POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA CIVILE,
AMBIENTALE E TERRITORIALE

Design and Implementation of a QGIS Plugin for Efficient Glacier Feature Monitoring and Reporting

TESI DI LAUREA MAGISTRALE IN
ENVIRONMENTAL AND LAND PLANNING ENGINEERING
INGEGNERIA PER L'AMBIENTE E IL TERRITORIO

Author: **Sebastián Fajardo Turner**

Student ID: 217089 / 10871219
Advisor: Prof. Federica Migliaccio
Co-advisor: Eng. Federica Gaspari
Academic Year: 2024-25

Abstract

This thesis presents the design and implementation of a QGIS plugin aimed at facilitating glacier monitoring, specifically focusing on the Belvedere Glacier in the Italian Alps. The plugin is designed to streamline and automate essential analysis processes, including elevation and volume change detection, displacement and surface velocity mapping, and the generation of Ground Control Point reports. The motivation behind this work stems from the need to simplify and standardize the data processing workflows that follow annual glacier survey campaigns, enabling wider accessibility to glacier data analysis while reducing the required expertise and time commitment.

The methodology consists in integrating geospatial data from Digital Surface Models and Ground Control Point measurements into the QGIS environment, providing a user-friendly interface that supports tasks such as calculating glacier volume changes and applying symbolism to visualize displacement and surface velocity. The plugin's core functionalities—automatic interpolation of glacier surface changes and the creation of standardized monographs for GCPs—were tested using data from the Belvedere Glacier monitoring project, providing an efficient solution for the long-term sustainability of the research program.

The results demonstrate the plugin's ability to significantly reduce the time required to analyze glacier dynamics while ensuring consistent and reliable outputs. In particular, the plugin has proven useful in visualizing trends in glacier displacement, surface velocity, and acceleration. These insights are crucial for understanding the glacier's response to climate change and evaluating potential hazards like glacier surges or collapses. The plugin's open-source nature ensures it can be adapted for other glaciers and contexts, making it a versatile tool for the glaciological community. Its application extends beyond research, providing an educational platform for students and a practical tool for professionals in glacier monitoring.

Key-words: Glacier monitoring, QGIS plugin, Belvedere Glacier, Digital Surface Models, Ground Control Points, surface velocity, glacier displacement, PyQGIS.

Abstract in italiano

Questa tesi presenta la progettazione e l'implementazione di un plugin per QGIS volto a facilitare il monitoraggio dei ghiacciai, con un approfondimento specifico sul caso studio del Ghiacciaio del Belvedere nelle Alpi italiane. Il plugin è stato progettato per semplificare e automatizzare i processi di analisi essenziali, tra cui il rilevamento delle variazioni di elevazione, la mappatura dello spostamento e della velocità superficiale, e la generazione di monografie di Punti di Controllo (GCP). La motivazione dietro questo lavoro deriva dalla necessità di semplificare e standardizzare i flussi di lavoro di elaborazione dei dati raccolti durante le campagne annuali di rilevamento dei ghiacciai, consentendo una maggiore accessibilità all'analisi dei dati glaciologici e riducendo il livello di competenza richiesto e il tempo necessario.

La metodologia prevede l'integrazione di dati geospaziali provenienti da Modelli Digitali di Superficie (DSM) e misurazioni dei GCPs in ambiente QGIS, fornendo un'interfaccia intuitiva che supporta attività come il calcolo delle variazioni di volume del ghiacciaio e l'applicazione della simbologia per visualizzare lo spostamento e la velocità superficiale. Le funzionalità principali del plugin—il calcolo automatico delle variazioni della superficie del ghiacciaio e la produzione di monografie standardizzate per i GCP—sono state testate utilizzando dati del progetto di monitoraggio del Ghiacciaio del Belvedere, offrendo una soluzione efficiente per il monitoraggio e l'analisi glaciologica a lungo termine.

I risultati dimostrano la capacità del plugin di ridurre significativamente il tempo necessario per analizzare la dinamica dei ghiacciai, garantendo al contempo risultati e prodotti finali coerenti e affidabili. In particolare, il plugin si è rivelato utile per visualizzare le tendenze di spostamento del ghiacciaio, velocità superficiale e accelerazione. Questi strumenti sono cruciali per comprendere la risposta del ghiacciaio ai cambiamenti climatici e valutare potenziali rischi, come le ondate o i collassi glaciali. La natura open-source del plugin garantisce che possa essere adattato ad altri ghiacciai e contesti, rendendolo uno strumento versatile per la comunità glaciologica. La sua applicazione si estende oltre la ricerca, fornendo una piattaforma educativa per gli studenti e uno strumento pratico per i professionisti del monitoraggio dei ghiacciai.

Parole chiave: Monitoraggio dei ghiacciai, plugin QGIS, Ghiacciaio del Belvedere, Modelli Digitali di Superficie, Punti di Controllo a Terra, velocità superficiale, spostamento del ghiacciaio, PyQGIS.

Contents

1 A QGIS plugin for Glacier Monitoring

This chapter introduces the motivation, objectives, and foundational structure of the QGIS plugin developed to facilitate glacier monitoring, with the Belvedere Glacier as a primary case study. The plugin addresses the increasing need for standardized, accessible tools that streamline glacier monitoring workflows, which are often time-intensive and complex due to the remote and challenging environments of glaciers. This chapter details the plugin's purpose and key features, including automated elevation change analysis, interpolation of point-based data, standardized Ground Control Point (GCP) reporting, and advanced symbology options for visualizing glacier dynamics. Additionally, it discusses the benefits of implementing this tool within the open-source QGIS platform, which allows for ease of use, adaptability, and broad accessibility for the global glacier monitoring community.

1.1. Introduction

The present work has as an objective the design and implementation of a QGIS plugin to facilitate and automate required analysis processes performed during glacier monitoring, using the Belvedere Glacier as a baseline.

The motivation for developing a QGIS plugin to aid in glacier monitoring stems from the increasing need for accessible, standardized tools that can support efficient analysis of glacier data. Glaciers are sensitive indicators of climate change, yet their remote locations and challenging terrain make regular monitoring complex and time intensive. This project seeks to streamline essential processes such as elevation change processing, field interpolations from measurement points, and data visualization, making these tasks more accessible to researchers, students, and practitioners. By creating a plugin within the QGIS environment, a wider array of people can contribute to data processing without the need for a deeper understanding of a Geographic Information System environment.

This tool not only reduces the manual effort required for these analyses but also fosters consistency across studies, supporting long-term glacier monitoring efforts critical for understanding climate impacts and predicting potential hazards.

Chapter 1 outlines the structure of this thesis, which focuses on the design and implementation of a QGIS plugin for glacier monitoring, using the Belvedere Glacier as a reference.

Chapter 2 begins by addressing the motivation and relevance of glacier monitoring in the context of climate change, where accelerated glacier mass loss is one of the most prominent consequences. The significant impact of glaciers on local ecosystems and human populations underscores the importance of effective monitoring tools. Furthermore, it introduces the Belvedere Glacier.

Chapter 3 delves into the concept of Geographic Information Systems (GIS) and QGIS, a widely used open-source GIS software. This chapter discusses the technical foundation of developing a plugin for QGIS, covering essential tools, data structures, and geospatial analysis techniques. It provides the necessary background for understanding the GIS environment and tools used in the project.

Chapter 4 explains the process of developing the plugin, detailing the different components and functionalities that were designed to streamline glacier monitoring tasks. This includes the automation of data processing steps like DSM elevation change calculations, interpolation of point-based values, and the standardization of Ground Control Point (GCP) report layouts.

Chapter 5 presents the results of using the plugin with data from the Belvedere Glacier, showcasing how it simplifies and accelerates the monitoring process, as well as discussing the potential for future improvements.

Finally, **Chapter 6** provides conclusions, summarizing the benefits of the plugin, its practical applications, and the significance of automating glacier monitoring for both research and safety purposes.

1.2. Objective of the Thesis

Currently, the processing operations for volume change and velocity interpolations in QGIS are manually carried out, as well as the map report layouts, which convey disaggregated steps. A QGIS plugin or script could automate these steps, leading to time-saving and standardized outputs.

The primary objective of this thesis is to develop an open-source QGIS plugin designed to streamline and automate key processes in glacier monitoring, making it accessible and useful for any glacier monitoring program worldwide. The goal is to create a versatile tool that can be easily adapted and applied to other glacier systems. Nonetheless, the plugin has been developed using the Belvedere Glacier monitoring project as a baseline, therefore, it will be the focus of the discussion of this project. The plugin aims to address several inefficiencies in current monitoring workflows, which often involve manual data processing, inconsistent reporting, and visualization challenges. The main objectives are as follows:

1. **Automation of DSM Elevation Change Calculation:** The plugin automates the comparison of Digital Surface Models (DSMs) to track changes in glacier surface elevation over time. This process, which is typically performed manually, is

time-consuming and error prone. By automating the calculation of elevation changes between two DSMs and allowing users to apply spatial constraints (such as bounding boxes), the plugin significantly improves the efficiency and accuracy of glacier surface change analysis, facilitating rapid and standardized data processing.

2. **Interpolation of Values from Point Layers:** Glacier monitoring often involves tracking surface velocity, displacement, or other point-based data (e.g., GNSS measurements). The plugin automates the interpolation of values from these point layers, providing a clearer picture of glacier behavior over time. This feature speeds up the analysis process and improves the accuracy of the understanding of the glacier's evolution, making it a valuable tool for both researchers and practitioners involved in glacier dynamics studies.
3. **Standardization of Ground Control Point (GCP) Report Layouts:** Ground Control Points are critical for long-term monitoring of glaciers. The plugin automates the creation of standardized GCP report layouts, ensuring consistent and repeatable outputs across multiple years and campaigns. This standardization reduces the risk of human error and allows for more reliable data comparisons, making it easier to maintain high-quality records for long-term glacier monitoring.
4. **Advanced Symbology for Visualizing Glacier Data:** The plugin includes an intuitive symbology option in the interpolation tab, which allows users to apply advanced symbology techniques to visualize glacier data. For example, users can apply graduated symbology to represent velocity fields or vector field markers to visualize glacier displacement. By improving the visual representation of complex datasets, the symbology feature enhances the interpretation and communication of glacier behavior.
5. **Open-Source Accessibility and Adaptability:** A key objective of this plugin is to ensure that it is accessible to all glacier monitoring programs globally. By making the plugin open source, it encourages collaboration, adaptation, and widespread use in glacier monitoring projects beyond the initial case of the Belvedere Glacier. The tool's modular design and adaptability mean it can be customized to suit different glacier environments, making it a practical solution for diverse research teams and field conditions.

The development of this open-source plugin will significantly improve the workflow efficiency of glacier monitoring efforts, reduce manual errors, and ensure standardized outputs. By providing a freely accessible tool for the broader glacier monitoring community, the plugin has the potential to become a valuable resource for both researchers and practitioners in cryosphere studies.

The decision to develop this tool as a QGIS plugin, as opposed to standalone scripts, is driven by key advantages:

1. **Open-Source Nature of QGIS**: QGIS is a widely adopted open-source geographic information system (GIS) that offers an extensive set of geospatial tools and libraries, making it suitable for both researchers and practitioners. Its open-source nature ensures accessibility, adaptability, and long-term sustainability, which aligns with the research goals of the glacier monitoring project [1].
2. **Extensive Functionality and Customization**: QGIS provides native support for various geospatial data formats and processing algorithms, which simplifies the integration of advanced spatial analysis techniques such as elevation change detection. Moreover, QGIS allows for the development of custom plugins through its Python-based API (PyQGIS), enabling the creation of tailored tools to meet specific project requirements [1].
3. **Convenience and Integration**: Developing a QGIS plugin offers seamless integration with the existing tools and functionalities within the QGIS environment. This provides users with a familiar interface for performing analyses, reducing the learning curve compared to a standalone script. Users can perform the desired operations without having to switch between different software or manually run scripts, thereby streamlining the workflow [2].
4. **Enhanced User Experience**: A plugin within QGIS allows for better user interaction, as it provides a graphical interface that makes complex operations like DSM elevation change calculation accessible with minimal technical knowledge. In contrast, using standalone Python or R scripts may require more programming proficiency and offer less user-friendly error handling and visualization options. A plugin, on the other hand, offers an intuitive, step-by-step process for the user [2].
5. **Standardization**: One of the major challenges in long-term monitoring projects is ensuring the consistency of output over time. By embedding specific functions into a QGIS plugin, the output formats, including the statistical files and the layout templates, can be standardized. This removes potential inconsistencies that could arise from users running different versions of a standalone script or using slightly modified methods year after year [2]

2 The Relevance of Glacier Monitoring

This chapter explores the critical role of glacier monitoring as glaciers worldwide experience accelerated retreat and mass loss due to climate change. Section 2.1 begins with an overview of the current state of glaciers, particularly in alpine regions, highlighting the severe environmental, social, and economic impacts of their decline, including increased risks of hazards like glacial lake outburst floods (GLOFs). Section 2.2 then delves into essential glacier metrics, such as volume change, displacement, and surface velocity, which are vital for understanding glacier dynamics and predicting future behavior. Finally, Section 2.3 reviews a range of monitoring techniques, from traditional in-situ methods to advanced approaches using photogrammetry and UAVs, underscoring how these methods contribute to comprehensive and reliable glacier monitoring. Together, these sections underscore the importance of systematic glacier observation for climate research and adaptation planning.

2.1. The Ongoing Transformation of Glaciers in a Warming Climate

Mountain ecosystems are among the key areas of concern for climate change. Due to their dynamism, they are especially vulnerable to climate change. The alpine cryosphere is no exception [3]. Glaciers throughout the world have suffered a series of profound transformations due to the ongoing climate crisis, such as accelerated loss of both surface and volume [4]. It is estimated that as much as 50% of the glacial surface of the European Alps has been lost between 1850 and 2000, but it is also important to remark that volume loss has been much higher, with an estimated loss of two-thirds of glacial volume in the same period [5]. Furthermore, glaciers are projected to continue losing significant parts of their volume in the coming century.

To better understand the potential future of glaciers, researchers use climate change scenarios known as Representative Concentration Pathways (RCPs). These scenarios provide standardized projections of future greenhouse gas concentrations and their effects on global warming, based on different assumptions about human activities and policy decisions. Each RCP represents a possible radiative forcing value (measured in watts per square meter) by the year 2100, reflecting the degree to which greenhouse gases could alter the Earth's energy balance. RCP2.6, for example, represents a stringent mitigation scenario with low emissions, while RCP8.5 describes a high-emission, "business-as-usual" scenario with minimal climate action [6]. These scenarios are briefly explained in *Table 1*. By modeling climate impacts under these pathways, scientists can estimate how glaciers, among other systems, might respond to various levels of greenhouse gas emissions in the coming decades.

Table 1. Description of the RCP scenarios. Adapted from [7].

Scenarios	**Radiative Forcing [W/m²]**	**CO_2-eq Concentration [ppm]**	**Description**
RCP 2.6	3.0	480-530	A scenario with a strict reduction of emissions. Policies are implemented to keep global warming likely below 2°C above pre-industrial temperatures.
RCP 4.5	4.5	480-720	A reduction scenario where significant greenhouse gas mitigation policies are implemented.
RCP 6.0	6.0	720-1000	A normal reduction scenario, where ordinary greenhouse gas mitigation policies are implemented.
RCP 8.5	8.5	>1000	High greenhouse gas emissions, where no or minimal efforts to constrain emissions are implemented.

Simulations have indicated that under climate change RCP2.6 scenario, by 2100 about two-thirds of the glacier volume and area will disappear compared to 2017. Under strong warming scenarios, the European Alps will be largely ice-free by the turn of the century with estimated losses of 79% and 94% under RCP4.5 and RCP8.5 respectively [8].

The continued loss of glacial bodies has serious environmental, social, and economic repercussions. The ongoing reduction of glacial masses raises challenges for water supply during dry periods, civil security, and tourism [9]. Amongst the challenges, the rapid retreat may lead to glacial lake outburst floods (GLOFs), which represent a major hazard and can result in significant loss of life. Since 1990, the number, surface, and volume of glacial lakes throughout the world has surged, rising by 53%, 51%, and 48%, respectively [10]. In conjunction to this, many downstream catchments have seen rapid increases in population, infrastructure, hydroelectric power, and intensified agriculture. It is estimated that globally, fifteen million people are highly exposed to impacts from potential GLOFs [10]. An example of a glacial lake formation occurred in the Belvedere Glacier (Macugnaga, Italy), which formed in 2001 and widely grew up in spring 2002 on the flat tongue of the glacier. During the summer of 2002, the lake reached a surface of 150,000 square meters and a volume of three million cubic meters, with a max depth of fifty-seven meters. Fortunately, the glacial lake was pumped and emptied to reduce the risk of a potential GLOF [11].

During the past decade there has been an increase of mass movements and hazardous events located in high-elevation regions of the European Alps due to climate change [12]. Relevant, recent, and tragic examples include an ice avalanche on the northern slope of the Marmolada Glacier (Dolomites, Italy) on July 3rd, 2022. In this occasion, 96,000 cubic meters of ice collapsed which resulted in eleven dead and seven injured mountaineers [13]. Furthermore, on August 27, 2023, a significant slope instability affecting the Belvedere Glacier led to large debris flow towards the Macugnaga municipality by the river Anza, causing extensive material damage [13]. This event profoundly impacted the glacier's dynamics, causing the collapse of the central part of the terminal lobe [13].

Examples like these are becoming increasingly common around the world, making systematic monitoring of glaciers and their processes crucial. By keeping track of and understanding glacier behavior, researchers can potentially gain early warning signs of impending instability or collapse events [14]. Nevertheless, glaciers are usually located in high mountainous areas which prove to be remote and with difficult terrain. This often presents logistical and safety challenges for any monitoring program.

Due to this, while there have been efforts to compile and extend glaciological records, significant gaps and variances in data quality persist. Amongst the sources of this data include the World Glacier Monitoring Service (a worldwide collection of information about ongoing glacier changes initiated in 1894), regional reports, and historical sketches, but records vary widely; some begin as early as 1534 and others only from the 20th century, with inconsistent temporal resolution. Furthermore, there is a notable disparity in record availability, with a dense concentration of data from the European Alps. This uneven coverage poses challenges for comprehensive glacier monitoring and highlights the need for improved and more consistent data collection across the globe [4].

2.2. The Importance of Glacier Volume Change, Displacement, and Velocity Monitoring

As discussed in the previous chapter, glaciers are critical indicators of climate change. Their health offers valuable insights into both local and global environmental processes. Monitoring glacier's mass balance, displacement, and velocity provides essential data that helps researchers understand glacial dynamics and predict potential impacts on sea levels, water resources, and regional hazards. These metrics are not only useful for climate scientists, but also for local communities that depend on glacial meltwater for agriculture, hydropower, and drinking water. Furthermore, understanding how glaciers behave is vital for predicting and mitigating hazards such as GLOFs and glacier surges, which can cause significant damage. This section aims to explore the importance of monitoring these three key aspects.

2.2.1. Glacier Mass Balance

The measurement of glacier volume change provides vital information about mass balance, which is directly linked to sea-level rise and regional hydrology. As glaciers lose volume, the meltwater they release significantly contributes to global sea-level rise, threatening coastal areas worldwide [15] [16].

In addition to its global impact, glacier volume loss influences hydrology. Glaciers act as natural water reservoirs, storing water during colder months and releasing it during warmer seasons. As glaciers shrink, there is less stored water available, which can exacerbate water shortages in areas dependent on glacial meltwater [17]. For example, regions in the Himalayas, Andes, and parts of the Alps rely heavily on this seasonal meltwater for agriculture, energy, and drinking supplies [16] [18]. Furthermore, monitoring volume change is also crucial for understanding the formation of glacial lakes, which present hazards like GLOFs [10]. In the context of long-term monitoring, volume change provides a comprehensive picture of how glaciers respond to atmospheric warming, making it a critical tool for both climate scientists and policy makers [19].

There are two ways to measure a glacier's mass balance, either by glaciological (stake) or geodetic methods [20]. The glaciological method involves setting up a grid of ablation stakes multiple times a year and interpolating data across the glacier's surface. These measurements are supplemented with snow density assessments and meteorological observations for additional support. The surveys are time-consuming, physically demanding and potentially dangerous [21]. On the other hand, the geodetic method uses volume changes to estimate the mass balance over several years. This is done by comparing Digital Surface Models [22]. Digital Surface Models (DSMs) represent the elevation of the Earth's surface, including all natural and built features. Given the scope of this project, the geodetic method is used, and thus, volume changes are calculated, however, due to lack of snow density the water equivalent is not estimated.

An example of a geodetic mass balance map is shown in *Figure 1* below, which illustrates the spatial distribution of mass changes for Waldemarbreen Glacier in Svalbard [23]. This type of map is generated by comparing Digital Surface Models (DSMs) taken 2019 and 2021 to measure changes in surface elevation across the glacier.

By converting these elevation differences into mass changes, researchers can visualize the gains and losses in glacial ice over time.

In this example, regions in red represent areas of significant ice loss, while blue areas indicate no ice loss. In this example the glacier had no ice accumulation. The map allows for a detailed analysis of where the glacier is thinning or growing, providing crucial data for assessing the glacier's overall mass balance. Such maps are essential tools in glaciological studies, as they offer high spatial resolution and enable scientists to observe not just the total volume changes but also how these changes are distributed across the glacier's surface.

This geodetic method is particularly advantageous for long-term monitoring because it provides a comprehensive view of glacier dynamics over large areas. Compared to traditional stake-based measurements, which are localized, and time-intensive, geodetic mass balance maps offer a more efficient and safer way to track glacier changes. This example highlights the precision and utility of geodetic methods in providing insights into glacier health and climate change impacts.

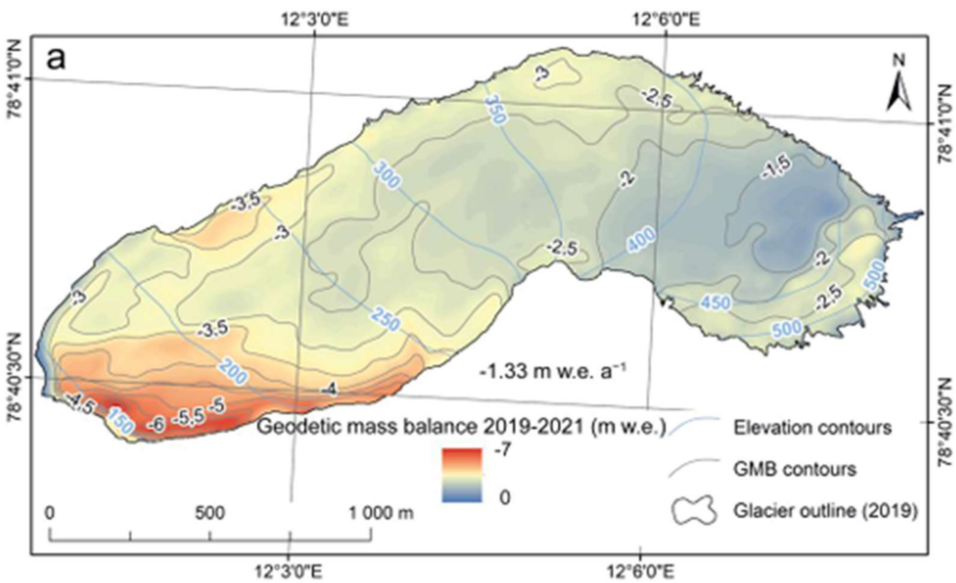


Figure 1. Geodetic mass balance of Waldemarbreen glacier, Svalbard, from 2019-2021 [23].

