencing historical imagery, laying the foundation for future model-building with these photographs.



Copyright © Yushan Liu, 2023
All rights reserved.

Preface

This thesis is an additional thesis under the requirements of course AES4011-10 Additional Thesis (2022/23). I am interested in the topic of this thesis and excited about what this work can bring to the project behind it. Felix Dahle came up with this topic, and provided great help throughout the whole process of my work. Roderik Lindenbergh also helped me a lot by giving constructive suggestions on the general direction of this work. I would like to express my gratitude to them. This work led me to see possibilities of historical aerial data, which is really worth exploring. In all, I really enjoyed this work.

Contents

List of Figures	v
List of Tables	vi
1 Introduction	1
2 Dataset	3
3 Method	6
3.1 The "Three-pointer" altimeter	6
3.2 Extraction of altitude information	7
3.3 Further analysis of altimeter readings: camera altitude change and footprint calculation	13
4 Results and Analysis	16
4.1 Evaluations of the altimeter reader.	16
4.2 Camera altitude change graph	18
4.3 Footprints calculation	20
5 Conclusion and Recommendations	24
5.1 Conclusion	24
5.2 Recommendations	24
References	26
A Python Source Code	27

List of Figures

2.1	Example photographs of Antarctica single frame collection	3
2.2	Flight lines in of Antarctica single frame collection	4
2.3	Cropped altimeters from the scanned images	5
3.1	"Three pointer" altimeter	7
3.2	Location of altimeters in images	8
3.3	An example of an unreadable image	10
3.4	Template matching	11
3.5	Images before and after histogram equalization	11
3.6	Sharpening, denoising and contrast enhancement, and binarization . . .	12
3.7	An example of line detection	12
3.8	Camera footprint	14
3.9	Roll, pitch, and yaw	14
4.1	Examples of incorrect reading results due to image enhancement and pointer identification.	18
4.2	Flight line 0347 and altitude change graph	19
4.3	Flight line 1815 and altitude change graph	20
4.4	Footprints of flight line 0347	21
4.5	Photo 260 in flight line 0347	22
4.6	Footprint of flight line 1815	23

List of Tables

4.1 Accuracy of the altimeter reader 16

Introduction

Observations and modeling of Antarctica is vital to a better understanding of glacier dynamics and global climate change. Over the past few decades, there has been an increasing number of glacier monitoring programs in Antarctica. However, the programs and the models only extend a few decades from the early 1990s to present, which covers the extent of the modern satellite era [1]. The locations covered by these glacier monitoring programs are also very limited. Many locations lack any records or only have short period of data [2]. Under this situation, earlier aerial surveys that dates back to the 1930s can be important resources to extend the temporal scale of Antarctic glacier observations. Due to the fact that there is often a lack of orientation metadata for historical aerial imagery, structure-from-motion (SfM) photogrammetry techniques were used to extract three-dimensional (3D) information such as point clouds and Digital Elevation Model (DEMs) derived from the point clouds with little or even no priori camera information [2] [3]. However, SfM is not directly applicable in our study. The historical aerial imagery used in this study was collected by the U.S. Navy between 1946 to 2000 in many parts of Antarctica. The images cover a large area, including the eastern half and most of the coastal areas in the western half. Such a data set brings opportunities to study Antarctica at a larger spatial and temporal scale, but also brings difficulties for modeling, since it is difficult to obtain uniformly distributed Ground Control Points (GCPs) in such a large area which is essential for SfM. Therefore, another method needs to be adopted to extract three-dimensional information from these images. Similar to SfM, a matching for the features (e.g. pixels, pixel groups, edges) will be done to generate 3D point clouds, but due to the lack of ground control, camera information is needed for locating the ground footprints of the images. The purpose of this thesis is to automatically extract the camera information (camera height) from the images in the data set, and further carry out preliminary geolocalization based on the extracted camera information.

Once the altitudes of the cameras for each image are computed, some analysis of the heights and further calculations of the footprints can be done base on the results. For example, line graphs can be made to observe how the height of the camera changes on a single flight line. These height change graphs are intuitive visualizations of the behavior of the airplanes, and are also helpful for a general check whether the height results are realistic. Furthermore, based on the camera height results and some approximate values of camera orientations and photo centers, footprints of the images can be computed. Although the computed footprints in this thesis are just approximate values, they are still important since it provides an initial localization of the exact footprints, which are

determined further by features matching.

Considering the background stated above, the purpose of this research is to design and create a workflow that can automatically extract altitude information from historical aerial photographs, and compute footprints of the photos based on the extracted altitude information. The research question of this study is as follows: How can a workflow be designed and created to automatically extract altitude information from historical aerial photos? And how can footprints of the photos be computed based on altitude information extracted from the photos and approximated orientations cameras used to take the photographs?

The thesis is structured as follows. Chapter 1 provides an overview of the background and purpose of the study. Chapter 2 includes detailed information and visualizations of the historical aerial imagery data set used in the study. Chapter 3 introduces the methods used for reading the altimeters and computing the approximate footprints. Chapter 4 presents the results of the altimeter reading, height change monitoring, and visualizations of the computed approximate footprints, along with evaluations of the altimeter reading approach. Chapter 5 summarizes the key findings of the study and draws conclusions, made recommendations for future research and practical applications based on the conclusions.

2

Dataset

Antarctic Single Frames (1946-2000) is a collection of trimetrogon aerial (TMA) photographs over the Antarctic from the United States Antarctic Resource Center (USARC) and the British Antarctic Survey (BAS) between 1946 and 2000. The camera system is TMA photography, which collects left-oblique, on-nadir(straight down), and right-oblique static frame. The TMA images are taken along a single flight line with the oblique cameras pointed at a depression angle of 30° . Each of the cameras has an angular field of view of 60° , which provides a 180° horizon to horizon coverage when the images are placed side-by-side [4]. There are black-and-white, natural color, and color infrared images included in this collection, however, the majority of the photographs are black-and-white, which is why in this thesis we only look into the black-and-white photographs. In Fig. 2.1 are four example photographs from the collection. These four photographs are on the same flight line and were taken adjacent in time, which is why the coastline in the four photographs are gradually moving with respect to the last one.

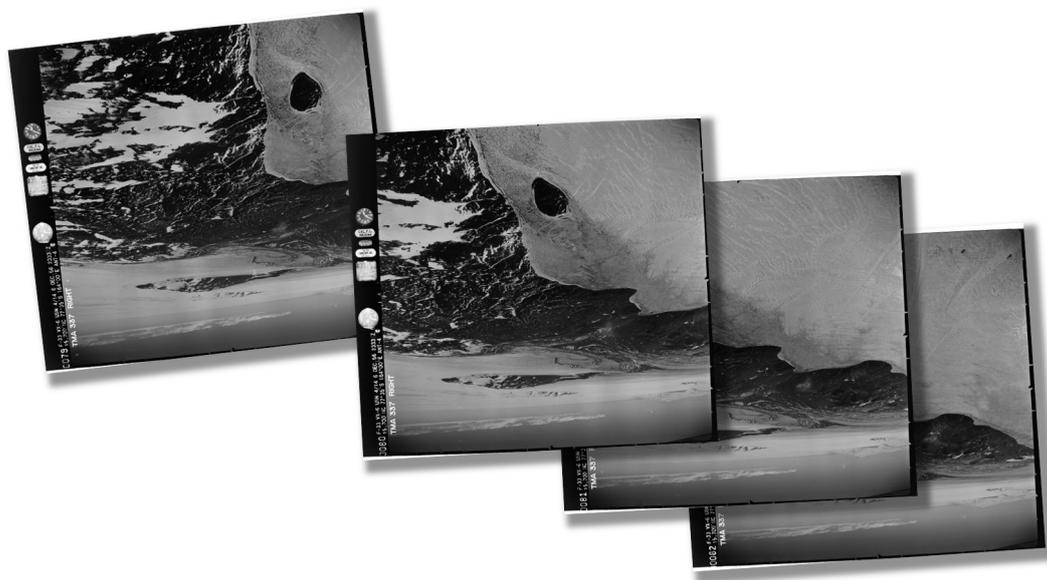


Figure 2.1: Example photographs of Antarctica single frame collection

As shown in Fig. 2.2, there are over 1000 flight lines, and over 330,000 single-frame

aerial photographs, covering most of the glaciers in Antarctica, including most areas in the eastern part of Antarctica, and some of the coastlines in the western part.

