TÉCNICO LISBOA

Georeferencing of Fire Front Aerial Images using Structure from motion and Iterative Closest Point

Francisco Goulão Sargento

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Prof. Alexandre José Malheiro Bernardino
Prof. Ricardo Adriano Ribeiro

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Prof. Ricardo Adriano Ribeiro
Member of the Committee: Prof. José Silvestre Serra Silva

December 2021

Dedicated to everyone who lost their lives fighting forest fires and to all who work so that fewer lives are lost.

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

My first words of gratitude go to my wonderful co-supervisors Alexandre Bernardino and Ricardo Ribeiro, who welcomed me into the FIREFRONT project and helped me plan and execute this work.

My words of gratitude also extend to the entire FIREFRONT team's members, whose influence and past work were pivotal for the making my own work.

To my sister, parents and grandparents, who helped me get where I am now and whose support was unwavering.

To my dear friends and colleagues Hugo Faustino, Ana Veiga, André Rocha, Sara Pessoa and Olívia Pinto, without whom my academic journey would have been longer and lonelier.

I would also like to acknowledge UAVision, whose forest fire monitoring work was key for optimizing my algorithm.

This work was made in the context of the FCT project FIREFRONT (PCIF/SSI/0096/2017).

Resumo

Esta tese propõe o uso de Sfm (*Structure-from-motion*) e de ICP (*Iterative Closest Point*) para georeferenciação em missões de patrulha de incêndio florestal, utilizando imagens capturadas por aeronaves não tripuladas. Sfm+ICP usa o video da câmara da aeronave, bem como os dados fornecidos pelo GPS e IMU (Unidade de medição inercial) para reconstruir uma nuvem de pontos 3D da área do desastre. A reconstrução Sfm é separada em dois passos: uma reconstrução esparsa usando *Speeded up robust features* (SURF) para estimar as poses da câmara ao longo do tempo, e uma reconstrução densa usando um rastreador de pontos *Kanade-Lucas-Tomasi* (KLT) inicializado com o algoritmo de valores próprios mínimos.

A reconstrução densa é depois registrada a um DEM (*Digital Elevation Model*) real da área à volta do fogo, e usada para obter as estimativas de georeferenciação, calculando a média das coordenadas 3D correspondentes aos *pixels* do alvo.

Este algoritmo foi validado em dois *datasets* artificiais criados no *Blender*, e dois *datasets* reais de aeronaves em missões de patrulha em incêndios florestais. Os resultados demonstram que Sfm+ICP consegue fazer reconstruções 3D fiéis, e, ao mesmo tempo, georeferenciar de forma precisa e exata numa situação de fogo florestal. Os resultados mostram também robustez a erros altos de IMU e GPS, tornando este algoritmo numa opção ágil para georeferenciação com UAVs com sensores pouco precisos.

Palavras-chave: fogo florestal, Veículo Aéreo Não Tripulado, structure from motion, georeferenciação, DEM, ICP

Abstract

This work proposes the use of Structure-from-motion (Sfm) and Iterative Closest Point (ICP) as a forest fire georeferencing algorithm to be used with images captured by an aerial vehicle. Sfm+ICP uses the real time video captured by an aircraft's camera, as well as its IMU and GPS measurements to reconstruct a dense 3D point cloud of the disaster area captured by the camera. The Sfm reconstruction is divided in two steps: a sparse reconstruction step using Speeded up robust features (SURF) for camera pose estimation, and a dense reconstruction step relying on a Kanade–Lucas–Tomasi (KLT) feature tracker initialized using the minimum eigenvalue algorithm.

This dense 3D reconstruction is then registered to a real Digital Elevation Model (DEM) of the surrounding area, thus refining the point cloud to better match the terrain. The reconstruction is then used as the basis of the georeferencing estimates, as any target's location can be estimated by averaging the 3D coordinates corresponding to its nearby pixels.

The algorithm was validated with two artificial Blender datasets and two real forest fire monitoring videos. The results demonstrate that Sfm+ICP can perform accurate 3D reconstructions while also georefering several targets in a forest fire environment. The results also show the algorithm is robust to high IMU and GPS errors, making it a far better option than optic-ray-based georeferencing for UAVs with unreliable telemetry.

Keywords: forest fire, UAV, structure from motion, georeferencing, DEM, ICP

Contents

List of Tables

List of Figures

List of Symbols

Greek Symbols

λ	scale factor.

Roman Symbols

c	principal point.
d	distortion parameter.
EP	extrinsic parameters.
f	focal length.
IP	intrinsic parameters.
P	projection matrix.
V	reprojection error.
x	horizontal pixel coordinates.
X	3D east coordinates.
y	vertical pixel coordinates.
Y	3D north coordinates.
Z	3D elevation coordinates.

Nomenclature

ASTER Advanced Spaceborne Thermal Emission and Reflection Radiometer

BA Bundle Adjustment

DEM Digital Elevation Model

EP Extrinsic parameters

GCP Ground Control Point

GNSS Global Navigation Satellite System

GPS Global Positioning System

ICP Iterative Closest Point

IMU Inertial Measurement Unit

IP Intrinsic parameters

IRT Iterative Ray Tracing

KLT Kanade Lucas Tomasi

LIDAR Light Detection and Ranging

MinEigen Minimum Eigeinvalue

MRE Mean Reprojection Error

MVT Multi-View Triangulation

NED North-East-Down

ORB Oriented FAST and rotated BRIEF

RANSAC RANdom SAmple Consensus

RMSE Root Mean Square Error

Sfm Structure from motion

SIFT Scale Invariant Feature Transform

SLAM Simultaneous Localization and Mapping

SRTM Shuttle Radar Topography Mission

SURF Speeded Up Robust Features

UAV Unmanned Aerial Vehicle

Chapter 1

Introduction

Contents

In recent years, the ever decreasing cost of Unmanned Aerial Vehicles (UAVs) has sparked major scientific interest in remote sensing and disaster monitoring. UAVs are now used for forest monitoring [1], coastal geohazard assessments [2] [3] and forest fire monitoring [4].

One of the major problems any remote sensing application must solve is Georeferencing: relating an image's internal coordinate frame to a ground system of geographic coordinates. Most applications use a direct georeferencing method, as the one already developed for the FIREFRONT project [5], in which Iterative Ray-Tracing is used.

Remote sensing is a set of techniques to obtain information about objects or areas from a distance, which is often performed using UAVs with a camera or other types of sensors, such as Light Detection And Ranging (LIDAR). Most UAVs used in remote sensing carry a GPS, a camera and an IMU to perform georeferencing, however, payload constraints force them to only carry small and error-prone IMUs, which significantly hinders these systems' georeferencing potential. In forest fire monitoring systems, this is compounded by an extremely demanding environment: heavy smoke, strong wind and a dynamic scenario.

Georeferencing is the process of assigning locations to geographical objects within a geographic frame of reference. In practice this means associating a set of 2D pixel coordinates from one image to a set of 3D world coordinates, most often expressed in latitude, longitude and altitude. There are two main methods to solve the georeferencing problem: direct and indirect georeferencing.

On the one hand, direct georeferencing uses navigational information and the camera to determine a target's geographic coordinates. Most direct georeferencing methods try to solve a single-ray back-projection problem: the process of determining the ground coordinates of pixels in a single aerial image with the support of a DEM.

On the other hand, indirect georeferencing methods do not rely on navigational information, instead,

they register photographs to georeferenced data, such as a DEM or satellite images. This registration can be done by directly comparing a photograph to a satellite image, but most state of the art algorithms process the image before the registration. For example, Structure from motion (Sfm) constructs a 3D point cloud from several sequential images, which can then registered to a DEM. Note that not all Sfm algorithms are examples indirect georeferencing, as some do rely on navigational data, but most do not.

The georeferencing method proposed in this work is designed to operate in medium altitude forest fire monitoring operations, using an UAV with a camera, GPS and IMU. The camera can be oriented remotely to track a fire, while the GPS and IMU provide camera pose estimates. The UAV's mission profile involves a loiter around the fire in which the algorithm is meant to georefer several ground targets chosen by the operator.

1.1 Motivation

Forest fires have reached unprecedented figures in Portugal. In the 1980's, 75000ha of Portuguese land burned in forest fires. That number increased to 100000ha in the 1990's and 150000ha in the 2000's, and that trend shows no signs of slowing down, with a series of socioeconomic and climate change related effects contributing to an ever increasing fire risk [6].

While downward economic cycles, budget scarcity, rural depopulation and forest land mismanagement have exacerbated Portugal's fire risk, other southern European nations have also been affected by forest fires. In the 2010's, over 3 million hectares of European land were burned as a result of forest fires. Whilst the main victims were Portugal and the Mediterranean countries of Spain, Italy, France and Greece, climate change will continue to cause rising temperatures and decreasing rainfall, which will likely increase the length and severity of the fire season [7]. Additionally, high emissions climate models predict that by 2070 most of Central Europe may also experience regular and lengthy fire seasons [8].

UAVs are relevant components of modern firefighting operations, which will help us fight the next waves of fire seasons. UAVs possess rapid maneuverability, extended operational range, improved personal safety and cost efficiency, when compared to other remote sensing solutions, making them particularly useful in fire monitoring and detection, given their ability to perform fire search, confirmation and observation [4]. Georeferencing algorithms are a critical aspect of these remote sensing systems: by locating a fire quickly and accurately, fire monitoring systems can rely on quality data to be used by fire propagation models and firefighting authorities, saving lives and property.

1.2 Objectives

This thesis entails two main objectives: the main objective is to develop an efficient and accurate georeferencing algorithm for use in a forest fire scenario, using a medium-altitude aerial vehicle equipped with a camera, GPS and an IMU.

The second objective is to design a robust Sfm (Structure from motion) approach that can be used, not only in DEM matching-based georeferencing, but also in other applications: such as disaster recon-

struction and as a first layer of an IRT georeferencing algorithm. Hence, the Sfm block must be thorough enough to build a detailed view of the scene, whilst also being fast and efficient so as to be used to quickly refine the camera's EP.

1.3 Thesis Outline

This work is structured as follows:

- Chapter 2 provides a background on the main theoretical aspects mentioned throughout this document, describes the current state of the art of georeferencing algorithms and finally presents the proposed approach.
- Chapter 3 explains the algorithm's implementation.
- Chapter 4 presents the results obtained in the simulations and experiments used to validate the algorithm.
- Chapter 5 concludes this work by listing its achievements and providing future work ideas.

Chapter 2

Background & State of the Art

Contents

This chapter is comprised of three main parts: section 2.1 provides a short theoretical background over the major concepts explored in this work. Section 2.2 presents and reviews the state of art georeferencing approaches relevant to disaster monitoring using UAVs. Lastly, sections 2.3 and 2.4 present the proposed approach and its contributions to the fields of aerial georeferencing and disaster monitoring.

2.1 Theoretical principles

The subsections below present a quick introduction to the main theoretical principles revisited across this document.

2.1.1 Pinhole camera model

The camera is the main sensor used in any Structure from motion project: The accuracy and consistency of the 3D reconstructions hinges on a precise knowledge of the camera's intrinsic parameters [9], which mathematically express the pixel coordinates of any 3D point captured by the camera. Bundle adjustment and point cloud registration may be able to greatly increase a reconstruction's accuracy, but only if the camera model is sound [10].

The camera model used in this work is the pinhole camera model, chosen for its simplicity and efficiency. This model is a simplification of a real camera in which there is no lens and the camera aperture is approximated by a single point, the pinhole [11].

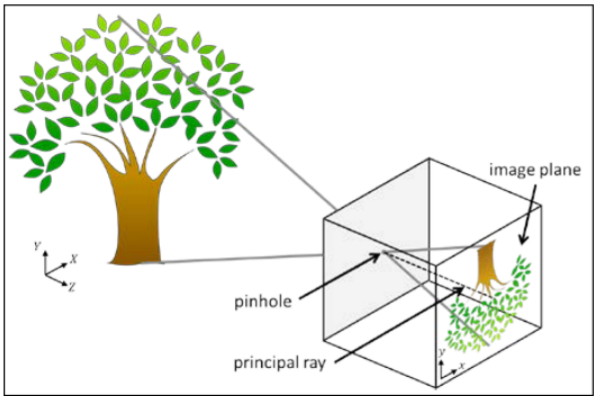


Figure 2.1: Pinhole camera model

The mathematical formula (presented by Hartley and Zisserman in [12]) that describes how a 2D pixel with coordinates (x,y) can be inferred from a 3D point with coordinates (X,Y,Z) is

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{2.1}$$

λ is the scale parameter , f_x and f_y are the horizontal and vertical focal lengths, c_x and c_y are the optical center coordinates expressed in pixels. The 3x3 matrix is often called the camera's intrinsic parameter matrix, since it expresses the linear relation between the 3D world (in the camera's reference frame) and the 2D pixel coordinates. Because these parameters are specific to the camera hardware (hence the name intrinsic), we need to calculate them only once in the lifetime of a camera[12]. This is called camera calibration.

Equation 2.1 is often a good approximation of the camera's behaviour, however, it ignores the effects of the camera lens, which causes radial and tangential distortions. These distortions can be accounted using the following equation [12]:

$$\begin{bmatrix} x \\ y \end{bmatrix}_U = d_t + d_r \cdot \begin{bmatrix} x \\ y \end{bmatrix}_D \tag{2.2}$$

, where $[x, y]_U^T$ is the undistorted pixel, $[x, y]_D^T$ is the distorted pixel, and d_t and d_r are the tangential and

radial distortion parameters, respectively.

It is also worth mentioning that this mathematical relationship naturally only works one way: a 3D point can only be expressed by a single 2D pixel, however, a 2D pixel can be an infinity of 3D world points.

2.1.2 Structure from motion

Structure from motion is the process of estimating the 3D structure of a scene from a set of images. It can produce high quality, dense, 3D point clouds of a landform for minimal cost [10]. Sfm is already well established in fields such as archaeology and cultural heritage [9], and its use as a topographic survey technique has surged in recent years.

The Sfm problem can be formulated as:

Given: m images of n fixed 3D points.
Problem: Estimate the m projection matrices, P_i, and n 3D points, X_j, from the mn correspondences, x_{ij}, assuming the following camera model holds true:

$$x_{ij} = P_i X_j, \quad i = 1, ..., m \ , \ \ j = 1, ..., n, \tag{2.3}$$

where P_i are 3x4 matrices and X_j and x_{ij} are, respectively, 3D and 2D points in homogeneous coordinates.

Figure 2.2 shows a simple Sfm example with only 3 images. In real applications, not all the 3D points, X_j, are present in every single image, and the point correspondences, x_{ij}, contain outlier matches, however, the basic idea behind Sfm still holds true.

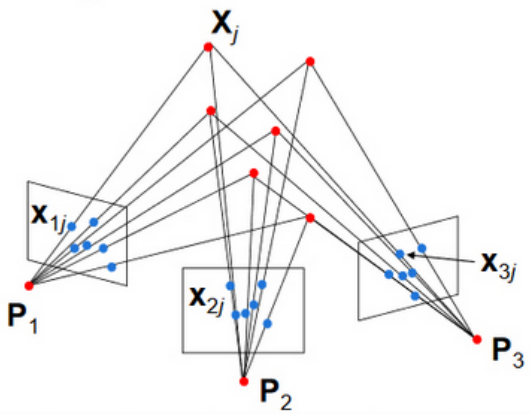


Figure 2.2: Structure from motion problem [12]

Sfm is not a single technique, it is a workflow, employing multiple algorithms developed from 3D computer vision, traditional photogrammetry and more conventional survey techniques. That being said, most published Sfm implementations follow the same 5 step process [10]:

1. **Keypoint detection:** The first step is to detect keypoints in all the m images. Each keypoint is a specific pixel that contains a distinct feature.

2. **Keypoint correspondence:** The keypoints are matched across all images, in order to find the correspondences x_{ij}.

3. **Multi-view triangulation:** MVT is used to initially estimate the 3D points, X_j, and the projection matrices, P_i.

4. **Bundle adjustment:** BA refines the initial X_j and P_i estimates provided by the previous step. Some Sfm applications iterate MVT and BA to converge on a better 3D reconstructions, while others do this just once.

5. **Linear similarity transformation:** The X_j estimates provided by the BA are expressed in an arbitrary reference frame relative to the camera. Hence, Sfm's final step is to scale, translate and rotate the 3D points, X_j, to get a reconstruction expressed in N-E-D reference frame.

At large scales, and when working with a physical environment, three main parameters have been shown to affect the quality of the 3D model recovered by a Sfm algorithm:

- **Lighting conditions:** glare and variable contrast across a scene can negatively impact point matching. This can be partially solved by using robust feature detectors and a rigorous keypoint matching scheme [13].

- **Changes in shadow length or albedo due to solar positioning:** these dynamic changes can severely reduce the quality and amount of feature matches. Other natural dynamic changes include: vegetation oscillation due to wind, smoke and fire. This effect can be easily minimized by limiting each survey to less than 30 minutes [13].

- **Errors in the position and orientation measurements:** a small error in the pose of the aircraft can cause a large bias in the subsequent reconstruction. However, since the relative reconstruction itself is consistent to its arbitrary reference frame, this bias can be removed by registering the 3D results with a known relief map of the scene.

2.1.3 Bundle adjustment

Bundle adjustment is the problem of refining a visual reconstruction to produce a jointly optimal 3D structure and viewing parameters (camera poses and/or calibration) estimates [14]. While Bundle Adjustment has been used in Aerial photogrammetry since the 1950s, its popularity has resurfaced in the past decades, mainly due to its ability to improve the accuracy and consistency of 3D reconstructions. The Bundle adjustment problem is classically formulated as a nonlinear least squares problem, in which the solution is found by minimizing the reprojection error of each feature point.

The idea behind the reprojection error is fairly straightforward: a feature point is a set o pixels in multiple images that correspond to the same 3D feature, whose 3D location can be estimated using a triangulation algorithm. The pinhole camera model (discussed in 2.1.1) can be used to determine the pixel coordinates of any 3D point , including the estimated 3D feature point locations. In real applications, the triangulation process will yield imperfect 3D location estimates, whose accuracy can be evaluated

by reprojecting the estimated 3D point to the image and comparing the reprojected pixels with the initial feature point pixels, hence the name reprojection error.

Table 2.1: Pros and cons of Bundle adjustment [14]

Pros	Cons
Flexibility: Bundle adjustment handles a variety of feature, scene and camera types, as well as different error models for minimization.	**Initialization:** Bundle adjustment relies on a good initial estimate. If this initialization contains a large amount of outliers, the Bundle adjustment will not produce a good result.
Accuracy: It is statistically optimal (assuming Gaussian noise), giving good results even for low quality reconstructions.	**Problem size:** Large problems may have millions of unknown variables, which is challenging even for the best of sparse equation solvers.
Efficiency: The Bundle Adjustment problem's structure enables sparse linear equation solvers to greatly outperform general-purpose optimization routines.	**Error accumulation:** In long reconstructions, error will accumulate over time, requiring the use of special types of Bundle Adjustment, which increase the algorithm's complexity.