2.2.2. Glacier Displacement and Surface Velocity

Glacier displacement refers to the movement of ice as glaciers flow across their beds. Glacier movement occurs as ice slides over the underlying terrain, driven by three main mechanisms: basal sliding, internal ice deformation, and subglacial bed deformation. Basal sliding refers to the glacier gliding over its bed, often facilitated by a thin layer of melted water acting as a lubricant. Internal deformation happens when ice crystals shift relative to one another. Lastly, subglacial bed deformation involves the deformation of unfrozen sediments beneath the glacier due to the weight of the overlying ice [24]. Monitoring displacement helps researchers assess how glaciers respond to changing environmental conditions, such as increased meltwater at the glacier base, which can accelerate ice flow and increase the risk of calving or surges [25].

The surface velocity of glacier movement is another key parameter that indicates the glacier's flow rate. Sudden increases in velocity often signal instability, which can lead to events like glacier surges or collapses. By tracking changes in velocity over time, researchers can predict and prevent catastrophic events, such as the formation of crevasses or glacier calving that contribute to sea-level rise. Remote sensing methods have become essential for monitoring these parameters, providing large-scale, high-precision data over long periods [25].

In *Figure 2*, a surface velocity map is presented, illustrating both the speed and direction of glacier movement over the Waldermarbreen Glacier between 2019 and 2021 [23]. The color gradient in the map represents the magnitude of the surface velocity, where areas with warmer colors indicate higher speeds, and cooler colors represent slower movements. Superimposed on the map are vector arrows, which provide information about the direction of glacier flow across different sections of the glacier.

The velocity shown in *Figure 2* is crucial to understanding glacier dynamics. As shown, certain areas of the glacier are moving faster than others, particularly along the centerline where the ice tends to flow more freely. Slower velocities are observed near the glacier's edges, where friction with the valley walls inhibits movement. The arrows indicate the direction of this flow, showing how the ice moves from higher elevations towards the terminus.

This map exemplifies how surface velocity measurements are used to assess glacier stability and behavior. Increased velocity in certain regions could indicate basal sliding due to increased meltwater at the base of the glacier, which can act as a lubricant and accelerate ice movement. Monitoring these velocity patterns over time is essential for predicting glacier surges, potential collapses, and the associated risks.

By providing both velocity magnitude and flow direction, this type of map offers a comprehensive view of glacier dynamics, contributing valuable insights into ongoing processes and helping to predict future behavior under changing climate conditions.

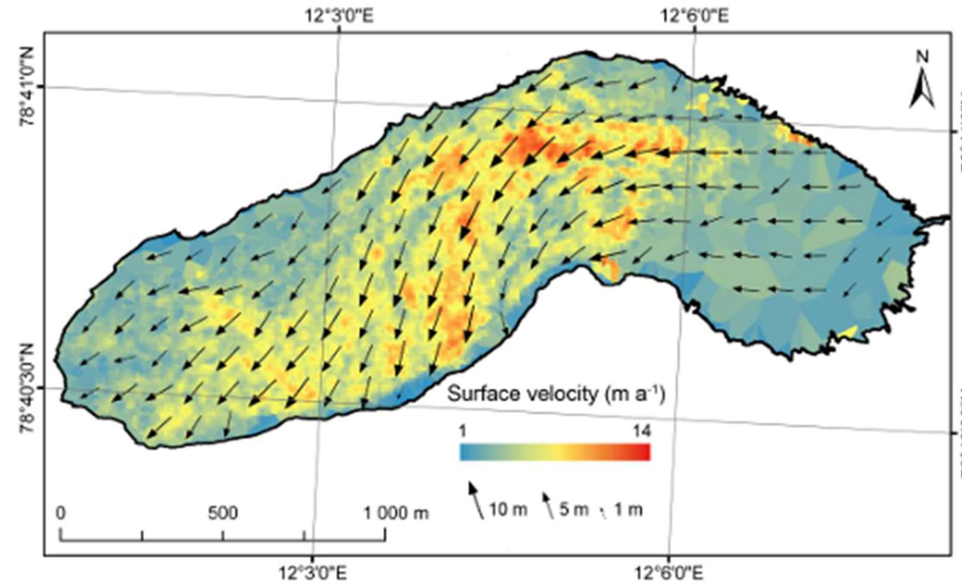


Figure 2. Surface velocity and displacement vectors of Waldemarbreen Glacier, Svalbard, from 2019 to 2021. [23]

2.3. Glacier Monitoring Techniques

There is a wide array of techniques which can be used for glacier monitoring, each with different spatial and temporal resolutions, logistical implications, and material requirements. The choice of techniques depends on the specific characteristics of the study site and the desired outcome.

Traditional glacier monitoring typically involves performing repeated in-situ measurements at designated stakes and snow pits scattered across the studied glacier to assess ice height variations and estimate an annual mass balance; these methods can be combined with geodetic GNSS to improve accuracy. However, in-situ techniques present a series of practical and logistical limitations. Amongst the practical limitations, accuracy is tightly linked with the number of measurements which must be interpolated. As for the logistical limitations, they can prove to be time-consuming, often necessitate mountaineering skills, and may imply higher risk to researchers due to the hazardous nature of glacial environments [26]. Furthermore, transporting heavy, and frequently expensive, specialized equipment to the glacier may prove to be a logistical challenge. Thus, glaciological studies based on in-situ measurements allow for glacier inventories to be updated with a frequency on the order of the decade [27].

The implementation and integration of photogrammetric techniques can help to overcome these limitations. Photogrammetry allows for 3D model reconstructions of the surface of the glacier from pairs (or groups) of overlapping images. These images can be acquired by terrestrial stations, aerial platforms carrying photogrammetric cameras, or satellites. They can be used to conduct geometrically accurate and recurrent studies on glacier evolution [27].

In the last two decades there has been rapid technological development and cost reductions in the field of Unmanned Aerial Vehicles (UAVs), informally known as drones [26]. UAVs made it possible to reduce the need for highly specialized and expensive equipment. Furthermore, they also significantly reduce economic, social, and environmental costs as compared to airplanes [28]. Therefore, even a single research team with simplified logistics and resource allocation can achieve high-spatial resolution as well as high-frequency studies of volume variations and displacement and surface velocity fields [27]. High-resolution UAV-derived DSMs have proven highly effective in tracking glacier surface changes. [23]

There are several examples of glacier monitoring using UAVs, amongst them, the Lirung Glacier, Nepal [29], Forni Glacier, Italy [30], Waldermarbreen Glacier, Svalbard [23], Gepatschferner Glacier, Austria [31], Alamkouh Glacier [32], Llaca Glacier, Peru [33], and the Belvedere Glacier, Italy [26], which is the area of study of this project.

2.4. The Belvedere Glacier

The Belvedere Glacier, located on the east face of the Monte Rosa, in the Anzasca Valley, Italy, constitutes the terminus of the glaciers descending the steep eastern slope of Monte Rosa in the Italian Alps [34]. It is a temperate alpine glacier, which extends from a maximum altitude of around 2250 meters above sea level down to 1800 meters a.s.l., where it splits into two distinct glacier tongues. It covers an area of approximately 1.8 square kilometers, and it is mainly elongated in the South-North direction, with a length of nearly 3 kilometers and a maximum width of 500 meters [26]. The Belvedere is a typical debris-covered glacier, similar to the Miage glacier (Monte Bianco, Valle d’Aosta). Furthermore, the glacier has a small accumulation basin, fed almost entirely by avalanches originating from the eastern slope of the Monte Rosa, which is more than 4500 m a.s.l. [34].

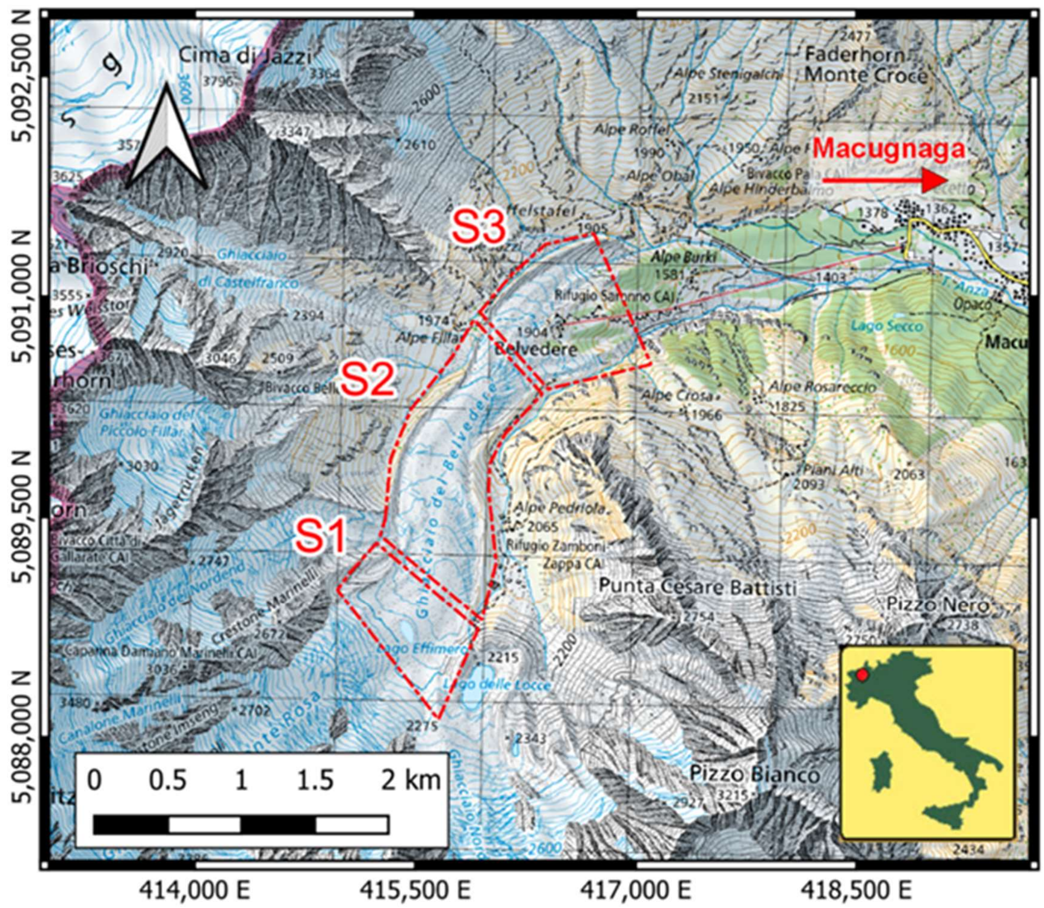


Figure 3. Position of the Belvedere Glacier with its three morphological sectors. S1 is the accumulation zone, S2 the transfer zone, and S3 is the low-relief zone where the glacier divides into two tongues. Coordinates are framed in ETRF2000(2008)-UTM 32N. [Basemap source Swisstopo (geo.admin.ch)] [26]

Morphologically, the Belvedere Glacier can be divided into three sectors:

1. Upper sector (labelled as S1 in *Figure 3*): It features a small accumulation zone situated at an elevation of 2,250 meters above sea level, at the base of Monte Rosa and the North Locce Glaciers. Frequent ice and snow movements from these glaciers supply the Belvedere. This upper section also serves as the primary deposition area for rocks and debris [26].
2. Central sector (labelled as S2 in *Figure 3*): This is the transfer zone, bordered by two sinuous moraines. It begins at an elevation of 2,250 meters above sea level and extends downward to 1,500 meters. This sector exhibits the highest ice flow velocities and the most irregular surfaces, characterized by numerous crevasses [26].

3. Lower sector (labelled as S3 in *Figure 3*): This is the low-relief sector, located near Belvedere Hill, where the glacier divides into two distinct tongues. The northwest tongue is the largest, descending to its lowest point at around 1,800 meters above sea level. From this tongue, the Anza River emerges. A smaller tongue extends eastward from Belvedere Hill, reaching an elevation of approximately 1,850 meters above sea level [26].

It is particularly relevant to note that until the first years of this century, the Belvedere was one of the few glaciers in the Alps that did not retreat, thanks in part to the cover provided by the rocks and debris. In fact, during the period 2000-2001, the Belvedere experienced an enormous surge, when a sudden increase in ice mass at the base of Monte Rosa flowed towards the valley in the form of a kinematic wave, as can be seen in *Figure 4* [35]. However, in recent years, the debris protection has not been enough to limit the glacier retreat and loss of volume. Between 2001 and 2009, the Belvedere lost an average of 5.97 million cubic meters. In the next decade from 2009 to 2019 the trend continued to be negative, but less severe, with an average loss of 2.72 million cubic meters per year [27].

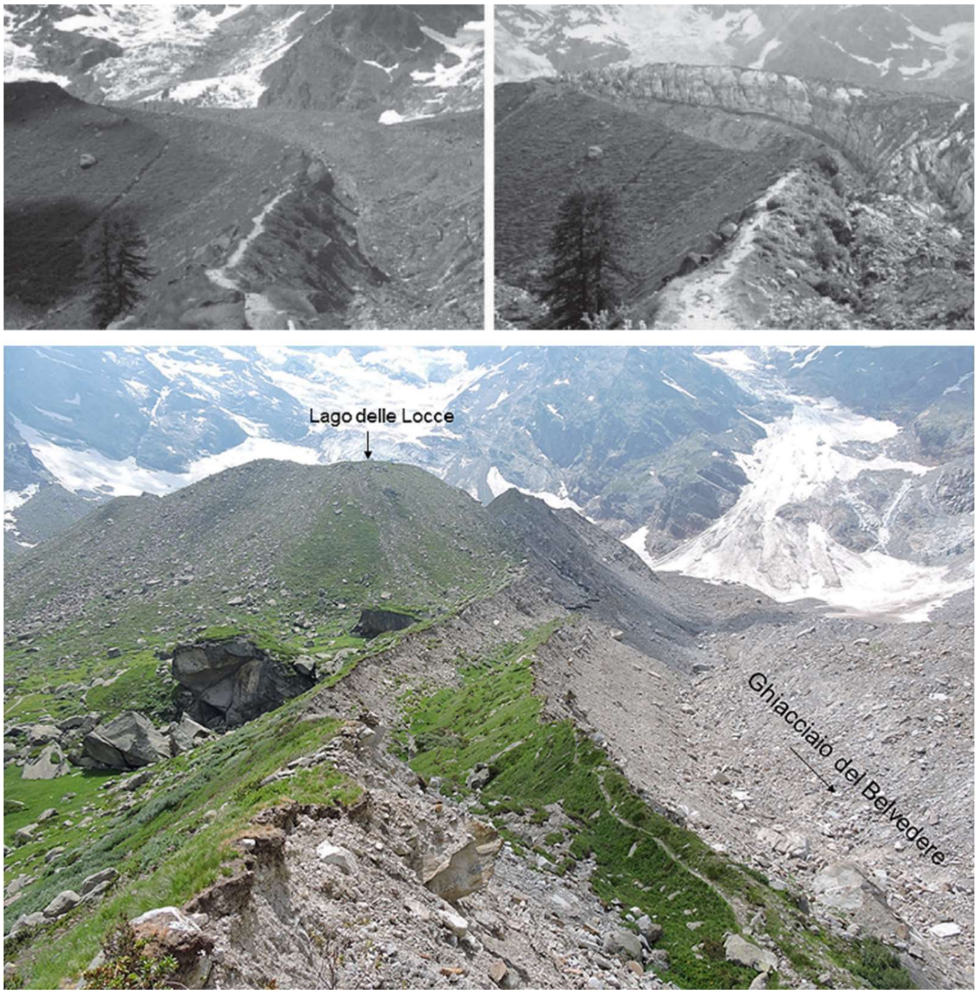


Figure 4. Orographic right lateral moraine and ice margin of the Belvedere Glacier as seen towards the upper direction (upper left summer 1996; upper right: end of June 2002; bottom: end of July 2019). The glacial surge can be appreciated, the ice wall to the middle of the right image is approximately twenty meters high [36] [37].

2.5. Monitoring the Belvedere Glacier

Due to its peculiarities and relative accessibility, the Belvedere Glacier has been studied extensively throughout the years. Photogrammetric data as early as 1977 is available for the glacier. However, since 2015, the Department of Civil and Environmental Engineering (DICA) from the Politecnico di Milano, in collaboration with the Department of Environment, Land and Infrastructure Engineering (DIIATI) from the Politecnico di Torino, has been carrying out extensive and continuous monitoring campaigns.

The project focuses on comprehensive and precise 4D monitoring of the Belvedere Glacier using photogrammetric techniques. It leverages varying spatial resolutions (ranging from centimeters to meters) and temporal resolutions (from daily intervals to 10-year periods) across different platforms, including UAVs, aerial photogrammetry, and terrestrial time-lapse cameras. Since 2015, extensive and continuous monitoring has been conducted through UAV-based photogrammetry and on-site GNSS measurements [26]. Each year, fixed-wing UAVs and quadcopters are deployed to remotely observe the glacier and create high-resolution photogrammetric models, enabling the estimation of annual changes in ice volume and flow velocities [38].

Additionally, the Geodetic and Photogrammetric Laboratory (LabMGF) of Politecnico di Milano succeeded in reconstructing the long-term evolution of the glacier, from 1977 up to date. To achieve this, historical images originally destined for regional mapping purposes were used. Historical analogue images were digitized and processed using a modern photogrammetric approach to obtain a 3D model of the glacier for 1977, 1991 and 2001 [13]. Using these models, Digital Surface Models and orthophotos were developed to estimate the variations in volume and mass balance, as well as surface displacements and velocities. In this thesis, DSMs are used to calculate elevation changes in glaciers. Furthermore, there is currently a high-frequency monitoring system on the northwest lobe of the glacier. Two cameras are installed using a time-lapse to capture daily photos and register the movements and changes in the glacier's structure [13].

2.6. Monitoring Activities during the "Belvedere Summer School"

Since 2015, the Department of Civil and Environmental Engineering (DICA) of the Politecnico di Milano has taken annual measurements of the glacier. To measure the glacier's movement and have reference points for photogrammetric processing, a network of ground control points was created. Ground Control Points (GCPs) are points on the Earth's surface with known coordinates, used to georeference spatial products and ensure their accuracy. This network is distributed evenly across the glacier; some points are inside the glacier, thus considered "mobile points" and subject to the glacier's dynamics; some others are considered "fixed" and located outside the glacier in zones with longer term stability along the moraines. In theory, these points are not subjected to the glacier's flow or rockslides, however, throughout the years some moraines have collapsed. Each monitoring campaign consists of two main parts. First, the data collection of ground control points is executed through a GNSS system. This step provides an accurate positioning of the points which serve a dual purpose. They can be used to measure the displacement, velocity, and acceleration of the glacier's surface when compared to previous years, as well as a georeferencing point and control point for the photogrammetric process. On the other hand, a photogrammetric survey is carried out by a drone flight, which collects images of the glacier, and is used to create a three-dimensional model of the terrain.

The ground control points consist of high-contrast plastic targets. They are fixed to large rock masses in adequately visible areas for the drone. Although durable, due to the severe alpine conditions, frequent rockslides, and other glacial events, the GCPs may suffer damage. Thus, during the GNSS positioning survey it is also necessary to determine whether certain GCPs need to be replaced. When a GCP becomes inaccessible or is deemed lost a new one is installed with a "bis" or "ter" added as a suffix.

Since 2016, fieldwork activities on the glaciers have been carried out also by Politecnico di Milano students, under the supervision of tutors. Following the daily fieldwork, students participate in preliminary exploration of the collected data (GNSS measurements, UAV images), including initial data processing using specialized software. At the end of the Belvedere Summer School, students conduct more detailed data processing at the Politecnico di Milano premises. This includes post-processing GNSS measurements with software packages such as Leica Infinity and RTKlib; populating the database with the new points and measurements; data processing for the 3D glacier reconstruction using Agisoft Metashape; calculating point displacements over time using QGIS by comparing measurements with previous years; documenting survey operations in monographies for archiving, amongst other activities [39]. During these final processing activities, there is a particular interest in facilitating, accelerating, and semi-automating the processing stage using QGIS.

3 Designing a QGIS plugin

This chapter outlines the foundational concepts and development process behind creating a QGIS plugin, focusing on its application in glacier monitoring. Section 3.1 begins with an exploration of the history of Geographic Information Systems (GIS), tracing its evolution from early spatial analysis methods to the development of modern, open-source GIS platforms like QGIS. This historical background establishes the relevance of GIS in environmental studies and the importance of spatial data for analyzing and managing natural resources. Section 3.2 introduces core GIS data types and structures, including vector and raster data, which are essential for organizing and analyzing geographic information. These data structures are crucial for glacier monitoring tasks, as raster data is used in the plugin for elevation change analysis, while vector data defines spatial extents and GCP layouts.

Section 3.3 discusses the basics of geospatial analysis, including methods like interpolation, which are integral to understanding glacier surface changes. Section 3.4 introduces QGIS as an open-source GIS platform, detailing its plugin architecture and the advantages of developing custom tools in this environment. Section 3.5 reviews the essential tools and file structure used in QGIS plugin development, including Python and PyQt, which form the backbone of the plugin's graphical interface and interactive features. Section 3.6 highlights useful tools, such as Qt Designer, Plugin Builder, and Plugin Reloader, which streamline the creation, testing, and interface design of the plugin. Section 3.7 explains the typical file structure for a QGIS plugin, including essential files for initialization, user interface, and core functionality, ensuring clarity, organization, and extensibility. Together, these sections provide a comprehensive overview of the GIS and programming concepts that support the development of the glacier monitoring plugin.

3.1. History of Geographic Information Systems

The term Geographic Information System (GIS) is applied to a system of hardware, software, and procedures that facilitate the management, manipulation, modelling, analysis, representation and display of georeferenced data to solve complex problems regarding planning and management of resources [40]. In other words, GIS allows to visualize and work with the relationship between the "what is happening" and the "where it is happening" [41]. Although examples of early GIS can be traced back to the 1850's when Dr Jon Snow mapped a cholera epidemic in London [42], it is not until the 1960's, when major technological advancements in computer technologies led to the birth of modern GIS. These breakthroughs were the capacity to output map graphics using line printers, advances in data storage and increased processing power of mainframe computers [41].

Roger Tomlinson, known as the 'father of GIS,' pioneered the development of the Canadian Geographic Information System (CGIS) in the 1960s while working for the Canadian Government, introducing the layered approach to map handling. Around the same time, the US Census Bureau began digitizing census boundaries, roads, and urban areas, while the Ordnance Survey in the UK developed digital mapping for large-scale maps by 1971 [42]. By the 1980s most advances in GIS technology had come from academia and the public sector; in fact, Tomlinson wrote in 1987 a speculation that future new innovations would come from academia and from the government, rather than the commercial sector. However, this was only partially true as in 1982 Esri released ARC/INFO, the first commercial GIS, which adapted the CGIS model of handling spatial and attribute data separately. By 1988 further advancements in computer technology spurred the commercial GIS market, with Esri emerging as a global leader by becoming a $40 million a year company by 1988 [42] [43].