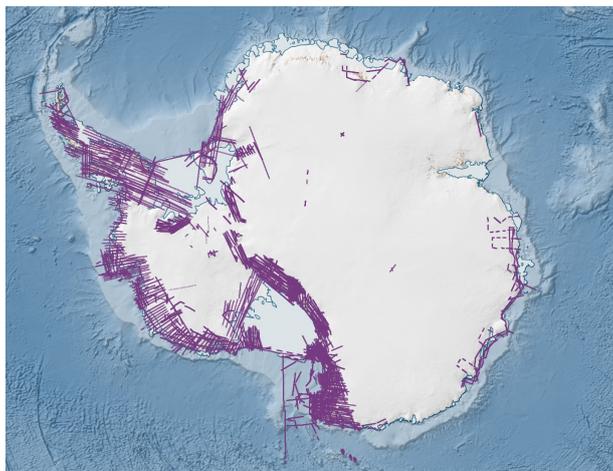


Figure 2.2: Flight lines in of Antarctica single frame collection

The USGS EROS Data Center, in cooperation with the British Antarctic Single Frames (BAS), scanned this collection of Antarctic aerial photography and made it available to the public. The scanned photographs are offered in high, medium, and low resolution versions. To achieve the best possible results, we used the high resolution version photographs for altitude reading. The length and width of one high resolution version of photo are roughly $10,000 \times 10,000$ pixels, while the space taken by one photo is about 100 MB. As for the altimeter in the image, a bounding box of an altimeter is generally about 700×700 pixels in length and width.

However, the scanning quality is usually not good. The lighting situations during the scanning process are different, resulting in different brightness of the scanned image, as shown in the figures in Fig. 2.3. Besides, sometimes the photograph was placed at an angle, therefore there might be only half or even two altimeters in a scanned image, as shown in the two figures at the bottom part in Fig. 2.3. This brings extra challenge for this study.

Other data from the Antarctic Single Frames Collection is also used in this study. According to information from the USGS official website, the oblique cameras were pointed at a depression angle of 30° and had an angular field of view of 60° . The Esri shapefile "TMA Photocenters" contains the photo centers for all flight lines and includes several attributes for each photo center, including Direction (text description of the flight direction), Altitude (camera height in meters), Azimuth_dd (camera azimuth, or rotation relative to north), TMA_num (flight number), and photo_ID (unique identifier for each photo). The attributes of Direction, Altitude, and Azimuth_dd are used to calculate the photo footprints, while TMA_num and photo_ID are used to identify the photographs.

This study also utilizes data from Quantarctica, a collection of Antarctic geographical datasets for research, education, operations, and management purposes [5]. It comprises of peer-reviewed data from ten scientific themes and a professionally designed basemap, which have been contributed by the community. To create visually appealing results, the

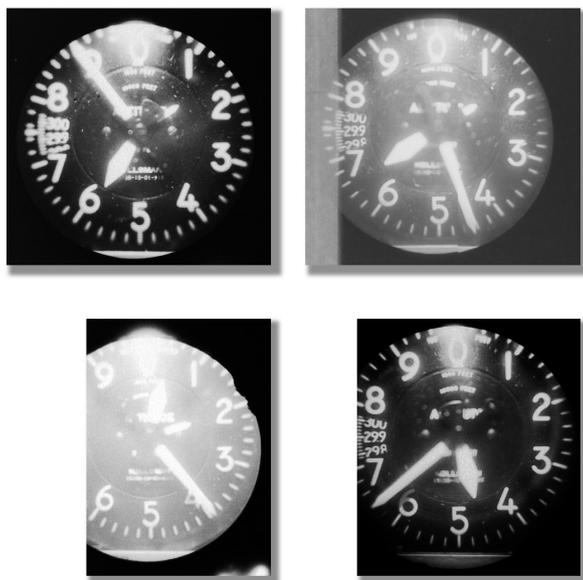


Figure 2.3: Cropped altimeters from the scanned images

"Detailed Basemap" created by the Norwegian Polar Institute in 2018 is utilized. Elevation information is obtained using the "BEDMAP2 (1km)" terrain model, which was created by the British Antarctic Survey (BAS) in 2013.

3

Method

In this chapter, the methods used in this study are thoroughly explained. First, the "three-pointer" altimeter is introduced in detail in Section 3.1, including its appearance, working principle, and other important information. Next, Section 3.2 outlines the steps involved in extracting altitude information from the altimeter. Finally, Section 3.3 provides a comprehensive explanation of the two additional analyses performed on the obtained altitude reading results. These include a camera altitude change graph and a calculation of the footprint.

3.1. The "Three-pointer" altimeter

The altimeter mounted on the aircraft is the barometric altimeter or pressure altimeter, and the display type is the conventional analogue display. The pointers of a barometric altimeter move in response to changes in barometric pressure, and the dial is calibrated in feet [6]. As shown in Fig. 3.1, in this altimeter, there are three pointers. The longest pointer reads hundreds of feet (the calibrations are 20 feet [7]), the next longest pointer reads thousands of feet, and the smallest pointer reads tens of thousands. Three-pointer altimeter can be easily misread, which is mainly because the three pointers often overlay each other. The smallest pointer is the most difficult to read, since it is at the bottom layer and often covered by the other two pointers. The longest pointer which is at the top layer also becomes problems because of its tail which is often covering other pointers. The fact that three-pointer altimeter can be so easily misread is the reason why many new airplanes are now being equipped with drum-pointer altimeter which has only one pointer and a counter. During this study, there are often situations where the three pointers become unreadable, and decreases the accuracy of the results.



Figure 3.1: "Three pointer" altimeter ¹

The calibration of an barometric altimeter follows the equation [8]:

$$z = cT \log (P_0/P) \quad (3.1)$$

Where c is a constant, T is the absolute temperature, and P is the barometric pressure at altitude z . z is the altitude above the level where the pressure is P_0 . In most cases, P_0 is set to be 29.92 Hg, which by international agreement is the barometric pressure at sea level on a standard day (with the temperature $15^\circ C$). Kollsman window allows the pilots to check the standard barometric pressure (P_0). In Antarctic Single Frames Collection, the P_0 of the altimeters was always set to be 29.92 Hg. This means the altitude we can read on the altimeter is the height above the sea level, which is also referred to as true altitude, while the altitude we need for camera position (EO, exterior orientation) is absolute altitude, which is the height above the ground surface. Therefore an extra data source [5] was used to get the approximate elevation in Antarctica, and then subtracted from the altitude readings to obtain the height from the camera to the ground surface.