In mathematical terms, the reprojection error $V_{x'_{ij}}$, of the point i in image j, is expressed by

$$x'_{ij} + V_{x'_{ij}} = \hat{\lambda}_{ij}\hat{P}_j(x_{ij}, IP, EP)\hat{X}_i \tag{2.4}$$

where x_{ij} is the real position of point i in image j, in pixels, IP and EP represent the camera's intrinsic and extrinsic parameters, respectively, and $\hat{\lambda}_{ij}$ is the scene's scaling factor, which can be either minimized by the Bundle adjustment, or remain fixed. Finally, $\hat{P}_j$ is the camera's projection matrix, which converts 3D points into 2D pixel coordinates in image j, $\hat{X}_i$ is the initial guess of the 3D position of point i, and x'_{ij} is the 2D pixel location of point i in image j, if we assume that $\hat{X}_i$ is correct.

This results in a very large number of unknown parameters to be minimized by the bundle adjustment:

- The 3D locations of all the points $\hat{X}_i$
- The 1D scale vector $\hat{\lambda}_{ij}$, which may vary image to image.
- The 6D exterior pose of the camera for each frame, given by the Extrinsic parameters (EP)
- The 5D intrinsic parameters of the camera are sometimes optimized, however this is often ignored in real time applications due to time constraints.

This high number of unknowns for each point/pixel can often lead to a total of millions of unknowns for a reconstruction with just a few dozen images. Notwithstanding, this linear system is very sparse, as not all the points are visible in all the images, and some unknowns are constant for each point, such as the intrinsic parameters.

There are several notable variations of Bundle adjustment, which seek to improve its efficiency in specific aspects:

- **Local Bundle adjustment:**

Local Bundle adjustment divides the image sequence, performing the minimization in smaller batches [15], instead of in one large global minimization, increasing the algorithm's time efficiency.

- **Incremental Bundle adjustment:**

  In Incremental Bundle adjustment, the minimization is performed right after each image is processed [16]. This is often coupled with Local Bundle adjustment, and is widely used in real time applications.

- **Constrained Bundle adjustment:**

  Constrained Bundle adjustment minimizes not only the reprojection error of each point, but also another relevant error variable, such as the GPS or gymbal error [17]. This method greatly increases the algorithm's accuracy, particularly the scale factor and camera's extrinsic parameters, if the navigational data is reliable enough.

2.1.4 Digital Elevation Model (DEM)

Digital Elevation Models are digital representations of elevation data to represent terrain. Most DEMs are represented as a regular grid with either Latitude and Longitude in degrees, or North and East in metres, and in which each grid's square has a certain elevation value.

There are two types of DEM: Digital Surface Models (DSM) represent the earth's surface, including buildings and trees and are mostly used in urban planning and visualization applications, while Digital Terrain Models (DTM) only represent the bare ground, and are amply used for land use studies, flood prevention, geological applications and planetary science. The DEMs mentioned in this work are all classified DTMs, however, they do include trees in their elevation data, since the Radar and satellite images used to create them cannot penetrate dense tree canopies [18].

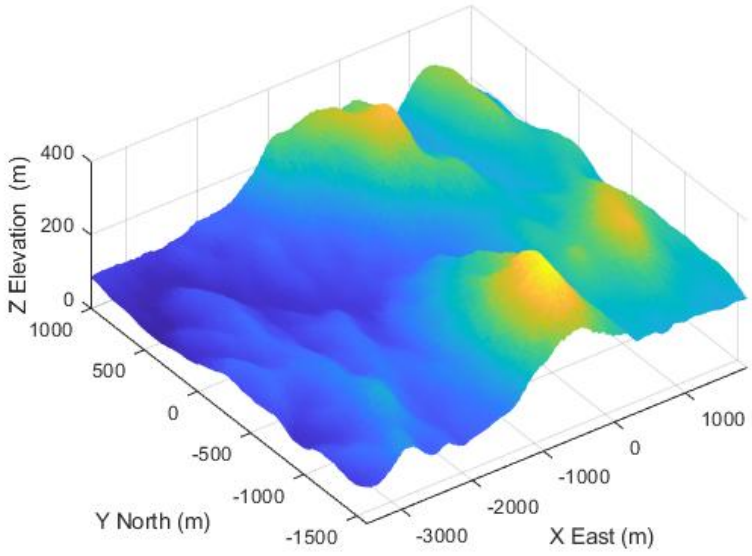


Figure 2.3: EU-DEM v1.1 around Serra da Archeira, Portugal

DEMs were traditionally created from existing topographic maps and aerial photos, however, most recent DEMs are generated using synthetic-aperture radar usually mounted on a satellite, or from even newer techniques such as LIDAR.

Europe's most accurate DEM is the EU-DEM v1.1 [19] which was made by merging elevation data from the Shuttle Radar Topography Mission (SRTM) and the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) DEMs. SRTM was generated using synthetic-aperture radar while ASTER was made from satellite images. EU-DEM v1.1 combines the perks of both ASTER and SRTM, resulting in a 25m resolution DEM with a 7m RMSE of most of Europe's countries.

DEMs are invaluable for most types of georeferencing algorithms: both direct and indirect methods rely on them, however, they do have some limitations in forest fire applications:

- The 25m resolution of the EU-DEM is remarkable, but in practice this means that any method relying on a DEM will struggle to have a better horizontal accuracy than 25m.

- Radar and satellite imagery cannot penetrate dense forests, but they can do that in areas with sparser forests, such as some which are planted for industrial use. This leads to situations where dense forests have higher elevations than sparser ones, which can be serious since, for example, eucalyptus reach heights between 10 and 40 metres.

2.1.5 Iterative Closest Point (ICP)

ICP is a point cloud registration framework used to find a local transformation to align two sets of points. Each step of the ICP algorithm attempts to align the two point clouds closer and closer, converging on a rotation matrix and translation vector that minimize the average distance between the two point clouds [20]. The ICP algorithm is based on a simple four step iterative process, designed to sequentially bridge the gap between the fixed and moving point clouds:

1. Select points on one point cloud.

2. Find the closest points on the other point cloud. This point cloud is called the ”fixed point cloud”, since it acts as a static reference.

3. Minimize the sum of the distances between the two sets of points. This produces a new point cloud more similar to the fixed one than in the previous iteration.

4. Repeat the previous 3 steps until the the two point clouds are similar enough.

Pros	Cons
Simple: ICP is easy to understand and implement.	**Initialization:** ICP relies on a good initial guess, and may fail to produce a good transformation if there are too many outliers or if the initial guess is too wrong.
Versatility: It is highly modifiable, allowing it to fit into diverse problems and solutions.	**Unpredictability:** Given it iteratively minimizes the distance error, the ICP's speed is not known *a priori*, instead the user has to specify a minimum tolerance for the algorithm to stop.
Accuracy: ICP performs well if a good initial guess is available and there are few outliers.	

Table 2.2: Pros and cons of Iterative closest point [21]

There are many ICP variants: most address the most difficult aspect of ICP, which is data association. Data association has a large impact on the convergence and speed of the algorithm, and is especially crucial in large and sparse point clouds with few features. There are also variants that change the minimization metric employed, point-to-plane being the most widely used [21]. Others perform different types of point sub-sampling to increase the algorithm's speed and accuracy. Finally, correspondence weighting is sometimes used to give more importance to certain points or features.

2.2 State of the Art

This subsection enumerates and analyzes the different aerial georeferencing methods. Georeferencing can be summed as the problem of aligning geographic data to a known coordinate system so it can be viewed, queried and analyzed.

2.2.1 Optic-ray surface intersection

Optic-ray surface intersection is the most efficient and versatile georeferencing method. Optic-ray-based algorithms perform georeferencing by solving a single-ray back-projection problem: the process of determining the ground coordinates of pixels in a single aerial image with the support of a DEM. There are three main methods to solve the single-ray back-projection problem, which are quantitatively evaluated in [22]:

- The **Iterative Photogrammetric** method is based on the inverse collinearity equations. It is the most computationally efficient method and is able to produce precise coordinates for simple surfaces, however, it suffers from divergence and occlusion induced problems in more complicated scenarios.
- The **Ray Tracing** method computes the coordinates by intersecting the view ray with the surface. It is very popular in computer graphics and it is the most accurate of the three methods. However, it is the most computationally intensive by far.

- **Iterative Ray Tracing** finds the intersection point by extending the view ray towards the surface by a certain step at a time, until it hits the surface. The IRT is intermediate in terms of efficiency and can produce coordinates with satisfactory precision

For these reasons, Iterative Ray Tracing (IRT) is the algorithm most widely used in real-time direct georeferencing applications. IRT is used in [23] for direct georeferencing in an agricultural application using a small low altitude drone. This technique achieves a high accuracy in low altitude flight, but the position and orientation sensor's errors are not accounted for, making this approach unscalable and ineffective for medium altitude forest fire monitoring.

Santana [5] proposes the use of the IRT in conjunction with an unscented transform to both estimate the target's position, and its uncertainty. This approach is better suited for medium altitude georeferencing, having an accuracy in the dozens of meters. This error is mostly due to sensor imperfections, which makes it difficult to decrease by purely algorithmically means.

Xu [24] proposes an electro-optical stabilization and tracking platform which integrates the camera with the UAV's navigational data, in order to stabilize the camera's orientation. Xu also proposes a CKF, a non-linear Kalman filter. These techniques greatly improve the georeferencing accuracy of the direct algorithm, however, they also increase its implementation cost, complexity and computation time.

In conclusion, optic-ray-based georeferencing methods are computationally efficient and achieve a satisfactory accuracy when fed with good quality navigational data. Some methods are able to account for this uncertainty, but none can reduce it *a posteriori*, which means their accuracy degrades quickly with altitude and orientation bias in particular.

2.2.2 Structure from motion

Sfm, which was initiated by the computer vision research community, has now been widely used for automated triangulation of overlapping UAV-based frame imagery. Sfm is a set of computer vision algorithms that facilitate the photogrammetric reconstruction of 3D scenes from images alone. It has gained popularity in recent years due to its ability to deal with sets of unordered and heterogeneous images without prior or accurate knowledge of the camera's intrinsic and extrinsic parameters.

Most of Sfm's variation happens in three aspects [9]:

- **Camera pose estimation**, which is usually performed with a sparse feature extractor, such as SURF, SIFT or ORB, followed by a sparse bundle adjustment to refine the initial camera pose estimations. Some approaches perform pose estimation and reconstruction at the same time, while others do them in two separate steps.

- **Exterior image parameters estimation**, which is employed to recover 3D points from the images. This was traditionally done with Ground Control Points (GCPs), however, most recent applications use GPS to perform this step [13].

- **Bundle adjustment**, which is the most computationally intensive part of the Sfm algorithm. Mouragnon [15] introduces a local bundle adjustment allowing 3D points and camera poses to be refined si-

multaneously through the sequence, which allows Sfm to run much faster. Lhullier [17] introduces a constrained bundle adjustment that minimizes both the reprojection error and the position error, using GPS data in real time. This fusion improves BA's accuracy on long image sequences, and increases the consistency of incremental BA. Incremental BA is also proposed by Lhullier in [25] as a way to reduce its computation time by dividing it in several minimization problems. This greatly improves Sfm's computation speed, but at the cost of decreased point cloud consistency.

Sfm is an agile algorithm, meaning it can easily be customized to fit a specific set of requirements. Forlani [26] uses it to perform indirect landing strip georeferencing under poor GNSS satellite coverage. Their approach achieves a satisfactory accuracy, but it is clear that either ground control points, or GNSS data is needed to improve a Sfm algorithm's accuracy. Sfm is used for forest remote sensing in [1] and [27]. These papers prove that Sfm can be used for medium altitude forest remote sensing with meter level accuracy and LIDAR equivalent point cloud density. They also prove that it can be run in real time, and perform georeferencing and data collection at the same time.

Notwithstanding, Sfm faces some obstacles in forest environments, as is highlighted by Iglhaut et al. [13]: feature extractors perform poorly in forests, where the scene is often dynamic, and sudden illumination changes are commonplace. All of these facts combine to make Sfm a tough proposition in wild forest environments.

2.2.3 Registration with georeferenced imagery

Image registration is the process of overlaying two images of the same scene taken at different times, from different viewpoints and with different sensors. The papers in this subsection use this process to register images taken by an UAV with georeferenced satellite/aerial images, and use the result of that registration to perform georeferencing.

Conte [28] proposes a correlation-based image registration approach. Correlation-based methods are efficient and can be applied to areas with no obvious landmarks, such as forests. However, these methods are not as accurate as pattern-based ones, being very sensitive to scene changes, such as smoke, fire or tree canopy height differences.

Zhuo [29] uses SIFT in a pattern-based image registration approach, matching UAV images with satellite images. This approach achieves a decimeter level accuracy, however, it is not practical for medium altitude tilted camera platforms, especially when operating in dynamic environments, such as wild forest fires.

Lindsten et al. [30] use environmental classification to classify and match superpixels. This method is more robust to orientation errors, but not to dynamic environment changes. Notwithstanding, environmental classification outperforms classic feature detectors in rural scenes.

Li et al. [31] propose a fusion of infra-red and visible light images to improve the registration's robustness to low illumination and smoke. This is obviously useful for forest fire scenarios, but it suffers from the same implementation issues that plague other image registration methods.

Hamidi and Samadzadegan [32] use database matching techniques to refine the coarse initial al-

titude and position parameters of the camera derived from its navigational data. Image registration is used as a first refinement step, while a direct IRT method is used to perform the georeferencing step itself, taking advantage of the refined camera poses. This is an intriguing approach, which maximizes the benefits of direct and registration based georeferencing, highlighting that an indirect georeferencing algorithm can either be an improvement to a classic direct method, or help improve it.

In conclusion, image registration can be more accurate than any other method, if the specific implementation suits it. On the other hand, it is much slower and less flexible than direct georeferencing, requiring more algorithmic complexity, while not being able to handle dynamic scenarios, such as a wild forest fire.

2.2.4 DEM matching

DEM matching is a technique used to match two DEMs : one being the ”true” DEM, made by a regulatory body, such as the EU-DEM v1.1, and the other being a DEM recovered from a set of 3D points measured with a camera, LIDAR or Radar.

DEM matching was originally proposed to replace the need for GCPs for absolute orientation of perspective images. Most methods solve the problem of adjusting the parameters of a 3D similarity transform (scale, translation, rotation) to minimize the sum of squares of height differences.

Sim et al. [33] uses an interpolation of a 25 m DEM to achieve a metre level accuracy. Rodriguez [34] proposes a quick stereo reconstruction and the use of cliff maps to perform a quick DEM registration.

Map registration is also used in military applications, to replace GPS when it is jammed [35], and in extra-planetary navigation, where GNSS is not available. DEM matching performs well across survey scales and flight levels, making it useful in medium altitude forest fire remote sensing.

2.3 Proposed approach

This work’s main objective is to design a georeferencing algorithm to be used in forest fire monitoring applications as a part of the FIREFRONT project. The algorithm has access to the GPS, altimeter and IMU data, which estimate the camera’s location and orientation, as well as a sequence of images captured by the camera as it moves around the fire. The final input is a series of user-selected target pixels, which are specific image regions that the user wants to georefer.

The algorithm must be able to accurately georefer several targets in real time, furthermore, as high telemetry data errors are expected, the algorithm needs to either correct them or estimate the result’s uncertainty.

There is no optimal solution for this problem, as there are several limitations in all the state of the art medium altitude georeferencing algorithms presented in section 2.2. Optic-ray-based georeferencing is time efficient and accurate when fed with good quality telemetry data, however, it produces wildly innacurate results when this data is unreliable, which happens often in disaster monitoring operations. And while they can estimate a result’s inaccuracy, optic-ray-based methods cannot lower it *a posteriori*,

which is especially bad when a large bias is present in any measurement.

Image registration is robust to telemetry errors and produces better georeferencing results than any other method when used in low altitude areas with enough landmarks. Nonetheless, most forest fires happen in areas with few landmarks, and while low altitude UAVs certainly have a role to play in disaster monitoring, fire, smoke and wind make low altitude georeferencing impractical for this project.

Sfm is able to reconstruct the operational area while also performing georeferencing, however, its workflow is designed for offline georeferencing, as the accuracy of medium altitude real time methods is lower than for other georeferencing types. Notwithstanding, Sfm can be paired with DEM matching to improve a reconstruction's accuracy, and its inner workings are highly modifiable, enabling the development of a specific Sfm approach to a specific problem.

DEM matching is by far the less popular georeferencing method for civil uses. It can greatly outperform any other method in high altitude georeferencing, hence its extensive use in military applications. However, these applications use extremely expensive equipment, and while high altitude aircraft loitering is sometimes used in forest fire monitoring, it is performed at low speed, whilst DEM matching is more suited to high speed and high altitude missions. Anyhow, the main premise of DEM matching is still useful: that terrain maps can be used to improve a georeferencing estimate's accuracy.

This thesis' proposed approach can be classified as a Sfm+ICP georeferencing method. Sfm is used to reconstruct a 3D model of the fire area using techniques most similar to [1] and [25]. This reconstruction is then improved by matching it to a known DEM of the terrain around the aircraft, similarly to [33] but with a more flexible registration algorithm that matches the downsampled reconstructed point cloud and the upsampled DEM. This registration is in essence similar to DEM matching, but instead of matching DEMs, the algorithm matches high density point clouds, which is more computationally intensive, but yields better registration results.

Using Sfm allows the algorithm to densely reconstruct the operational area and consequently georefer dozens of targets at the same time with no increase in computation time. Using it in real time is also challenging due to some of its time intensive routines, thus the algorithm needs to use a simple and quick kind of Sfm that still guarantees high georeferencing accuracy . On the other hand, using the ICP makes this algorithm robust to telemetry errors, even if they are exceedingly large.

2.4 Contributions

This work's main contribution is a georeferencing algorithm designed for medium altitude forest fire monitoring operations. It has a comparable accuracy to state of the art algorithms, while also being more robust to telemetry errors than other direct georeferencing methods.

The Sfm half has two Sfm blocks: one initial sparse one to estimate the camera's pose throughout the image sequence, using SURF features and a single bundle adjustment, and a second dense reconstruction to build a dense point cloud of the surrounding terrain, using a KLT point tracker with low quality MinEigen features. This setup allows Sfm+ICP to perform well even with low resolution and shaky footage captured above 900m altitude.

The ICP half of the algorithm is designed to register a point cloud spanning 50-300ha to a reference DEM. While there is no novelty in the ICP, its use with Sfm is new in the field of georeferencing.

Chapter 3

Implementation

Contents

This chapter provides a deep explanation over all phases and aspects of the proposed implementation. Section 3.1 details the algorithm's inputs and outputs. Section 3.2 presents a brief overview of the core blocks of the georeferencing algorithm. Sections 3.3 and 3.4 focus on the individual parts of the algorithm: Structure from motion and Registration. Finally, section 3.5 defines the metrics used to optimize and evaluate the algorithm.

3.1 Inputs and outputs

The algorithms' inputs are:

- **Images**:

  Being primarily a Sfm implementation, this georeferencing algorithm's main input is a set of sequential images. In fact, the extrinsic measurements are not strictly necessary for the algorithm to work, they just increase its accuracy and convergence time.

  The algorithm was designed to work with video frames sampled at a regular time interval, however, it can also work with regular aerial photos or with images taken by human observers in high

elevation areas. The only constraint is that the images need to be taken sequentially and have overlapping regions, so that the same feature points can be detected in several images.