Between 1990 and 2010, GIS adoption expanded rapidly due to IT advancements, including cheaper and faster computers, a growing number of GIS software options, and increased availability of digitized mapping data. This period also saw the integration of remote sensing technology and Earth observation satellites in GIS environments [42].

By 2010, open-source GIS, such as QGIS, had emerged, as GIS data became more ubiquitous, available online, in the cloud, and on mobile devices. Today, GIS and spatial analysis are deeply integrated into everyday life, powering location-based search, logistics, crime analysis, and environmental policy. Rooted in geography, modern GIS uses spatial data to reveal deeper insights into patterns and relationships, aiding smarter decision-making. With hundreds of thousands of organizations utilizing GIS worldwide, the market is projected to reach $9 billion by 2024 [41].

3.2. GIS Data Type and Structures

The main data of interest when working with GIS is geospatial data, which contains both spatial and thematic components. Conceptually, geographic data consists of two main components, the observation or entity, and the attribute or variable. A GIS must be able to manage both elements. The spatial component deals with two aspects in the entity's localization: the absolute position, which is based on a coordinates system, and the topological relationship, which refers to the spatial relationship of a single observation to other ones. On the other hand, thematic components refer to variables or attributes of the entity [40]. There are two main data structures to store spatial data, vector-based and raster-based.

Vector data is comprised of points, lines, and polygons. Lines are defined as the connection between two points, while polygons are an enclosed area by interconnected lines. The locations of the nodes and their topological structure are usually stored explicitly. In a vector-based GIS, geographic data is represented through defined points, lines, and boundaries, with curved lines stored as connected arcs. Although storing explicit topology increases overhead, only the points that define features are saved, and space outside these features is not recorded. These systems often use a hybrid structure, where spatial data is managed separately from thematic attributes. The spatial data is stored in a topological system, while the attributes are managed by a relational database. A unique identifier for each object connects both databases, ensuring smooth integration of spatial and thematic information [40].

On the other hand, raster is a method of storing, processing and displaying spatial data. The area is divided into rows and columns, which form a regular grid structure. Each cell within its matrix has location coordinates and an attribute value. The spatial location is implicitly contained in the ordering of the matrix. An example of raster data is a Digital Surface Model. Raster data is an abstraction of the real world. In the raster data model, spatial data is not continuous but divided into discrete units in each cell. Thus, raster data is particularly suitable for some types of spatial operations, overlays, or area calculations [40].

In this case, the plugin’s elevation change stage will use raster data to perform operations and vector data to define a bounding box. The layout section works entirely with point vector data. While in the interpolation stage, QGIS’s algorithm depends on an extent, which is calculated using the vector point layer.

GIS uses an entity-relation model to store data. This model consists of three elements: Entities, Attributes, and Relations. Entities are the objects of interest for the database, specifically for GIS, it is something that can be spatially localized. Attributes are attached to the entities and have a limited domain of values. Finally, relations allow entities to be related to each other, e.g. "intersects" or "contained in" [40].

The Database Management System can be either relational or object-oriented. Relational databases store data in tables, where rows represent an entity and columns an attribute. The database may consist of several tables, related to each other through a primary key (which is unique and univocal to the entity) and external keys, which link an entity with the primary key of an entity in another table [40] [44]. A visual representation can be seen in *Figure 5*.

Relational databases offer several advantages, including a strong theoretical foundation that supports evolution, straightforward implementation compared to other models, and flexibility in adding new tables. Many DBMS using this approach also support query languages like SQL, facilitating integration with GIS [40]. However, relational databases have limitations, such as scalability issues requiring distribution beyond certain data volumes, increased complexity when data does not fit into tables, and unused features that add cost. Additionally, SQL is complex when managing unstructured data, and partitioning databases across servers presents challenges in joining tables [44].

Most common databases in GIS are relational, however object-oriented databases are progressively incorporated [40]. Object-oriented databases are based on objects, which can be defined as an entity with a localization represented by values and by a set of operations. A visualization is presented in *Figure 6*. The advantage it conveys is that when defining the object, not only are its attributes defined but also the methods or operations that act on this object. Furthermore, objects belong to classes that can have their own variables, and these classes can belong to super-classes. Object-oriented programming is discussed further in Section 3.5, as it is the basis of Python programming.

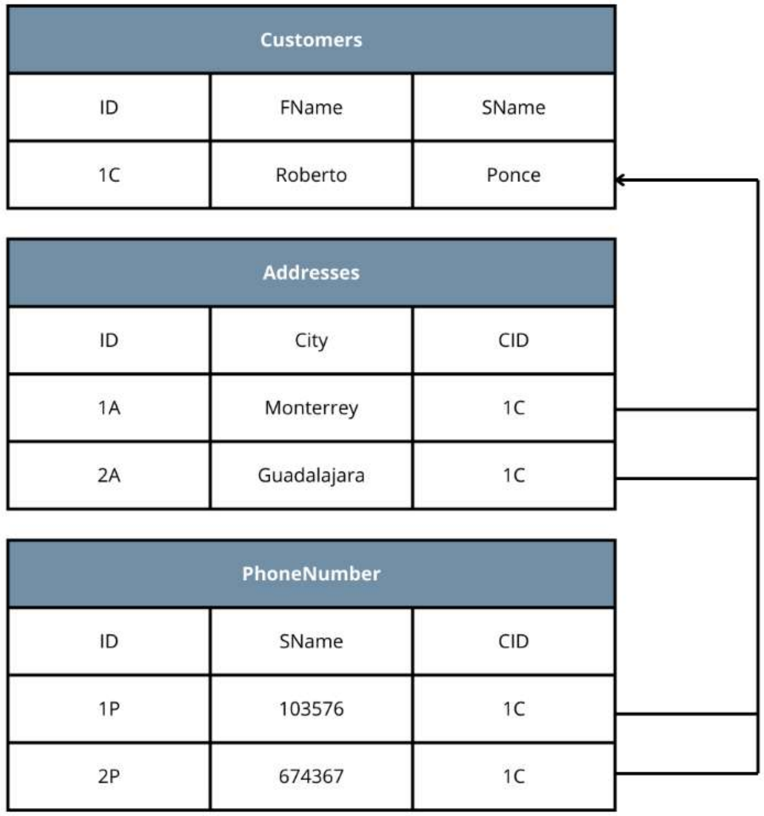


Figure 5. Example of a relational database. Adapted from [45].

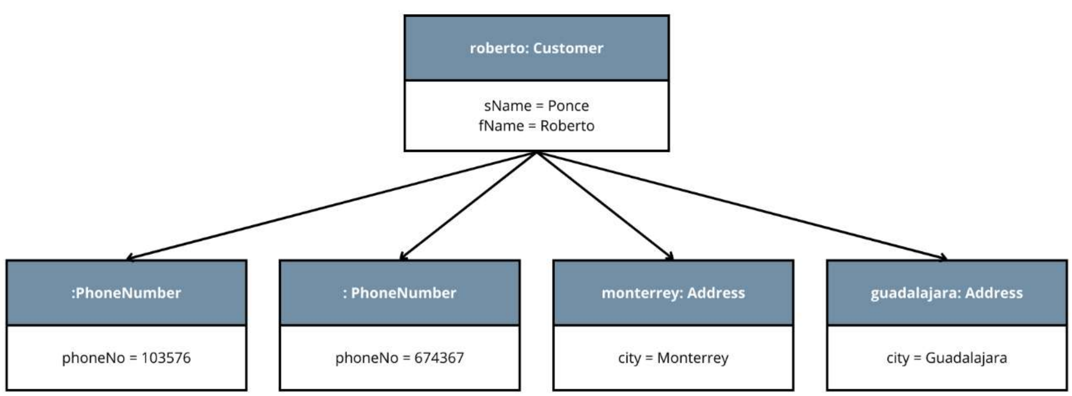


Figure 6. Example of an Object-Oriented database. Adapted from [45].

3.3. Basis of Geospatial Analysis

Geospatial analysis is the foundation of Geographic Information Systems (GIS), enabling the processing and visualization of spatial data for diverse applications, including glacier monitoring. This section introduces the key concepts and methods fundamental to understanding and performing geospatial analysis.

3.3.1. Coordinate Reference System (CRS)

A Coordinate Reference System (CRS) defines how spatial data points on the Earth's surface are represented in a two-dimensional or three-dimensional coordinate system. Each CRS is based on a specific geographic model (datum) that approximates the Earth's shape and provides a framework for mapping and spatial data analysis.

Using a consistent CRS is crucial in geospatial analysis to ensure that different data layers align properly. In the plugin developed for this thesis, raster and vector layers must have the same CRS to avoid misalignment and inaccuracies in calculations.

3.3.2. Bounding Box and Extent

- **Bounding Box**: A bounding box refers to the smallest rectangular area that can completely enclose a spatial dataset or feature. It is defined by minimum and maximum x and y coordinates and is often used to clip or restrict spatial operations to a specific region of interest. In this thesis, the bounding box is used to limit calculations such as interpolation and elevation change to defined areas within the glacier.
- **Extent**: The extent represents the full geographic area covered by a dataset. It is defined by the minimum and maximum coordinates in the x and y axes. When working with raster or vector layers, the extent is used to determine the spatial boundaries of the data and is important for visualization and processing. For example, when generating a raster from point data, the extent of the points determines the size of the output raster.

3.3.3. Interpolation Methods in GIS

Interpolation is a fundamental process in GIS used to estimate unknown values at unsampled locations based on known values from surrounding points. This technique is particularly useful for creating continuous surfaces, such as elevation models or temperature gradients, from discrete data points.

Inverse Distance Weighting (IDW)

The IDW interpolation method assigns weights to sample points so that their influence decreases with distance from the unknown point being estimated. The weights are determined using a weighting coefficient, which controls how quickly the influence of each sample point decreases as the distance from the unknown point increases. A higher weighting coefficient reduces the effect of distant points, causing the interpolated value to closely approximate the nearest sample point [46].

However, IDW interpolation has some limitations. The accuracy of the results can decline if the sample points are unevenly distributed. Additionally, the maximum and minimum values in the interpolated surface can only occur at the sample points themselves, often creating small peaks and depressions around those locations. In GIS, interpolation results are typically displayed as a two-dimensional raster layer [46].

The general formula for IDW is:

Equation 1.

$$Z(x_0) = \frac{\sum_{i=1}^{n} \frac{Z(x_i)}{d_i^p}}{\sum_{i=1}^{n} \frac{1}{d_i^p}}$$

Where:

$Z(x_0)$ is the interpolated value at point x_0.

$Z(x_i)$ is the known value at point x_i.

d_i is the distance between the known point x_i. and the interpolated point x_0

p is the power parameter that controls how quickly the influence of distant points diminishes.

Triangular Irregular Networks (TIN)

TIN interpolation is another widely used tool in GIS. A common algorithm for TIN is Delaunay triangulation, which aims to create a surface made up of triangles connecting the nearest neighbor points. This is done by generating circumferences around selected sample points and connecting their intersections to form a network of non-overlapping triangles that are as compact as possible [46].

The main drawback of TIN interpolation is that the resulting surfaces are not smooth and can appear jagged. This occurs due to the discontinuous slopes at the edges of the triangles and around the sample points. Additionally, TIN is not well-suited for extrapolating beyond the area covered by the sample data points [46].

Unlike IDW, which generates a smooth surface, TIN preserves the original points by using them as vertices of the triangles, making it better at capturing terrain features. However, its accuracy can vary based on the spatial distribution of the points [46].

3.4. Introduction to QGIS

QGIS, previously known as Quantum GIS, is an open-source GIS written in C++ available under the conditions of the GPL license. This means that the user may inspect and modify the source code. QGIS is based on the C++ cross-platform library Qt from Trolltech, which defines the Graphic User Interface (GUI) [47]. Gary Sherman founded the project in 2002, in pursuit of a fast geographic data viewer that would run on Linux and support a broad range of data sources. Today QGIS runs on most Unix platforms, Windows, and macOS. The project has made GIS software, which is traditionally expensive proprietary software, available to anyone with access to a personal computer [48]. This aspect is of particular importance for the project, as proprietary GIS software diminishes sharing, access, and collaboration.

Plugins can be written in Python, a popular language in the geospatial world [49]. A brief explanation of what Python is given in Section 3.5. Python was first introduced in QGIS 0.9, the current version is 3.34. The main advantage of Python plugins over C++ plugins is simplicity of distribution, as it requires no compiling for each platform, and easier development. There are several ways to use Python within the QGIS interface (explained in the following sections) [50]:

- Execute commands in the Python console within QGIS
- Develop and use plugins
- Automatically execute Python code when QGIS starts
- Build processing algorithms
- Create custom functions for expressions in QGIS
- Develop custom applications using the QGIS API

Since the introduction of Python support, numerous plugins with diverse functionalities have been developed. The plugin installer enables users to easily download, update, and remove Python plugins [50]. It is important to note that independent organizations and developers develop plugins, thus the QGIS organization takes no responsibility for them [51]. Furthermore, it is important to note there is a specific API documentation for the classes from the QGIS libraries, the Pythonic QGIS API (pyqgis) [52].

The basic steps for developing a QGIS plugin are [53]:

- **Idea**: Start with a clear concept of what the functionality of the plugin is.
- **Script vs. Plugin**: Consider whether the functionality of the plugin is better suited as a script or a plugin.
- **Setup**: Create the necessary plugin files. Some are mandatory depending on the plugin type, while others remain optional. (Further discussion in Section 3.7)
- **Develop**: Write the code in the appropriate files. (Further discussion in chapter 4)
- **Document:** Prepare the documentation for the plugin.

- **Translate**: (*optional*) Translate the documentation into different languages.
- **Test**: Reload the plugin to ensure everything works correctly.
- **Publish**: Share the plugin in the QGIS repository or create a private repository for internal use.

3.5. Python and the Qt library for plugin development

As stated in the previous section, QGIS plugins are based on Python, thus, a brief explanation on Python is included. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics [54]. Explanations on each characteristic are presented below [55]:

1. **Interpreted:** Python is an interpreted language, meaning that its code is executed line by line by an interpreter at runtime, rather than being compiled into machine code beforehand. This makes Python more flexible and easier to debug but can be slower than compiled languages.
2. **Object-Oriented:** Python follows the object-oriented programming (OOP) paradigm, which means it organizes code around objects, which can represent data (attributes) and behaviors (methods). This structure promotes modularity, code reusability, and makes it easier to manage complex systems.
3. **High-Level:** Python is considered a high-level language because it abstracts away many of the complex details of the computer's hardware, allowing developers to write programs in a way that is more human-readable and closer to natural language. This also means fewer lines of code are required to perform complex tasks.
4. **Dynamic Semantics:** Python uses dynamic typing, which means that the type of a variable is determined at runtime, not in advance. This allows for greater flexibility in coding, as variables can change types, but it also means errors related to type mismatches may only appear during execution.

Furthermore, as mentioned in chapter 3.4, QGIS is based on the Qt library. Qt is a cross-platform application development framework; it is not a programming language on its own. In other words, it is a collection of libraries for developing Graphic User interfaces (GUIs) and other applications [56]. The importance of Qt in QGIS plugin development cannot be overstated, as it provides the foundation for creating the interactive components of the plugins, such as windows, dialogs, buttons, tables, and other visual elements.

3.5.1. The role of PyQt in QGIS plugins

The GUI components of QGIS and its plugins are powered by the Qt library. When developing a QGIS plugin, developers take advantage of Qt's extensive toolboxes to design and manage user interfaces. Qt Designer (which is further explained in Section 3.6), is a visual tool that eases and speeds up the GUI creation process.

Typically, in QGIS, a plugin requires interaction and geographic data input by the user. This interaction may be done through custom forms, dialogs, and tools built using Qt. As an example, the plugin may provide a custom dialog box where the user can define a specific bounding box where the spatial analysis operation should take place, or a button to trigger map updates. These interface elements are all powered by the Qt framework, making it the backbone of QGIS's interactive capabilities and its usual user-friendly appearance.

To bridge Python and the Qt framework, QGIS uses PyQt, which is a set of Python bindings for the Qt application framework [57]. PyQt allows Python code to interact with the Qt libraries, thus, allowing developers to write Python code that seamlessly integrates with the Qt libraries, providing both flexibility and complex GUIs [50].

This integration allows developers to combine Python's simplicity and power with Qt's extensive GUI-building potential. For example, Python code can be used to handle back-end processes like data analysis, while PyQt manages the user interface, allowing for a clear separation between logic and presentation. This is particularly useful for QGIS plugins that require both data manipulation and an intuitive user interface.

Furthermore, Qt's cross-platform nature ensures that the plugin will be able to run consistently across Windows, macOS, and Linux. This compatibility is critical for GIS tools, as users may be working on various platforms. Qt ensures that the designed interface will look and function the same regardless of the underlying operating system [56].

3.5.2. Signal and Slot Mechanism within Qt

One of the key features of the Qt framework is the signal and slot mechanism, which plays a vital role in developing interactive applications. In GUI programming, typically, when a widget is interacted with, another one should be notified [58]. A simple example to understand Signals and Slots is, in the same way as one interacts with the lights in a room. When one moves the light switch (signal) a result is obtained which may be that the lightbulb turns on or off (slot) [59]. To achieve this Qt uses signals and slots.

In the context of Qt, signals are emitted by objects when a specific event occurs, for instance, when a button is clicked, the user changes a value in a text box or selects an item from a drop-down menu. These signals don't need to know which function or object is called; they simply announce that an event has occurred. This ensures that truly independent components can be created with Qt [59]. On the other hand, slots can be used for receiving signals. When a signal is emitted, it is "connected" to a slot, meaning that the designated function is triggered as a response to the signal. In the same way as objects don't know if anything receives its signal, a slot does not know if it has any signals connected to it, thus, ensuring truly independent components. One can connect as many signals as desired to a single slot, and a single signal can be connected to as many slots as needed [59]. A visual reference is presented in *Figure 7*.

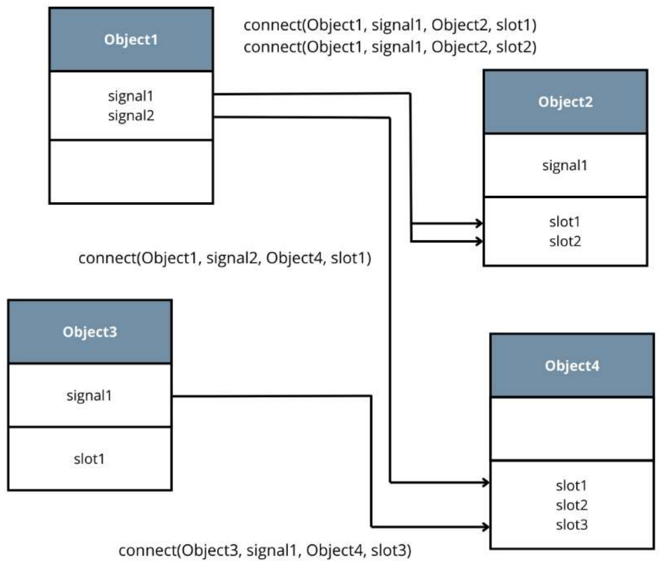


Figure 7. Visual diagram of the Signals and Slots mechanism. Adapted from [58].

3.6. Useful tools for QGIS plugin development

Given that QGIS is an open-source platform with a large, active community, it offers multiple tools that streamline the process of plugin development. These tools are essential for ensuring efficiency and enhancing the development workflow. For this project, the three main tools used for this project are **Qt Designer with QGIS custom widgets**, **Plugin Builder 3**, and **Plugin Reloader**. Each of these tools plays a crucial role in different stages of plugin development, from designing the interface to building the structure and testing the plugin.

3.6.1. QtDesigner with QGIS Custom Widgets

Qt Designer serves as a visual development tool within the Qt framework, specifically designed for the creation of user interfaces. This tool enables developers to construct graphical user interface (GUI) components—such as buttons, text fields, and combo boxes—through a visual interface, eliminating the necessity for manual coding of these elements. This capability greatly improves the efficiency of the design process and promotes uniformity across various plugins. A key benefit of Qt Designer is its intuitive drag-and-drop functionality, which allows developers to swiftly modify the layout and aesthetics of the interface. Furthermore, Qt Designer supports custom widgets, which is particularly beneficial for the development of QGIS plugins.

In the context of this project, the integration of QGIS custom widgets into Qt Designer proved to be essential. These widgets, including the Field Combo Box and the Map Layer Combo Box, are specifically designed to interact with QGIS data structures. By utilizing these preconfigured components, developers can effectively engage with geographic data within the plugin interface, thereby minimizing the need for manual coding of these functionalities. The integration of these custom widgets not only guarantees the operational effectiveness of the plugin within the QGIS framework but also ensures compatibility with QGIS's native tools and layers.

The design of the graphical user interface (GUI) in Qt Designer was pivotal to this project, facilitating the development of an accessible and user-friendly interface. An illustration of the Qt Designer interface, featuring QGIS custom widgets, is presented in *Figure 8*.

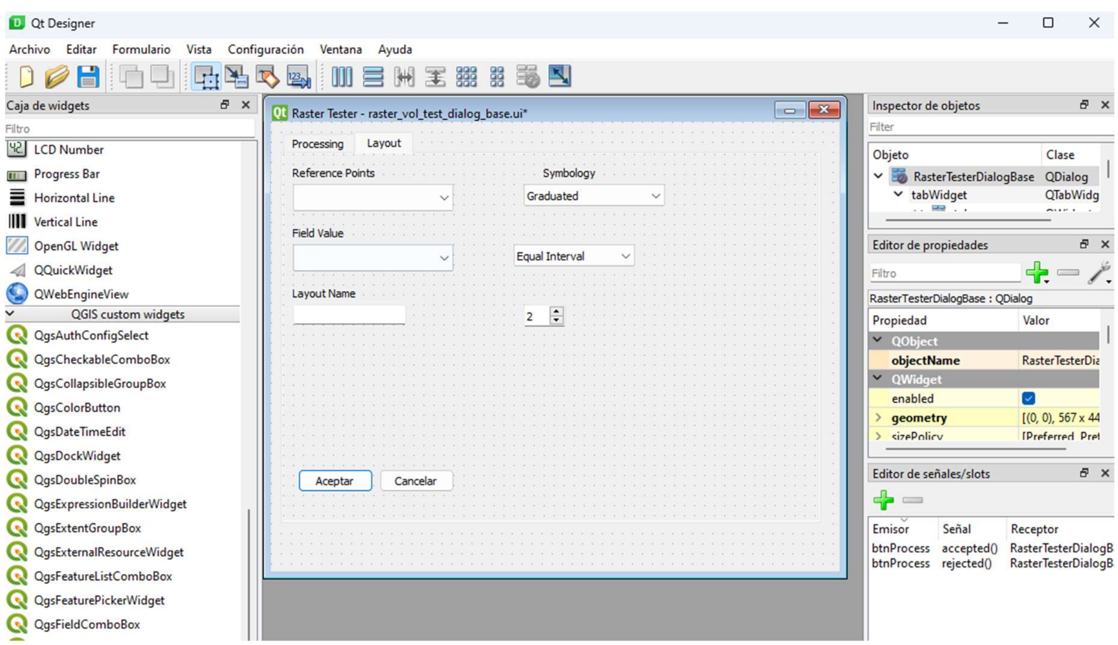


Figure 8. A screenshot of the QtDesigner with QGIS custom widgets interface.