3.2. Extraction of altitude information

A "three pointer" altimeter can be approached from a computer vision perspective similarly to a standard clock. Both have circular dials and three hands or pointers, with the 360° on the dial divided into equal parts. The three pointers provide readings of different orders of magnitude. The methods used to read an analog clock can be applied to read the "three pointer" altimeter. There have been numerous discussions on how to read an analog clock using computer vision libraries such as OpenCV, which suggest detecting the hands or pointers as lines and determining the readings based on the angles of these lines, as reported in [9]. Based on these discussions and the specific features of the "three pointer" altimeter, the following steps can be outlined for reading the altimeter: (1) locate the altimeter in the image, (2) perform image enhancement on the cropped altimeter image, (3) detect the lines representing the pointers from the enhanced image, and (4) calculate the altitude from the detected lines.

¹<https://www.ukairsports.com/news/how-to-set-up-your-instrument-altimeter-settings/>

3.2.1. Locating the altimeter

As shown in Fig. 3.2, the altitude is usually located at the bottom left in an image. However, due to scanning error, the position is not always the same. Therefore, an automated way of detecting the position of the altimeter is necessary.

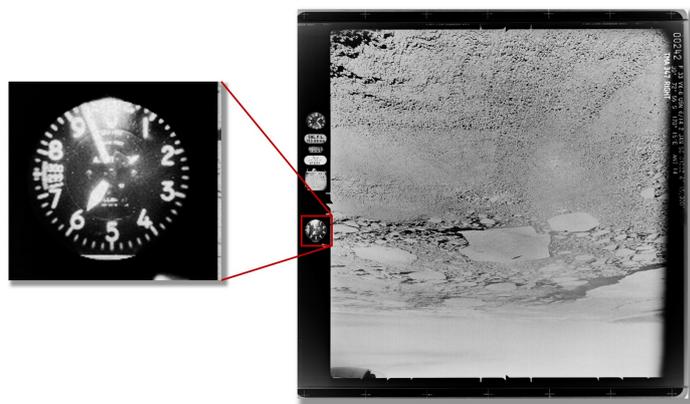


Figure 3.2: Location of altimeters in images

Dlib `object_detector`

The task of locating a bounding box for the altimeter in an image is done with help of Dlib. Dlib is a C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems [10]. `Object_detector` is a tool in Dlib for detecting the positions of objects in images or videos. It uses a machine learning algorithm, specifically a deep learning-based method known as the HOG (Histogram of Oriented Gradients) features combined with a linear SVM (Support Vector Machine), to perform object detection [11]. The `Object_detector` in dlib can detect a wide variety of objects, including faces, pedestrians, cars, and more, depending on the type of data used to train the SVM.

HOG is a feature extraction method that describes the local object appearance and shape based on the image gradient information. SVM is a supervised machine learning algorithm used for classification and regression tasks, in which a hyperplane is trained to separate data into different classes or predict continuous values. In `Object_detector`, the HOG features are computed from the image gradient information and describe the local object appearance and shape, while the SVM is trained on these features for object/non-object classification.

HOG is well-suited for detecting altimeters in images, as it provides a robust representation of the object's shape and appearance. HOG features are based on the gradient information of the image, making them invariant to changes in illumination and small deformations. This robustness to lighting changes and small variations in object position makes HOG particularly useful for detecting objects in cluttered scenes or with partial occlusion. In the case of altimeter detection, HOG can effectively capture the clear circular shape of the altimeters, allowing for accurate detection even when parts of the altimeter are missing in the image due to scanning errors.

Create training dataset and the training process

The training dataset and testing dataset were made with a simple graphical tool named Imglab. Imglab allows users to annotated images with object bounding boxes. The number of images in the training and testing dataset are 54 and 6, respectively. The ratio between items in training and testing datasets is 9:1.

The training process was performed using the python file `train_object_detection.py`, which is provided in the `dlib` library. Upon completion of the training, the detector is able to identify the location of the altimeter within an image and draw a bounding box around it. Due to the limited computer memory available, it was necessary to reduce the size of the high resolution version of photos in training dataset before the training process could be performed. This was achieved by converting the images to JPEG format with reduced quality. Although this reduction in quality may result in a slight decrease in detection accuracy, it is deemed acceptable due to the significant improvement in efficiency gained.

The training and testing accuracy of the `Object_detector` model are both 1.0, which raises questions about the model's ability to generalize to new data. Overfitting, in which a model memorizes the training data rather than learning general patterns, is often a result of having too few training examples, a complex model with many parameters, or a lack of regularization. In this case, the small size of the training dataset, with only 54 images, suggests that overfitting may be a concern. However, the fact that there is always only one altimeter in each image, positioned in a similar location, suggests that the model may be performing perfectly rather than overfitting. Despite the small size of the training dataset, the model appears to be correctly detecting every object in every image, as evidenced by the 1.0 testing accuracy. However, this result may still be unrealistic and raises questions about the generalizability of the model to new data.

The true performance of the `Object_detector` model can only be determined through testing on a larger, unseen dataset. The model was later evaluated on 106 additional images and was found to have an accuracy of 98.11%, with only 2 of the 106 images not being detected correctly. No false detection of altimeters were observed. Upon examination of the two images that were not recognized, it was discovered that the images were of extremely poor quality (as shown in [Fig. 3.3](#)), making it difficult for even the human eye to discern the altimeter or other information. Given that these cases are rare and would likely be filtered out in practical use, it can be concluded that the `Object_detector` is performing very well and is not overfitting the training data.



Figure 3.3: An example of an unreadable image

Template matching

After obtaining the bounding box of the altimeter, the next step is to determine its exact position. The altimeter can be represented as a circle, so the initial approach was to use the Hough Circle method to detect it. The Hough Circle method is a computer vision technique that transforms the image into a parameter space and identifies peaks that correspond to circles. However, due to scanning errors, the Hough Circle method may not provide a precise position of the circle. For example, during scanning, images may become tilted and the round "frame" detected by the Hough Circle method around the altimeter may not actually correspond to the correct circle, with the center of the detected circle not matching the center of the dial of the altimeter.

Template matching is a solution to the problem of detecting the precise position of the altimeter. As shown in [Fig. 3.4](#), three templates are used to represent the numbers three, five, and eight, which were chosen because they are located at relatively similar distances from each other. This design ensures that even when one or two of the numbers are not recognizable due to reflection or the blocking of the 100 feet pointer, the remaining number(s) is more likely to remain recognizable. The template matching process involves transforming a black and white image of the altimeter into a set of templates and finding the best match. The returned match gives the exact position of the number on the dial in the image, which can then be used to compute the exact position of the circle representing the altimeter using the fixed relative position between the number and the center of the circle

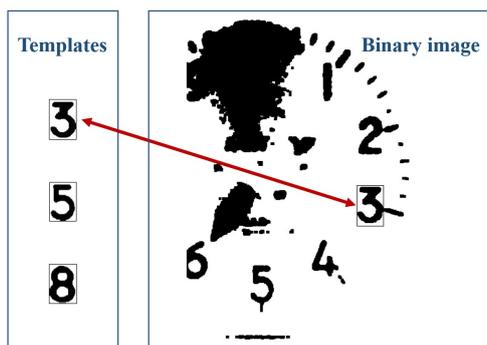


Figure 3.4: Template matching

3.2.2. Image Enhancement

Image enhancement is a crucial step in preparing the altimeter images for line detection. The images can vary greatly in brightness and clarity due to varying lighting conditions and reflections during scanning. Proper image enhancement can improve these issues and provide better results for the subsequent processing steps.