While forest fire monitoring videos typically do have a 1080x1920 resolution, some regions of the original frame are cluttered with flight information, so only a smaller part of each frame may be used as an input image. Figure 3.1 shows an original frame of one the videos used in this work. This image has a 1080x1920 resolution, however, the usable part of it is only 630x930.

The video frames' sampling rate depends mainly on three factors: altitude, speed and image resolution. Intuitively, high altitude, high speed and low image resolution all make it harder to perform Sfm, hence a higher sample rate is needed to overcome this. For example, a sampling rate of 2Hz was used for the 900m altitude dataset and one of 4Hz for the 1900m one. In truth, the only drawback of having a higher sampling rate is an increased algorithm run time, so while it is necessary to increase the sampling rate when faced with adverse conditions, it is not mandatory to lower it when in more favourable ones.

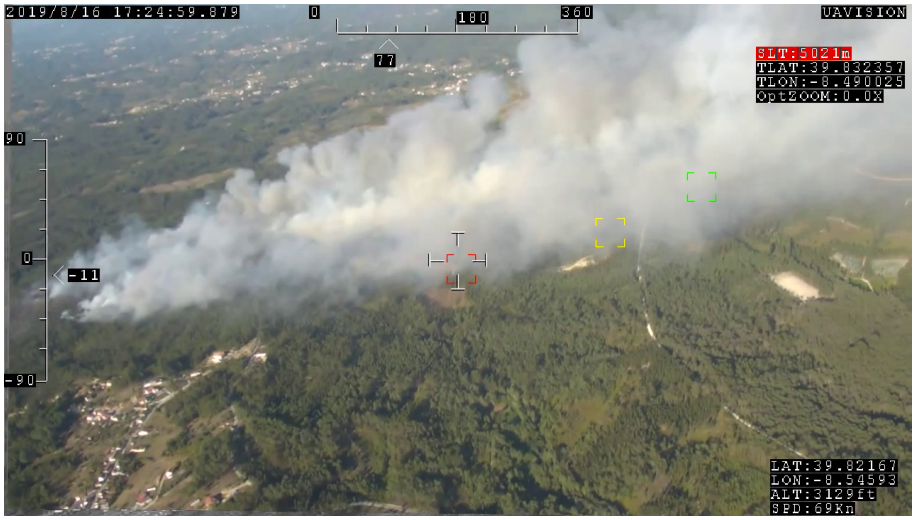


Figure 3.1: Frame from the UAVision dataset

- **DEM**:

  A DEM of the surrounding area is used to register the reconstructed point cloud to the terrain, thereby improving the point cloud's accuracy. The DEMs used in this thesis' tests were all cropped versions of the EU-DEM v1.1. This DEM has a 25m resolution and a vertical RMSE of 7m, making it Europe's most accurate terrain map available for civilian purposes.

- **GPS**:

  The GPS is used to initialize the ICP algorithm, providing an initial relationship between the aircraft's reference frame and a global reference frame.

  However, the GPS's main purpose is to help determine the reconstruction's scale factor: since Sfm cannot directly estimate its scale factor, external measurements have to be used. The traditional way of determining the scale factor is to use GCPs, which provide the real location of a specific

feature point, but GCPs are not flexible enough to be used in forest fire missions, so a different method must be used.

The aircraft measures its airspeed, altitude, orientation and GPS coordinates, so they may all be used in conjunction with the reconstruction itself to determine its scale factor. Since the speed measured is an air speed, not a ground speed, it cannot be used to accurately estimate the scale. The next option is to use the camera's pitch, altitude and a DEM to determine the correct scale factor for the reconstruction to have the same elevation as the DEM, however, the camera's pitch has a very high error, which would force the use of a long iterative process to refine the pitch estimate and the scale factor using several ICP loops.

Therefore, the method chosen to determine the scale factor was to compare the estimated trajectory given by the motion estimation step with the measured trajectory given by the GPS, and use the ratio between them to compute the scale factor.

- **Altitude**:

  The camera's altitude is only used to initialize the ICP algorithm. Since forest fire monitoring is performed in level flight, the algorithm does not actually need the altitude measurements in order to work, as the ICP can easily correct an altitude error by translating the entire point cloud vertically.

- **Camera pitch and heading**:

  The cameras used by forest fire monitoring drones typically have their own stabilization system which provides somewhat reliable pitch and heading measurements. It is important to remark that since the camera is independent from the UAV itself, the aircraft's orientation (which is a far more accurate estimate) cannot be used to estimate the camera's pitch and heading. Instead, the camera's pitch and heading angles are defined as in Figure 3.2, and the roll angle is assumed to be null, since the camera attempts to stabilize it in real time.

  The camera's pitch and heading are only used to transform the reconstructed point cloud from the camera's reference frame to a global inertial reference frame. Much like the altitude and the GPS, Sfm+ICP can work without initial orientation estimates, however, reliable measurements do speed the ICP's convergence and increase its success chance. This is especially true for the pitch angle, which is the hardest for the ICP to correct.

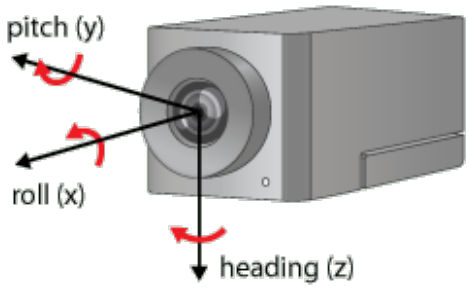


Figure 3.2: Camera orientation axis (taken from [36])

- **Targets' pixel coordinates**:

The user can choose several targets to be georeferenced at once. Each target is fed to the algorithm as a specific pixel in a frame. Since the dense reconstructions can find dozens of features next to each pixel, and each feature's 3D location is determined by Sfm+ICP, the estimated 3D locations of all nearby features can be averaged to approximate the target's real location. And since all the features are georeferenced during Sfm+ICP, there is no limit to the amount of targets the user can georefer in a given algorithm iteration.

Using those inputs, the algorithm's outputs are:

- **Target's geodetic coordinates**:
  Since its main purpose is georeferencing, the algorithm's main output is the target's location, expressed as latitude, longitude and altitude. This output's accuracy is discussed in chapter 4, by comparing the algorithm's georeferencing estimates with manual georeferencing performed using georeferenced satellite imagery. Unlike optic-ray-based georeferencing, Sfm+ICP cannot directly compute its result's uncertainty, it can only qualitatively estimate the reconstruction's accuracy using its reprojection error, and estimate the ICP's result by its RMSE.

- **3D reconstruction of the fire area**:
  The 3D point cloud containing the estimated locations of all the inlier feature points can be used for more than just georeferencing a few targets: it can be used to study the fire's evolution, create an updated 3D map of the disaster area, which may differ drastically from what is shown in older satellite images and it can even help other georeferencing algorithms.

  Since this Sfm+ICP implementation was designed to operate in real time, each reconstruction's size was purposefully kept at less than 50 thousand points per algorithm run, since this is enough to get accurate reconstructions and georeferencing estimates, however, the reconstruction can have a much higher amount of points if the algorithm's time constraints are relaxed.

3.2 Algorithm overview

This section provides quick rundown of each of the algorithm's main components, contextualizing them in the broader scheme of things. Figure 3.3 shows a schematized view of the Sfm+ICP algorithm.

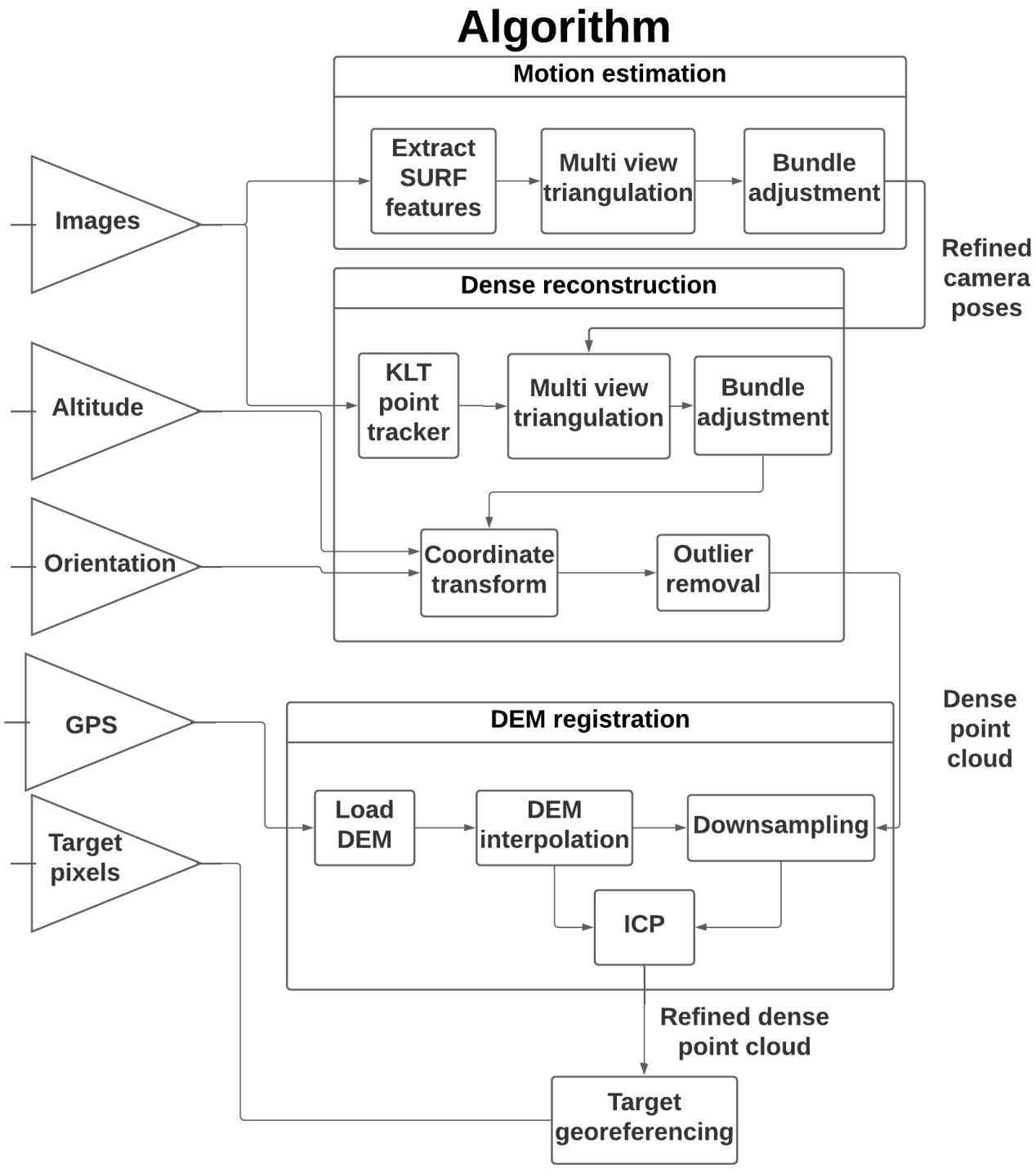


Figure 3.3: Algorithm diagram

The algorithm has four main blocks:

1. **Motion estimation**:

   This step estimates the camera's pose at the time each photograph was taken. The camera poses are estimated using a simple and quick Sfm, often called sparse Sfm in contrast to the dense Sfm used in the next step. Sparse Sfm is a Sfm algorithm that uses sparse features, allowing it to be

run several times faster than its dense counterpart.

The attribute sparse comes from the fact that this Sfm step extracts only the best quality features, rejecting lower quality ones, which allows it to accurately estimate the camera's pose in each of the images. These poses are this block's main output, and will be used in the next step to produce the dense point cloud, which is then used as the basis for the georeferencing estimates.

2. **Dense reconstruction**:
   Dense reconstruction generates a dense point cloud of the area captured in the images. This is achieved in five steps: it starts by detecting features in each image and matching them across all the images. Multi-view triangulation is then used to coarsely estimate the 3D location of these features, which is then refined using a single bundle adjustment. These three steps produce an unscaled point cloud in the camera's reference frame, that is transformed to a scaled inertial reference frame using the telemetry data. The final step is to remove outliers from the point cloud, using their location and reprojection error.

3. **Point cloud registration**:
   Point cloud registration improves the point cloud's precision by matching it to a real DEM of the surrounding area using ICP, *i.e.* translating and rotating the point cloud in a way that minimizes the distance between the real map and the reconstruction. The DEM is interpolated to match the point cloud's resolution, which is downsampled to a 2.5m resolution. Finally, the ICP's RMSE is used to evaluate the algorithm success up to this point: if the RMSE is too high, it means that the 3D reconstruction is of poor quality, thus the algorithm discard the current point cloud and goes back to the first step, motion estimation.

4. **Target georeferencing**:
   This final step computes the targets' geolocation using the refined dense reconstruction from the last step. This the simplest and quickest of the four main blocks, as it simply averages the 3D position of the 10 closest features to the target in order to estimate its location.

3.3 Structure from Motion

Structure from motion uses the the photographs taken by the aircraft and its telemetry to reconstruct the scene captured in the images.

The Structure from motion block has two steps: motion estimation, used to estimate the camera's pose throughout the images, and dense reconstruction, which generates a dense point cloud of the area captured in the images.

3.3.1 Motion estimation

As depicted in Figure 3.3, the motion estimation step is composed of three smaller blocks, the first of which is extracting Speeded Up Robust Features (SURF) from the input images.

SURF is a scale and rotation invariant interest point detector and descriptor. It is a quicker version of SIFT, while also being more accurate than most state of the art feature extractors, due to its scale and rotation invariability [37].

SURF's only major downside is that it extracts much fewer features when compared to SIFT, KAZE and corner feature detectors such as the Minimum eigenvalue algorithm. However, SURF greatly outperforms these algorithms in speed, and its lack of feature density is actually an advantage in motion estimation: as the main purpose of this step is to estimate the camera's pose at each of the instants it took a photograph, not to reconstruct the captured scene. In this way, having fewer but high quality features as provided by SURF is much better than having a lot of low quality SIFT, KAZE or Min Eigen features.

Figure 3.4: Features extracted by SURF

SURF's sparsity is clear in Figure 3.4, as it only extracts 250 features in one image. However, most of these features are in regions favourable to feature matching, such as buildings, roads and the smoke's outline.

Algorithm 1 describes the motion estimation algorithm: each camera pose is initially estimated relative to the previous pose, using the SURF feature matches found between each sequential image pair. After performing this initial pose estimation loop, a bundle adjustment is used to refine the camera poses. To improve the accuracy, an intermediary step was added to also estimate the 3D location of all the SURF features matched across all images, however, the refined camera poses are the only relevant

output from this multi-view triangulation and bundle adjustment sequence, since the recovered 3D point cloud is very sparse and only used inside the motion estimation subroutine.

Algorithm 1 Motion estimation algorithm

Inputs: images, IP.
Outputs: camera poses.

1: Initialize the first camera pose as the global reference frame's origin.
2: Extract SURF features from the first image.
3: **for all** remaining images **do**
4: Extract SURF features from the image.
5: Match these features to the last image's features.
6: Estimate the pose of the camera relative to the previous camera, using the matched features.
7: Convert this relative pose to a global pose.
8: **end for**
9: Estimate the 3D location of all the features matched between all the images, using Multi-view triangulation.
10: Refine the camera poses and feature 3D locations using Bundle adjustment.

This is a simple but quick configuration: the use of a fast feature detector alongside a single global bundle adjustment ensures that it runs smoothly even when dozens of images are processed in real time. And while it is not as accurate as a configuration with a denser feature detector and multiple bundle adjustment steps, 3D feature accuracy is not the priority here, since the dense reconstruction step refines the estimates provided by the motion estimation step.

This Motion estimation method does what it is supposed to do: quickly estimate the camera's pose throughout the image sequence, considerably reducing the dense reconstruction's run time. Additionally, the motion estimation step can be run alongside the initial parts of the dense reconstruction algorithm without any loss in accuracy or point cloud density.

The results of the Motion estimation step can also be used to predict the final result's georeferencing accuracy: the motion estimation's bundle adjustment has a certain mean reprojection error, corresponding to the average image distance between the reprojected points and the measured ones. This value can be easily calculated after the motion estimation step and used as a way to discard the current algorithm run and start over, as a high reprojection error often leads to inconsistent 3D reconstructions and a poor ICP registration, culminating in a bad georeferencing estimate.

3.3.2 Dense reconstruction

As shown in Figure 3.3, the dense reconstruction block uses the refined camera poses computed in the motion estimation step alongside the images and telemetry data to produce a dense point cloud of the photographed area.

Algorithm 2 provides a step by step overview of the dense reconstruction algorithm. It starts by initializing a Kanade-Lukas-Tomasi point tracker using Min Eigen features. The KLT is a state of the art feature tracking algorithm [38], which, like other point trackers, is rarely used in Structure from motion projects, since it cannot handle long scenes with low overlap. However, the forest fire footage used in fire monitoring scenarios is often focused on a single operational area, so the camera captures the same

Algorithm 2 Dense reconstruction algorithm

Inputs: images, IP, refined camera poses, GPS, altitude and camera orientation (given by the IMU).
Outputs: dense point cloud in an inertial North-East-Down reference frame.

1: Initialize the KLT point tracker with Min Eigen features extracted from the first image.
2: **for all** remaining images **do**
3: Find the previous tracked features on the next image, using the KLT.
4: **end for**
5: Estimate the 3D location of all the points tracked, using multi-view triangulation.
6: Exclude points with reprojection error higher than 2 pixels.
7: Refine the camera poses and feature 3D locations using Bundle adjustment.
8: Rotate the point cloud using the pitch and heading of the first camera pose.
9: Translate the point cloud using the altitude of the first camera pose.
10: Scale the point cloud using the ratio. between the refined camera poses and the GPS.
11: Remove outliers from the point cloud using their distance and elevation.

scene from multiple angles, which suits the KLT.

Min Eigen features were selected for the KLT's initialization, since they greatly outperform other state of the art feature extractors in feature density for low resolution images. This concern is especially pressing given the difficulties faced by SURF, SIFT and KAZE in extracting features from forest environments, which offer low color variability (they are often quite green). These difficulties are amplified by the aircraft's altitude, usually between 800 m and 2000 m, and the disturbances common in forest fire scenarios: heavy smoke and wind induced camera shaking. All these factors make the KLT the most sensible dense feature extraction method for this type of medium altitude disaster monitoring.