Another powerful tool used during the development process is the Plugin Builder 3 plugin. This plugin is an essential tool for developers as it helps to generate the basic structure needed to create a QGIS plugin. Developed by GeoApt LLC and maintained by gsherman [60], Plugin Builder 3 has become a widely used tool in the QGIS development community. The latest stable version, 3.2.1, has been downloaded over 82,000 times, demonstrating its popularity and reliability. [61]

The primary function of Plugin Builder 3 is to automate the creation of a QGIS plugin template. This eliminates the need for developers to manually set up the required file structure, significantly reducing the initial development time. When starting a new plugin, the developer can use the Plugin Builder's GUI to input key information such as the plugin name, author details, version number, and a brief description. Based on this input, the tool generates all the necessary files, including the *init.py* file, which initializes the plugin, and other essential scripts for running the plugin within QGIS.

This prebuilt structure allows developers to focus on writing the core functionality of the plugin without having to worry about setting up the underlying infrastructure. Plugin Builder 3 simplifies the process of getting a plugin off the ground and provides a solid foundation for further development. The basic structure of the generated files is explained in detail in Section 3.7.

An example of the Plugin Builder 3 interface can be seen in *Figure 9* which illustrates the template generation process.

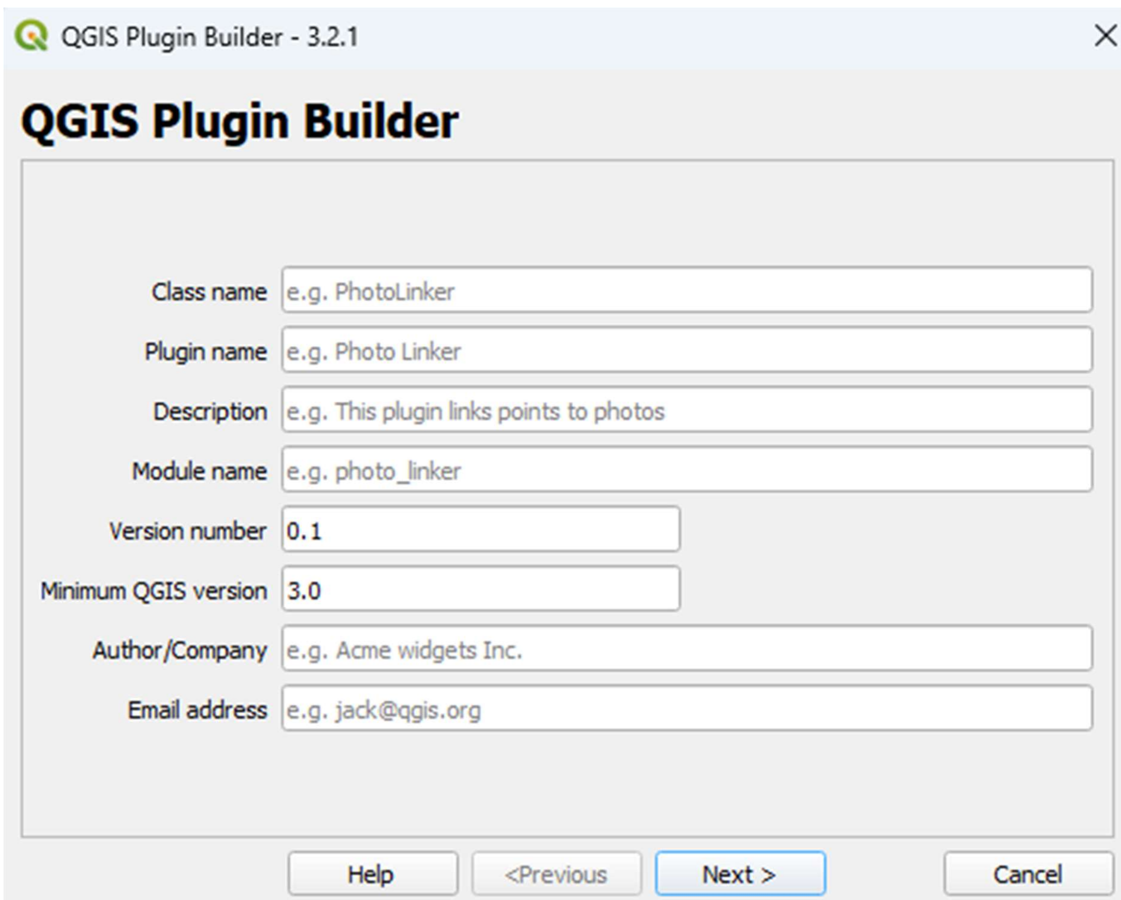


Figure 9. A screenshot of the Plugin Builder GUI template.

The final tool that has proved to be indispensable in plugin development is the Plugin Reloader. Created by Borys Jurgiel, Plugin Reloader is a simple yet highly effective tool that streamlines the testing process by allowing developers to reload a plugin with a single click [62] [63]. When developing QGIS plugins, developers frequently need to test changes and updates to their code. In a typical workflow, every time a change is made, the developer would have to manually reload the plugin by navigating to the "Manage and Install Plugins" window and reactivating the plugin. This process can become repetitive and time-consuming, especially during the debugging and testing phases when frequent changes are made.

Plugin Reloader solves this issue by allowing the user to instantly reload the selected plugin, directly from the QGIS interface. While this functionality may seem trivial, the time saved by eliminating repetitive manual reloading quickly accumulates, significantly improving the workflow. This tool is especially beneficial during the iterative testing phase of development, where quick reloads ensure that changes can be tested immediately and that any bugs or issues can be identified and resolved without delay.

The Plugin Reloader also helps maintain focus on coding and testing without being interrupted by unnecessary navigation through menus, enabling a more fluid and efficient development cycle. In *Figure 10* it is possible to see how simple it is for the developer to reload the desired plugin.

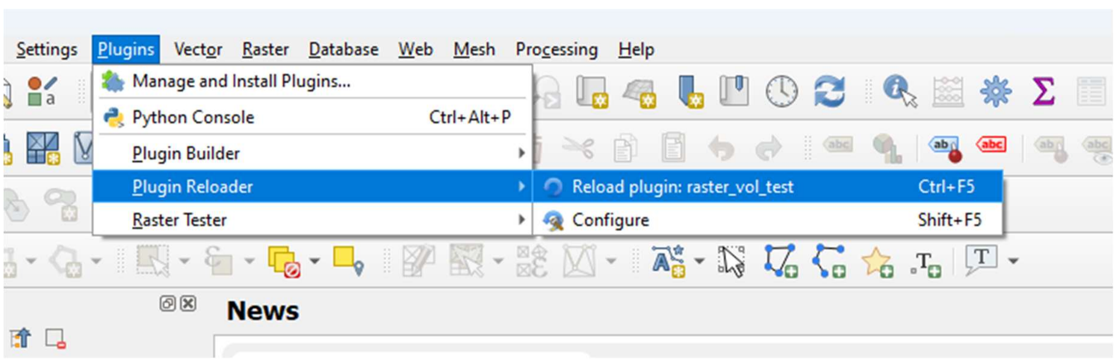


Figure 10. A screenshot of the Plugin Reloader from the QGIS interface.

3.7. Plugin file structure

Clear file organization is essential for a QGIS plugin to maintain clarity, functionality, and extensibility of the project. Each QGIS plugin typically follows a standardized directory structure, ensuring that all required components are contained and readable. The Plugin Builder facilitates this process. *Table 2* shows a breakdown of the core files usually found in a typical plugin directory, as well as their respective roles [53]:

metadata.txt – *required*

This crucial file contains the general characteristics of the plugin. The file includes the following information:

Table 2. Required and Optional information in the metadata file. Adapted from [53].

Metadata name	Type	Notes
Name	Required	A short string containing the name of the plugin.
QgisMinimumVersion	Required	Dotted notation of minimum QGIS version.
QgisMaximumVersion	Optional	Dotted notation of maximum QGIS version.
Description	Required	Short text which describes the plugin, no HTML allowed.
About	Required	Longer text which describes the plugin in detail, no HTML allowed.
Version	Required	Short string with the version dotted notation
Author	Required	Author's name
Email	Required	Email of the author, only shown on the website to logged-in users, but visible in the Plugin Manager.
Changelog	Optional	String, can be multiline, no HTML allowed.
Experimental	Optional	Boolean flag, *True* or *False*. *True* if this version is experimental.

Deprecated	Optional	Boolean flag, *True* or *False*. Applies to the whole plugin.
Tags	Optional	Comma separated list that provides keywords.
Homepage	Optional	A valid URL pointing to the homepage of the plugin.
Repository	Required	A valid URL for the source code repository.
Tracker	Optional	A valid URL for tickets and bug reports.
Icon	Optional	A file name or relative path of a web friendly image.
Category	Optional	One of *Raster*, *Vector*, *Database*, *Mesh*, or *Web*
Plugin_dependencies	Optional	PIP-like comma separated list of other plugins to install.
Server	Optional	Boolean flag, *True* or *False*. Determines if the plugin has a server interface.
HasProcessingProvider	Optional	Boolean flag, *True* or *False*. Determines if the plugin provides processing algorithms.

__init__.py – *required* [53]

This is the starting point of the plugin. It must contain the classFactory() method, which is responsible for creating an instance of the plugin's main class. The file may also include other initialization codes. In other words, this creates the plugin within the QGIS interface.

In Python, *__init__.py*, typically signifies that the directory is a Python package, allowing the plugin to be treated as a module that can be imported. In the case of QGIS, this file ensures that the plugin is correctly loaded into the interface.

mainPlugin.py – *core code* [53]

This file contains the main working code of the plugin. It defines its main functionality, such as what happens when a user interacts with the plugin through the interface. This could involve geoprocessing tasks, map rendering, or interacting with spatial data.

The mainPlugin.py typically includes:

- Methods for user actions
- Connections to the GUI
- Core functions

For larger or more complex plugins, the functionality can be modularized across multiple Python files to keep the code readable and understandable. These files can then be imported as modules into the mainPlugin.py file.

form.ui – *for plugins with custom GUI* [53]

Plugins that offer a custom graphic interface contain this file, which defines the layout and components of the user interface. The file is generated using QtDesigner. In this file developers can visually arrange widgets like buttons, checkboxes, labels, and text fields. This file is in XML format and describes how the interface should look and behave but does not contain any functional code.

form.py – *compiled GUI* [53]

This file is a Python file generated by compiling the form.ui file into code that can be understood and used by the plugin. This file is responsible for rendering the GUI inside QGIS and making it interactive and functional.

It is useful to divide the design of the interface and its underlying logic into two distinct files. This allows developers to update the interface in QtDesigner and recompile it without touching the core Python code.

resources.qrc – *optional* [53]

An XML file created by Qt Designer. It lists all the relative paths to resources used in the GUI, such as images, icons, and other assets. If the plugin contains custom icons for buttons or any other graphical elements, this file helps manage their inclusion.

resources.py – *compiled resources, optional* [53]

Like the form.ui and form.py relationship, resources.qrc is converted into a Python file which is readable by the plugin.

Additional files – *optional* [53]

Beyond these core files, the directory may include several additional files based on the specific needs of the project such as:

- Icons (*.png or *.svg): An icon file can be added to serve as the visual identifier for the plugin within the QGIS interface. The icon is referenced in the plugin's menu or toolbar and is typically defined in the metadata.txt file.
- Functions (*.py): For larger plugins, it is common practice to breakdown the functionality into smaller, more manageable parts. The functions.py file can be used to store helper functions or utility methods that can be imported into the mainPlugin.py file. This improves code readability, maintainability, and modularity.
- Test Files: Some plug-ins may include test files, which help in validating functionality and ensuring that updates don't introduce bugs. Python's unit test framework or similar tools can be used for testing.

4 Development of the GlacioTools plugin

This chapter details the technical aspects of the QGIS GlacioTools plugin development, focusing on the core functionalities and the coding process that conveys its implementation. The plugin is designed to streamline glacier monitoring processes and its consequent reporting. It automates several tasks, such as calculating elevation changes between DSMs, interpolating velocity values from points within the glacier, processing raster data within a bounding box, and generating standardized layouts for monographies of ground control points. The plugin design keeps modularity in mind, thus allowing it to handle multiple types of data processing through separate functional units. For user-friendliness, it was decided to keep all processing tasks within a single plugin, instead of developing four different ones. Thus, the user can select which type of data processing is desired by navigating through the different tabs of the plugin.

The development relied on both QGIS's extensive API and custom Python scripts. The development was done using Visual Studio Code and QGIS Desktop 3.28.2. Version changes were kept through GitHub.

4.1. The Graphic User Interface

The Graphic User Interface (GUI) is the point of interaction between the user and the program. Thus, the GUI plays a crucial role in enhancing user interaction by providing an intuitive and accessible way to perform complex geospatial analyses. By streamlining workflows and minimizing manual coding, the GUI allows users of varying technical backgrounds to efficiently utilize the plugin's features. Keeping this in mind, it was deemed necessary to keep all functions within a single plugin and design it in a clear and easy way to navigate it. To achieve this the QTabWidget [64] played an important role in keeping each function compartmentalized in its specific tab, which can be easily navigated by the users and work independently from each other. The GUI for each tab is reported in *Figure 11*:

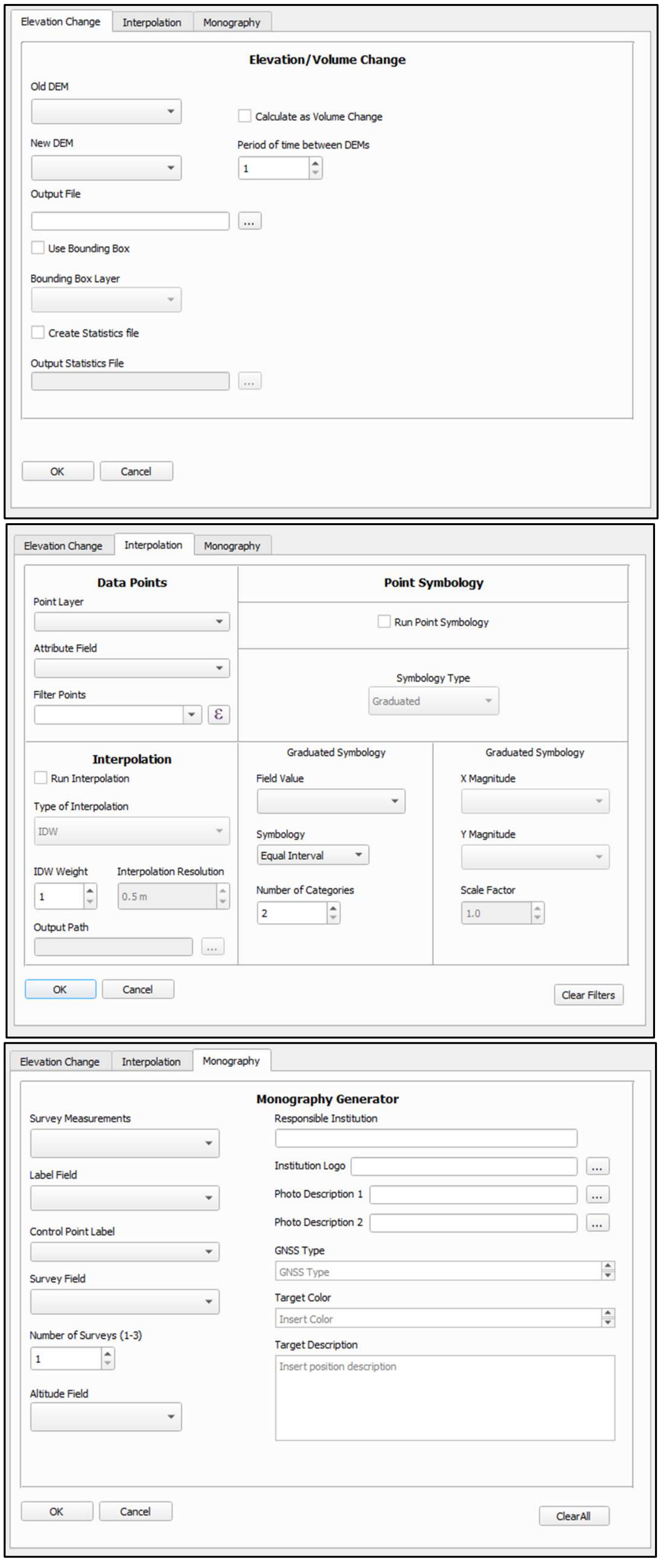


Figure 11. Screenshots of the GUI in the Elevation Change tab (upper), Interpolation and Symbology tab (middle), Monography tab (bottom)

PyQt offers a wide array of objects that facilitate user interaction. Furthermore, there are specific widgets for interacting with QGIS that have specialized methods that make working with user input easier for the developer. These widgets serve different purposes, from organizing the interface layout to capturing user inputs and displaying information. A brief explanation of those used is detailed in *Table 3*.

Table 3. PyQt Widgets used in the interface of the plugin.

Widget Type	Widget Name	Description
QTabWidget	Tabbed Layout	Provides a tabbed interface for organizing different sections of the GUI, such as Elevation Change, Layout, Monograph, and Interpolation tabs [64].
QToolButton	Tool Button	A customizable button is often used for file selection or clearing inputs, which can include icons and dropdowns for flexible interaction [65].
QDialogButtonBox	Dialog Button Box	Groups standard dialog buttons (e.g., OK, Cancel, Apply) and handles their signals in a unified manner, ensuring consistent interaction with dialogs [66].
QCheckBox	Checkbox	Allow users to toggle options on or off (e.g., enabling bounding box clipping or saving statistics) and dynamically enable or disable related options [67].
QLabel	Label	Displays static text or images, guiding the user by labeling input fields or explaining the purpose of different sections of the GUI [68].
QLineEdit	Text Input Field	Provides a single-line text input [69].
QComboBox	Dropdown Menu	Displays a list of predefined options, commonly used to select map layers,

		fields, or other parameters dynamically [70].
QPlainTextEdit	Multi-Line Text Editor	Used for entering or displaying longer text inputs, such as descriptions for the monograph section of the plugin [71].
QgsMapLayerComboBox	Layer Selector	Provides a dropdown menu to select available map layers in the current QGIS project, ensuring valid layers are chosen for tasks like elevation change [72].
QgsFieldComboBox	Field Selector	Displays a list of fields (attributes) from the selected layer, allowing users to select specific fields for operations such as symbology or interpolation [73].
QgsFeaturePickerWidget	Feature Selector	Allows users to select individual features (e.g., points or polygons) from a vector layer, particularly useful for the Monography Tab [74].
QgsSpinBox	Integer Input Field	Allows input of integer values for parameters like the number of classification classes or surveys in the monography [75].
QgsDoubleSpinBox	Decimal Input Field	Like QgsSpinBox but allows precise input of floating-point values for settings like interpolation resolution or vector field scaling [76].

4.2. The Elevation Change Tab

The main purpose of this plugin feature is to calculate elevation changes between two DSMs (Digital Surface Models), which is a key metric in glacier monitoring. By analyzing changes in surface elevation over time, researchers can observe glacier dynamics, track mass loss or gain, and identify patterns of retreat or advance. It is important to note that this feature does not provide the glacier's total mass; rather, it estimates elevation changes between two time periods by performing a raster calculation. Additionally, users have the option to select a checkbox to display the result as a volume change instead of an elevation change. When this option is chosen, the plugin multiplies the elevation change by each pixel's surface area to estimate volume change. The function responsible for this calculation, *elevation_change()*, is in the *functions.py* file and operates in several key steps outlined below. Below is the pseudocode for the function and a detailed explanation of the various components included in the elevation difference calculation.

```
Start Function
    # Log the start of the process
    Log: "Processing task started"
    # Step 1: Validate the output file path
    Get output filename from User
    If output filename is empty:
        Show error message: "Missing Output Save Name"
        Exit function
    # Step 2: Load the old and new raster layers
    Old raster = Load raster layer from User
    If the old raster is not loaded:
        Exit function
    New raster = Load raster layer from User
    If the new raster is not loaded:
        Exit function
    # Step 3: Check if the CRS of both raster files matches
    If the CRS of the new raster is different from the CRS of the
old raster:
        Ask user: "The Layers are in different CRS. Do you want to
continue?"
        If the user chooses "No":
            Exit function
    # Step 4: Clip the raster files if the bounding box option is
enabled
    If Bounding Box Checkbox is checked:
        Get the bounding box from User
```

```
        If the bounding box is loaded:
            Clip the new raster using the bounding box as a mask
            Clip the old raster using the bounding box as a mask
            Log: "Raster files clipped using bounding box"
    # Step 5: Set up the raster calculator entries
    Create old raster entry
    Create new raster entry
    # Step 6: Perform the raster calculation (New Raster - Old Raster)
    If Volume Change is Checked:
        Set formula string = "(new Raster – old Raster) * pixel area"
    Else:
        Set formula string = "new Raster – old Raster"
    Set raster calculation parameters with old raster and new raster
entries
    Run raster calculation
    # Step 7: Check if the raster calculation is successful
    If raster calculation is successful:
        Log: "Raster calculation finished successfully"
        Add the resulting raster to the QGIS project
    # Step 8: Save statistics if the option is enabled
        If Stats Checkbox is checked:
            Log: "Saving statistics is checked"
            Get the difference raster layer by name
            If difference raster is found:
                Calculate and save statistics to Stats Output Path
    Else:
        Show error message: "Raster calculation failed"
        Log: "Raster calculation failed"
End Function
```

Step 1: Input Raster Loading

The process starts by allowing the user to select two raster layers that are present in the current QGIS project. These raster files should represent DSMs referring to different epochs. This is done through the implemented function *get_raster_layer()*. The raster layers are read from the project and loaded as a **QgsRasterLayer** object [77]. Furthermore, as an error handler, the function verifies the selected layers' validity before proceeding with further steps. If a raster is missing or invalid, a critical error message is displayed to the user using **QMessageBox** [78], informing the user that the process cannot continue until both raster layers are properly selected.

Step 2: Coordinate Reference System Check

Once the raster layers are properly loaded, the plugin verifies that both raster layers have the same CRS. Although the main processing method used ahead can convert the layers' CRS on the fly, it was found that the method natively used is inaccurate. Thus, it can lead to misaligned data and inaccurate calculations. The plugin uses the *crs()* method from the **QGSRasterLayer** [77] class to retrieve the appropriate information from each layer.

If the CRS check fails, the plugin prompts the user with a message box, providing them with the option to either proceed with the mismatch (potentially leading to errors) or cancel the process to reproject the layers manually. This interaction ensures that the users are aware of potential issues in the data handling.

Step 3: Clipping Raster Layers (Optional)

The plugin allows the user to restrict the elevation change analysis to a specific region by clipping the raster layers using a bounding box. This option is prompted to the user in the GUI. The bounding box must be in a vector format in the same CRS. The clipping is performed by the implemented *clip_raster()* function, which utilizes the **gdal** tool available through the QGIS Processing Framework [79].

Thanks to the **QGSProcessing** class in PyQGIS, the plugin can efficiently crop the input raster layers to the spatial extent of the selected bounding box. This feature allows the user to focus on a single section of the glacier and reduces the calculation time used by the **QgsRasterCalculator** class.