The image enhancement process includes histogram equalization, which helps to balance the brightness of the images and bring them to a more consistent level. As shown in Fig. 3.5, histogram equalization improves the images' brightness and makes them more suitable for further binarization. The equalization process is only applied to the central area of the altimeter, which helps to eliminate the impact of reflected light, typically found in the upper edge of the altimeter images.

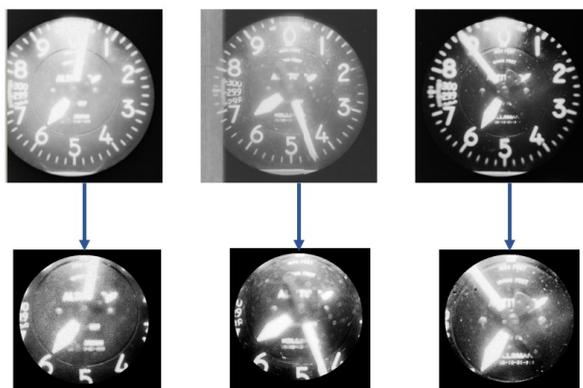


Figure 3.5: Images before and after histogram equalization

The next step after histogram equalization is image sharpening to improve the clarity of the image. This is followed by denoising to remove any noise or graininess in the image. Further, contrast enhancement is performed to increase the difference between the bright and dark areas of the image. Finally, the image is binarized, meaning that the pixels are transformed into either black or white based on a set threshold. The results of each step can be seen in Fig. 3.6.

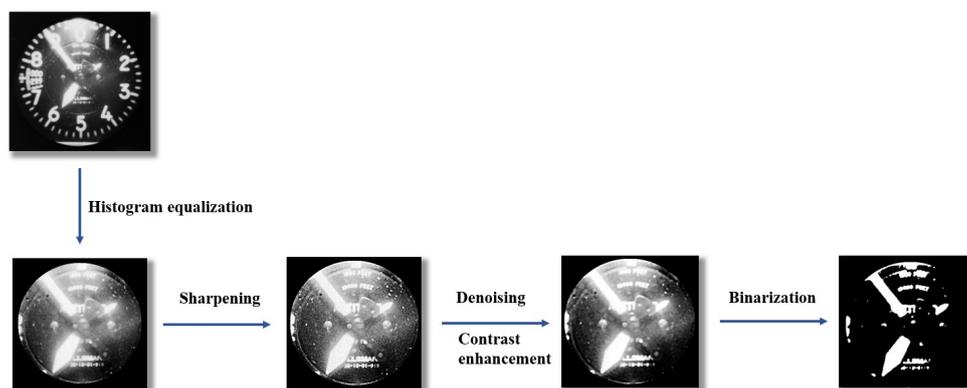


Figure 3.6: Sharpening, denoising and contrast enhancement, and binarization

3.2.3. Line Detection

The Hough line detection method involves transforming the image into a parameter space and searching for peaks that correspond to lines. This process allows for the detection of lines within the image. However, it is important to note that not all detected lines may represent the edges of the pointers, as depicted in Fig. 3.7. Thus, the next step is to select the lines that are meaningful and filter out the unwanted lines.

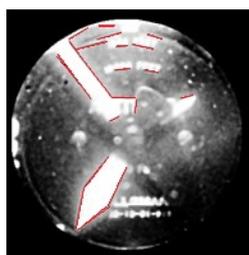


Figure 3.7: An example of line detection

There are a total of six lines that could potentially represent the edges of the pointers. The 100 feet pointer should be represented by two parallel lines, which are likely to be the longest among all the lines. The 1000 feet pointer should be represented by two intersecting lines, which are the edges of its tip, and two additional lines representing the sides.

However, due to its small size and bottom layer location, the 10,000 feet pointer is often obstructed and difficult to detect through line detection. An alternative method will be introduced in Chapter 3.1.4 to address this challenge.

The approach to selecting the correct lines is based on their relative positioning. For example, the two lines that represent the edges of the 1000 feet pointer's tip should intersect and the distance between the intersection point and the circle's center should be within a defined range. Additionally, the angle between these two lines should also be within a specified range. By considering these positional relationships, the lines of the pointers can be accurately determined.

3.2.4. Computing the altitude

With the lines representing the pointers, we can determine the position of the 100 and 1000 feet pointers. To determine which number the pointers are pointing to, we calculate the angle between the line of the pointer and the line pointing to position zero. The angle range is from 0° to 360° , with 0° indicating position 0 and 180° indicating position five. This way, we obtain the readings of the 100 and 1000 feet pointers.

As discussed in Chapter 3.1.3, the detection of the 10,000 feet pointer is challenging due to its small size and bottom layer placement. However, given that the flight height is always under 30,000 feet, the 10,000 feet pointer will always be pointing between zero and one, one and two, or two and three. Based on the readings of the hundreds and thousands of feet, there are only three possibilities for the tens of thousands of feet. By computing the position of the 10,000 feet pointer under these three situations, we can search for the 10,000 feet pointer around these three positions on the image. Further assumptions and tests are applied to determine the pointing of the 10,000 feet pointer. By combining the readings of the three pointers, the final height reading can be calculated.

3.3. Further analysis of altimeter readings: camera altitude change and footprint calculation

This section provides a further analysis of the altimeter readings from a more holistic perspective. Two key aspects of this analysis are the camera altitude change and footprint calculation.

3.3.1. Camera altitude change graph

To gain insight into the changes in flight altitude, camera altitude change line graphs were created for two flight lines. The purpose of these graphs is to display the general altitude changes throughout the flight and provide a preliminary assessment of the altimeter readings. Significant changes in altitude can indicate inaccurate readings, which often appear as outliers on the line graph. The interquartile range (IQR) method was used to identify and eliminate these incorrect readings. The IQR method considers data points that fall below the first quartile (25th percentile) minus 1.5 times the interquartile range or above the third quartile (75th percentile) plus 1.5 times the IQR as outliers. By removing these incorrect readings, the line graph can offer a clearer representation of the changes in flight altitude.

3.3.2. Footprint calculation

The footprint of each image can be computed based on the altitude readings. Typically, an image's footprint on the ground is a trapezoid shape. The shape, size, and position of the trapezoid are determined by the camera's external orientation (EO) parameters, including X, Y, Z, roll, pitch, and yaw, as well as the camera's horizontal and vertical field of view (FoV).

The process of computing the image's footprint is relatively straightforward. First, the four rays from the camera must be computed, which represent the four edges of a quadrangular pyramid in three-dimensional space, as shown in [Fig. 3.8](#). The viewing range of the camera is defined by this pyramid. The four rays are then rotated to

compensate for the camera's EO. Finally, the points of intersection of the four rays with the ground are found, and these four points define the corners of the trapezoidal footprint.