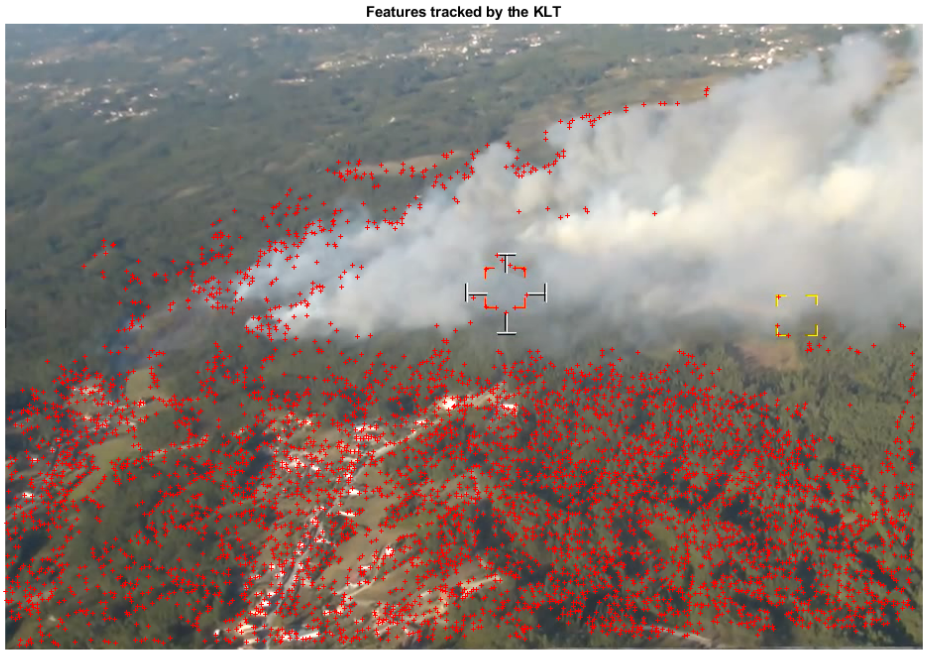


Figure 3.5: Features extracted by KLT

Figure 3.5 shows the features tracked by the KLT in one of the datasets. The features shown in 3.5 are just the ones selected as inliers at the end of the dense reconstruction, not the entire reconstruction

block, hence why faraway regions have no features. The KLT shows a good dispersion of features, tracking them accurately in forests and in the town, while having a relatively low percentage of smoke outliers.

The KLT is used to find feature matches across all the images, which are then used alongside the refined camera poses estimated in algorithm 1 to compute the 3D locations of all the matched features, using multi-view triangulation. Only one global Bundle adjustment is used to refine the 3D feature locations, since the alternative would be multiple local Bundle adjustments, each right after finding the tracked features in each image, which would severely increase the algorithm's computation time. However, multiple local bundle adjustments would work well in an approach with a low number of images, at the cost of a smaller reconstruction area.

The bundle adjustment produces a dense unscaled point cloud in the reference frame of the first camera pose, which is then converted into an inertial North-East-Down reference frame. This non-rigid transformation uses the measured camera orientation, GPS and altitude to rotate, translate and scale the point cloud. This operation will produce a highly inaccurate result, since the orientation given by the on-board sensors is off by several degrees, which is enough to render the resulting point cloud useless for georeferencing purposes, however, the registration block is able to rectify these errors.

On the other hand, the reconstruction's scale factor cannot be directly estimated by the ICP in the next step , since it is a rigid point cloud registration method. Notwithstanding, a bad scale factor leads to a poor DEM registration, which can be easily identified as a high RMSE ICP value, therefore, bad dense reconstructions cane be either discarded, or further rectified by iteratively performing DEM registration with slightly different scale factors.

The scale factor is clearly the most critical parameter to accurately estimate: the translation and rotation errors can be easily corrected by the ICP, even if they are extremely large, however, from empirical evidence, a small 1% scale error leads to a 20% increase in georeferencing error. The best way to estimate the scale factor is to compare the estimated trajectory from the motion estimation step and the measured trajectory given by GPS.

The last operation in algorithm 2 is to remove outliers from the final dense N-E-D point cloud, using their 3D attributes. Two types of outliers are detected and removed:

- **Smoke outliers**: these are points that correspond to the smoke outline tracked by the KLT, which are not relevant for this georeferencing approach, since the goal is to geolocate ground points, not smoke. Most of these points can be removed by filtering out high elevation points.
- **Distant points with high reprojection error:** points that are more than a few kilometres away from the camera are not relevant for this georeferencing approach. They are not only isolated from the main body of the dense point cloud, but they also have a higher reprojection error than most points, and are not useful for the registration step, since they are isolated and prone to high error.

3.4 DEM Registration and georeferencing

DEM registration is the last major block of the proposed georeferencing algorithm, as shown in Figure 3.3. The main goal of this block is to rectify the dense reconstruction obtained in the previous step by applying a geometric transformation to the estimated point cloud, thereby improving the algorithm's georeferencing accuracy. This transformation is obtained by registering the point cloud to a DEM.

The next subsections present and explain each of the components of this registration and georeferencing approach.

3.4.1 EU-DEM v1.1

The EU-DEM v1.1 was selected as the reference DEM used by the ICP. This DEM is provided for free by European Environment Agency (EEA) under the Copernicus program. The EU-DEM v1.1 covers all the EEA member states, as well as some countries to the east. However, it does not cover some Mediterranean countries that suffer from forest fires [39], such as Morocco, Algeria, Jordan and Israel, nor some eastern European countries such as Ukraine and Russia.

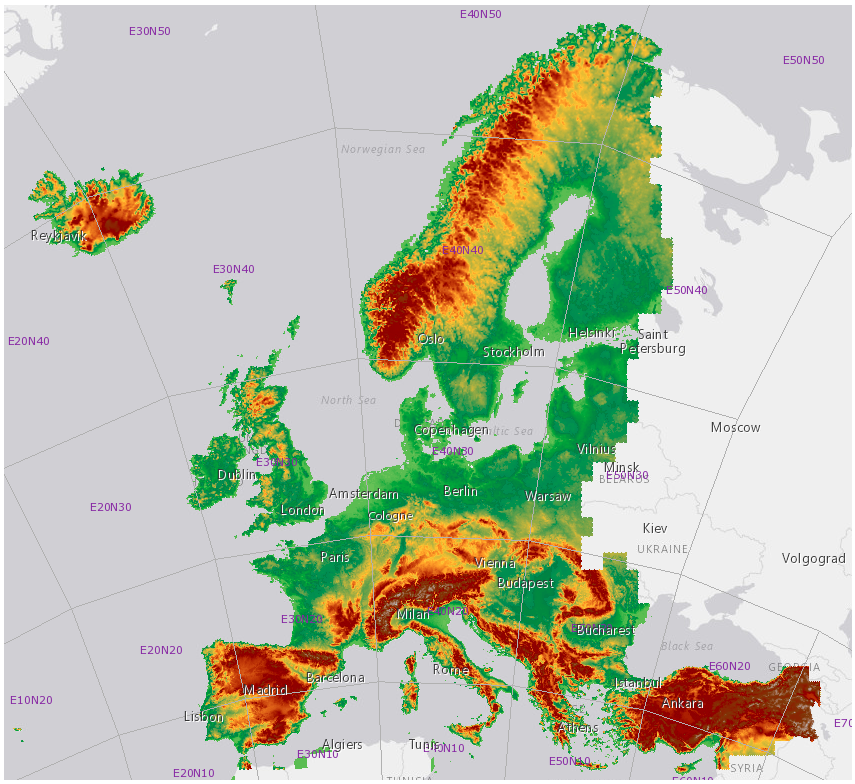


Figure 3.6: Extent of the EU-DEM v1.1

The EU-DEM v1.1 was created by merging elevation data from the SRTM and ASTER global datasets, as well as from Soviet topographic maps at high latitudes. It has a 25 metre spatial resolution and a vertical RMSE of 7 metres, which is quite good for such a large DEM. Recent validation studies such as [40] conclude that the EU DEM v1.1 has the higher resolution of ASTER GDEM as well as the vertical accuracy of both SRTM DEM and ASTER GDEM in low and high slope areas, respectively. It has also

been used for georeferencing in forest fire scenarios by Santana in [5].

The DEM used as a reference for the ICP is an interpolation of the original EU-DEM v1.1, since a 2.5 metre resolution interpolated DEM yields better ICP registration results than the original 25 metre one.

3.4.2 ICP

After interpolating the DEM and downsampling the dense point cloud obtained in algorithm 2, the ICP is employed to compute a rotation matrix and translation vector that better rectify the point cloud. This transformation's goal is to counter the effect of the telemetry and camera model errors, which produce a slightly innacurate reconstruction. For example, a pitch error produces a higher than normal slope on the reconstruction, and an altitude error produces a point cloud with a certain vertical offset relative to the real terrain. The rotation and translation corrections can also be used to refine the aircraft's telemetry, which is especially useful for low cost camera rigs.

Algorithm 3 Registration and target georeferencing algorithm

Inputs: DEM, dense point cloud of the terrain, GPS coordinates, target pixels.
Outputs: target coordinates, rectified point cloud.

1: Select a relevant subsection of the EU DEM v1.1 using the GPS coordinates.
2: Interpolate the selected DEM to a 2.5 metre resolution.
3: Downsample the dense point cloud to a 2.5 metre resolution.
4: Perform ICP between the downsampled point cloud and the interpolated DEM. Use a point to plane minimization metric.
5: Apply the rigid transformation provided by the ICP to the full dense point cloud.

6: **for all** target pixels **do**
7: Find the 10 closest features to each target pixel and their corresponding 3D locations in the rectified point cloud.
8: Average the location of the 10 closest features, using Inverse distance weighting, where each location estimate is weighted according to its pixel distance from the target.
9: **end for**

Algorithm 3 provides a brief overview of the registration and georeferencing algorithm. It starts by preparing the two point clouds for registration. The DEM used is a 4km by 4km square centered around the aircraft's GPS coordinates, extracted from the EU-DEM v1.1. This may seem an excessive size, however, the reconstructions usually capture several hundred hectares, which depending on the camera's altitude and pitch, may be several kilometers from the aircraft.

The DEM is interpolated to a 2.5m resolution to match the downsampled point cloud's own resolution. The primary reason for this downsampling is to reduce the algorithm's run time, but empirical evidence suggests that matching the two point cloud's resolutions also reduces the horizontal georeferencing error. The specific resolution was set at 2.5m since medium altitude surveys can comfortably extract several features per square metre, however, a 15m resolution was used in the high altitude tests (above 1500m).

Table 3.1 presents the ICP's parameters used for the medium altitude datasets.

Point-to-plane minimization was chosen, as it greatly outperforms point-to-point in both run time and accuracy in surface to surface registration. Point-to-plane treats the fixed point cloud as a surface, rather

Table 3.1: ICP parameters

Parameter	Value
Dense point cloud's resolution	2.5m
DEM's resolution	2.5m
Minimization metric	point-to-plane
Maximum iterations	120
Inlier ratio	0.5
Extrapolation step	yes

than a set of points, by projecting the distance error onto a plane normal to the fixed surface, so as to minimize the distance between surfaces. Hence, the ICP's metric may be defined as

$$min \sum_{i=1}^{N} ((s_i - d_i) \cdot n_i)^2, \tag{3.1}$$

where s_i and d_i are, respectively, the moving and fixed points and n_i are the surface normal unit vectors at each fixed point. This idea is also schematized in Figure 3.7.

A 0.5 inlier ratio was used, since it delivered the best georeferencing results in the medium altitude tests. This ratio is used by the ICP as to only consider the indicated fraction of possible inlier point pairs: if it is too high, the ICP will be unable to remove outlier pairs (caused by smoke, high elevation anomalies, etc), and if its too low, the ICP will reject too many good pairs and end up in a local minimum. The optimal inlier ratio depends mostly on the camera's altitude, as lower surveys reconstruct the terrain more accurately, while high altitude ones will have plenty of anomalies.

An extrapolation step was included at the end of each ICP iteration, which traces out a path in the registration state space, resulting in a faster convergence, as the original pointcloud has a good initial correlation with the DEM.

The parameters used in the medium altitude version of Sfm+ICP (shown in Table 3.1) had the lowest georeferencing error and ICP RMSE. Since they correlate with both, they can be optimized during a long survey by performing some initial reconstructions and computing the specific array of parameters that minimize the RMSE. This is especially true for the inlier ratio and resolution, whose optimal values depend mostly on the reconstruction's quality and camera altitude.

The ICP's effect can be seen in Figure 3.8, which shows the reconstruction before and after applying the ICP transform in a specific dataset. The refined point cloud (in green) shifts several dozen metres southwest (seen in 3.8a and 3.8c), indicating a large GPS error. The green point cloud is also pitched down considerably (seen in 3.8b and 3.8d), thus fixing some orientation error. The resulting point cloud is visibly much closer to the DEM than the original one, which results in a much lower georeferencing error, as shown in the next chapter.

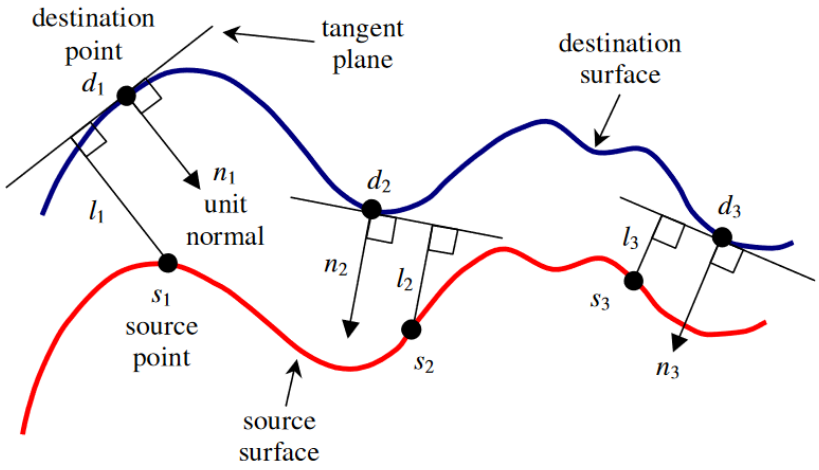
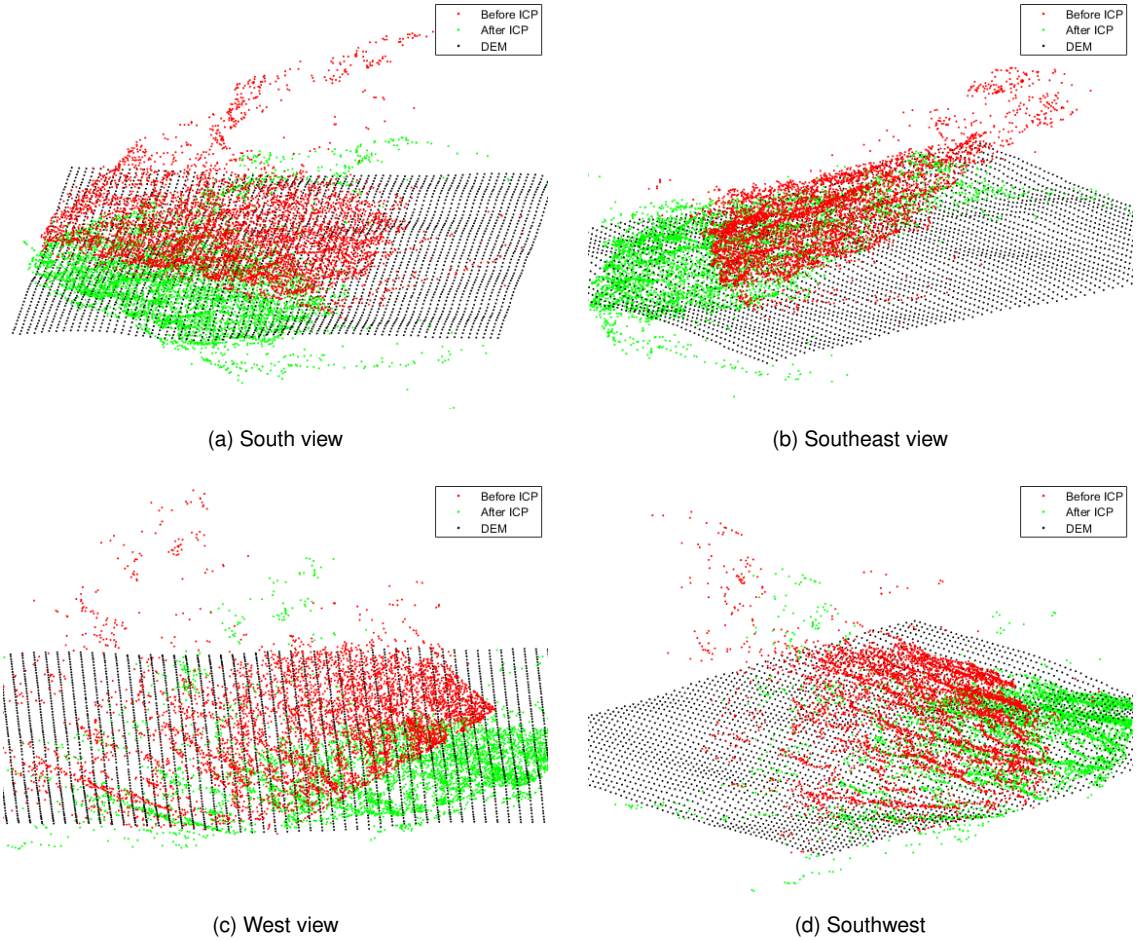


Figure 3.7: Point-to-plane minimization (taken from[20])



(a) South view

(b) Southeast view

(c) West view

(d) Southwest

Figure 3.8: Point cloud before (in red) and after (in green) applying the ICP transform, and DEM (in black)

3.4.3 Target georeferencing

The final loop in Algorithm 3 estimates each target's 3D location in an inertial NED reference frame independent of the aircraft. Each target is provided by the user as a pixel in one image. The KLT used

in the dense reconstruction step tracks a set of features identified in the images, some of which are very close to the input pixels. Since there are dozens of features less than a few pixels away from the target pixel, the best target location estimate is an average of the surrounding features' corresponding 3D location.

However, not all features are weighted the same in this average, in fact, each feature's weight for the final target location estimate is inversely proportional to the square of its pixel distance to the target. This method ensures that outlier locations are not as detrimental to the final result, while also ensuring that distant measurements are not given too much relevance.

Thus, the 3D location of the target, P_{target} is computed as

$$P_{target} = \frac{\sum_{i=1}^{N} w_i P_{target}^i}{\sum_{i=1}^{N} w_i} \tag{3.2}$$

where N is the number of nearby features, P_{target}^i is the 3D location of feature i, and w_i is the weight given to feature i's estimate, which is given by:

$$w_i = \frac{1}{\| \mathrm{px}_{target}, \mathrm{px}_i \|^2}, \tag{3.3}$$

where px_{target} and px_i are the pixel coordinates of the target and feature i, respectively.

Figure 3.9 shows one dataset's target and the nearby feature points. The target in question is a large house on the edge of a village. As can be seen, several features are extracted around the house, and the 10 closest features to it could all be considered as part of the house, therefore, averaging all these features' locations provides a better georeferencing estimate than simply choosing the closest feature to the target pixel.