Step 4: Raster Calculation

This step fulfills the core functionality of the tab, where the plugin computes the difference between the older DSM and the newer DSM. This is done utilizing the **QgsRasterCalculator** class [80]. The elevation change is calculated by subtracting the old DSM from the new one, pixel by pixel.

This process results in a difference raster that represents the change in elevation. In case the user chooses so, the result will be given as a volume change. This is done by multiplying the difference between DSMs by the area of the pixel. The result is saved as a GeoTIFF file, which can be easily loaded and visualized in QGIS. The **QgsRasterCalculator** class handles heavy lifting in terms of geospatial calculations, ensuring consistency and efficiency even for large raster datasets. However, as mentioned above, it is important to have both raster layers in the same CRS to avoid potential errors.

Step 5: Output Generation

When the raster calculation is complete, the result is automatically added to the QGIS project [81]. This allows the user to visualize the output difference raster directly within the QGIS interface. Additionally, if the user has opted to save statistical summaries, the plugin computes various statistics such as mean, standard deviation, minimum, and maximum values using the **bandStatistics()** method provided by the **QgsRasterBandStats** class [82].

These statistics are saved as a CSV file, which can be used for further analysis or reporting. The plugin ensures that all necessary steps for raster processing and output generation are handled within a single workflow, making it user-friendly and efficient for glacier monitoring.

4.3. The Monography Tab

This section is dedicated to the generation of monographies in the layout using Ground Control Points (GCPs). Monographies are standardized, detailed reports designed to provide essential information about GCPs, which play a critical role in the accuracy of geospatial data and analysis. In the specific case of glacier monitoring, accurate GCP data is crucial for anchoring spatial measurements and ensuring that datasets remain consistent over time. Furthermore, in the case of the Belvedere Glacier, there are points within the glacier that move following the glacier's dynamics, thus finding differences in positions and keeping track on where they are for future data collection is relevant.

The plugin simplifies the often-times consuming process of creating these monographies by automating the process through the implemented function *create_monograph()*. This function not only ensures consistency across reports, but also significantly reduces the time required to generate them.

The implemented function *create_monograph()* generates a comprehensive layout that includes multiple elements, allowing the user to focus on data collection and analysis rather than spending valuable time on manual report generation. The format was taken from the Belvedere Glacier monographies from the 2023 campaign. Below is the pseudocode for the function and a detailed explanation of the various components included in the monography generation process.

```
Start Function
    # Log the start of the process
    Log: "Monography task started"
```

```
    # Step 1: Validate Input Data
    layer = Get current layer from User
    If no layer is selected:
        Show error message: "No Point Layer selected"
        Exit function

    feature = Get selected feature from User
    If no feature is selected:
        Show error message: "No Feature selected"
        Exit function

    # Step 2: Get Survey Data and Sort by Date
    Label field = Get selected label field from User
    Survey field = Get selected survey date field from User
    Number of surveys = Get number of surveys from User
    Height field = Get selected height field from User

    surveys = Get matching features from layer for the selected feature
based on label field
    Sorted surveys = Sort surveys by date from survey field

    For each survey in sorted surveys:
        Get geometry of the survey
        Get East and North from Geometry
        Transform coordinates to EPSG:4326 (WGS84) for lat/long
        Get altitude field from chosen attribute

    # Step 3: Get User Input for Detailed Description
    Target color = Get text from User
    Target description = Get text from User
    Gnss mode = Get text from User
    institution = Get text from User
    logo path = Get logo file path
```

```
    photo_1_path = Get photo 1 file path
    photo_2_path = Get photo 2 file path

    # Step 4: Create the Monograph Layout
    Layout name = Construct layout name using feature's label and survey date
    layout = Create layout with layout name

    # Add title and institution information
    Add title to layout with feature's label
    Add institution name and logo to layout

    # Add target color, description, and GNSS mode
    Add target color and description text to layout
    Add GNSS mode text to layout

    # Step 5: Add Survey Data to Layout as Table
    Create a table layout for the survey data
    Add columns: Latitude, Longitude, Height, Easting, Northing

    For each survey in the limited number of surveys (up to number of surveys):
        Add corresponding values to the table

    # Step 6: Add Map with Monitoring Point
    Add a map to the layout
    Set map symbology to highlight the selected feature (e.g., red dot for the point)
    Adjust the map extent to fit the point with some buffer
    Add scale bar and north arrow to the map

    # Step 7: Add Photos to Layout
    If photo_1_path exists:
        Add photo 1 to the layout
```

```
    If photo_2_path exists:
        Add photo 2 to the layout

    # Step 8: Finalize and Open Layout for User Review
    Add the layout to the QGIS project
    Open the layout designer for further customization if needed
    Log message: "Monograph layout created successfully"

End Function
```

Step 1: Selecting the Monitoring Point

The process begins by allowing the user to select a vector layer containing monitoring points using the QgsMapLayerComboBox. Each point represents a feature that corresponds to a specific GCP or other key locations on the glacier. The user then selects the specific feature (point) from the layer using the QgsFeaturePickerWidget. This feature will be the focus of the monograph, and the associated data (such as coordinates, elevation, and label) will be retrieved from the vector layer.

The **QgsFieldComboBox** is used to select the field that contains the labels or names of the monitoring points, and another field for the survey dates. This helps in identifying and organizing the data for each point.

Step 2: Retrieving Survey Data

The next step involves retrieving relevant data from the selected feature and its associated surveys. The user specifies the number of surveys to be included in the monograph using the QgsSpinBox. Each survey is a record of the feature's coordinates, elevation, and other attributes over time. The plugin sorts the surveys by date, ensuring that the most recent data is displayed first.

This step is crucial for monitoring the changes in the glacier's surface over time, as it allows researchers to track the movement and evolution of specific points on the glacier. The plugin retrieves the coordinates (latitude, longitude, easting, northing) and ellipsoidal height from the vector layer using PyQGIS's **QgsGeometry** and **QgsCoordinateTransform** classes to ensure accurate geographic information.

Step 3: Customizing the Monograph Layout

After retrieving the relevant data, the user is prompted to add relevant information to the layout of the monograph. The following information is asked for:

- **Target Color and Description**: The user can input a description of the monitoring point, its color, and most crucially an explanation of its location in the field so that it can be found in following campaigns.
- **GNSS Information**: The user can input details about the GNSS mode used to record the point's position, which is crucial for understanding the accuracy and reliability of the survey data.
- **Institution Logo and Photos**: The user can upload an institutional logo and up to two photographs related to the monitoring point. These images are added to the monograph layout to provide a visual reference of the monitoring point's surroundings or conditions at the time of the survey.

Once the customization fields are filled, the plugin compiles all the input data and prepares it for the layout.

Step 4: Creating the Monograph Layout

The plugin generates a new monograph layout using the *create_layout()* function. The layout is structured to display all relevant data, including:

1. **Title and Header Information**: The name of the monitoring point, the institution name, and the logo are displayed at the top of the monograph.
2. **Target Description**: The target color, description, and GNSS, and CRS information are displayed next, providing additional context about the point.
3. **Survey Data Table**: The survey data, including geographic coordinates (latitude, longitude, easting, northing) and ellipsoidal height, is presented in a table format. The table is organized chronologically, with the most recent survey listed first.
4. **Map Representation**: A map of the monitoring point is included in the monograph. The map shows the selected feature with labels and customized symbology (e.g., red dots) for easy identification. The extent of the map is adjusted to ensure that the feature is clearly visible, and a scale bar and north arrow are added for reference.
5. **Photographs**: If provided, the photos are displayed next to the map, giving a visual snapshot of the monitoring point.

The layout is generated using the PyQGIS **QgsPrintLayout** class, which ensures that the resulting monograph is of high quality and ready for printing or exporting as a PDF.

Step 5: Exporting the Monograph

Once the layout is generated, it is automatically added to the QGIS project. The user can open the layout in the QGIS layout designer for further customization, if needed, or export the monograph as a PDF or image file. This allows for easy sharing and inclusion in reports, ensuring that the data collected during glacier monitoring campaigns is documented clearly and consistently.

The monograph serves as an essential tool for tracking changes at specific glacier monitoring points over time, making it easier to compare data across different survey periods and identify trends in glacier behavior.

Step 6: Managing Multiple Monographs

The plugin also allows the creation of multiple monographs within the same QGIS project. Each monograph is saved under a unique layout name based on the feature's label and survey date, ensuring that researchers can generate and save detailed reports for multiple monitoring points in an organized manner.

The Monography tab streamlines the process of documenting glacier monitoring points, providing a structured and visually rich format for presenting geographic data. By combining survey data, geographic coordinates, images, and maps, the plugin ensures that researchers can generate comprehensive reports to support glacier monitoring and analysis.

4.4. The Interpolation and Symbology Tab

The Interpolation and Symbology tab is divided into two main processes. The core process is the interpolation of an attribute field within a point vector layer. Applying symbology is an optional functionality provided. Using this tab, the user can create interpolated raster surfaces from point data, which is essential in glacier monitoring for estimating values such as displacement velocity, temperature, or snow depth over an area based on limited point measurements. This function uses two common interpolation methods—Inverse Distance Weighting (IDW) and Triangular Irregular Networks (TIN)—allowing the user to generate a continuous surface from discrete data points. The interpolated raster can help researchers visualize spatial variations across the glacier and fill in gaps where direct measurements were not taken.

Furthermore, the user may choose to complement the visualization of the interpolation by applying symbology to the points which were used to interpolate the field. This helps to generate a more robust and understandable map which portrays the behavior of the glacier. The user can choose between two types of symbology for the points:

- Graduated Symbology allows the user to classify numeric values (such as velocity or displacement) into different ranges and apply a color ramp to visualize these classifications.

- Vector Field Symbology visualizes movement, such as glacier displacement, by drawing lines indicating magnitude and direction. A limitation of the current plugin is that adjustments to certain vector symbology properties (such as symbol type) may require manual fine-tuning within the QGIS environment. Future versions of the plugin aim to automate this process and provide additional symbology customization options through the GUI.

The primary implemented function responsible for interpolation in the plugin is *interpolator*(). This function reads point data from a selected vector layer and interpolates values based on user-defined parameters such as resolution and interpolation method. This chapter explains how the interpolation is performed and outlines future improvements to include options for masking the raster output and merging additional point layers. Below is the pseudocode for the function and a detailed explanation of the various components included in the interpolation and symbology process.

```
Start Function
    # Step 1: Validate Input Data
    Output path = Get output file path from User
    layer = Get current layer from User
    If no layer is selected:
        Show error message: "No Point Layer selected"
        Exit function
    field = Get selected field from User
    If no field is selected:
        Show error message: "No Field selected"
        Exit function
    Interpolation type = Get selected interpolation type from User
    resolution = Get resolution value from User
    weight = Get weight value from User (for IDW only)

    # Step 2: Setup for Interpolation
    Get bounding box of the layer (extent)
    Calculate columns and rows based on extent and resolution

    # Step 3: Prepare Layer Data for Interpolation
```

```
    Layer data = Create QgsInterpolator.LayerData object
    Set source to layer data
    Set interpolation field to layer data
    Set source type to "Point"

    # Step 4: Perform Interpolation Based on Selected Method
    If the interpolation type is "IDW":
        Idw interpolator = Create QgsIDWInterpolator object with layer
data
        Set distance coefficient (weight) for IDW interpolator
        output = Create IDW Interpolation with QgsGridFileWriter
        Write interpolated raster to file
    Else If the interpolation type is "TIN":
        Tin interpolator = Create QgsTinInterpolator object with layer
data
        output = Create TIN Interpolation with QgsGridFileWriter
        Write interpolated raster to file

    # Step 5: Add Interpolated Raster to QGIS Project
    Add the resulting raster layer to the QGIS project with the name
based on interpolation method and field
    Log message: "Interpolation process completed successfully"
    # Step 6: Symbology of Points (Optional Symbology)
    If Symbology Checkbox is checked:
      Get Symbology Type from User
      If Symbology Type is 'Graduated'
        Get Graduation Method from User
          Options: (Equal Interval, Jenk, or Quantile)
        Get Number of Categories from User
        Apply Graduated Symbology
      Else if Symbology Type is 'Vector Field Marker':
        Get X Magnitude from User
        Get Y Magnitude from User
        Get Scale Factor from User
```

```
        Apply Vector Field Markers
    Apply Repaint
End Function
```

Step 1: Selecting Input Data and Interpolation Method

The interpolation process begins by allowing the user to select a point layer from the current QGIS project. This layer contains the discrete data points from which the interpolation will be performed. The user then selects the field that holds the attribute values to interpolate (e.g., displacement, velocity, acceleration, temperature, or other relevant measurements).

Next, the user chooses the interpolation method in the GUI:

1. IDW (Inverse Distance Weighting): This method assumes that the values at unsampled locations are more influenced by nearby points than by those farther away. The user can set the distance coefficient (weight) to control how the influence of nearby points decreases with distance.

2. TIN (Triangular Irregular Networks): This method creates a surface by connecting points to form triangles and interpolating values within each triangle. The TIN method is especially useful when the data points are irregularly spaced.

The plugin uses the PyQGIS **QgsIDWInterpolator** and **QgsTinInterpolator** classes for these respective methods, ensuring high performance and accuracy [83].

Step 2: Defining Interpolation Parameters

Once the interpolation method is chosen, the user must specify additional parameters:

- Resolution: The grid resolution for the output raster is chosen by the user using a spin box. A higher resolution results in a more detailed raster but requires more processing time.

- Weight (for IDW): For IDW interpolation, the user sets the distance coefficient using a spin box. This weight determines how quickly the influence of a point decreases with distance.

These parameters give the user control over the level of detail and precision in the interpolated raster, allowing them to tailor the output to the specific needs of the glacier study.

Step 3: Interpolation Calculation

Once all parameters are set, the user can run the plugin to start the interpolation process. The *interpolator()* function handles this by first retrieving the input data and then performing the interpolation using the selected method. The plugin uses the PyQGIS **QgsGridFileWriter** class to write the interpolated raster to a file.

For IDW, the function calculates each raster cell's value by averaging the values of nearby points, with closer points having more influence. For TIN, the plugin constructs a surface of triangles from the points, and the values are interpolated within each triangle.

The interpolation results in a continuous raster surface that fills in the gaps between the known point data, providing a comprehensive visual representation of the glacier's characteristics over a given area.

Step 5: Output Generation and Visualization

Once the interpolation is complete, the resulting raster file is automatically added to the QGIS project. The interpolated surface can then be visualized using standard QGIS symbology and analysis tools. The raster provides a smooth, continuous surface that reveals patterns in elevation or other glacier-related measurements that would be difficult to see with discrete point data alone.

Researchers can also export the resulting raster as a GeoTIFF file for further analysis or for use in external applications. The addition of masking and layer merging in future versions will enhance the utility of the output, allowing for even more customized and precise interpolations.

Step 6: Exporting and Saving the Results

The interpolated raster is saved to the file path specified by the user. This can be done directly through the file input widget, where the user defines the output path. The raster is saved as a GeoTIFF file, a common format for raster data that is compatible with various GIS applications. This makes it easy to share the results of the interpolation or use them in further geospatial analyses.

Step 7: Selecting the Symbology Type for the Points (Optional)

The plugin offers two primary symbology types:

- Graduated Symbology: This method is used to represent continuous data, such as velocity values, using color gradients. The user can select a classification method (Equal Interval, Jenks, Quantile) from the QComboBox, and the plugin divides the data into distinct classes based on the selected method. The symbology is applied by the *graduated_symbology()* function, which uses the **QgsGraduatedSymbolRenderer** class [84] to map data ranges to colors from a predefined color ramp.
- Vector Field Symbology: This method is designed for vector layers representing displacement data, such as glacier flow direction and magnitude. The user selects the respective fields for the magnitude in the horizontal axis and the vertical axis. Using these, the plugin visualizes the magnitude and direction of movement as arrows. The *vector_field_symbology()* function applies the

symbology using **QgsVectorFieldSymbolLayer** [85], which draws lines that scale according to the values in the selected fields. It would have been desired to also change the symbology of these lines into arrows automatically, however, there is sparse documentation on how to achieve this from PyQGIS, thus at this stage the change in symbology for these lines must be done manually after running the plugin.

The plugin dynamically updates the interface based on the user's choices. For instance, if the user selects Graduated Symbology, they will be prompted to choose the classification method and the number of classes. On the other hand, selecting Vector Field Symbology will prompt the user to specify fields representing the x and y magnitudes of displacements, as well as a scaling factor.

Future Improvements – Masking and Merging Layers

There are plans to enhance the Interpolation Tab with additional functionality to make the tool more flexible and powerful. Two key features under consideration include:

1. Masking the raster output: In future versions, the user will be able to use a mask layer (such as a glacier boundary) to clip the interpolated raster to a specific area of interest. This would allow the user to focus on regions of the glacier, excluding areas outside the boundary. This feature will be useful for reducing unnecessary computation and generating more targeted outputs.
2. Merging additional point layers: Another planned enhancement is the ability to merge additional point layers to complement the input data. This would allow users to interpolate values from multiple datasets, such as combining historical and recent measurements to improve the accuracy of the interpolated surface. By merging point layers, the interpolation would have more comprehensive input data, resulting in more reliable and precise outputs.
3. Vector marker customization: Finally, another planned enhancement is giving the user the chance to choose from the plugin's GUI the desired symbology for the Vector Marker. Currently the plugin generates standard black lines, however, providing the user with the option of using, for example, arrows, may save time and help standardize the process.

5 Results and Discussion

This chapter presents the results of the implementation and use of the QGIS plugin developed for glacier monitoring, using data from the Belvedere Glacier. The plugin automates and streamline key processes in glacier monitoring, including the calculation of volume changes, the generation of an interpolated model for displacements and velocities, the symbolized maps for GCPs, and the creation of standardized GCP monographies. The primary goal of this chapter is to showcase the various products generated by the plugin and provide an initial interpretation of these outputs in the context of glacier dynamics. In the discussion, the significance of these outputs is analyzed. Additionally, the practical implications of the plugin, its limitations and potential future improvements are considered.

5.1. The Belvedere Glacier Datasets

The dataset used to demonstrate the functionalities of the QGIS plugin originates from the Belvedere Glacier Monitoring Program conducted by Politecnico di Milano. This long-term monitoring initiative provides a rich dataset that includes various types of geospatial and topographic data, collected over several years. The dataset is particularly suitable for the development and testing of glacier monitoring tools like the plugin presented in this thesis, as it includes both high-resolution Digital Surface Models (DSMs) and Ground Control Point (GCP) data, among other elements [26] [27] [86] [87].

Table 4 provides an overview of the available datasets, outlining the data type, characteristics, and temporal coverage. This information will help contextualize the plugin's capabilities and the products it generates, showcasing how it can handle and process different forms of data for glacier monitoring purposes. It is important to note that not all available data was used.

Table 4. Overview of Datasets from the Belvedere Glacier Monitoring Program [87].

Data Type	Description	Characteristics	Temporal Coverage
Historic Digital Surface Models	Historic elevation data of the glacier surface from aerial photography, photogrammetrically constructed.	Spatial resolution: 1977, 1991, 2001: 50 cm. 2009: 40 cm.	1977, 1991, 2001, 2009
Digital Surface Models	High resolution elevation data of the glacier surface from UAV photography, photogrammetrically constructed.	Spatial resolution: 20 cm	Annual data since 2015
Orthophotos	Aerial and UAV photographs corrected for topographic distortion.	Resolution: 1977, 1991, 2001: 50 cm. 2009: 40 cm. 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023: 20 cm	1977, 1991, 2001, 2009, 2015, 2016,2017,2018, 2019, 2020, 2021, 2022, 2023
Ground Control Points	Geodetic points used for reference in DSM generation and validation.	In-person GNSS measurements.	Updated annually since 2015
Glacier Velocity and Displacement Data	Surface displacement and velocity measurements derived from GNSS and remote sensing.	Derived from photogrammetric flights' scatter points and GNSS measurements.	Annual data since 2016.

5.2. Volume Change

As discussed in Section 2.2.1, it is relevant for glacier monitoring to be able to compute the volume change of a glacier through DSMs. The volume change between each survey was computed, however, for conciseness it was determined to present only data for the years 1977, 1991, 2001, 2009, 2015, 2017, 2019 2021, and 2023.

Figure 12 shows that the Belvedere Glacier has suffered a general trend of volume loss since 1977, particularly in the S1 Accumulation zone. However, if both maps are compared, it can be noted that the major loss of ice volume in the S1 zone must have occurred between 1977 and 2015, since the period 2015-2023 presents small losses, and instead concentrates the losses in the S3 zone, particularly at the end of the glacier's northwest tongue.

Using the Statistics file produced with the plugin, it can be easily obtained that the total volume loss in the period 1977-2023 was almost 81 million cubic meters of ice, of which 22 million cubic meters were lost in the period 2015-2023. In other words, nearly 27% of the ice loss occurred in 17% of the period. Using this initial data, it can be deduced that the rate at which the glacier is losing volume has increased in the last decade.

However, it should be observed that the positive volume accumulation values occur outside what is the proper glacier. These values could come from rock accumulation or vegetation growth. These increases in volume resulting from non-glacier sources are problematic for the estimation of the total volume change, as they are a constant source of error over the years. Possible solutions will be discussed in Section 6.1. Furthermore, given the Belvedere glacier has a debris cover, there might be some unknown volume losses as increased debris may compensate for ice loss.

In addition to these absolute volume changes, it is important to assess the rate of volume loss over specific time periods to better understand the glacier dynamics. While the total volume loss provides an overall picture of the glacier retreat, analyzing the volume change per year over different intervals can identify periods of accelerated ice melting.

The next maps delve deeper into the annual volume change during various intervals, such as 1977-1991, 1991-2001, and more recent periods like 2021-2023. By comparing these intervals, it becomes possible to identify specific trends in glacier behavior, which helps in understanding whether the accelerated loss observed in the last decade is part of a longer trend or a more recent development linked to environmental changes.