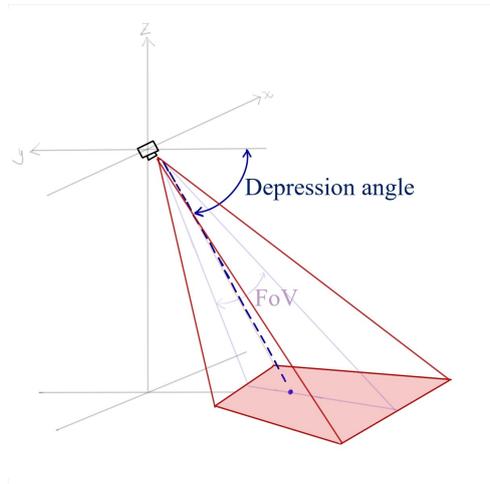


Figure 3.8: Camera footprint

As shown in Fig. 3.9, in our case, the X axis is the direction of the flight line which the airplane is flying into, Y axis is the left side to the flying direction, and Z axis is perpendicular to the X-Y plane up. Correspondingly, roll (θ) is the rotation in degrees around the X axis, pitch (ϕ) is the rotation in degrees around the Y axis, and yaw (ψ) is the rotation in degrees around the X axis.

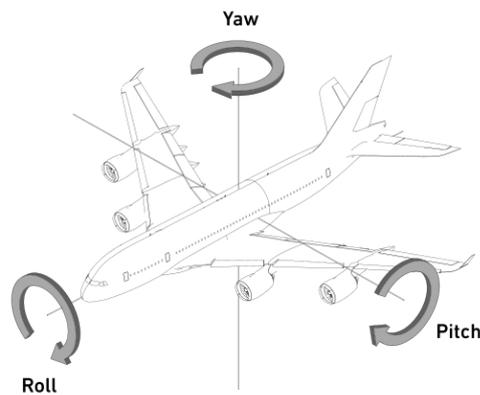


Figure 3.9: Roll, pitch, and yaw ²

Four rays from the camera

The first step in computing the footprint is to compute the four rays that represent the edges of the quadrangular pyramid. In the camera frame, the rays have the directions as

²<https://www.versluis.com/2020/09/what-is-yaw-pitch-and-roll-in-3d-axis-values/>

described in [12]:

$$\begin{aligned}
 r_{bl} &= \left(1, \tan \frac{fov_h}{2}, \tan \frac{fov_v}{2}\right) \\
 r_{tl} &= \left(1, -\tan \frac{fov_h}{2}, \tan \frac{fov_v}{2}\right) \\
 r_{br} &= \left(1, \tan \frac{fov_h}{2}, -\tan \frac{fov_v}{2}\right) \\
 r_{tr} &= \left(1, -\tan \frac{fov_h}{2}, -\tan \frac{fov_v}{2}\right)
 \end{aligned} \tag{3.2}$$

where fov_v and fov_h are the vertical and horizontal field of view (FoV), respectively. The variable r represents a ray, and the subscript of r indicates the position of the ray with respect to the camera, i.e., top or bottom, left or right.

The four rays are then transformed into the ground frame by rotating the camera frame in three axes. The roll (ϕ) depends on the depression angle, the yaw (θ) depends on the flight line direction, and the pitch (ψ) is assumed to be zero at all times. The full rotation matrices for the camera can be constructed based on the rotations around the Z-axis, X-axis, and Y-axis, as described by equations in Eq. (3.3):

$$\begin{aligned}
 R_z &= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 R_x &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\
 R_y &= \begin{bmatrix} \cos\psi & 0 & -\sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{bmatrix}
 \end{aligned} \tag{3.3}$$

Computing intersection points

With the four rays representing the edges of the quadrangular pyramid known, the next step is to calculate the points of intersection with the ground using algebraic methods. The ground surface is assumed to be a flat plane for the purposes of this estimate, and the height of the plane is calculated as the difference between the altitude reading and the ground elevation, which is obtained from Quantarctica [5].

As the FoV is known to be 60° in both the horizontal and vertical directions, and the depression angle is 30° according to information provided by USGS, all the necessary parameters are known to perform the calculation.

The resulting four intersection points are then recorded as polygons in geojson files, which can be viewed in QGIS for visualization purposes.

4

Results and Analysis

In this chapter, the results of the study are presented and analyzed. The evaluation of the performance of the altimeter reader created in this study is presented in section 4.1. In section 4.2, the camera altitude change graph of two flight lines is presented and analyzed. In section 4.3, the illustrations of the footprints calculated for the same two flight lines are presented. One image is selected for comparison with its corresponding footprint to assess the validity of the calculated positions of the footprints.

4.1. Evaluations of the altimeter reader

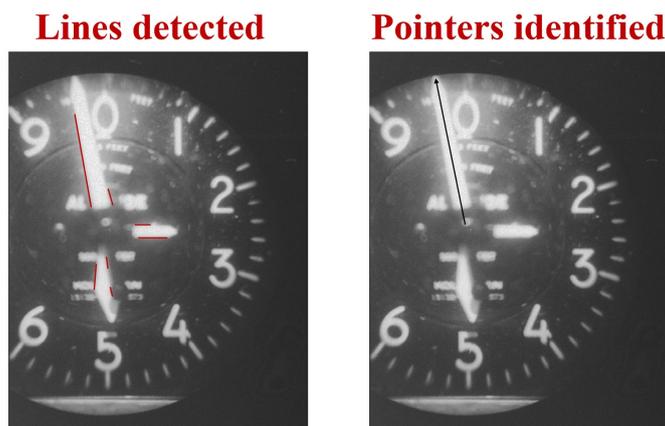
To assess the performance of the altimeter reader developed in this study, a simple scoring system was established. A total of 106 altimeter images were manually read by two human operators to minimize the impact of human error. The same 106 images were then read by the altimeter reader and the results were compared. The difference threshold for a correct reading was set at 300 ft, as height fluctuations within this range were commonly observed in height change graphs (as shown in Fig. 4.2b and Fig. 4.3b). The results of the comparison are presented in Table 4.1.

	total	106
	correct	81
incorrect	template matching	0
	object detector	2
	image enhancement	5
	pointer identification	18
	accuracy	0.764

Table 4.1: Accuracy of the altimeter reader

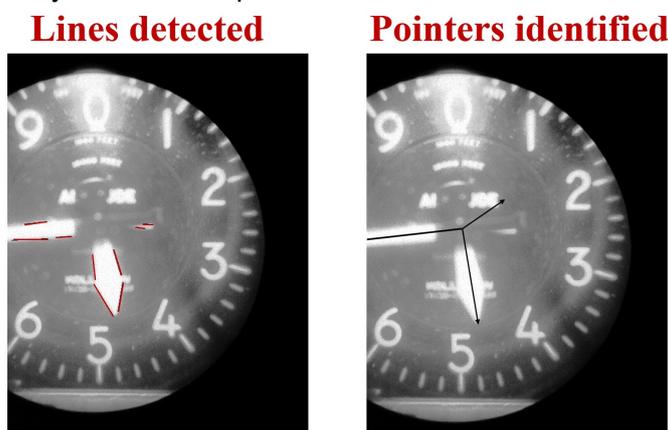
The data in the table illustrates the four potential causes for incorrect readings. Template matching errors occurred when the matching process failed to identify a match for the three numbers used as templates, but no such errors were observed in the study, which suggests that the template matching methodology used is robust. Altimeter detection errors occurred when the altimeter was not detected by the dlib's `Object_detector`. The table shows that two altimeters were not detected, which was likely due to the poor

quality of the scanned images, as discussed in Chapter 3.2.1. These images would likely not be used in practice, therefore the object detector is performing effectively. Image enhancement errors happened when the altimeter image was not properly enhanced, making it difficult for the Hough line detection to locate the edges of the pointers and leading to "cannot read" results. An example of failed image enhancement is shown in [Fig. 4.1a](#). Out of the incorrect readings, five were due to insufficient image enhancement. Lastly, pointer identification errors resulted from mistakes made during the identification of the three pointers. For example, misidentifying the 1000 feet pointer as the 100 feet pointer would result in an incorrect reading. The most often case for this type of error is when the 10,000 feet pointer is not recognizable. When the 10,000 feet pointer is not identified, the workflow would simply assume it is pointing at position one, which is not always correct. An example of failure of identifying the 10,000 feet pointer is shown in [Fig. 4.1b](#). Of all incorrect readings, 18 were caused by incorrect pointer identification, accounting for 72% of all the incorrect readings, suggesting that the pointer identification process requires improvement.