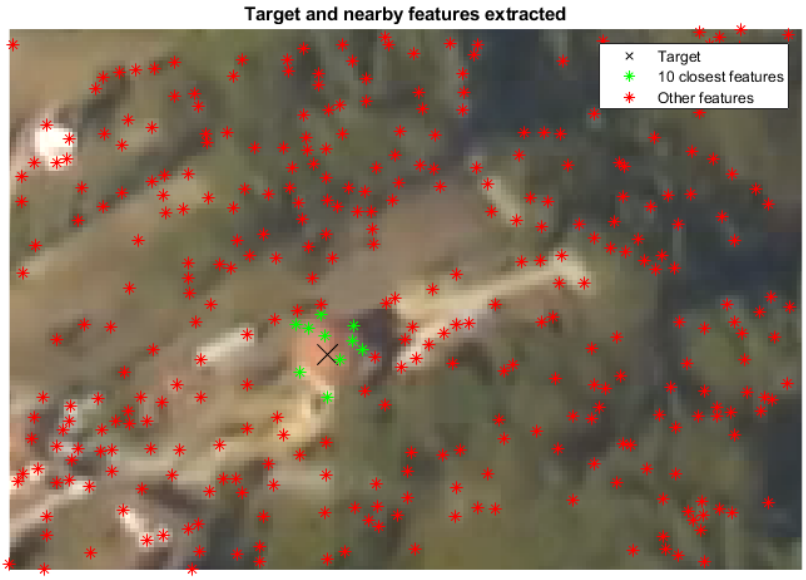


Figure 3.9: Target georeferencing example

3.5 Metrics

This section introduces the metrics used to evaluate the performance of the Sfm+ICP algorithm, which influenced how it was optimized throughout its development. There are three types of metrics, depending on what they evaluate: those which evaluate the algorithm's georeferencing accuracy, the reconstruction's quality and the ICP registration's success.

Georeferencing accuracy metrics are the most relevant algorithm performance indicators, since they directly measure the difference between the target's estimated geolocation and the true one. Each dataset had several targets chosen *a priori*. Most targets are either crossroads, large houses, churches or cemeteries, which are easily manually georeferenced using google earth, which allows the estimated geolocations to be compared with the real ones with precision.

Two accuracy metrics were used:

- **Mean horizontal error (XY)**:
  A target's horizontal error is the distance between its real position and the estimated one, in the East (X) and North (Y) components. Therefore, a dataset's mean horizontal error, XY, is defined as
  $$XY = \frac{\sum_{i=1}^{N} \|(x_i^e, y_i^e) - (x_i^r, y_i^r)\|}{N}, \tag{3.4}$$
  where N is the dataset's number of targets, x_i^e and y_i^e are the estimated east and north components of target i's location, while x_i^r and y_i^r are its real ones.

- **Mean vertical error (Z)**:
  Similarly, a target's vertical error is the absolute elevation difference between its real and estimated positions, hence, the mean vertical error, Z, is defined as
  $$Z = \frac{\sum_{i=1}^{N} \|Z_i^e - Z_i^r\|}{N}, \tag{3.5}$$
  with Z_i^e and Z_i^r being the real and estimated elevations for target i.

The XY is the most relevant of the two, because georeferencing algorithms are mostly relied on to estimate longitude and latitude, while elevation can be consulted in a DEM. Additionally, since the ICP registers the point cloud with the DEM, the XY error is usually one degree of magnitude higher than the Z error, nonetheless, the Z error can provide insights into how well the registration was performed.

The mean reprojection error, MRE, was used to evaluate each reconstruction's quality. The reprojection error was already defined in subsection 2.1.3, as the pixel distance between a feature's actual pixel, defined as x_{ij} and the reprojected pixel, represented as x'_{ij}. The former is determined simply by detecting the feature with a feature extractor, while the latter is the result of the reprojection of the feature's 3D location, $\hat{X}_i$, back to the image. In theory, x'_{ij} should be as close as possible to x_{ij}, since x_{ij} is used to estimate $\hat{X}_i$, however, in practice, the camera has an inherent mean reprojection error which is amplified by feature pair outliers. All high reprojection error points are excluded from the reconstruction,

however, a poor quality point cloud can still be identified using the MRE.

The last metric to be presented is the ICP's RMSE, which is used to determine how similar to the real terrain the 3D reconstruction is. As mentioned in subsection 2.1.5, each ICP's iteration attempts to minimize the RMSE by rotating and translating the reconstructed point cloud to match the real DEM. Since both point clouds have the same density and are both square grids, the RMSE can be defined simply as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M} \|Z_i'^e - Z_i'^r\|}{M}}, \tag{3.6}$$

where $Z_i'^e$ and $Z_i'^r$ are the elevations on the estimated and real point clouds being registered, for grid square number i, and M is the number of inlier grid squares. Note that the estimated point cloud used here is a downsampled version of the original reconstruction, hence the notation Z' instead of Z, while the real point cloud is an interpolated version of the original DEM, to match the estimated point cloud's resolution.

Chapter 4

Simulations and results

Contents

This chapter presents the algorithm's results using four different datasets. The algorithm is subjected to different image sources in diverse operational scenarios to evaluate its georeferencing accuracy. These four datasets consist of:

1. A dataset obtained using Blender, a 3D simulation tool. This tool is used to simulate an UAV flying over a series of targets which the algorithm attempts to georefer. This dataset was used to determine the algorithm's performance in ideal conditions.

2. A second Blender dataset with a slightly more complex camera trajectory and orientation.

3. A dataset obtained from real footage taken by an UAV operating in a forest fire environment. This dataset was used to assess the algorithm's accuracy in a real medium altitude scenario.

4. A dataset obtained from real footage taken by an air force aircraft in a forest fire environment. This last dataset was used to evaluate the algorithm's performance when subjected to low quality images and a high operational altitude.

4.1 Blender datasets

Blender is a popular free and open source 3D creation tool used by researchers to simulate 3D environments. It was particularly useful in this work for two reasons:

- Blender can be used to animate a camera along a path. Moreover, the camera can be modelled using the pinhole model, enabling the simulation to run with precise EP and IP.
- Blender has an extensive list of open source Addons, such as BlenderGIS. BlenderGIS is used to import satellite images and elevation map data to construct a 3D terrain model, which the simulated camera can capture from different viewpoints. These artificial photographs can then be fed to the algorithm to test its accuracy in ideal conditions.

Both Blender datasets were obtained from the same artificial environment: Serra da Archeira, a small mountain range south of Torres Vedras, at 39.029005°N -9.219768°E, as shown in Figure 4.1. This location was chosen for its terrain complexity: the simulation covers a large flat valley, slopes, mountain tops, small villages, high canopy forests and deforested areas.

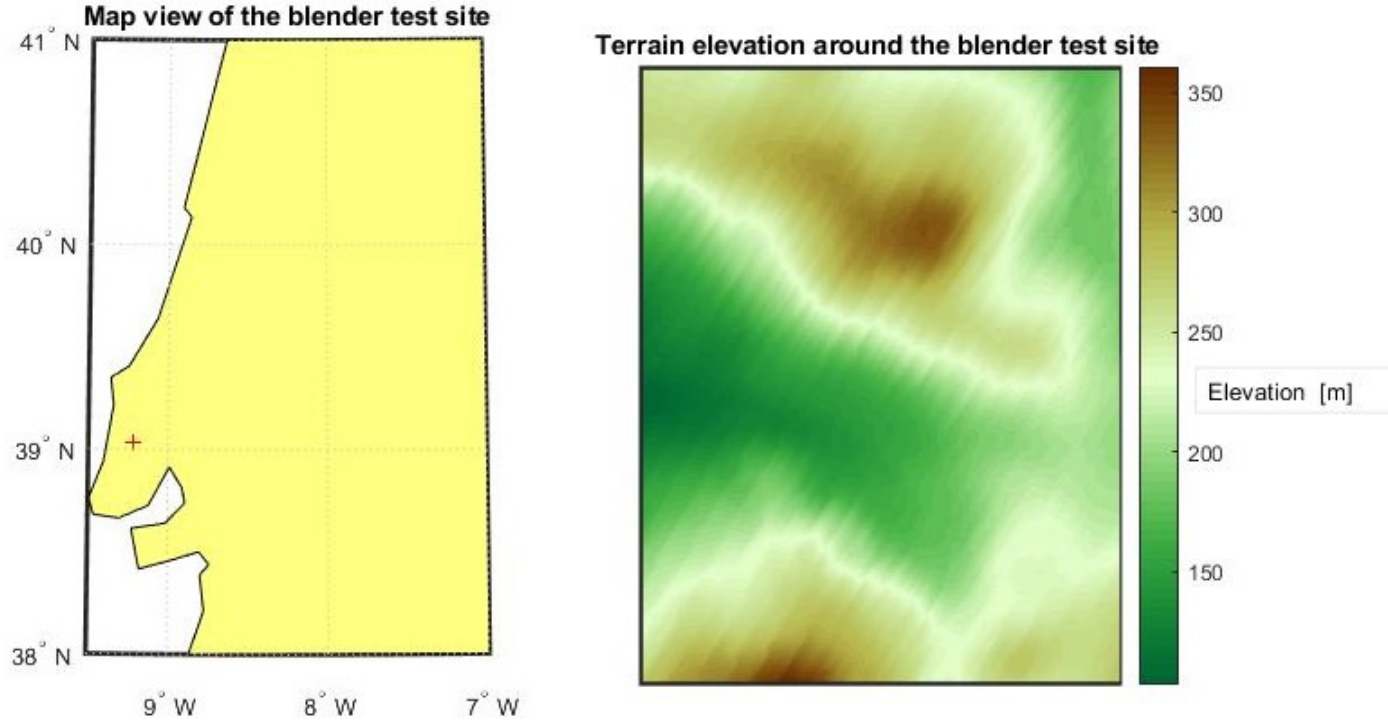


Figure 4.1: Map and DEM view of the blender environment

The camera used in the blender simulations had the same focal distance as the one used in the real datasets presented in 4.2 and 4.3, however, it had a slightly larger image size, 1080x1920. This ideal camera used in blender has no distortions and is not subject to the rolling shutter effect. These factors make this dataset better suited to the georeferencing algorithm, since its images have a higher quality.

4.1.1 Blender test 1 - Linear trajectory

The first blender dataset is the simplest one: the camera moves with constant velocity and orientation in a straight line over the targets. The camera has a fixed pitch of -45°, a perfect North heading and no roll angle, as shown in Figure 4.6. It is important to mention that the position and orientation data fed to the algorithm for these blender tests was not the one given by blender. Instead, each algorithm run

was fed the camera pose with a Gaussian error, so as to better simulate a real scenario. The GPS and altitude errors follow a Gaussian with μ=0 m and σ=10 m, while the pitch and heading error follow a one with μ=0 º and σ= 1°.

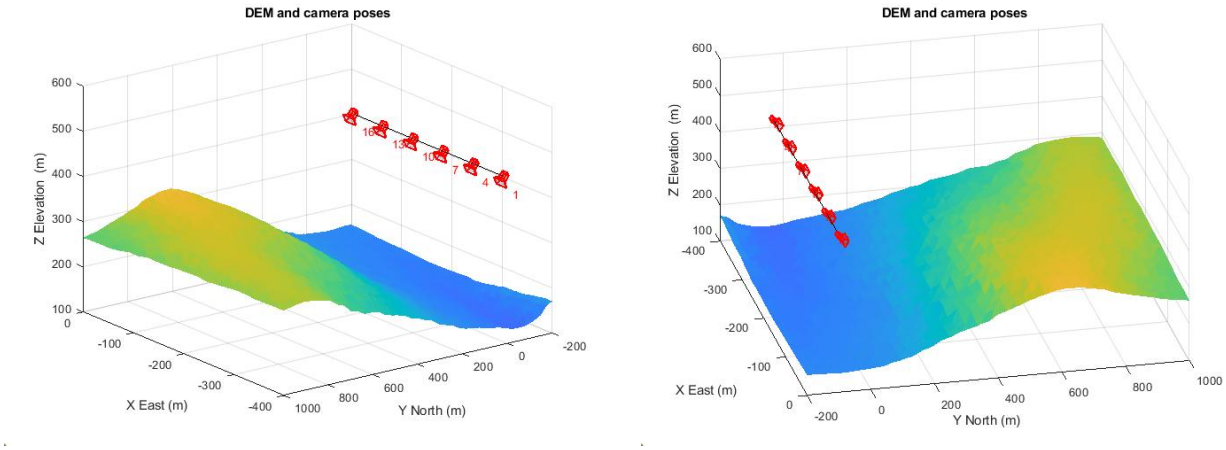
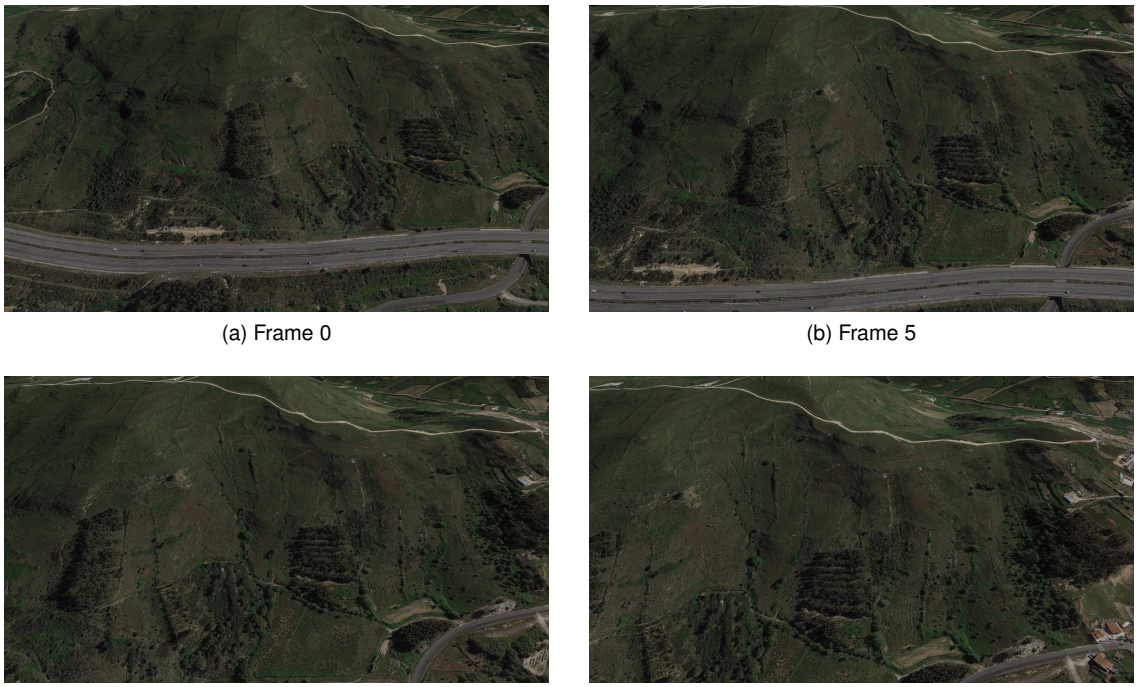


Figure 4.2: Camera poses relative to blender environment's terrain for blender test 1

The camera captured several photographs of the small mountain north of it. Sixteen images were taken during the simulation, all equally spaced by 25 metres, so as to emulate a 2 Hz video sampling frequency, since the aircraft used in the real datasets fly at roughly 50 m/s. Figure 4.3 shows four images taken by the camera, which were used by the Sfm block to perform dense reconstruction. These sixteen



(a) Frame 0

(b) Frame 5

(c) Frame 10

(d) Frame 15

Figure 4.3: Four equally spaced frames from the first blender dataset

images were processed by the algorithm, creating a dense point cloud of the captured environment which can be seen in Figure 4.4. The algorithm reconstructs the operational area with very few gaps:

the point cloud's density is larger near roads and houses, where the feature detector is able to extract a larger number of features, and the density in more remote vegetated areas is sufficient to perform georeferencing.

The point cloud covers 225ha, which highlights the monitoring capabilities of this genre of georeferencing algorithms, which are able to quickly reconstruct and georefer the disaster area with remarkable range. However, the point cloud's density is significantly lower in specific regions due to the camera's limitations: areas behind mountains, and regions suffering from occlusion effects.

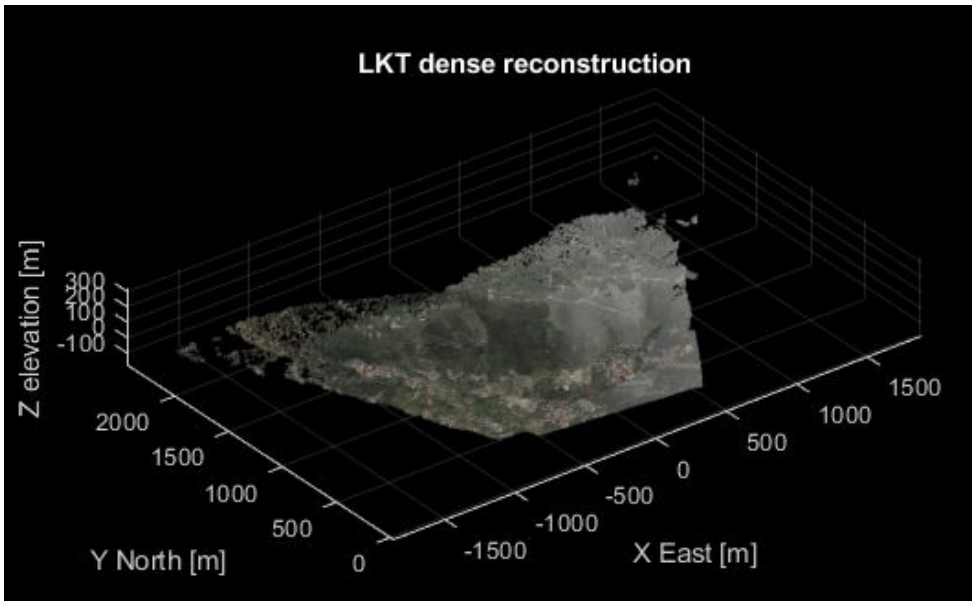


Figure 4.4: Dense point cloud computed from blender dataset 1

As mentioned in the implementation section, and displayed in Figure 3.3, the dense point cloud was improved using ICP. This improved point cloud is then used to georefer several targets inputted by the user as pixel coordinates in any of the images used to reconstruct their location. To assess this process' accuracy, the estimated 3D coordinates of each target were compared to the real target coordinates, given by blender. These targets include crossroads, windmills and houses.

The georeferencing error before and after applying the ICP transform is shown in Figure 4.5 for several algorithm runs, and accuracy statistics are shown in Table 4.1.

Figure 4.5 shows the mean horizontal and vertical errors for ten algorithm runs before and after applying the ICP transform. Eight targets were georeferecend by the algorithm during each algorithm run. Figure 4.5 shows a clear error convergence, forcing most of the scattered pre-ICP values to a precise location, however, while this correction lowers the vertical error, it slightly increases the horizontal error in some instances. This convergence shows that the ICP produces coherent and reproducible results even on slightly different point clouds with different induced telemetry noises. Moreover, while some pre-ICP results have a large altitude error, the ICP is able to rectify it, at the cost of some additional XY error.