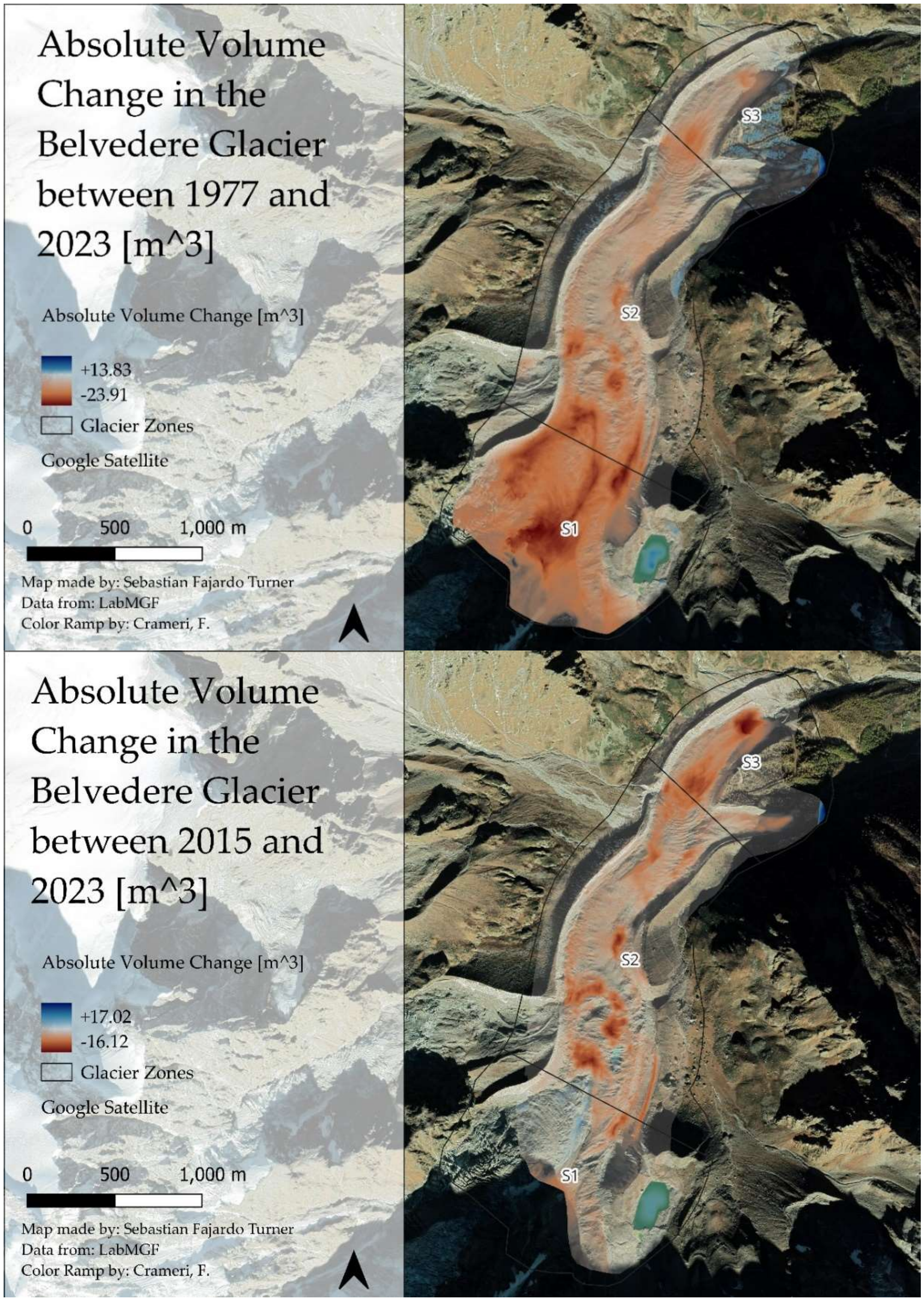


Figure 12. Absolute Volume Change in cubic meters in the Belvedere Glacier in the period 1977 - 2023 (upper) and 2015 – 2023 (lower). Resolution: 0.5 m. Processed with GlacioTools plugin. Data from LabMGF [87]. *Color Ramp Scientific Color maps* [88].

Figure 13 illustrates the annual changes in the periods from 1977 to 2015 in the glacier, where a different trend can be observed. In the period 1977-1991 a net positive accumulation of ice volume can be observed. This accumulation is particularly concentrated in the south-eastern portion of the S1 zone, and then evenly distributed in the northern half of the S2 zone and all throughout the S3 zone. No significative volume loss can be appreciated in this period. During the period 1991-2001 there is an even greater ice accumulation throughout the glacier, especially concentrated in the S2 zone. This period corresponds to the glacial surge shown in *Figure 4*. As discussed then, this glacial surge was product of a strong flow of ice coming from the upper sections of zone S1. Corroborating this, it can be observed that in the 1991-2001 period there is already a significant loss in the upper levels of zone S1.

However, the ice gain in these periods was not enough to compensate for the volume losses in the period 2001-2009. This period was characterized by a strong ice loss concentrated in the intersection between zones S1 and S2. The rate of loss for the selected periods can be observed in *Figure 14*. Overall, in this period there were losses of nearly 6.5 million cubic meters per year. This number is around 10% higher than the one found in the literature discussed in Section 2.4, which estimated loss on average of 5.97 million cubic meters per year [27]. This error could be due to different extents used for the glacier. A more accurate extent and mask may eliminate noise in elevation changes from outside the glacier, as the ones caused by rock accumulations, moraine raises or collapse, and vegetation growth or loss.

Nonetheless, the ice volume loss in other periods is more in line with the one found in the literature. The rate of loss in the period 2009-2019 is in a similar range to the one found in literature. The plugin estimated the loss to be 2.77 million cubic meters per year on average, while the literature stated a loss of 2.72 million cubic meters per year [27]. Ahead, the period between 2015 and 2023 is assessed by subdividing it into two-year gaps.

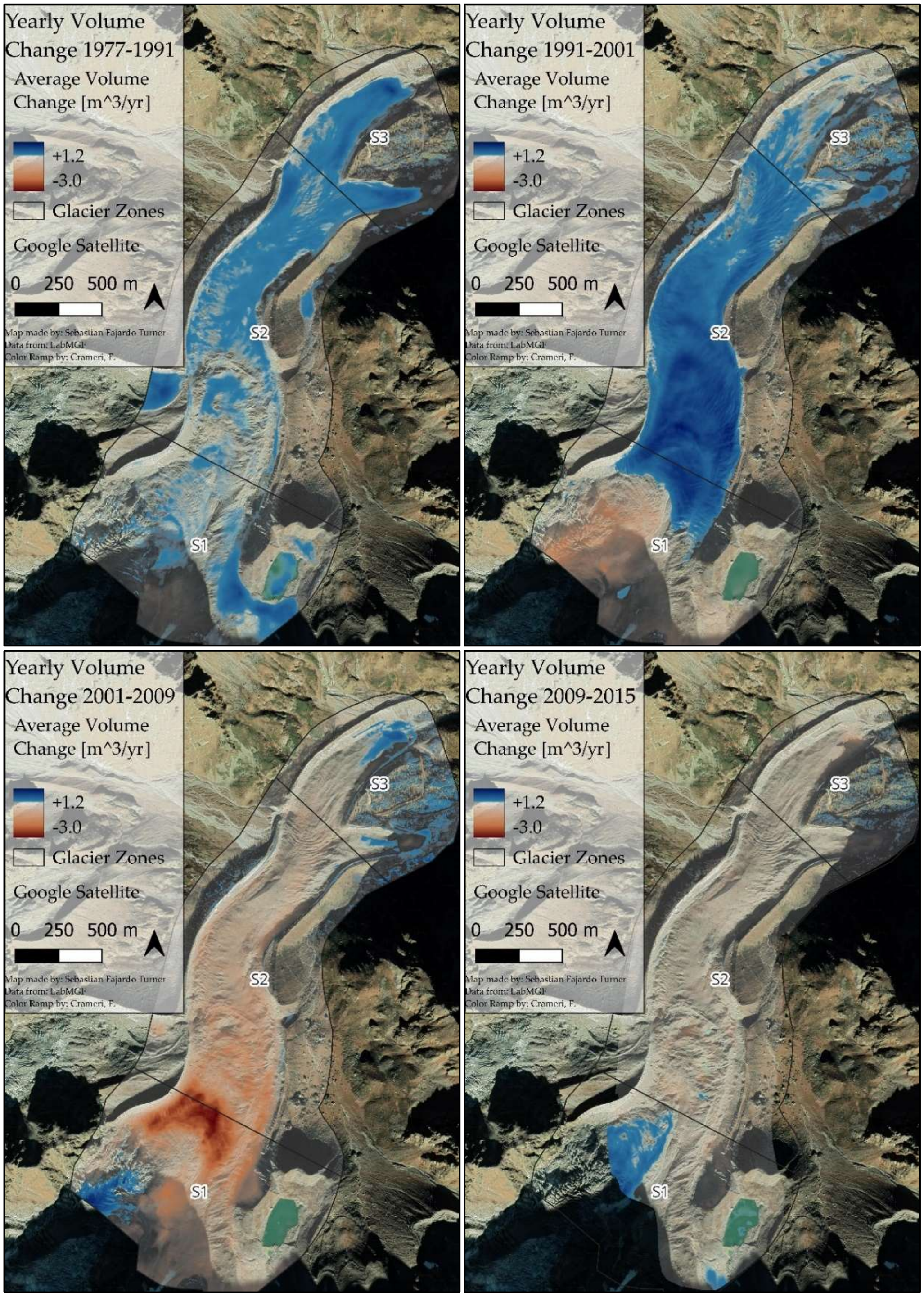


Figure 13. Average Yearly Volume Change in cubic meters per year in the Belvedere Glacier in the period 1977-1991 (upper-left), 1991-2001 (upper-right), 2001-2009 (lower-left) and 2009-2015 (lower-right) Resolution: 1 m. Processed with GlacioTools plugin. Data from LabMGF [87]. *Color Ramp Scientific Color maps* [88].

At first glance it can be observed in *Figure 15* that the volume changes after 2015 have been significantly less dramatic. However, they have followed a constant negative trend throughout the whole glacier. There are no significative accumulation zones in any part of the glacier, in fact the only apparent positive change between the DSMs occurred outside the glacier, particularly in Lake Locce. This may be due to new ice floes or simply errors in data collection due to the lake's surface characteristics.

On the other hand, there is a slight constant loss along the glacier. A small yet constant loss can be appreciated in the central part of zone S2. These losses constantly appeared throughout all periods. Furthermore, in the north-western tongue of zone S3 there is a constant and more significant loss of volume.

Coming back to *Figure 14*, obtained from the Statistics generated by the plugin, it can be observed that in fact there has been a constant loss of volume in the period 2015-2023. The rate of loss is less significant than the ones the glacier suffered between 2001-2015. However, starting in 2019 the rate of loss has increased once again, with it reaching similar levels to the 2009-2015 period by the end of 2023. Whether this accelerating loss trend continues remains to be seen with future monitoring campaigns.

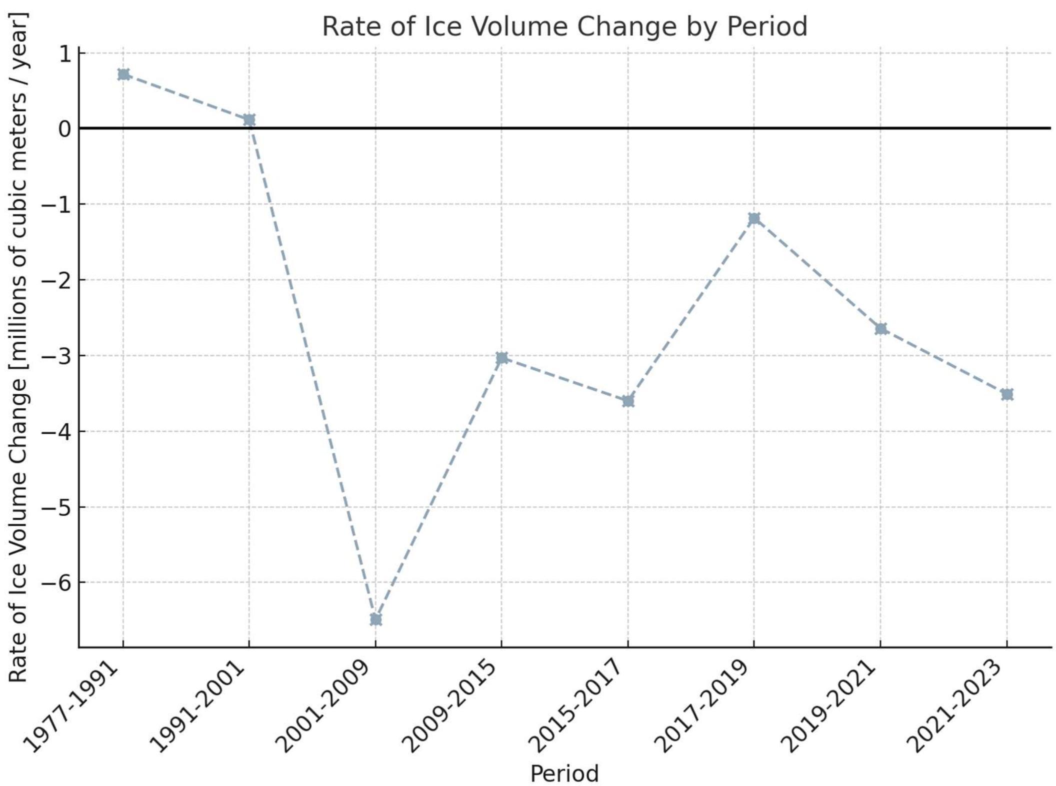


Figure 14. Rate of Ice Volume Change by Period from 1977 to 2023 [millions of cubic meters per year].

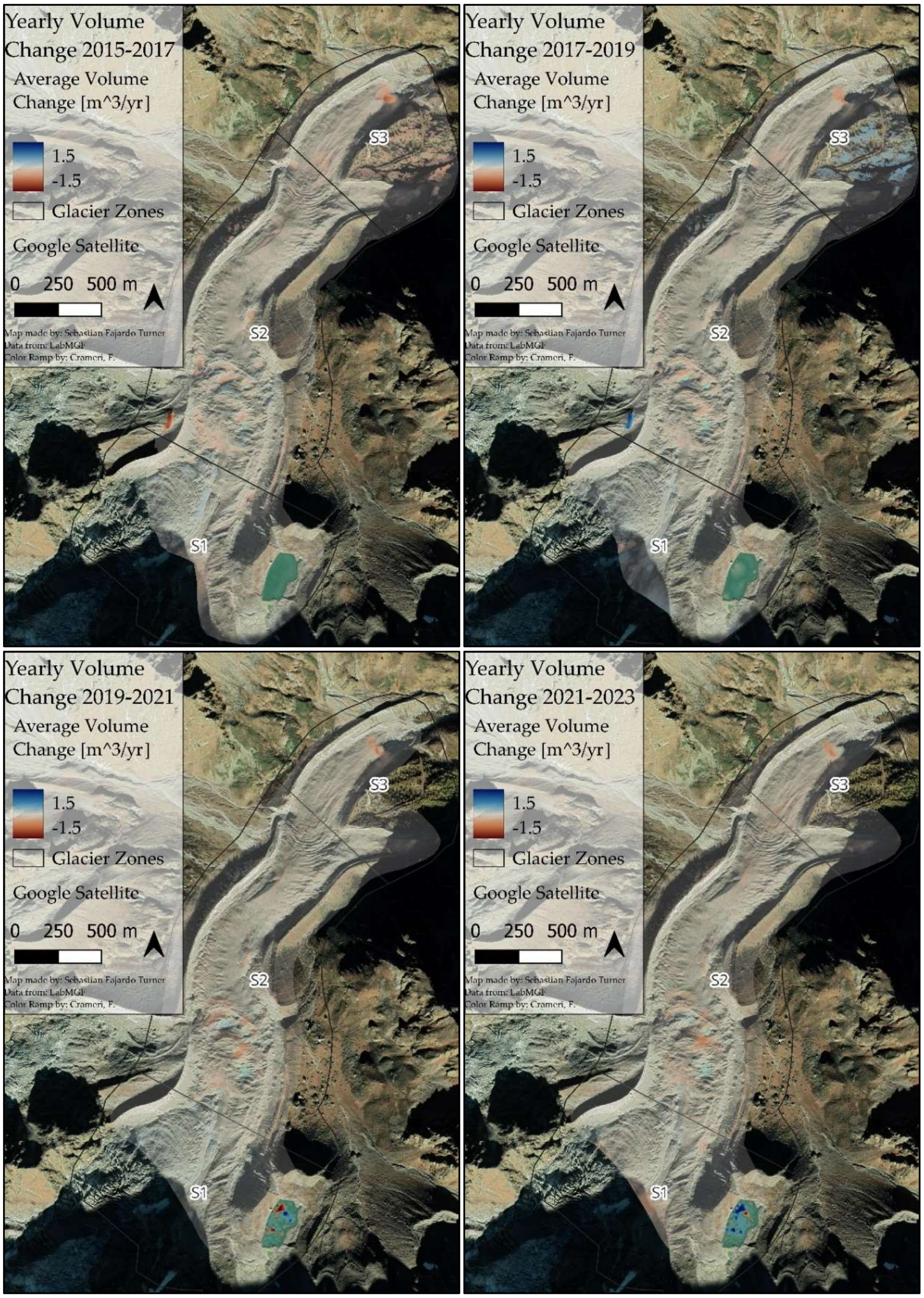


Figure 15. Average Yearly Volume Change in cubic meters per year in the Belvedere Glacier in the period 1977-1991 (upper-left), 1991-2001 (upper-right), 2001-2009 (lower-left) and 2009-2015 (lower-right) Resolution: 1 m. Processed with GlacioTools plugin. Data from LabMGF [87]. *Color Ramp Scientific Color maps* [88].

5.3. Displacement, Surface Velocity and Acceleration of the Glacier

Monitoring glacier displacement, surface velocity, and acceleration is another relevant aspect for understanding the dynamic behavior of glaciers. These metrics may provide specialists with the necessary insights into how ice moves, deforms, and responds to environmental forces such as gravity and climate-induced changes in temperature and precipitation. In this case, displacement refers to the total movement of a glacier's surface over time, while surface velocity measures the rate of this movement in each period. On the other hand, acceleration highlights changes in the velocity, indicating whether a glacier is speeding up or slowing down. Together, these measurements are essential for predicting glacier surges, identifying potential instabilities, and assessing the overall health of a glacier system. Using the plugin's functionalities, the surface velocity field was interpolated for the glacier, as well as its acceleration. Furthermore, using its symbology function, the vector field marker creation was simplified. In *Figure 16* maps depicting surface velocity with displacement vectors are presented for the years 2017, 2018, 2019, and 2020. Additionally, *Figure 17* presents maps with the interpolated acceleration field and velocity vector markers.

The interpolation was done with data from the Belvedere Glacier's yearly monitoring. This dataset includes scattered points across the glacier, specifically homologous points—features identified consistently across orthophotos from different years—and GCPs, which serve as reference points for precise spatial alignment. These combined points provide a basis for interpolating glacier surface changes over time. An Inverse Distance Weighting interpolation with a weight of one was applied using the plugin. Later, due to the current limitations of the plugin, the resulting raster file was masked manually in QGIS using the glacier's mask that was used for the volume change calculations. Unfortunately, there are not enough points throughout the whole glacier to completely interpolate a field. Therefore, the upper sections of zone S1 and the tips of the tongues in zone S3 are not visualized in this interpolation. Nonetheless, certain trends can be observed using the available data. It is also important to note that currently there is no estimated error for the interpolation calculated. This is a possible improvement to the plugin's functionality.

In 2017 a high surface velocity can be observed in the visible portion of S1 and all along the central regions of zone S2, with values of up to 26 meters per year. This behavior is consistent with the one discussed in literature, since glaciers tend to have a higher displacement in the central spine as it is impacted to a lesser extent by friction with the moraines. Another relevant observation is that notwithstanding the higher velocities in S2, once the ice reaches the terminal section S3, the velocity is greatly reduced. This is also consistent with literature, as it has been observed that the tongues have reduced movement but have suffered volume loss instead.

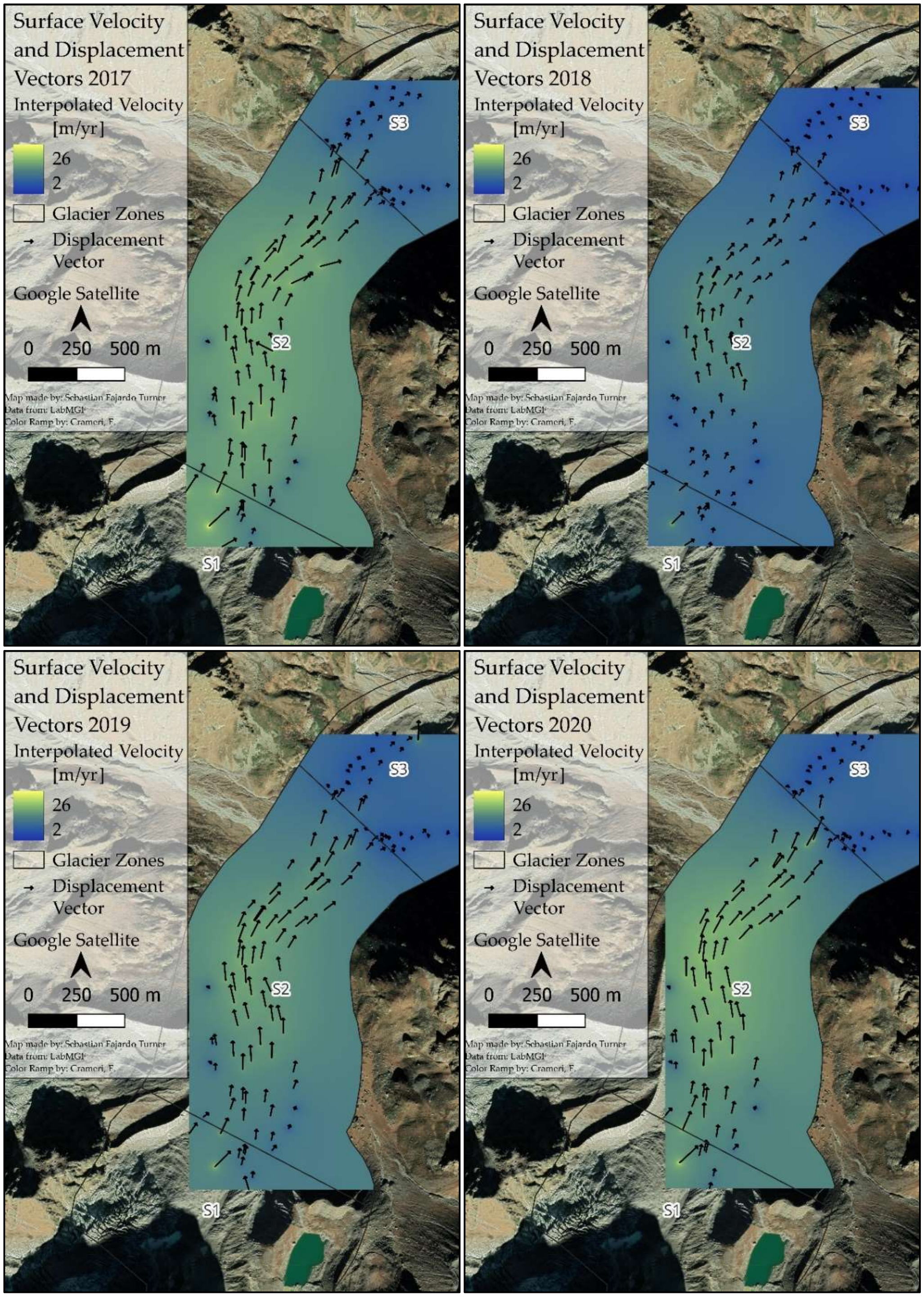


Figure 16. Surface Velocity and Displacement Vectors in the Belvedere Glacier for the years 2017 (upper-left), 2018 (upper-right), 2019 (lower-left), and 2020 (lower-right). Resolution: 0.5 m. Processed with GlacioTools plugin. Data from LabMGF [87]. *Color Ramp Scientific Color maps* [88].