(a) Image enhancement

As shown in the figure on the left, the edge lines of the 1000 feet pointer is not detected, which is due to inadequate image enhancement in previous steps. As a result the 1000 feet pointer is not identified in pointer identification process. The "cannot read" result was produced because the identification of only the 100 feet pointer was not sufficient to calculate a reading.



(b) Pointer identification

As shown in the figure on the left, the lines detected from the altimeter image accurately capture the edge lines of the 100 and 1000 feet pointers. The figure on the right demonstrates the successful identification of these two pointers. However, the 10,000 feet pointer was not identified correctly due to limitations in the pointer recognition process, resulting in an incorrect reading.

Figure 4.1: Examples of incorrect reading results due to image enhancement and pointer identification.

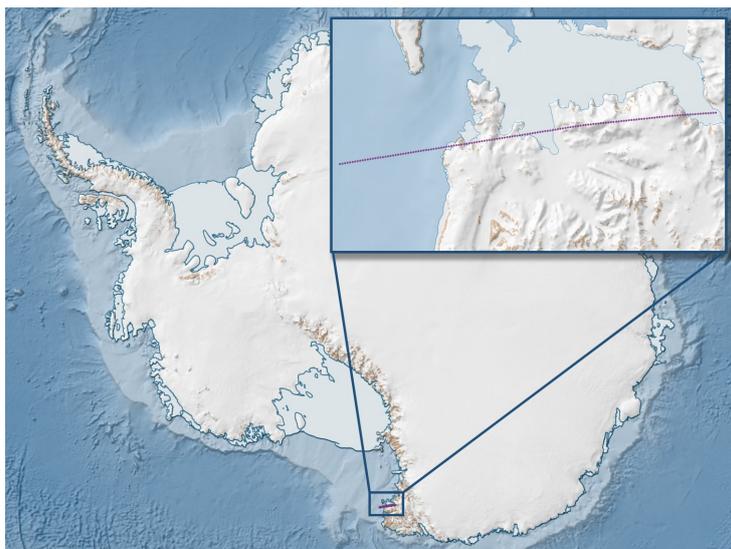
The accuracy of the altimeter reader is 0.764, which is an acceptable result. It should be noted that two images with extremely poor scanning quality were included in the evaluation, which would most likely not be used in practice. This means the actual accuracy of the altimeter reader is likely to be slightly higher.

4.2. Camera altitude change graph

Two flight lines are analyzed in this study: flight line 0347 and flight line 1815. The camera altitude change graphs are made for both flight lines. Below more detailed information about the flight lines and the generated graphs can be found.

4.2.1. Flight line 0347

The flight line 0347, located around Coulman Island in Antarctica, is depicted in Fig. 4.2a and is flown in the direction from left to right and upwards. A total of 149 right-oblique images were taken along this line, from which the camera height information was obtained. Fig. 4.2b shows that the camera altitudes generally ranged from 15,700 to 16,200 ft, with six readings that were significantly larger than the others and considered to be incorrect (photo-IDs of the incorrect readings are: 278, 284, 355, 370, 373, and 386). The application of the Interquartile Range (IQR) method to remove these readings resulted in a more realistic line graph with gradual height changes within a reasonable range, as shown in Fig. 4.2c.



(a) Flight line 0347

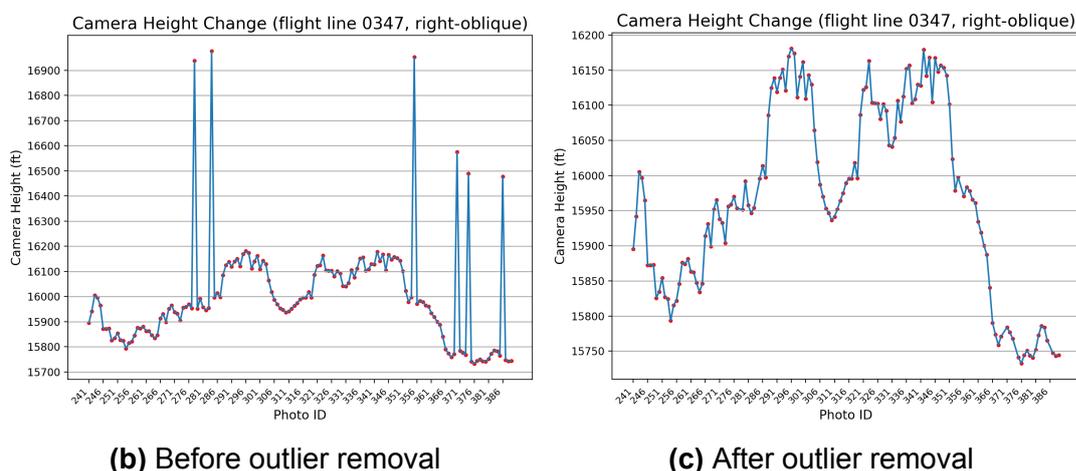
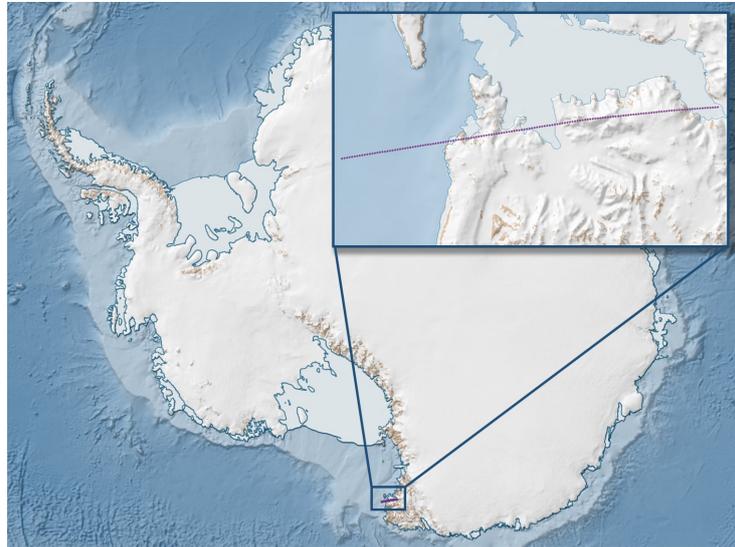