The real datasets' results in sections 4.2 and 4.3 do not show this consistent increase in horizontal error, so this additional XY error in the blender tests may be due to inaccuracies in the blender model

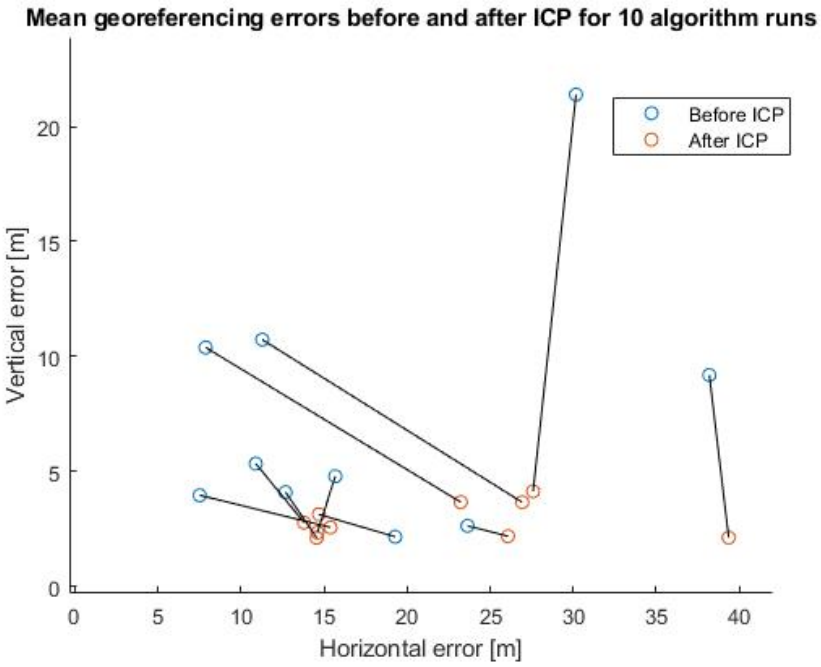


Figure 4.5: Georeferencing accuracy before and after ICP for blender test 1

Table 4.1: Georeferencing error statistics after ICP for blender test 1

	XY [m]	Z [m]
Max	39.4	4.1
Mean	21.6	2.9
Median	19.3	2.7
Standard deviation	8.5	0.7
Range	25.6	2.0

itself.

Table 4.1 presents basic statistics regarding the post-ICP results shown in Figure 4.5. Overall, these results prove that this Sfm+ICP algorithm can accurately estimate a target's position, even in a medium altitude flight with noisy sensor readings. They also show that even though the camera is roughly 600m above the targets, the vertical georeferencing error is remarkably small. On the other hand, the horizontal accuracy shows less consistency, however, it is still under the resolution of the DEM used (25m).

This test measured the algorithm's behaviour under the most favourable conditions: a simple linear trajectory, fixed camera orientation, static environment and good quality imagery. The next subsections concern more demanding tests and further evaluate the algorithm's performance when these favourable conditions are absent.

4.1.2 Blender test 2 - Complex trajectory

This second Blender dataset is slightly more complex than the one presented in subsection 4.1.1: in this simulation, the camera moves in a curved trajectory and with a slight rate of descent. Moreover, the camera's orientation is not fixed as in 4.1.1, instead, the camera is locked on to a target for the duration of the simulation, altering its heading and pitch as its distance to the target changes.

These simulation parameters were designed to emulate a real forest fire scenario, in which the aircraft loiters around the fire, with the camera changing its orientation to track it. This test's results provide valuable insights into the algorithm's behaviour when subjected to a sequence of complex camera poses. Figure 4.6 shows the camera's pose along the simulation:

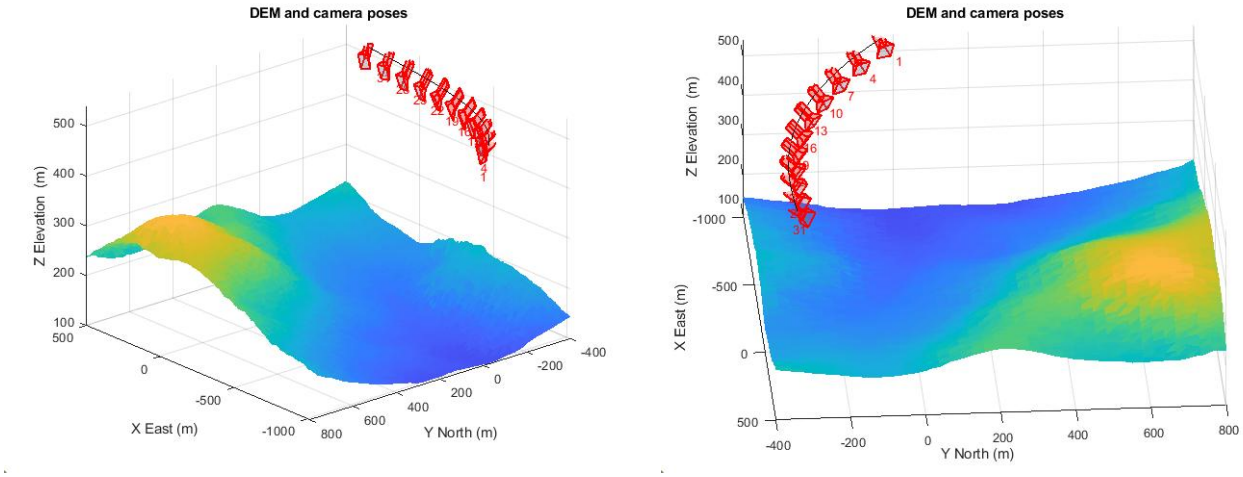


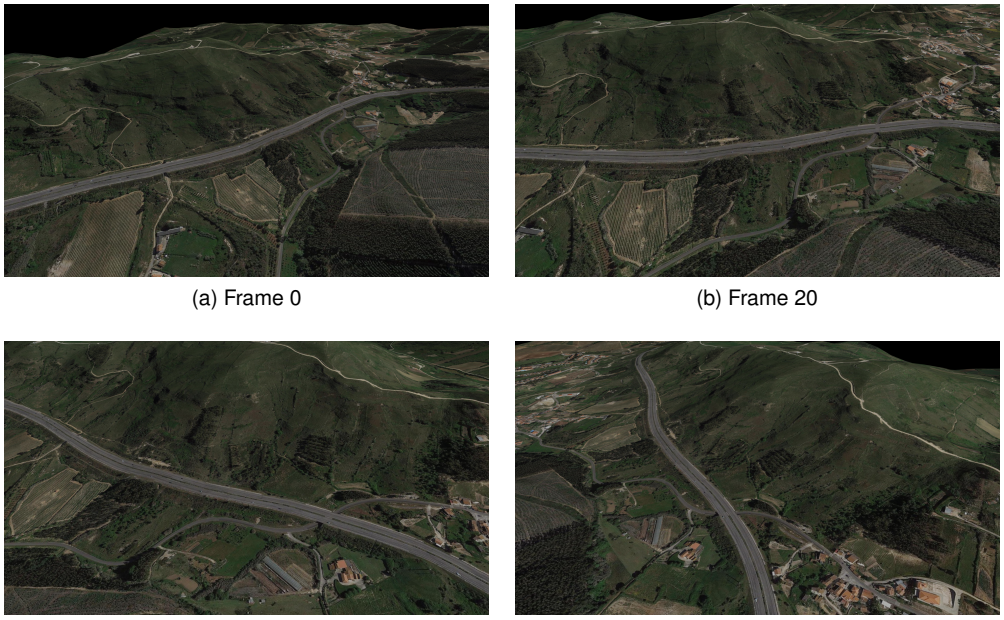
Figure 4.6: Camera poses relative to blender environment's terrain for blender test 2

Figure 4.7 shows four of the sixty frames used in this experimental algorithm test. These sixty frames were taken at regular intervals as the camera moved 500 m east and 400 m south. The captured area is the same as the one photographed in the previous blender test, however, in this one the camera views it from different angles and at different distances. This puts more pressure on the motion estimation step, as it needs to determine the 6 parameters of each camera pose accurately, otherwise the subsequent dense reconstruction step fails and provides an inconsistent and inaccurate point cloud, leading to a high georeferencing error.

Figure 4.8 shows the mean georeferencing errors for ten algorithm runs using blender dataset 2 and Table 4.2 provides statistics regarding those results. Figure 4.8 shows the georeferencing error for each algorithm run, before and applying the ICP transform. Each pair of connected points shows the mean georeferencing error of the targets estimated for each algorithm run, before and after applying the ICP transform.

The pre-ICP results are significantly more inaccurate than as in the first blender test in which the camera followed a linear trajectory, however, the ICP is able to rectify this large error: the ICP improves the XY accuracy by 20% and the Z accuracy by 50%. This is a stark contrast to figure 4.5, in which the ICP did not dramatically increase the point cloud's georeferencing accuracy, which can be explained by the more demanding nature of this second dataset, in which the camera's EP change in 5 spatial dimensions (3 translation and 2 angular directions), while it only moved west with constant orientation in the first dataset.

In other words, the Sfm process yields a less accurate point cloud, however, the ICP is able to consistently improve its accuracy: the post-ICP results bested the pre-ICP ones in every single algorithm run. This proves that this algorithm is not sensitive to poor initial reconstructions, or to complex trajec-



(a) Frame 0 (b) Frame 20

(c) Frame 40 (d) Frame 60

Figure 4.7: Four equally spaced frames from the blender dataset 2

Table 4.2: Georeferencing accuracy statistics after ICP for blender test 2

	XY [m]	Z [m]
Max	44.5	6.5
Mean	38.0	4.4
Median	37.6	4.3
Standard deviation	3.0	1.0
Range	9.6	3.4

tories. On the other hand, it also shows that using a simple trajectory, avoiding long reconstructions and complex object tracking algorithms leads to better georeferencing results. Notwithstanding, in real forest fire scenarios, the camera is often subject to environmental disturbances and flight constraints that force it to move in a complex manner, therefore any georeferencing algorithm needs to be robust to these complex movements. There is also often the need for the camera to be focused on a particular spot of the fire area, so having an algorithm that can process images taken from different viewpoints is also important.

Table 4.2 presents a statistical overview of the results shown in Figure 4.8. Unsurprisingly, the mean and median errors are larger when compared to the linear trajectory results presented in subsection 4.1.1, which can be explained by the more demanding nature of the dataset.

On the other hand, Table 4.2 shows that the second dataset leads to a much smaller horizontal error dispersion, evident by the reduced standard deviation and range. This increased consistency is due to the diversity of the images taken by the camera: as the camera changes its orientation and focuses on a specific region, it is able to capture the same features from very diverse viewpoints, leading to a more consistent result, even if these results are on average less accurate due to the increased motion complexity.

As for the vertical error, it is almost as small as in the previous dataset, showing yet again that a

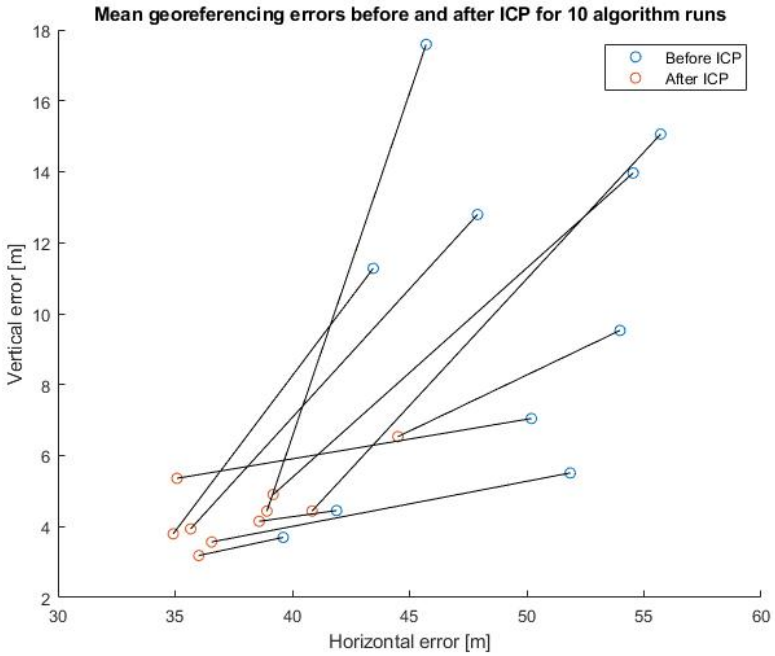


Figure 4.8: Georeferencing accuracy before and after ICP for blender test 2

good ICP registration leads to a very small vertical error, even if the camera is flying several hundreds of metres above the targets.

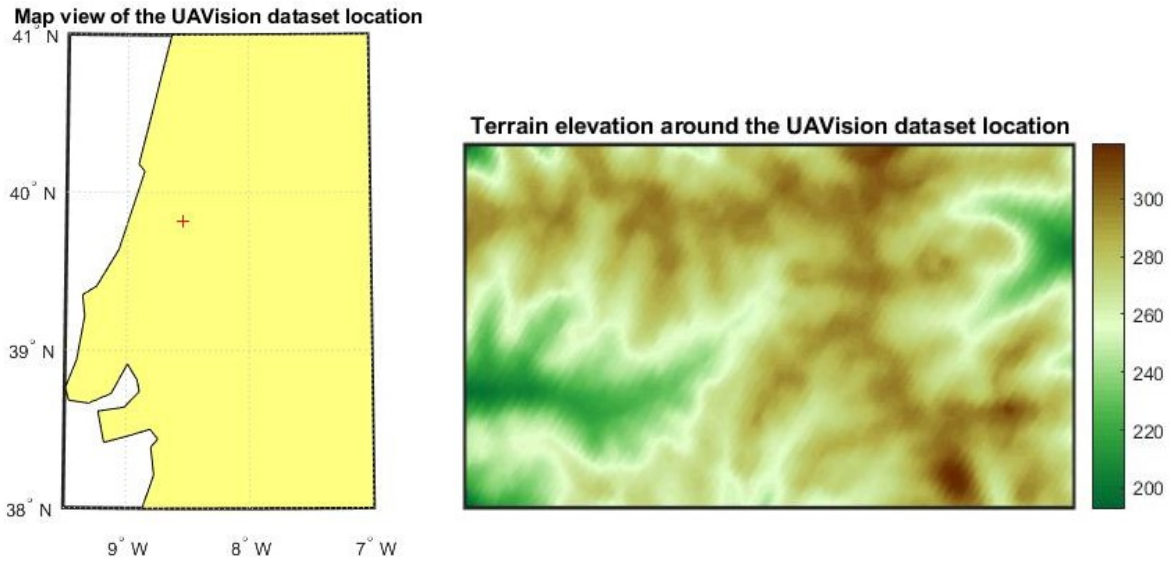
Overall, the results presented in subsections 4.1.1 and 4.1.2 show that a Sfm+ICP georeferencing algorithm works in favourable conditions. Blender test 1, the simplest of the two datasets presented so far, showed that this algorithm can be used to accurately and simultaneously reconstruct hundreds of hectares and georefer a large number of targets. After that, blender test 2 proved the algorithm's robustness to complex camera motion. This test also proved the merits of using an ICP step after Sfm, as it consistently reduced the georeferencing error.

The next two subsections present the algorithm's response to real datasets captured in real forest fire scenarios.

4.2 UAVision dataset

The UAVision dataset is a series of video frames taken from a video captured by a fixed wing UAV loitering above a forest fire near Pombal, Leiria, at 39.832856N -8.519885E at the 16th of August 2019. Figure 4.9 shows the fire's location on a map and the terrain elevation around it.

Figure 4.10 shows four equally spaced frames from the UAVision dataset. This dataset was sampled at a rate of 2Hz from 15s of continuous footage. This is the same sampling rate as in the last blender test, but this dataset was taken from just 15s of footage, instead of blender test 2's 30s. The footage is unstable, with some sharp camera movements and video cutoffs that degrade the reconstructions



(a) Map view of the Pombal fire

(b) Terrain elevation around the Pombal fire

Figure 4.9: Map and DEM of the Pombal fire

quality, hence only a short part of the video was usable.

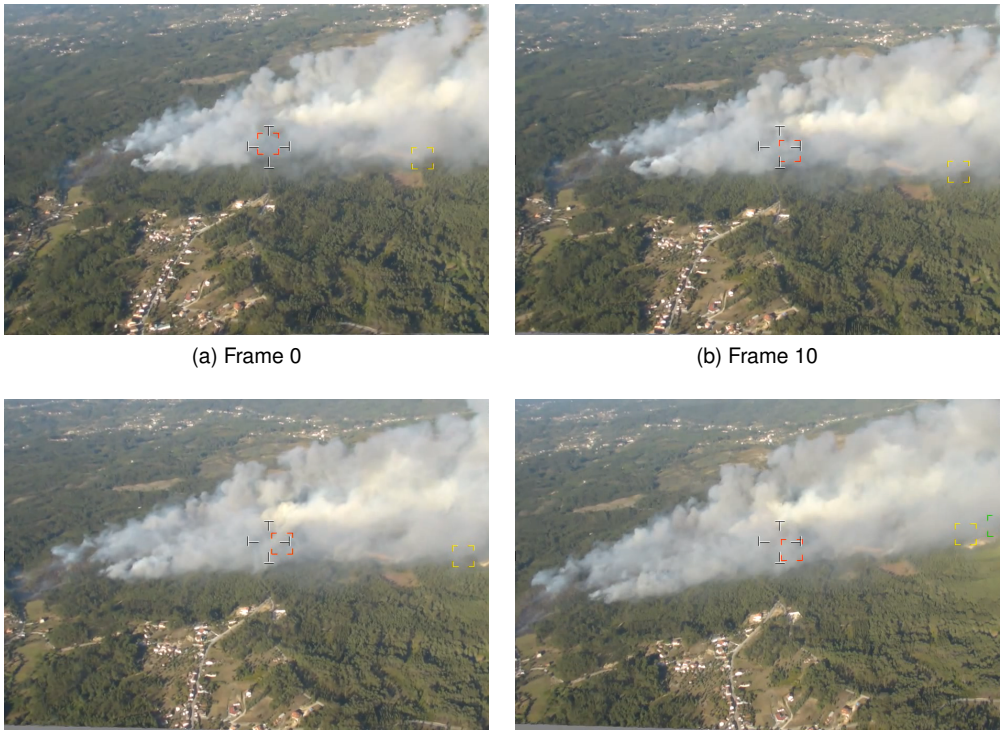
The four frames in Figure 4.10 show a typical Portuguese forest fire: a small village surrounded by a dense and vast forest and a large column of smoke above it. This column of smoke obscures part of the terrain under and behind it, hindering the ICP registration. At the same time, the forest surrounding the town is dense and has few distinct features, further complicating the reconstruction process. The last hindrance can be seen in Figure 4.9: the terrain around the fire is not as distinct as in the blender datasets, as the terrain elevation range is only 100 m and there are no large hills to assist the ICP registration. However, the terrain is still complex and distinct enough to perform Sfm+ICP georeferencing.

Figure 4.11 shows the reconstructed point and the true camera trajectory: the aircraft performs a coordinated turn heading northwest with a slight bank angle, naturally changing its distance to the fire in all three spatial coordinates. The camera changes its heading and pitch throughout the video in order to track the fire's progress, and inadvertedly changes its rotation due to flight induced disturbances.

Figure 4.12 shows a closer look on the point cloud already displayed in 4.11. These plots show each inlier feature and its color in three dimensions, after applying ICP transform. Comparing these plots with Figure 4.10, one can see that all of the village is reconstructed with a high point density.