For the year 2018 there is a reduction in the velocity shift. It is particularly observable in the border zone between S1 and S2. There was still a higher velocity in the central region of S2, however, noticeably lower than the one observed in 2017. However, the trend seems to reverse once again in 2019, as velocities throughout the glacier increase again. The positive trend remains in 2020, where parts of the glacier's surface have reached velocities comparable to 2017. Nonetheless, the glacier tongues still have a lower surface velocity than the one estimated in 2017. In general, the glacier's tongues have velocity values as low as 2 meters per year.

The maps shown in *Figure 17* provide a clearer image of how the surface velocity has changed in the glacier over the years. The year 2017, which had a high velocity, also experienced a significant acceleration compared to 2016. Once again, the most accelerated zone was the border region between zones S1 and S2. Though it is interesting to note that the glacier also experienced a slight deceleration in 2017 around the border zone between S2 and S3.

The acceleration map reinforces the previous statement, showing that in fact for 2018 the glacier had a general decelerating behavior. The greatest deceleration was in the border area between S1 and S2, reaching negative values up to 12 meters per year squared. This area suffered high acceleration in 2017 of up to 7 meters per year squared. It is also interesting to note that the terminus areas, although constantly slowing down, have not experienced a proper decelerating behavior, but simply remained constant.

The deceleration behavior does not remain constant, and reverses again for the year 2019, when the glacier goes through its highest acceleration phase of the four years reported, reaching peaks of 8 meters per year squared. The trend slows down for the year 2020, but it remains positive, indicating a still increasing velocity in the glacier's surface.

These maps offer an example of the potential to produce visually effective maps for glacier dynamics using the plugin. The maps offer a clear representation of the spatial variability of the glacier's movement, providing important clues about areas of rapid ice flow and potential regions of instability. By studying these visualizations, it is possible to better understand the underlying processes driving glacier dynamics and how they may evolve in response to environmental changes.

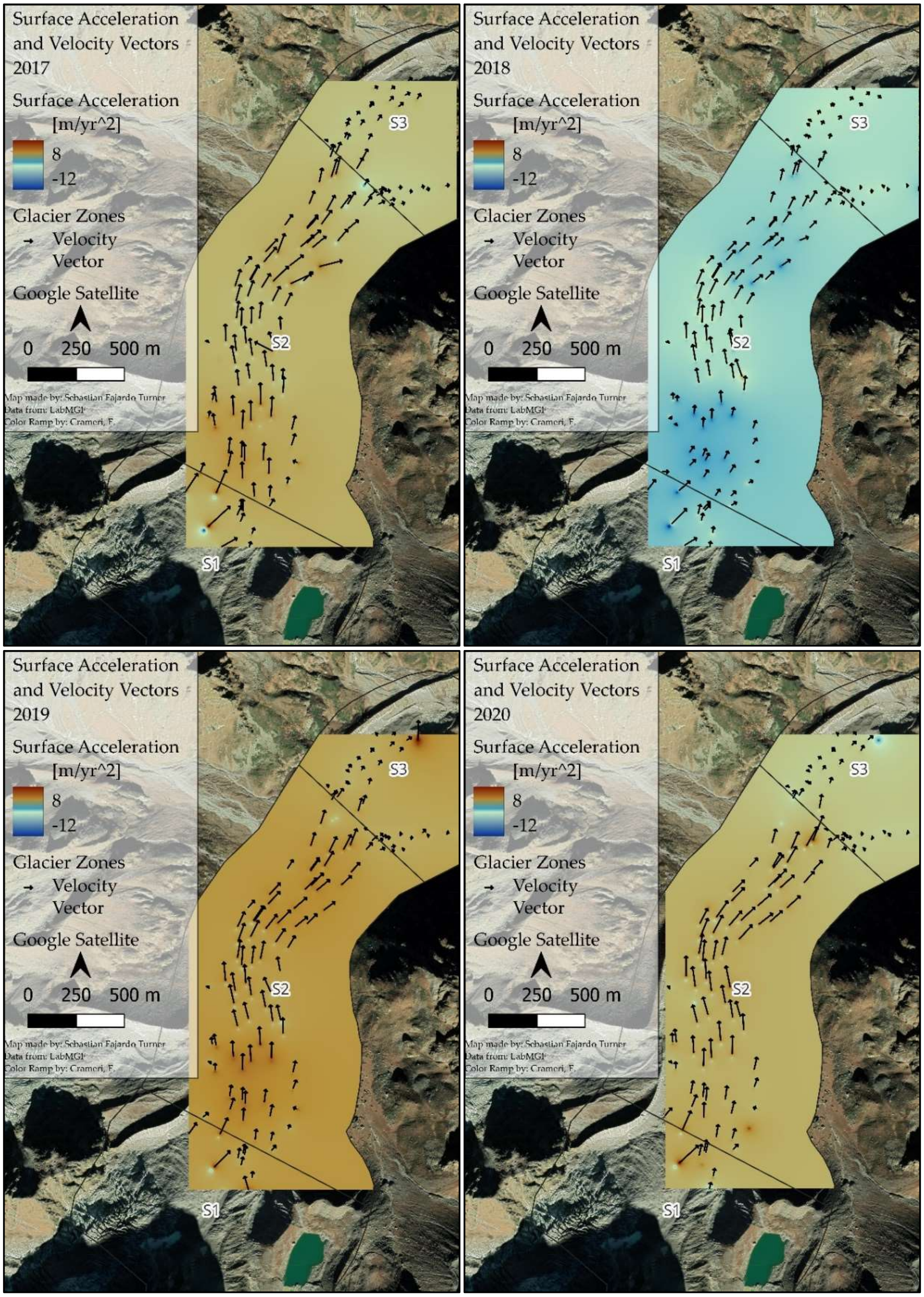
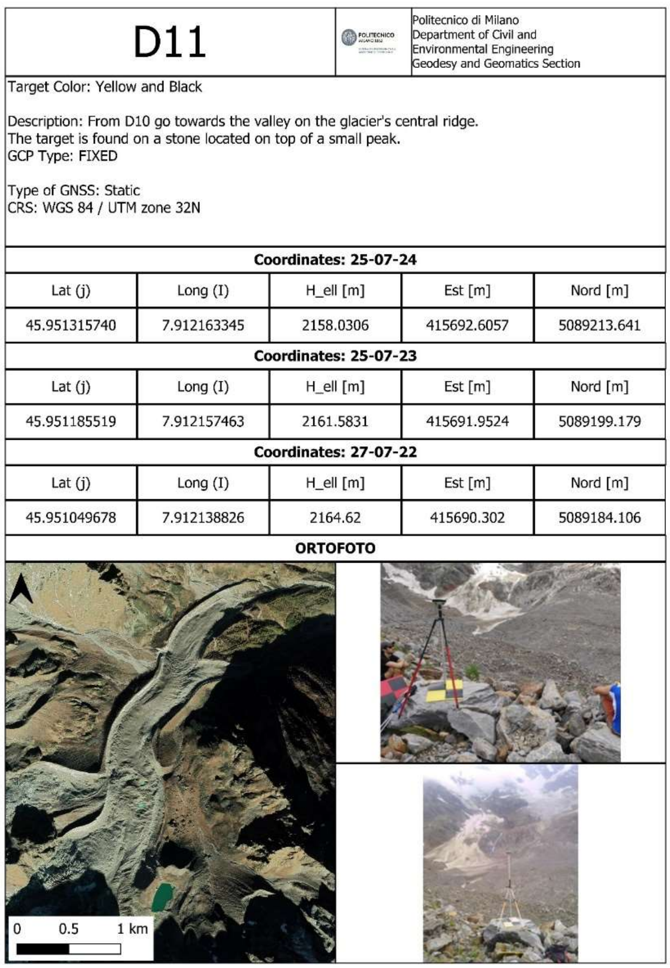


Figure 17. Surface Acceleration and Velocity Vectors in the Belvedere Glacier for the years 2017 (upper-left), 2018 (upper-right), 2019 (lower-left), and 2020 (lower-right). Resolution: 0.5 m. Processed with GlacioTools plugin. Data from LabMGF [87]. *Color Ramp Scientific Color maps* [88].

5.4. GCP Monography

The GCP Monographies play an important role in keeping track of the targets monitored every year in and around the glacier. As previously discussed, the elaboration of the monographies is an onerous task, prone to typographical and human error. The plugin eases this process and frees up time for other tasks. An example of a monography is shown in *Figure 18*.

The aim of the plugin is to be useful to any glacier monitoring program. However, this monography format is specific to the Politecnico di Milano. Nonetheless, efforts were made to make it as adaptable as possible to any user. Improvements to this section will depend on user feedback.



D11

Politecnico di Milano
Department of Civil and Environmental Engineering
Geodesy and Geomatics Section

Target Color: Yellow and Black

Description: From D10 go towards the valley on the glacier's central ridge.
The target is found on a stone located on top of a small peak.
GCP Type: FIXED

Type of GNSS: Static
CRS: WGS 84 / UTM zone 32N

Lat (j)	Long (I)	H_ell [m]	Est [m]	Nord [m]
Coordinates: 25-07-24				
45.951315740	7.912163345	2158.0306	415692.6057	5089213.641
Coordinates: 25-07-23				
45.951185519	7.912157463	2161.5831	415691.9524	5089199.179
Coordinates: 27-07-22				
45.951049678	7.912138826	2164.62	415690.302	5089184.106

ORTOFOTO

Figure 18. Example of the GCP Monography for point D11.

6 Conclusions

This chapter presents a comprehensive overview of the development, application, and future potential of the GlacioTools plugin designed for glacier monitoring within the QGIS environment. By automating complex geospatial analysis tasks, the plugin simplifies the calculation of glacier volume changes, visualization of displacement, and analysis of surface velocity and acceleration, providing a more accessible and efficient solution for researchers, practitioners, and educators. Section 6.1 examines the current limitations of the plugin, such as its dependency on data quality, QGIS processing speed, and areas where user interface improvements could enhance usability. It also discusses opportunities for future development, including the integration of additional libraries and advanced visualization features. Section 6.2 reflects on the plugin's performance in monitoring the Belvedere Glacier and its potential to expand glacier monitoring practices. Emphasizing the plugin's open-source nature, the chapter concludes by highlighting its contribution to both scientific research and education, as well as its broader impact as a tool for tracking glacial changes in response to climate change.

6.1. Limitations and Future Development

While the developed QGIS plugin has proven to be a valuable tool for automating and streamlining various processes, there are certain limitations and areas of opportunity that emerged during its application. Evidently, a key limitation lies in the inherent complexity of glacier dynamics, which may not always be fully captured by the plugin's current functionality. For example, the plugin completely relies on the accuracy of the input data to calculate velocity fields and volume changes. Variations in the spatial and temporal resolution of the DSM data can introduce errors, especially when working with older or lower resolution datasets. This discrepancy can lead to minor inaccuracies in estimating volume changes.

Another aspect to consider is the dependency of the plugin on QGIS's internal algorithms, as it is built using the PyQGIS library and compiled within the QGIS environment. This has made it slower and more difficult implementing unit testing and developing new sections of code. Furthermore, the plugin operates within the somewhat bloated system of QGIS, which can slow down processing times, particularly with large or complex datasets. Moreover, by running through QGIS's predefined functions, much of the process becomes a "black box, where the user has limited control or visibility into the internal workings of the algorithm.

Nonetheless, the reliance on QGIS offers certain significant advantages as well. It allows the plugin to leverage established and proven methods within QGIS, such as the QgsInterpolator for data interpolation. Users benefit from the fact that these algorithms have been extensively tested and are widely trusted, notwithstanding certain limitations and disappointing results. Furthermore, working within QGIS ensures universality between devices if the correct version of QGIS is installed, as opposed to having to create specific environments to run Python libraries.

For more rigorous and customized analysis, a standalone Python script or package using other geospatial libraries such as *geopandas* or [89] [90] [91] [92] [93] [94] [95] [96] [97], may offer more flexibility and efficiency in handling data. These libraries allow for more granular control over data processing, which could be advantageous in specific contexts where high accuracy and adaptability are required.

Nevertheless, the shift towards using standalone Python libraries also raises the entry barrier for potential users. While QGIS provides an accessible platform with a user-friendly interface, transitioning to more advanced Python scripting requires a higher level of technical knowledge in both programming and geospatial analysis. Therefore, the balance between ease of use and analytical rigor must be considered carefully when thinking about future development. A clear distinction must be made between the use cases of a user-friendly plugin for routine glacier monitoring and more complex analyses that require a higher degree of customization and precision.

Another potential limitation lies in the plugin's user interface. Although designed to be intuitive, the wide variety of possible inputs and outputs may confuse users who are unfamiliar with geospatial data processing. While the QGIS interface is accessible, the complexity of glacier dynamics and data processing may overwhelm less experienced users. To address this, future versions could include more comprehensive in-plugin documentation or tooltips that guide users through each step of the analysis, reducing the learning curve and minimizing the potential for user error.

Other examples of the trade-off between using QGIS's built-in algorithms and external libraries can be seen in areas such as spatial analysis and raster calculations. For instance, QGIS offers powerful built-in tools for raster operations, but these tools may not be as efficient as alternatives in libraries like GDAL when working with very large datasets. Similarly, QGIS's built-in symbology options, while intuitive and flexible for most users, may lack the advanced customization possibilities offered by more programmatic approaches using matplotlib or seaborn for data visualization.

It is possible to integrate certain external libraries into the QGIS script. However, due to time constraints, these options were not fully explored for this version of the plugin.

It is encouraged for future developers to investigate these options as it may greatly improve the quality of the output and efficiency of the rendering process.

Concerning specific limitations and future development of the plugin, while developing the products used in Chapter 5, certain areas of opportunity were identified, as well as certain flaws in the design of the plugin. These observations are detailed along the following paragraphs.

Concerning the Elevation and Volume Change process, it was found that although the script correctly calculates the raster difference even if the input raster files have different resolutions, the output resolution is not explicitly stated, and it is not controlled by the user. This can cause confusion, as interpreting volume changes with different pixel resolutions can lead to inaccuracies. Future development should allow users to specify the output resolution and even apply a standardized symbology to the output raster for easier interpretation. Additionally, optimizing how the plugin handles these operations could improve its performance with larger datasets or lower resolution DSMs.

Concerning the Interpolation and Symbology process, it must be noted that it is the functionality that went through the most changes out of all embedded in the plugin. There are simply too many possible desired outcomes. Currently, the areas of opportunity identified are the following. In the interpolation section, it would greatly streamline the process if the option to use a vector layer as a mask for the interpolated raster file was given. This option was not added due to specific peculiarities of how the QgsInterpolator and saving a raster to file works. Nonetheless, it is considered a crucial aspect that would greatly improve the quality of life for the user. Additionally, the plugin currently works by using a Query filter to only run with the selected survey year. This was chosen to offer more flexibility to the user, in case the user wanted to include filters not only related to the year. However, resetting filters every time the plugin must run, although seemingly inconsequential, can prove to be a tedious task. Finally, pertaining to the symbology section, there is the already discussed limitation in which the vector field marker only can create simple lines from code. There is simply not enough clear documentation available to find a workaround, but certainly there is a possible solution. Finally, it may be necessary to add an option to only run the symbology without needing to interpolate points first.

Concerning the Monography process, there are not many comments to be made at this stage. Possible improvements and areas of opportunity will arise when it is finally utilized by the end user. One potential enhancement could be the integration of external python libraries, for example, Matplotlib. This would allow the creation of plots showing desired attributes for the selected target over time, such as displacement. This feature would provide users with a clear visual representation of movement patterns for specific points, further enhancing the utility of the monographies for glacier monitoring and analysis.

To account for all these considerations and any possible bugs that may arise, the plugin has a GitHub repository [98] where end users can report bugs and suggestions. Additionally, it is open source, meaning that any user and developer can contribute to improving the plugin. This is particularly crucial, as the aim of the plugin is to be a helpful tool for any glacier monitoring project, and the more feedback and experience that collaborates in it will lead to a much higher quality tool for researchers.

Finally, the plugin is not yet ready for release in the QGIS Plugin Repository. It is expected that in the following months after the publication of this text, the necessary steps, such as complete metadata, translation features, and bug testing will be complete to submit it to the QGIS community. For the time being the distribution of the plugin remains solely within the community and shared as a compressed file.

6.2. Final Conclusions

The continuous and accelerating changes in the global climate have a profound impact on the cryosphere around the world. Thus, it is ever more essential to develop and make accessible efficient and reliable monitoring tools. Glaciers serve as vital indicators of climate change. The ability to track their volume changes, surface displacement, and velocity is essential for understanding both local environmental changes and broader global trends. This thesis aimed to contribute to this effort by designing, developing, implementing, and applying a custom QGIS plugin tailored to automate and streamline glacier monitoring processes, using the Belvedere Glacier Monitoring Project as a baseline and source of data. The goal was not only to improve the efficiency of data processing but also to make glacier monitoring more accessible to a wider range of users, from researchers and scientists to field technicians.

The GlacioTools plugin developed within the framework of this project addresses key challenges associated with traditional glacier monitoring workflows, which often rely on manual data handling and inconsistent methods. The plugin automates tasks such as calculating DSM elevation changes, interpolating displacement values, and generating standardized report layouts, thus providing a more seamless and standardized approach to glacier monitoring. This automation reduces the potential for human error and significantly shortens the time required for data analysis, allowing for quicker times between data collection and the generation of valuable insights.

Moreover, the GlacioTools plugin is integrated into QGIS, a widely used open-source platform, which ensures that it can be adopted by researchers and monitoring programs around the world without the need for costly, proprietary software. Its open-source nature also encourages collaboration and further development by the broader scientific community, ensuring that the tool can continue to evolve in response to new challenges and technological advancements.

In Chapter 5 the GlacioTools plugin was tested using data from the Belvedere Glacier. This provides valuable insights into its recent behavior, including significant ice volume losses and shifts in glacier dynamics over time. The results generated by the plugin are consistent with global trends of glacial retreat, emphasizing the critical role that such monitoring tools can play in tracking environmental changes over the long term. While the plugin has demonstrated considerable utility, the process has also revealed limitations and opportunities for future development. The plugin is by no means a finished product and the intention is to continue developing it and allow collaboration from multiple users to fine-tune and expand its functionalities.

One of the most significant contributions of the plugin is its ability to integrate multiple stages of the data analysis process, providing a user-friendly interface that removes much of the complexity traditionally associated with glacier monitoring. By reducing the reliance on manual data processing and integrating established geospatial algorithms, the plugin simplifies what would otherwise be a laborious and time-consuming process.

From the results discussed in Chapter 5, the plugin has demonstrated its effectiveness in accurately estimating ice volume changes across different time periods, specifically between 1977 and 2023. The glacier's retreat is marked by noticeable fluctuations in volume loss, with an accelerated rate of loss evident in recent years (2019-2023). These findings are consistent with the global trends of glacial retreat, underlining the importance of having reliable and efficient tools like the plugin for continuous glacier monitoring.

While the rate of ice volume change for earlier periods (e.g., 1977-2001) showed some periods of stabilization or even accumulation, the latter half of the data set, particularly between 2015 and 2023, showcased a consistent and worrying decline. This sharp drop in glacier volume is reflective of both local climatic conditions and the broader impacts of climate change.

The observations on volume loss are supported by the analysis on glacier displacement, surface velocity, and acceleration, which has provided insights into the dynamic behavior of the Belvedere Glacier. The surface velocity maps generated with the plugin illustrate the glacier's variability in movement across different time periods, offering a detailed look at areas of rapid flow and zones where velocity slows down. Data from 2017 to 2020 shows noticeable fluctuations in velocity and acceleration, particularly in the central regions of the glacier, where high velocities align with reduced friction from moraines, and in terminal zones, where ice movement slows down considerably, highlighting areas of volume loss.

The ability to map acceleration adds further depth to the understanding of the glacier’s behavior, indicating whether certain parts of the glacier are speeding up or slowing down. The observed trends, such as the strong acceleration in 2017 followed by deceleration in 2018 and another surge in 2019, provide important clues about the glacier's behavior, which could potentially be paired with external factors, such as temperatures, for further research.

The GlacioTools plugin developed offers a robust tool for simplifying complex geospatial analysis tasks, such as calculating glacier volume changes, visualizing displacements, and generating surface velocity and acceleration maps. By streamlining these processes, the plugin provides an accessible and efficient solution for researchers, practitioners, and students alike, enabling a deeper understanding of glacier behavior without the burden of technical software complexities. An added advantage is its educational potential for students attending the Belvedere Glacier Summer School hosted by Politecnico di Milano. As a hands-on tool, the plugin allows students to work with real data from the Belvedere Glacier, applying methods like interpolation, symbology, and geodetic analysis. This experience helps students gain practical skills in glaciology and geospatial analysis, as well as learn how to communicate their findings effectively through maps and reports. Ultimately, the plugin supports both research and learning, making glacier monitoring tasks more accessible to a wider audience.

The potential of this plugin extends beyond the technical scope of glacier monitoring at the Belvedere Glacier. In an era where climate change is rapidly transforming natural environments, efficient and scalable tools for environmental monitoring are more crucial than ever. By automating and standardizing key aspects of the glacier monitoring process, the plugin can help ensure that monitoring programs are more accessible, scalable, and capable of delivering reliable data on a consistent basis.

Moreover, as glacier monitoring becomes increasingly essential for predicting the impacts of climate change—such as the potential for glacial lake outburst floods (GLOFs) or long-term water supply reductions—tools like this plugin will be invaluable for both scientific research and practical risk management.

In conclusion, the plugin has high potential to contribute significantly to advancing glacier monitoring techniques. Its application to the Belvedere Glacier has demonstrated both its strengths and limitations, providing a solid foundation for future work in this field. Moving forward, ongoing refinements to the plugin and its underlying methods will not only enhance its effectiveness but also broaden its applicability to other glacial environments, aiding in the global effort to monitor and mitigate the impacts of climate change.

Figure 19. Students and faculty at the Belvedere Glacier Summer School 2024. Photo credits to Pinto, L.

7 References

[1] D. Hoptsii, A. Siedov, T. Anopriienko, D. Khainus and D. Yaremko, "Advantages of using QGIS to solve spatial planning tasks," *Baltic surveying: International Scientific Journal,* vol. 18, no. Poland : Lithuania, p. 50'56, 2023.

[2] M. Miller, "When do you need a plugin vs. a script?," 11 2020. [Online]. Available: https://www.udemy.com/course/qgis-plugin-development-with-python/learn/lecture/22885834#content. [Accessed 16 09 2024].

[3] S. Winkler, T. Chinn, I. Gärtner-Roer, S. U. Nussbaumer, M. Zemp and H. J. Zumbühl, "An Introduction to Mountain Glaciers as Climate Indicators with Spatial and Temporal Diversity," *Erdkunde,* vol. 64, no. 2, pp. 97-118, 2010.

[4] J. Oerlemans, "Extracting a Climate Signal from 169 Glacier Records," *Science,* vol. 308, pp. 675-677, 2005.

[5] G. Nigrelli, S. Lucchesi, S. Bertotto, G. Fioraso and M. Chiarle, "Climate variability and Alpine glaciers evolution in Northwestern Italy from the Little Ice Age to the 2010s," *Theor Appl Climatol,* vol. 122, pp. 696-608, 2015.