Figure 4.2: Flight line 0347 and altitude change graph

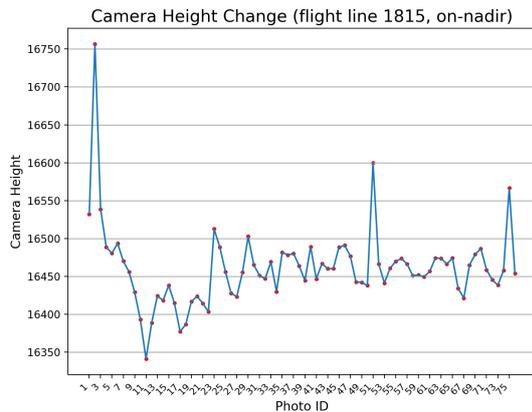
4.2.2. Flight line 1815

As shown in Fig. 4.3a, flight line 1815 is located around the Antarctic and is flown from left to right and downwards. The camera altitudes are computed based on the 76 on-nadir photos taken along this flight line. As seen in Fig. 4.3b, there are several points with

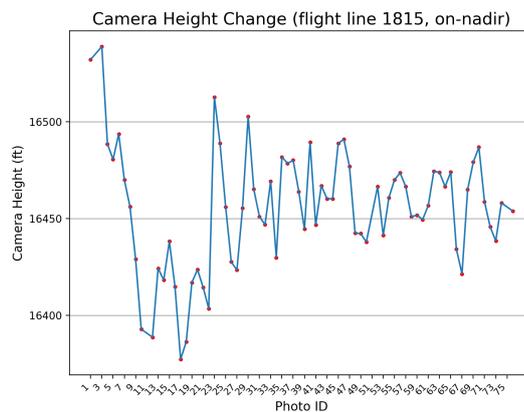
sudden value changes that are suspected to be errors. However, these changes are not as pronounced as those in flight line 0347, so the IQR method was used to determine the incorrect readings. The IQR method identified four incorrect readings (photo IDs: 2, 11, 50, and 75), which were removed, resulting in relatively stable altitudes between 16,350 and 16,500 ft, as shown in Fig. 4.3c.



(a) Flight line 0347



(b) Before outlier removal



(c) After outlier removal

Figure 4.3: Flight line 1815 and altitude change graph

4.3. Footprints calculation

The footprints of the photos on flight lines 0347 and 1815 were computed and visualized in QGIS. The visualizations for each flight line will be presented. To validate the accuracy of the computed footprints, photo 260 on flight line 0347 and its corresponding footprint were selected for comparison to assess if the footprint accurately covers the intended area on the ground.

4.3.1. Flight line 0347

The footprints of the right-oblique photographs on flight line 0347 are illustrated in [Fig. 4.4](#). With a depression angle of 30° , the camera is oriented to the right. The camera's field of view of 60° extends to the horizon, resulting in footprints that extend downward to infinity. The adjacent photos have a substantial overlap along the flight direction, estimated to be over 90% as shown in the figure. This high degree of overlap along the flight direction helps to mitigate the impact of terrain on the images.

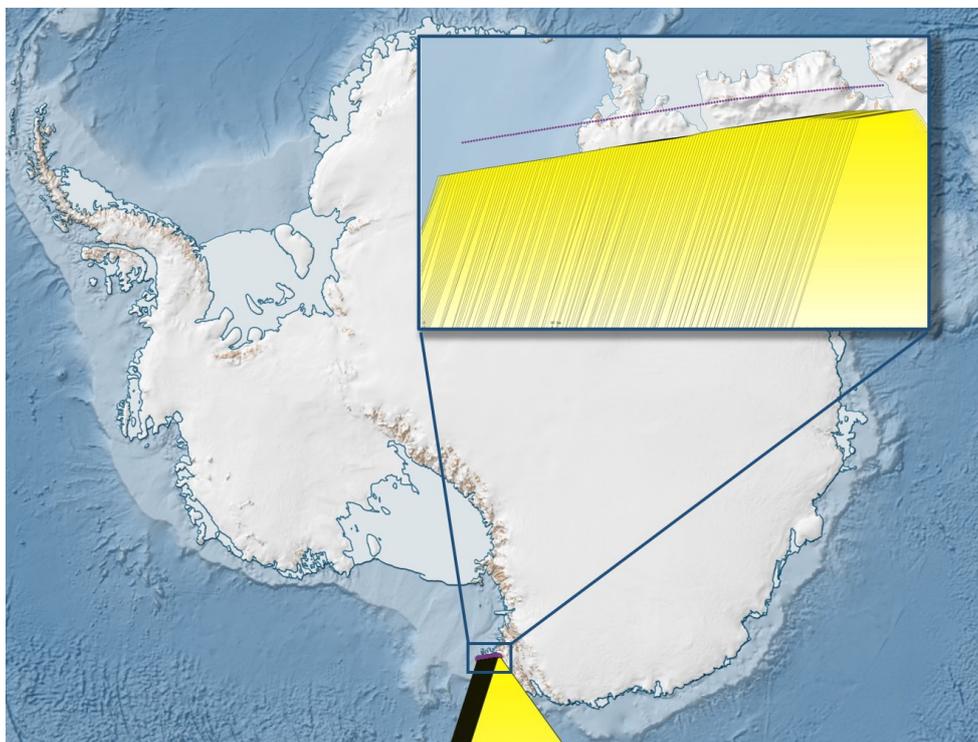


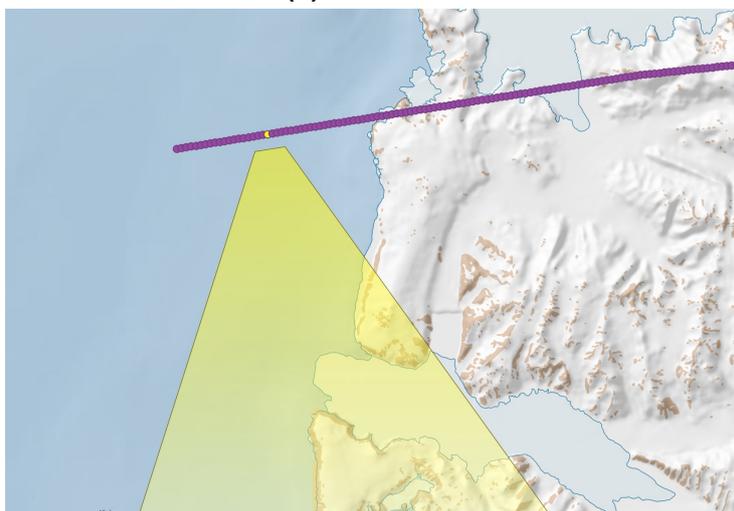
Figure 4.4: Footprints of flight line 0347

Photo 260

To confirm the validity of the footprint locations, a comparison was made between the footprint of photo 260 and the actual photo 260, as shown in [Fig. 4.5](#). The photograph depicts a clear dividing line between land and sea in the upper left part. In [Fig. 4.5b](#), the footprint accurately covers this coastline. As the airplane was flying from left to right and upwards, we can confidently assert that the coastlines in the two figures are the same, and the footprint provides an accurate representation of the features captured in the photo.



(a) Photo 260



(b) Footprint of photo 260

Figure 4.5: Photo 260 in flight line 0347

4.3.2. Flight line 1815

As for flight line 1815, footprints of on-nadir photos are shown in [Fig. 4.5](#). With the camera facing straight down, and a FoV of 60° , the on-nadir images are expected to cover the gaps in between the right-oblique and left-oblique images.