The forests surrounding it are also well represented, although with a smaller point density. The smoke seen in 4.10 obscures the terrain northeast of the town, and most of the smoke is not considered as a feature, only a small amount of gray points hover north of the town at 400m altitude. These smoke points should be considered outliers, as they serve no purpose for the georeferencing algorithm, but the fact that a small amount of them managed to bypass the outlier removal block is not serious, as the ICP can easily classify these points as outliers, if the rest of the reconstruction is close enough to the real DEM.

Table 4.3 provides statistics regarding Sfm's performance with the UAVision dataset: 48ha were



(a) Frame 0 (b) Frame 10

(c) Frame 20 (d) Frame 30

Figure 4.10: Four equally spaced frames from the UAVision dataset

reconstructed, with an average of 96 points per hectare. While this density is not as high as in the artificial datasets, this point density is more than enough to accurately georefer all the buildings in the village and several targets outside of it, as well as the firefront itself. The low reconstruction area is mostly due to the smoke obscuring the terrain behind it, so the camera is unable to capture it.

Table 4.3: UAVision reconstruction statistics

Reconstruction area [ha]	Average points per reconstruction	Points per hectare	Points per DEM square
48	4616	96	6

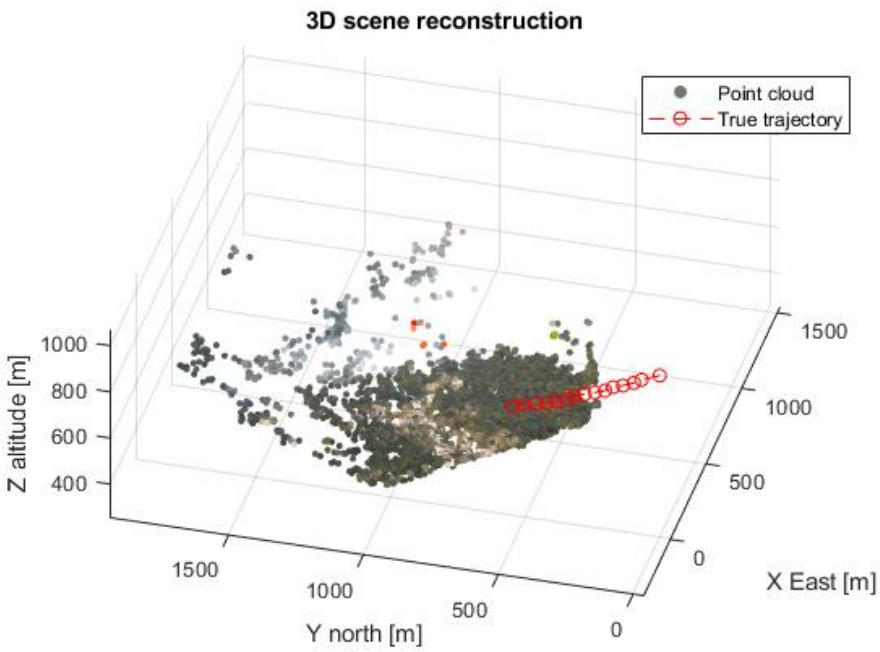


Figure 4.11: Reconstructed point cloud and real UAV trajectory

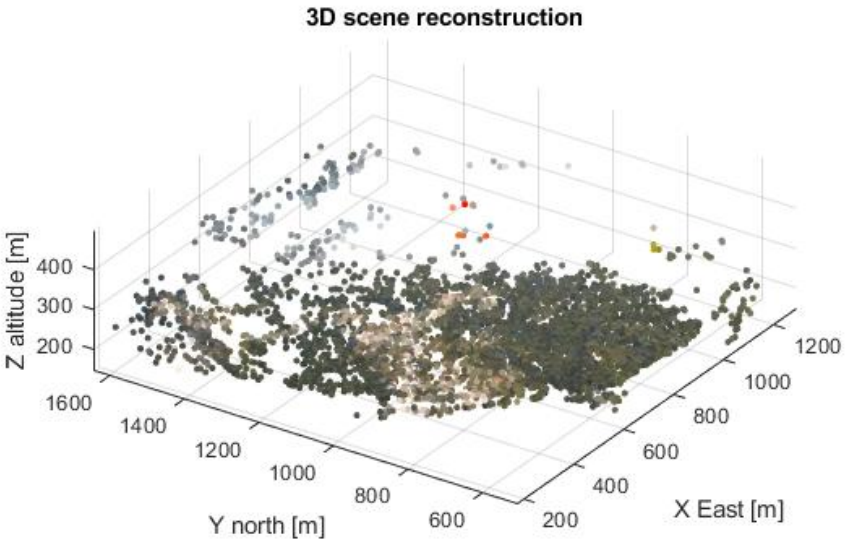


Figure 4.12: Reconstructed point cloud

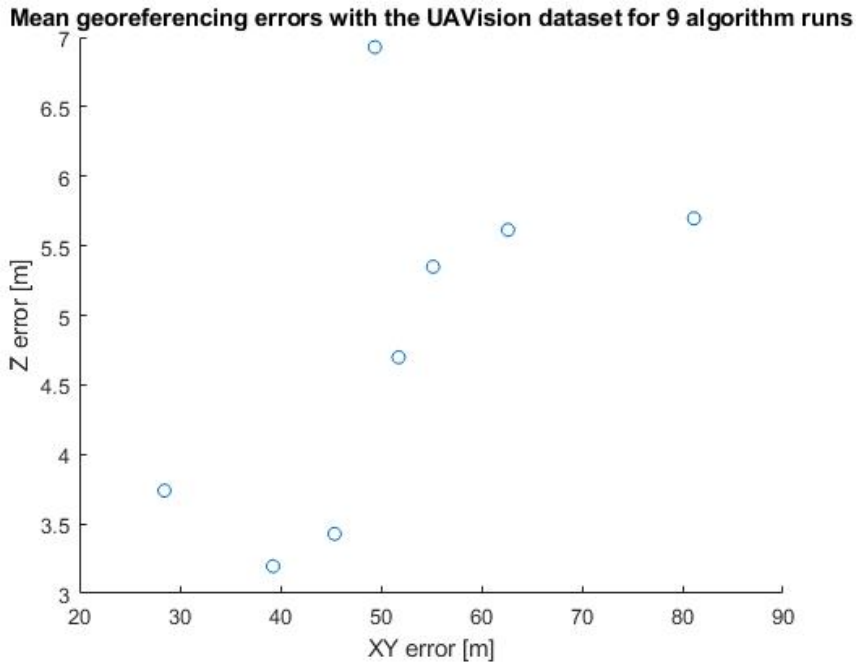


Figure 4.13: Georeferencing accuracy after ICP for the UAVision dataset

Table 4.4: Georeferencing accuracy statistics after ICP for the UAVision dataset

	XY [m]	Z [m]
Max	81.2	6.9
Mean	49.1	4.7
Median	49.4	4.7
Standard deviation	16.7	1.3
Range	52.7	3.7

Figure 4.13 shows the average XY and Z errors for several algorithm runs, and Table 4.4 presents basic statistics regarding those results. They show an average XY georeferencing error of just 49.1m, which is on par with state of the art medium altitude georeferencing algorithms. The XY results show a good error dispersion, with a relatively small standard deviation, meaning that the algorithm is, not only accurate, but also consistent when geolocating using this dataset. There is one clear outlier with 81.2m XY error, which is discussed later on, but overall, most results are roughly in the 30m-60m range.

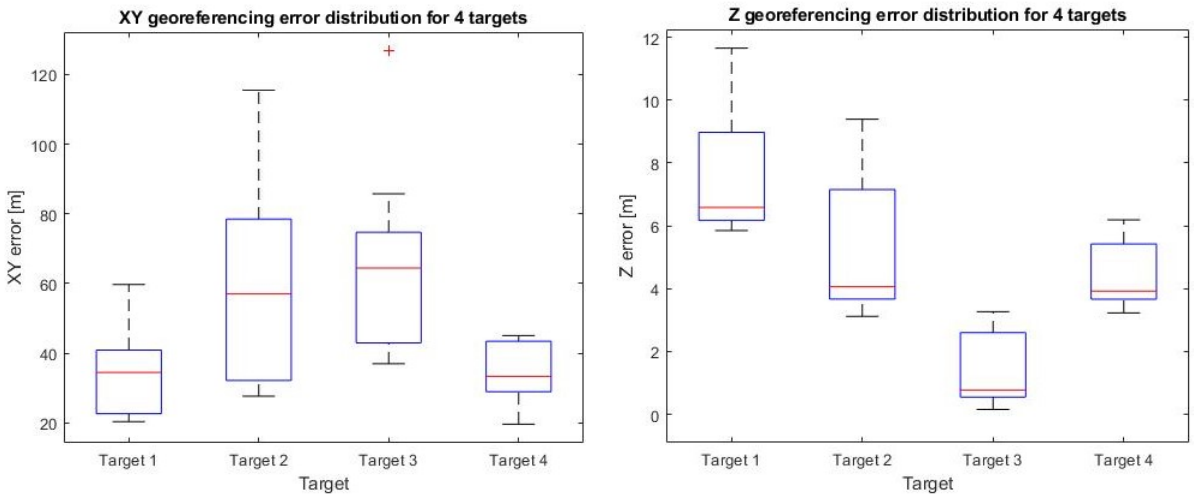
The average Z error shown in figure 4.13 and Table 4.4 is 4.7m, and its standard deviation is only 1.3m. These statistics demonstrate that Sfm+ICP can consistently and accurately determine a target's elevation, even if the camera is flying at 900m altitude. While this Z accuracy can be a useful property in some instances, a target's elevation can also be estimated by sampling a DEM at the desired coordinates. Notwithstanding, the low Z error also means that the reconstruction is consistent with the terrain, therefore the dense reconstruction is useful for more than just georeferencing, for instance, it can be used as a new dynamic map.

Figure 4.13 reveals a correlation between the XY and Z errors: reconstructions with high XY also

tend to have a high Z error, which is unsurprising due to the way that the ICP corrects both the XY and Z errors. However, it shows that, since the Z error can be in part estimated using a DEM, this correlation can be used to assess if a reconstruction is horizontally accurate or not.

Figure 4.14 shows the XY and Z error distribution for the four targets georefered with the UAVision dataset. The results show that the four targets have a low average XY error, and that some even have a similar accuracy as the DEM itself. There is one clear outlier reconstruction that had a very poor accuracy, however, apart from it, the algorithm performed consistently and accurately across all targets.

The Z error also shows a low error somewhat consistently across the four targets, however, here the outlier's impact is more clear: the mean Z error of each target (represented by red lines) is much lower than the median, for every single target. While the difference between the mean and the median for each target is no more than 2m, it shows that the vertical error is much more sensible to bad registrations than initially thought. This is likely due to this dataset's high pitch error (around 5 degrees), which makes the registration step all the more difficult. Despite this, the vertical results are still on par with the state of the art, and could be further improved if more stringent MRE and RMSE constraints were applied, at the cost of a longer algorithm run time and sparser reconstruction.



(a) XY error by target

(b) Z error by target

Figure 4.14: XY and Z error for each target

Figure 4.15 shows the mean reprojection error and ICP RMSE of the algorithm runs also shown in the previous plots, and Table 4.5 displays basic statistics regarding the MRE and RMSE. Figure 4.15 has one clear outlier with a much higher MRE and RMSE, which corresponds to the outlier in the georeferencing error shown previously. In a real scenario, the maximum allowed RMSE and MRE could be tuned so as to discard outlier reconstructions, or the algorithm could be run in parallel, after which the georeferencing estimate would either be an average of all the different threads, or a specific reconstruction would be chosen as the best, using the MRE and RMSE.

The tests performed using this Sfm+ICP algorithm have shown that, looking at the 8 inlier reconstructions shown in the bottom of figure 4.15, one cannot predict which specific one offers the best georeferencing estimate. However, clear MRE and RMSE outliers, such as the one on the top right of

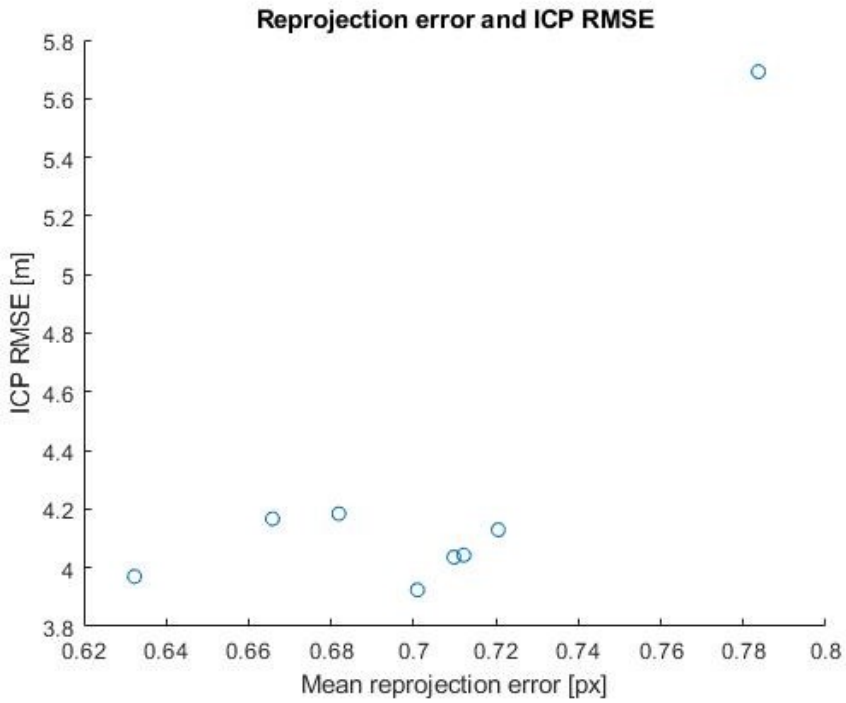


Figure 4.15: Reprojection error and ICP RMSE for several algorithm runs

figure 4.15, can be confidently classed as georeferencing outliers as well, which is unsurprising, as the MRE measures a Sfm reconstruction's quality, and the RMSE measures the registration's performance: if Sfm or ICP have sub-par outcomes, then so does Sfm+ICP.

Table 4.5 shows an average MRE across all 9 reconstructions of just 0.70px, which is satisfactory, but a lower MRE would be achievable, if the camera's IP were known more precisely, especially its distortion values. The average RMSE (from the ICP) is just 4.25m, but this low value is mostly due to the flatness of the surveyed terrain.

Table 4.5: Average Mean reprojection error and ICP RMSE for the UAVision dataset

	MRE [px]	RMSE [m]
Min	0.63	3.92
Max	0.78	5.69
Mean	0.70	4.25
Median	0.71	4.13

Overall, the results from this dataset prove this algorithm can perform georeferencing accurately and precisely with a real video captured in a forest fire environment. This subsection has shown Sfm+ICP can accurately reconstruct a large dynamic disaster scenario even with low quality imagery and severely noisy sensors which would render any optic-ray-based method useless. It was also demonstrated that the aircraft does not need to be flown at low altitude to get a good accuracy, as these results were achieved with images from an aircraft flying at 900m, well above the fire. This increased target distance did result in a larger error and standard deviation, however these estimates were still on par with medium altitude state of the art papers. Even more importantly, these results were obtained from just 15s of footage and still had a good georeferencing accuracy, which proves Sfm+ICP could be used in real time

firefighting applications in the future.

Subsection 4.3 further analyzes the algorithm's performance in a real situation in a high altitude survey above a forest fire.

4.3 Air force dataset

The air force dataset is comprised of a series of frames from a video taken aboard a manned fixed wing aircraft on the 3rd September 2019 above a forest fire raging near Freixiel, Bragança, at 41.310040N -7.225683E. These frames cannot be shared in this work for legal reasons, nonetheless, a satellite image of the fire area cane be seen in figure 4.16.

Figure 4.16: Satellite image of the Freixiel fire

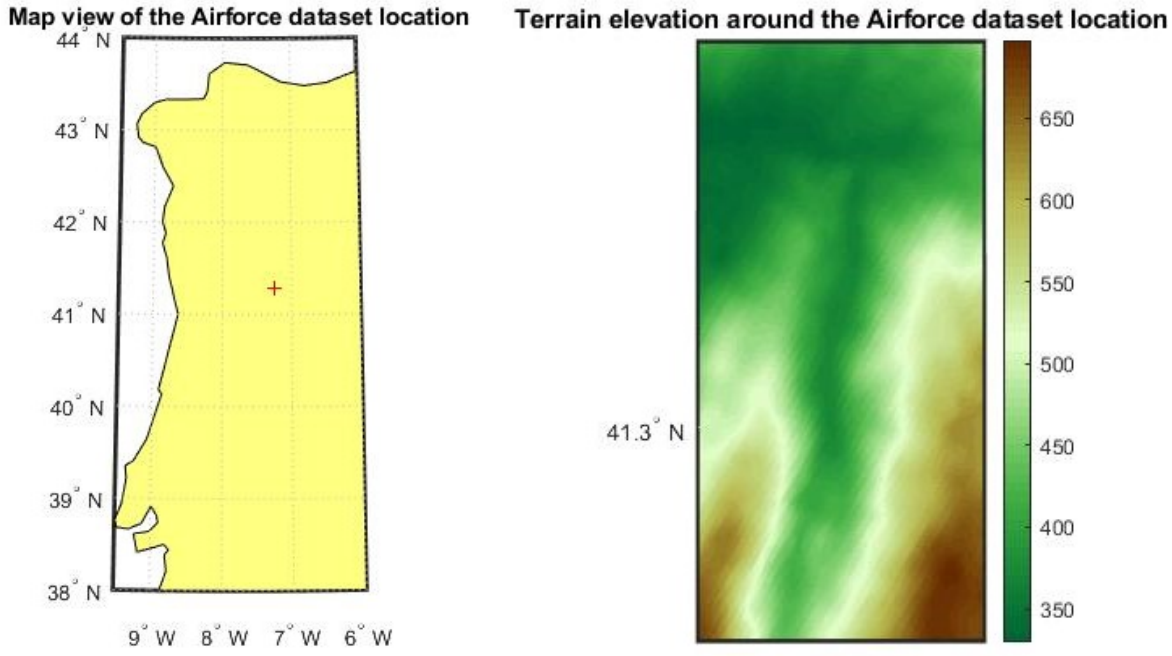
These frames are similar to the ones in Figures 4.3, 4.7 and 4.10, with two notable exceptions: the aircraft is flying at 1900m of altitude and the camera is operating with 40% optical zoom, which increases the focal length by the same amount.

Table 4.6: Airforce dataset reconstruction statistics

Reconstruction area [ha]	Average points per reconstruction	Points per hectare	Points per DEM square
215	8686	40.4	2.53

The aircraft flies at 1900m above the terrain shown in Figure 4.17, heading east with the camera focused on the eastern most hill, where the fire is raging. This terrain is much more hilly than as in the UAVision test, as the top of the mountain is more than 300m above the valley. This clear terrain feature helps the ICP registration, which now has a simple terrain pattern to match. On the other hand, the aircraft is flying over a remote region, with no houses and only scarce roads, unlike the previous dataset, where the feature detection algorithm had plenty of clear image features to track across images due to the abundance of houses and roads.