[6] D. van Vuuren, J. Edmonds, M. Kainuma, K. Riahi, A. Thomson, K. Hibbard, G. Hurtt, T. Kram, V. Krey, J.-F. Lamarque, T. Masui, M. Meinshausen, N. Nakicenovic, S. Smith and S. Rose, "The representative concentration pathways: an overview," *Climatic Change,* vol. 109, no. 5, 2011.

[7] Y. Kwon, J. Hwang and Y. Seo, "Performance of a RBSN under the RCP Scenarios:A Case Study in South Korea," *Sustainability,* vol. 10, no. 1242, 2018.

[8] H. Zekollari, M. Huss and D. Farinotti, "Modelling the future evolution of glaciers in the European ALps under the EURO-CORDEX RCM ensemble," *The Cryosphere,* vol. 13, pp. 1125-1146, 2019.

[9] C. Sommer, P. Malz, T. C. Seehaus, S. Lippl, M. Zemp and M. H. Braun, "Rapid glacier retreat and downwasting throughout the European Alps in the early 21st century," *Nature Communications,* vol. 11, no. 3209, 2020.

[10] C. Taylor, T. R. Robinson, S. Dunning, J. R. Carr and M. Westoby, "Glacial lake outburst floods threaten millions globally," *Nature Communications,* vol. 14, no. 487, 2023.

[11] G. Mortara and L. Mercalli, "Il lago epiglaciale "Effimero" sul ghiacciaio del Belvedere, Macugnaga, Monte Rosa," *Nimbus,* pp. 10-17, 2002.

[12] G. Nigrelli, R. Paranunzio, L. Turconi, F. Luino, G. Mortara, M. Guerini, M. Giardino and M. Chiarle, "First national inventory of high-elevation mass movements in the Italian Alps," *Computers & Geosciences,* vol. 184, no. 105520, 2024.

[13] F. Ioli, *Multi-scale and multi-temporal photogrammetry for alpine glacier monitoring,* Milano, MI: IngCat, Polimi, 2024.

[14] J. Faillettaz, M. Funk and C. Vincent, "Avalnching glacier instabilities: Review on processes and early warning perspectives," *Reviews of Geophysics,* vol. 53, no. 53, pp. 203-224, 2015.

[15] E. Lee, "Calculating glacier volume change from Space," Antarctic Glaciers, Newcastle University, 2023.

[16] D. Elsberg, W. Harrison, K. Echelmeyer and R. Krimmel, "Quantifying the effects of climate and surface change on glacier mass balance," *Journal of Glaciology,* vol. 47, no. 159, pp. 649-658, 2001.

[17] Washington Water Science Center, "Glaciers and Snowpack," USGS, 21 12 2017. [Online]. Available: https://www.usgs.gov/centers/washington-water-science-center/science/glaciers-and-snowpack#overview. [Accessed 2024 10 16].

[18] B. Su, C. Xiao, D. Chen, Y. Huang, Y. Che, H. Zhao, M. Zou, R. Guo, X. Wang, L. Xin, W. Guo, S. Liu and T. Yao, "Glacier change in China over past decades: Spatiotemporal patterns and influencing factors," *Earth-Science Reviews,* vol. 226, no. 103926, 2022.

[19] R. Hugonnet, R. McNabb, E. Berthier, B. Menounos, C. Nuth, L. Girod, D. Farinotti, M. Huss, I. Dussaillant, F. Brun and A. Kaab, "Accelerated global glacier mass loss in the early twenty-first century," *Nature,* vol. 592, pp. 726-731, 2021.

[20] G. J. Cogley, "Geodetic and direct mass-balance measurements: comparison and joint analysis," *Annals of Glaciology,* vol. 50, no. 50, pp. 96-100, 2009.

[21] M. Shahgedanova, "Chapter 3 - Climate change and melting glaciers," in *The Impacts of CLimate Change*, Bath, Elsevier, 2021, pp. 53-84.

[22] P. Wang, Z. Li, H. Li, W. Wang and H. Yao, "Comparison of glaciological and geodetic mass balance at Urumqi Glacier No. 1, Tian Shan, Central Asia," *Global and Planetary Change,* vol. 114, pp. 14-22, 2014.

[23] K. Lamsters, J. Jeskins, I. Sobota, J. Karuss and P. Dzerins, "Surface Characteristics, Elevation Change, and Velocity of High-Arctic Valley Glacier from Repeated High-Resolution UAV Photogrammetry," *Remote Sensing,* vol. 14, no. 1029, 2022.

[24] M. Hambrey and N. Glasser, "Sedimentary Processes of Glaciers," in *Encyclopedia of Geology*, Elsevier, 2005, pp. 663-678.

[25] X. Zhang, M. Feng, H. Zhang, C. Wang, Y. Tang, J. Xu, D. Yan and C. Wang, "Detecting Rock Glacier Displacement in the Central Himalayas Using Multi-Temporal InSAR," *Remote Sensing,* vol. 13, no. 23, 2021.

[26] F. Ioli, A. Bianchi, A. Cina, C. De Michele, P. Maschio, D. Passoni and L. Pinto, "MId-Term Monitoring of Glaciers Variations with UAVs: The Example of the Belvedere Glacier," *Remote Sensing,* vol. 14, no. 28, 2022.

[27] C. I. De Gaetani, F. Ioli and L. Pinto, "Aerial and UAV Images for Photogrammetric Analysis of Belvedere Glacier Evolution in the Period 1977-2019," *Remote Sensing,* vol. 13, no. 3787, 2021.

[28] S. Park, H. Lee and J. Chon, "Sustainable monitoring coverage of unmanned aerial vehicle photogrammetry according to wing type and image resolution," *Environmental Pollution,* vol. 247, pp. 340-348, 2018.

[29] W. Immerzeel, P. Kraaijenbrink, J. Shea, A. Shrestha, F. Pellicciotti, M. Bierkens and S. de Jong, "High-resolution monitoring of Himalayan glacier dynamics using unmanned aerial vehicles," *Remote Sensing of Environment,* vol. 150, pp. 93-103, 2014.

[30] D. Fugazza and G. Diolaiuti, "Glacier volume change monitoring from UAV observations: issues and potential of state-of-art techniques," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. XLIII, no. B2, 2020.

[31] A. Keutterling and A. Thomas, "Monitoring glacier elevation and volume changes with digital photogrammetry and GIS at Gepatschferner glacier, Austria," *International Journal of Remote Sensing,* vol. 27, no. 19, pp. 4371-4380, 2006.

[32] N. Karimi, S. Sheshangosht and R. Roozbahani, "High-resolution monitoring of debris-covered glacier mass budget and flow velocity using repeated UAV photogrammetry in Iran," *Geomorphology,* vol. 389, no. 107855, 2021.

[33] B. Mark and O. Wigmore, "Monitoring tropical debris-covered glacier dynamics from high-resolution unmanned aerial vehicle photogrammetry, Cordiller Blanca, Peru," *The Cryosphere,* vol. 11, pp. 2463-2480, 2017.

[34] G. Diolaiuti, C. D'Agata and C. Smiraglia, "Belvedere Glacier, Monte Rosa, Italian Alps: Tongue Thickness and Volume Variations in the Second Half of the 20th Century," *Arctic, Antarctic, and Alpine Research,* vol. 35, no. 2, pp. 255-263, 2003.

[35] W. Haeberli, A. Kääb, F. Paul, M. Chiarle, G. Mortara, A. Mazza, P. Deline and S. Richardson, "A surge-type movement at Ghiacciaio del Belvedere and a developing slope instability in the east face of Monte Rosa, Macugnaga, Italian Alps," *Norsk Geografisk Tidsskrift,* vol. 56, no. 2, pp. 104-111, 2002.

[36] A. Kääb, C. Huggel, S. Barbero, M. Chiarle, M. Cordola, F. Epifani, W. Haeberli, G. Mortara, P. Semino, A. Tamburini and G. Viazzo, "Glacier Hazards at Belvedere Glacier and the Monte Rosa East Face, Italian Alps: Processes and Mitigation," in *Internationales Symposion INTERPRAEVENT*, Riva / Trient, 2004.

[37] A. Tamburini, M. Chiarle and G. Mortara, "Il Collasso delle Morene del Ghiacciaio del Belvedere (Monte Rosa, Ossola)," 16 08 2019. [Online]. Available: http://www.nimbus.it/ghiacciai/2019/190816_BelvedereCollassoMorene.htm. [Accessed 15 09 2024].

[38] LABMGF, "The Belvedere Glacier Project - Monitoring Project," Politecnico di Milano, [Online]. Available: https://thebelvedereglacier.it/pages/monitoring.html. [Accessed 15 09 2024].

[39] F. Gaspari, F. Ioli, F. Barbieri, S. Bonora, R. Fascia, L. Pinto and F. Migliaccio, "Bridging geomatics theory to real-world applications in alpine surveys through an innovative summer school teaching program," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.,* vol. XLVIII, no. 4/W12, pp. 59-66, 2024.

[40] F. Escobar, G. Hunter, I. Bishop and A. Zerger, *Introduction to GIS,* Melbourne: Department of Geomatics, The University of Melbourne, 2008.

[41] R. Steenson, "The history of Geographic Information Systems (GIS)," BCS, 25 04 2019. [Online]. Available: https://www.bcs.org/articles-opinion-and-research/the-history-of-geographic-information-systems-gis/. [Accessed 16 09 2024].

[42] N. Waters, "GIS: History," in *The International Encyclopedia of Geography*, Wiley Online Library, 2017, pp. 1-12.

[43] Library of Congress, "History of Geographic Information Systems," Library of Congress, [Online]. Available: https://guides.loc.gov/geospatial/computer-cartography-archive/history-of-gis. [Accessed 16 09 2024].

[44] N. Jatana, S. Puri, M. Ahuja, I. Kathuria and D. Gosain, "A Survey and Comparison of Relational and Non-Relational Database," *International Journal of Engineering Research & Technology,* vol. 1, no. 6, 2012.

[45] C. Codes, "ELI5: Object-Relational Mapping," Medium, 12 07 2020. [Online]. Available: https://candacecodes.medium.com/eli5-object-relational-mapping-b91f71f13a88. [Accessed 16 09 2024].

[46] QGIS, "Spatial Analysis (Interpolation)," QGIS, 14 10 2024. [Online]. Available: https://docs.qgis.org/3.34/en/docs/gentle_gis_introduction/spatial_analysis_interpolation.html. [Accessed 19 10 2024].

[47] M. Hugentobler, "Quantum GIS," in *Encycolpedia of GIS*, Boston, MA, Springer, 2008, pp. 935-939.

[48] QGIS, "QGIS User Guide - Foreword," QGIS, 15 09 2024. [Online]. Available: https://docs.qgis.org/3.34/en/docs/user_manual/preamble/foreword.html. [Accessed 16 09 2024].

[49] QGIS, "QGIS User Guide - QGIS Python Console," QGIS, 16 09 2024. [Online]. Available: https://docs.qgis.org/3.34/en/docs/user_manual/plugins/python_console.html. [Accessed 16 09 2024].

[50] QGIS, "PyQGIS Developer Cookbook - Introduction," QGIS, 16 09 2024. [Online]. Available: https://docs.qgis.org/3.34/en/docs/pyqgis_developer_cookbook/intro.html. [Accessed 16 09 2024].

[51] QGIS, "QGIS plugin web portal," QGIS, [Online]. Available: https://plugins.qgis.org/. [Accessed 16 09 2024].

[52] QGIS, "QGIS Python API," QGIS, 06 10 2024. [Online]. Available: https://qgis.org/pyqgis/3.34/. [Accessed 08 10 2024].

[53] QGIS, "PyQGIS Developer Cookbook - Structuring Python Plugins," QGIS, 07 10 2024. [Online]. Available: https://docs.qgis.org/3.34/en/docs/pyqgis_developer_cookbook/plugins/plugins.html. [Accessed 08 10 2024].

[54] Python, "What is Python? Executive Summary," Python, 2024. [Online]. Available: https://www.python.org/doc/essays/blurb/?external_link=true. [Accessed 16 09 2024].

[55] G. Van Rossum and F. Drake, An Introduction to Python, Bristol: Network Theory Limited, 2003.

[56] Qt, "About Qt," Qt Wiki, 18 07 2022. [Online]. Available: https://wiki.qt.io/About_Qt. [Accessed 16 09 2024].

[57] The Qt Company, "PyQt Documentation v5.15.7 - Introduction," Riverbank Computing Ltd, 2023. [Online]. Available: https://www.riverbankcomputing.com/static/Docs/PyQt5/introduction.html. [Accessed 2024 10 08].

[58] The Qt Company, "Qt Documentation - SIgnals & Slots," The Qt Company, 2024. [Online]. Available: https://doc.qt.io/qt-6/signalsandslots.html. [Accessed 08 10 2024].

[59] The Qt Company, "Qt for Python - Signals and Slots," The Qt Company, 2024. [Online]. Available: https://doc.qt.io/qtforpython-6/tutorials/basictutorial/signals_and_slots.html. [Accessed 2024 10 08].

[60] QGIS, "QGIS Python Plugins Repository - PluginBuilder3," QGIS, 28 06 2019. [Online]. Available: https://plugins.qgis.org/plugins/pluginbuilder3/#plugin-versions. [Accessed 08 10 2024].

[61] g-sherman, "Qgis Plugin Builder," GitHub, 2019 06 28. [Online]. Available: https://github.com/g-sherman/Qgis-Plugin-Builder. [Accessed 08 10 2024].

[62] QGIS, "QGIS Python Plugins Repository - PluginReloader," QGIS, 05 10 2024. [Online]. Available: https://plugins.qgis.org/plugins/plugin_reloader/#plugin-versions. [Accessed 08 10 2024].

[63] borysiasty, "plugin_reloader," GitHub, 05 10 2024. [Online]. Available: https://github.com/borysiasty/plugin_reloader. [Accessed 08 10 2024].

[64] The Qt Company, "Qt Documentation - QTabWidget Class," The Qt Company, 2022. [Online]. Available: https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QTabWidget.html. [Accessed 16 10 2024].

[65] The QtCompany, "QToolButton," The QtCompany, 2022. [Online]. Available: https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QToolButton.html#qtoolbutton. [Accessed 16 10 2024].

[66] The Qt Company, "QDialogButtonBox," The QtCompany, 2022. [Online]. Available: https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QDialogButtonBox.html#qdialogbuttonbox. [Accessed 16 10 2924].

[67] The Qt Company, "QCheckBox," The Qt Company, 2022. [Online]. Available: https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QCheckBox.html#qcheckbox. [Accessed 16 10 2024].

[68] The Qt Company, "QLabel," The Qt Company, 2022. [Online]. Available: https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QLabel.html#qlabel. [Accessed 16 10 2024].

[69] The Qt Company, "QLineEdit," The Qt Company, 2022. [Online]. Available: https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QLineEdit.html#qlineedit. [Accessed 16 10 2024].

[70] The Qt Company, "QComboBox," The Qt Company, 2022. [Online]. Available: https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QComboBox.html#qcombobox. [Accessed 16 10 2024].

[71] The Qt Company, "QPlainTextEdit," The Qt Company, 2022. [Online]. Available: https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QPlainTextEdit.html#qplaintextedit. [Accessed 16 10 2024].

[72] QGIS, "Class: QgsMapLayerComboBox," QGIS, 14 10 2024. [Online]. Available: https://qgis.org/pyqgis/3.38/gui/QgsMapLayerComboBox.html. [Accessed 16 10 2024].

[73] QGIS, "Class: QgsFieldComboBox," QGIS, 14 10 2024. [Online]. Available: https://qgis.org/pyqgis/master/gui/QgsFieldComboBox.html. [Accessed 16 10 2024].

[74] QGIS, "Class: QgsFeaturePickerWidget," QGIS, 14 10 2024. [Online]. Available: https://qgis.org/pyqgis/master/gui/QgsFeaturePickerWidget.html. [Accessed 16 10 2024].

[75] QGIS, "Class: QgsSpinBox," QGIS, 14 10 2024. [Online]. Available: https://qgis.org/pyqgis/master/gui/QgsSpinBox.html. [Accessed 16 10 2024].

[76] QGIS, "Class: QgsDoubleSpinBox," QGIS, 16 10 2024. [Online]. Available: https://qgis.org/pyqgis/master/gui/QgsDoubleSpinBox.html. [Accessed 16 10 2024].

[77] QGIS, "QGIS Python API - Class: QgsRasterLayer," QGIS, 06 10 2024. [Online]. Available: https://qgis.org/pyqgis/master/core/QgsRasterLayer.html. [Accessed 14 10 2024].

[78] The Qt Company, "PyQt - QMessageBox," The Qt Company, 2022. [Online]. Available: https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QMessageBox.html. [Accessed 14 10 2024].

[79] QGIS, "QGIS User Guide - 27.7 Using processing algorithms from the console," QGIS, 14 10 2024. [Online]. Available: https://docs.qgis.org/3.34/en/docs/user_manual/processing/console.html. [Accessed 14 10 2024].

[80] QGIS, "QGIS Python API - Class:QgsRasterCalculator," QGIS, 14 10 2024. [Online]. Available: https://qgis.org/pyqgis/master/analysis/QgsRasterCalculator.html. [Accessed 14 10 2024].

[81] QGIS, "PyQGIS Developer Cookbook - 3 Loading Layers," QGIS, 14 10 2024. [Online]. Available: https://docs.qgis.org/3.34/en/docs/pyqgis_developer_cookbook/loadlayer.html. [Accessed 13 10 2014].

[82] QGIS, "QGIS Python API - Class:QgsRasterBandStats," QGIS, 14 10 2024. [Online]. Available: https://qgis.org/pyqgis/master/core/QgsRasterBandStats.html. [Accessed 14 10 2024].

[83] QGIS, "Class: QgsInterpolator¶," QGIS, 14 10 2024. [Online]. Available: https://qgis.org/pyqgis/master/analysis/QgsInterpolator.html. [Accessed 16 10 2024].

[84] QGIS, "Class: QgsGraduatedSymbolRenderer," QGIS, 14 10 2024. [Online]. Available: https://qgis.org/pyqgis/3.38/core/QgsGraduatedSymbolRenderer.html. [Accessed 16 10 2024].

[85] QGIS, "Class: QgsVectorFieldSymbolLayer," QGIS, 14 10 2024. [Online]. Available: https://qgis.org/pyqgis/3.38/core/QgsVectorFieldSymbolLayer.html. [Accessed 16 10 2024].

[86] F. Ioli, N. Dematteis, D. Giordan, F. Nex and L. Pinto, "Deep Learning Low-cost Photogrammetry for 4D Short-term Glacier Dynamics Monitoring," *PFG,* 2024.

[87] F. Ioli, C. De Gaetani, F. Barbieri, F. Gaspari, L. Pinto and L. Rossi, *Belvedere Glacier long-term monitoring Open Data,* Zenodo, 2024.

[88] F. Crameri, *Scientific Colour maps,* Zenodo, 2018.

[89] "Tab widgets in QGIS plugin - run different scripts," Gis StackExchange, 2020. [Online]. Available: https://gis.stackexchange.com/questions/341737/tab-widgets-in-qgis-plugin-run-different-scripts. [Accessed 20 09 2024].

[90] "How to use QgsFieldComboBox in building QGIS plugin," GIS StackExchange, 2022. [Online]. Available: https://gis.stackexchange.com/questions/435654/how-to-use-qgsfieldcombobox-in-building-qgis-plugin. [Accessed 20 09 2024].

[91] jakobgager, "QGIS 3.14.15: QgsMapLayerComboBox with enabled filters -> NoSuchClassError: Unknown C++ class: QgsMapLayerProxyModel," GitHub, 27 08 2020. [Online]. Available: https://github.com/qgis/QGIS/issues/38472. [Accessed 20 10 2024].

[92] "Apply a color ramp to vector layer using PyQGIS3," Gis StackExchange, 2020. [Online]. Available: https://gis.stackexchange.com/questions/342352/apply-a-color-ramp-to-vector-layer-using-pyqgis3. [Accessed 25 09 2024].

[93] "Adding QgsLayoutItemTextTable to Print Layout," Qgis StackExchange, 2019. [Online]. Available: https://gis.stackexchange.com/questions/330120/adding-qgslayoutitemtexttable-to-print-layout. [Accessed 26 09 2024].

[94] "How to change page orientation and number of pages of print composer layout using pyqgis in QGIS 3.4?," GIS StackExchange, 2019. [Online]. Available: https://gis.stackexchange.com/questions/325360/how-to-change-page-orientation-and-number-of-pages-of-print-composer-layout-usin. [Accessed 27 09 2024].

[95] "Setting width for label.adjustSizeToText() in PyQGIS," GIS StackExchange, 2023. [Online]. Available: https://gis.stackexchange.com/questions/459233/setting-width-for-label-adjustsizetotext-in-pyqgis. [Accessed 30 09 2024].

[96] "Adding north arrow using PyQGIS," GIS StackExchange, 2020. [Online]. Available: https://gis.stackexchange.com/questions/347026/adding-north-arrow-using-pyqgis. [Accessed 25 09 2024].

[97] geomatics, "PyQGIS Tutorial: Automating Map Layout," GEODOSE, 07 02 2022. [Online]. Available: https://www.geodose.com/2022/02/pyqgis-tutorial-automating-map-layout.html. [Accessed 20 09 2024].

[98] S. Fajardo Turner, "QGIS-Thesis," GitHub, 20 10 2024. [Online]. Available: https://github.com/sfajardot/qgis-thesis.

[99] LabMGF, "The Belvedere Glacier," Politecnico di Milano, [Online]. Available: https://thebelvedereglacier.it/potree/index.php. [Accessed 15 Septembeer 2024].

[100] D. J. Maguire, "An overview and definition of GIS," in *Geographical information systems: Principles and applications*, 1991, pp. 9-20.

8 List of Figures

9 List of Tables

10 Acknowledgments

The completion of this thesis was not achieved in isolation, and I am deeply grateful to the many individuals whose support, guidance, and encouragement made this work possible. First and foremost, I wish to extend my heartfelt thanks to **Professor Federica Migliaccio** and **Engineer Federica Gaspari** for inviting me to participate in this project. Their invaluable patience, expertise, and thoughtful feedback were instrumental throughout the entire process.

I am also deeply indebted to the entire team at the **Belvedere Glacier Summer School**, whose insights and provision of essential data were critical for the development and testing of this plugin. The opportunity to visit the glacier and participate in the surveying firsthand was an inspiring experience that motivated me to pursue this project with enthusiasm.

This work would not have been possible without the generous support of the **Invest Your Talent in Italy Program**, which provided the funding for my master's degree at **Politecnico di Milano**. I am sincerely grateful for their support.

I would also like to acknowledge **Dr. Roberto Ponce** and **Dr. Fabian Lozano** from **Tecnológico de Monterrey**, whose mentorship introduced me to the world of geospatial data sciences and set me on the path to where I am today. Their teaching and guidance have been invaluable to my academic journey.

I would like to extend my gratitude to my classmates and friends, particularly my flatmates at **Porpora 90**, for their invaluable help with editing, late-night feedback sessions, and constant moral support. Their encouragement and companionship played an essential role in my studies at **Politecnico di Milano** and the completion of this thesis.

Finally, I must express my deepest appreciation to my family, especially my parents, **Francisco Fajardo** and **Laura Turner**, and my brother, **Miguel Fajardo**. Their unwavering belief in me has been a source of strength and motivation throughout this journey, and I am truly grateful for their support.