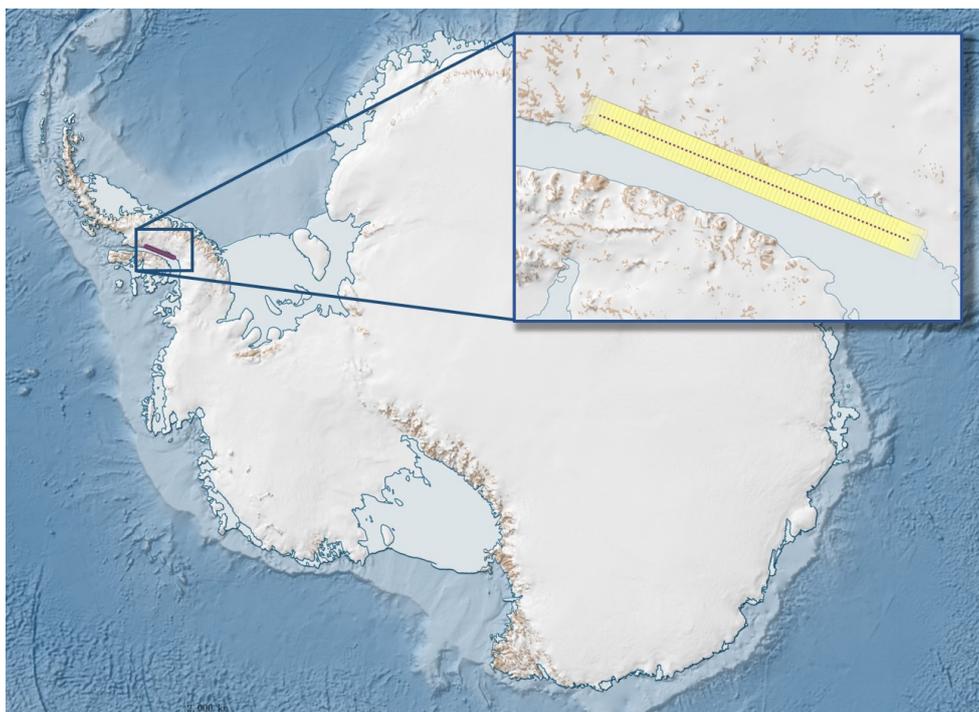


Figure 4.6: Footprint of flight line 1815

Conclusion and Recommendations

5.1. Conclusion

In this thesis, a workflow was developed for the automatic extraction of altitude information from historical aerial photographs. This information was used to compute the camera altitude change of flight lines and photo footprints. The workflow's performance was evaluated using a simple scoring system that compared the readings of 106 randomly selected photos obtained by the reader to those obtained by two human evaluators. The overall accuracy of the altitude reader was found to be 0.764.

However, the results of this study indicate that the accuracy of the altitude reader can vary widely between flight lines. While the overall accuracy of the reader was found to be 0.764, the accuracy of readings taken from individual flight lines can be significantly higher or lower. For example, in [Fig. 4.2b](#) and [Fig. 4.3b](#), the accuracy was found to be 0.960 and 0.947, respectively, for readings taken from two separate flight lines. These values were significantly higher than the overall accuracy of 0.764, but other flight lines could potentially have lower accuracy values. This variability is likely due to differences in the scanning quality of the images between flight lines.

The footprints generated by the workflow provide useful information about the locations within the images and serve as a starting point for further image matching and georeferencing procedures. Furthermore, the footprints in [Fig. 4.4](#) and [Fig. 4.5](#) demonstrate that the workflow is capable of producing a horizon-to-horizon coverage of 180° when the field of view is set to 60° and the depression angle is set to 30° .

5.2. Recommendations

As demonstrated by [Table 4.1](#), the leading cause of incorrect readings is the failure of pointer identification, accounting for the largest proportion of incorrect cases. This suggests that the current pointer identification process in the workflow is inadequate, and improvements are necessary to accurately identify the three pointers. The second most frequent cause of incorrect readings is inadequate image enhancement, which results in significant noise and uneven background brightness. This causes the lines representing the pointers to be unclear and difficult to detect, and can also contribute to errors in pointer identification. Hence, improving both the image enhancement process and the mechanism for pointer identification would be highly beneficial.

To further improve the results, utilizing information that is present within the results

themselves should also be considered. For example, taking into account altitude readings from photos taken in close proximity in time on the same flight line or from all photos taken on the flight line may be useful. Given that sudden changes in altitude during flight are uncommon, these flight altitude changes can be utilized to validate the reliability of a reading and potentially assist in determining the ten thousands digit of the altitude (i.e. where the 10,000 feet pointer is located).

References

- [1] Anders A Bjørk et al. “An aerial view of 80 years of climate-related glacier fluctuations in southeast Greenland”. In: *Nature Geoscience* 5.6 (2012), pp. 427–432.
- [2] Jordan R Mertes et al. “Using structure-from-motion to create glacier DEMs and orthoimagery from historical terrestrial and oblique aerial imagery”. In: *Earth Surface Processes and Landforms* 42.14 (2017), pp. 2350–2364.
- [3] Sarah F Child et al. “Structure-from-motion photogrammetry of Antarctic historical aerial photographs in conjunction with ground control derived from satellite data”. In: *Remote Sensing* 13.1 (2020), p. 21.
- [4] Earth Resources Observation et al. *USGS EROS Archive - Aerial Photography - Antarctic Single Frame Records*. 2018. URL: https://www.usgs.gov/centers/eros/science/usgs-eros-archive-aerial-photography-antarctic-single-frame-records?qt-science_center_objects=0#overview (visited on 01/20/2023).
- [5] Norwegian Polar Institute. *Quantarctica*. 2022. URL: <https://www.npolar.no/en/quantarctica/> (visited on 01/20/2023).
- [6] SKYbrary. *Altimeter*. 2021. URL: <https://www.skybrary.aero/articles/altimeter> (visited on 01/20/2023).
- [7] ASA. *Flight Instruments: The Altimeter and Altitudes*. 2015. URL: <http://learntoflyblog.com/2015/06/22/flight-instruments-the-altimeter-and-altitudes/> (visited on 01/20/2023).
- [8] Wikipedia contributors. *Altimeter — Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/w/index.php?title=Altimeter&oldid=1108737671>. [Online; accessed 28-January-2023]. 2022.
- [9] Joseph Butler et al. *python-cv-samples/examples/analog-gauge-reader/*. [https://github.com/intel-iot-devkit/python-cv-samples/tree/master/examples/analog-gauge-reader/](https://github.com/intel-iot-devkit/python-cv-samples/tree/master/examples/analog-gauge-reader). 2023.
- [10] Dlib contributors. *Dlib C++ Library*. 2022. URL: <http://dlib.net/> (visited on 01/20/2023).
- [11] Navneet Dalal et al. “Histograms of oriented gradients for human detection”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. Ieee. 2005, pp. 886–893.
- [12] user3235832. *Calculating coordinates of an oblique aerial image*. 2017. URL: <https://stackoverflow.com/questions/38099915/calculating-coordinates-of-an-oblique-aerial-image> (visited on 01/20/2023).



Python Source Code

The source code for this project can be found on GitHub at https://github.com/yushan613/Altimeter_Reader