This helps explain the reconstruction obtained and shown in Figure 4.18, which has a much lower point cloud density than the reconstructions presented in the previous subsections. The high camera altitude also helps explain the higher point sparsity: being farther away from the surface makes reliably tracking features across images more difficult. Table 4.6 shows statistics regarding the reconstruction shown in Figure 4.18. A total area of 215ha was reconstructed with 2.53 point per DEM square, which is enough to perform the ICP registration, but with laxer parameters, as the point density is less than



(a) Map view of the Freixiel fire

(b) Terrain elevation around the Freixiel fire

Figure 4.17: Map and DEM of the Freixiel fire

half than in the UAVision dataset. Despite the reduced point density, the reconstructed point cloud is far bigger and is able to georefer targets several kilometers apart.

This highlights the usefulness of high altitude surveys: the higher the camera is, the faster it can survey a specific area. The major downside of high altitude imagery is that its resolution and overall quality needs to be significantly higher than for medium altitude, which increases deployment costs and the algorithm's run time. Unfortunately, the frames used for this test were only 990x1220, hence why the results were sub-optimal.

Figure 4.19 shows the average XY and Z errors obtained after running the algorithm several sequential times with the same data. The horizontal error is much larger in this dataset when compared to the previous datasets' results, however, this was to be expected as the camera is more than two times more distant to its targets in this dataset. In fact, the average error is similar to the one presented by Santana in [5], in which a similar air force dataset was used.

While the XY error increased quadricatically with the altitude, the Z error is similar to the last real dataset. This is an important aspect of this algorithm: because it matches the recovered point cloud with a real DEM, the elevation error can only be high if the Sfm process fails, or if the telemetry errors are too severe. In theory this should also happen with the horizontal error, if the image's resolution is high enough.

Table 4.7 provides statistics on the results shown in Figure 4.19. The XY mean and median is around 2.5 times larger than in the UAVision dataset, which shows the massive accuracy reduction accrued when performing high altitude surveys, however these values are still on par with the few high altitude georeferencing papers. As expected, the vertical error's mean, median and standard deviation

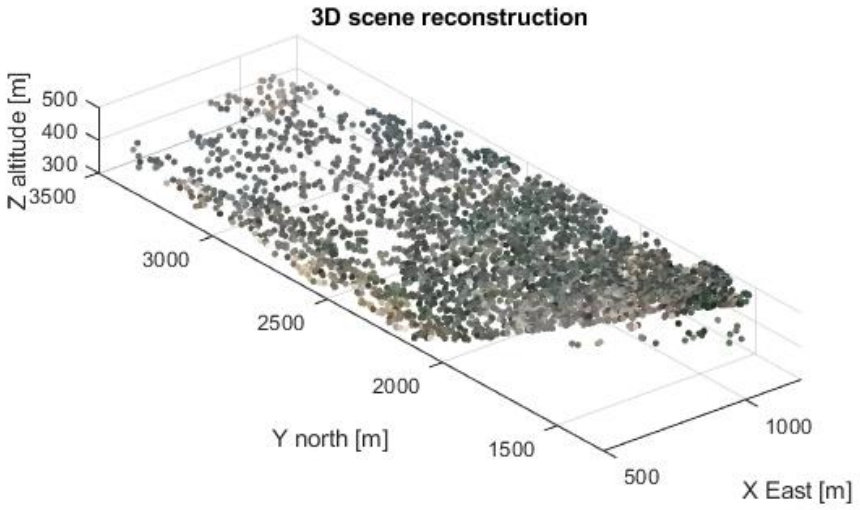


Figure 4.18: Reconstructed point cloud

Table 4.7: Georeferencing accuracy statistics after ICP for the air force dataset

	XY [m]	Z [m]
Max	235.8	19.6
Mean	136.3	11.7
Median	119.2	12.2
Standard deviation	58.9	5.0
Range	155.4	14.3

are extremely small when compared to the 1900m flight altitude, nonetheless, the vertical error's range is much higher than in the medium altitude datasets.

Figure 4.20 shows the XY and Z error distribution for four targets whose location was estimated during several algorithm runs. The XY errors have uniformly much bigger means and uncertainties than in the previous datasets, and reach very high values. This can be explained by both the increased camera to target distance and the increased focal length of the video, making its frames even more susceptible to the aircraft's vibrations. Notwithstanding, the error relative to the distance to the target is satisfactory.

The vertical error is even more satisfactory, as the four targets have similar vertical error means, indicating that the reconstruction is more even than the one from the UAVision dataset. Similarly to the XY error, the Z error's max value is rather large, stemming from the same issue: some algorithm iterations perform poor reconstructions that lead to bad ICP transforms which do not rectify the point cloud's misalignment correctly, leading to small errors on some targets, and to larger errors on more distant ones.

Figure 4.21 shows the mean XY error with each run's MRE and RMSE. These types of plots provide insights into why the georeferencing error is so much higher than in the previous datasets, and what

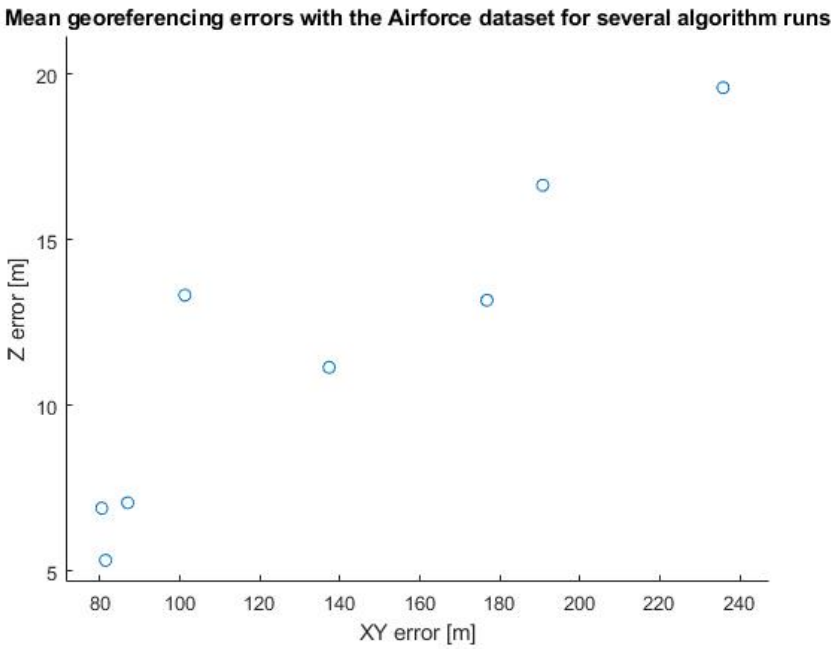


Figure 4.19: Georeferencing accuracy after ICP for the air force dataset

Table 4.8: Average Mean reprojection error and ICP RMSE for the air force dataset

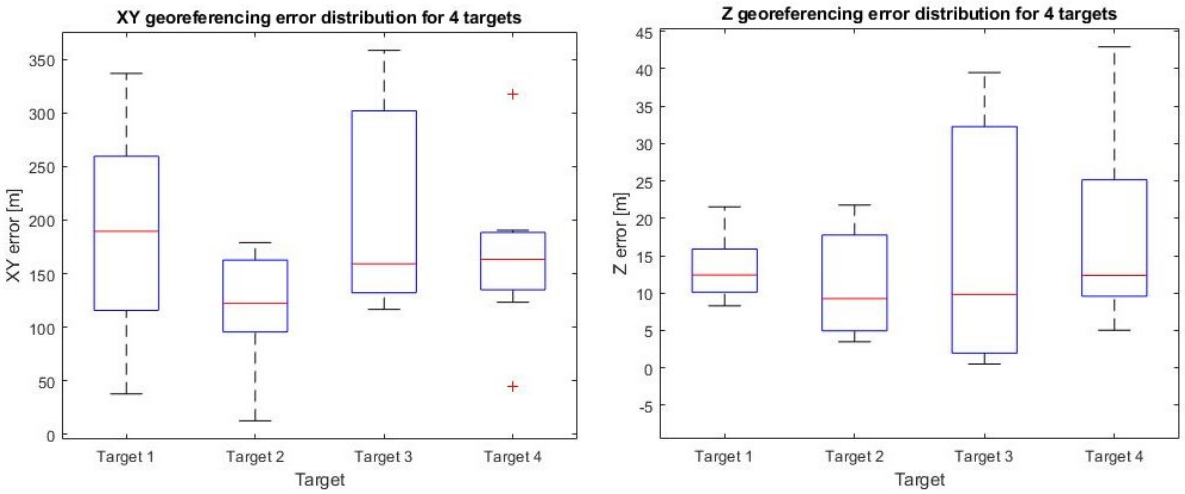
	MRE [px]	RMSE [m]
Min	1.89	8.05
Max	2.55	15.65
Mean	2.24	11.42
Median	2.21	11.31

could be done to reduce it. This figure shows a clear correlation between mean reprojection error and XY error: all low MRE reconstructions have a good accuracy. This is of course to be expected, but it proves that having a stricter reprojection error threshold could increase the algorithm's accuracy, at the cost of it being much slower.

On the other hand, Figure 4.21 shows that there is not clear relationship between the ICP's RMSE and the XY error. While the best reconstructions tend to have a lower RMSE, not all low RMSE data points have a low XY error. This demonstrates that whilst the ICP can improve Sfm's accuracy, it struggles to do so when the reconstruction has a large reprojection error (which here is 3 times larger than in the UAVision dataset).

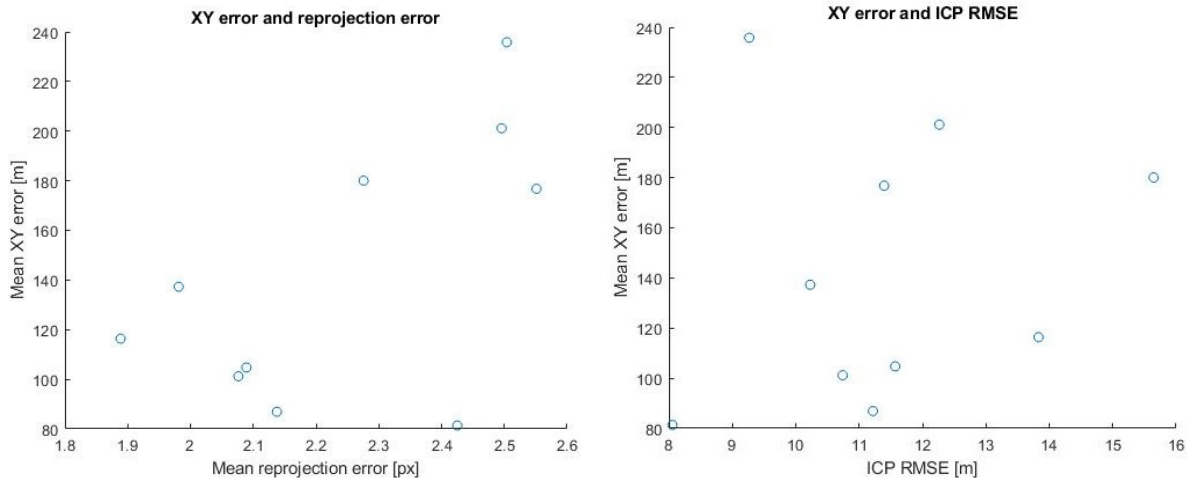
Figure 4.22 shows the Z georeferencing error for several algorithm runs, alongside their MRE and RMSE. Similarly the XY error analyzed in the last paragraphs, this figure points to a clear correlation between mean reprojection error and vertical georeferencing error, with most high accuracy data points having a low MRE. Interestingly, Figure 4.22 also shows correlation between Z error and the ICP's RMSE, which does not happen with the $XYerror$. This is due to the fact the ICP can more easily rectify the point cloud's elevation rather than its heading and translation.

Overall, this dataset produced mixed results: on the one hand, they proved that the algorithm can be used to for high altitude georeferencing with a satisfactory accuracy and preciseness. On the other



(a) XY error by target (b) Z error by target

Figure 4.20: XY and Z error for each target of the air force dataset



(a) XY error with MRE (b) XY error with RMSE

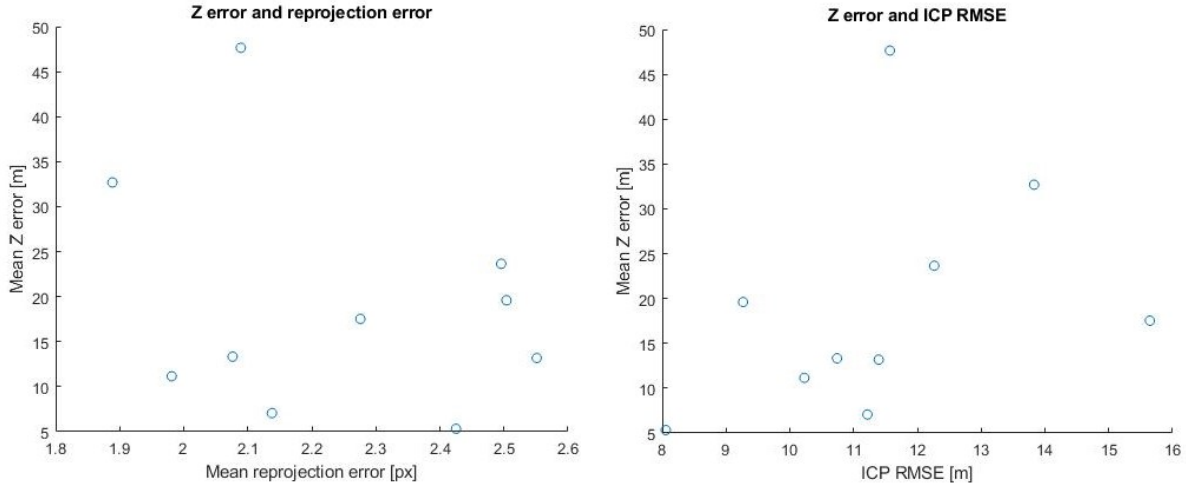
Figure 4.21: XY error with MRE and RMSE

hand, the results were significantly worse than the ones presented in sections 4.1 and 4.2, showing how quickly the georeferencing accuracy degrades with altitude.

Some of this degradation is due to the low image quality and the demanding environmental conditions faced, thus new tests with better quality high altitude datasets should be performed. Unfortunately, no such datasets are available yet.

4.4 General result discussion

Section 4.1 presented the outcome of two tests performed using Blender datasets. These tests proved that this Sfm+ICP algorithm can be used to perform medium altitude georeferencing in ideal conditions, rivalling state of the at methods in its accuracy and preciseness.



(a) Z error with MRE

(b) Z error with RMSE

Figure 4.22: Z error with MRE and RMSE

Section 4.2 analyzed results obtained with a UAVision dataset. This test proved the algorithm's merits in a real medium altitude survey: in 3D reconstruction and georeferencing.

Section 4.3 evaluated the algorithm's performance using a high altitude forest fire air force dataset. This proved the algorithm can still work satisfactorily in adverse high altitude conditions, however, the increased altitude brought a much higher XY georeferencing error.

Overall, these experiments show that this Sfm+ICP algorithm has significant disaster monitoring potential for its ability to:

- Perform dense reconstructions quickly and accurately
- Accurately georefer multiple targets at the same time
- Work even with highly noisy navigational data and low resolution images
- Act as an initial EP refinement step for optic-ray-based georeferencing algorithms

Chapter 5

Conclusions

Contents

This chapter presents this work's final conclusions: section 5.1 presents its achievements in the fields of georeferencing and Sfm, and section 5.2 proposes future improvements to the algorithm.

5.1 Achievements

This work presented a new and robust Sfm+ICP georeferencing algorithm designed for medium altitude forest fire monitoring. Its secondary achievement is the algorithm's ability to perform a reconstruction of the area around the fire. Both these features are useful to the FIREFRONT project, as they further the project's ability to perform real time georeferencing in adverse conditions, while also opening the door to other Sfm based research in the field of forest fire monitoring.

The results obtained with the blender datasets proved the georeferencing potential of a Sfm+ICP method: the algorithm was able to georefer several targets with state of the art precision, even with considerably high induced telemetry errors. The georeferencing precision shown by these blender results is specially remarkable, since the simulated aircraft was operating at 900m altitude, and still had a 30m horizontal precision.

The blender reconstructions also prove the algorithm's reconstruction ability and potential. Several hundred hectares were reconstructed in real time with state of the art precision, all while flying well above most paper's stated altitude. This hints on one of the major possible uses of this kind of algorithms: real time disaster reconstruction. The ability to seamlessly georefer and reconstruct a disaster provides civil authorities with tools to know how and where a certain disaster, minute by minute, which until now was impossible to do.

The UAVision dataset was used to validate the algorithm with a real medium altitude forest fire mon-

itoring footage. This dataset's georeferencing results were naturally worse than the simulated blender datasets', however, they were still on par with state of the art algorithms. These results proved that Sfm+ICP can be used for real time forest fire georeferencing, which is something very few georeferencing papers can claim.

The UAVisison dataset's reconstruction was similar in both density and precision to the blender one, which highlights the algorithm's robustness to poorly calibrated cameras and low quality telemetry. The reconstruction was obtained from just a few seconds of real footage, and was still able to capture several dozen hectares with good point density. This specific aspect is particularly useful for firefighting authorities, since reconstructing the fire area quickly and precisely can be useful to study a fire's progress and plan its extinction. This form of dense reconstruction is also useful to fine tune other georeferencing algorithms that might rely on surface maps.

The air force dataset presented the algorithm with its most difficult task: shaky footage taken at high altitude, of a remote area, using a poorly calibrated camera. The georeferencing precision was 4 times worse than the previous datasets, however, this level of precision is well above other high altitude georeferencing tests. This precision is especially remarkable, since Sfm is rarely used at such high altitudes, but the ICP block enables this Sfm algorithm to perform decently well even at high altitude. In theory, this Sfm+ICP method should not degrade in precision as altitude increases, provided the camera is well calibrated and the footage has enough resolution, however, this dataset failed to meet both this clauses.

The Air fore reconstruction encompassed a 215ha area, well above the previous dataset's reconstructions. This is the major pro of using high altitude footage: the algorithm can reconstruct an incredibly vast area using only a few seconds worth of footage. The reconstruction's density and precision was naturally lower than as in the medium altitude datasets, but it was still good enough for georeferencing purposes. This high altitude capability is specially useful in large forest fires, in which the monitoring aircraft are forced to operate above 1000m.

This work's main novel contribution is a Sfm method that can operate in remote disaster areas in real time. No other paper can claim this, making this the first Sfm-based real time georeferencing paper.

Its secondary achievement is its use of DEM registration to refine the dense point cloud provided by Sfm. While using ICP after Sfm is not a novelty in itself, this paper is the first to use ICP between a DEM and a dense point cloud for medium altitude georeferencing.

Sfm+ICP's main pro is its precision, when compared to other medium altitude georeferencing solutions, which perform poorly with high error telemetry and degrade quickly with altitude. The algorithm is also able to perform dense scene reconstructions while performing the georeferencing itself, which might be helpful for other georeferencing algorithms. This integration potential is particularly relevant, since it hints that direct georeferencing methods might greatly benefit from an initial Sfm step, to counteract their inherent dependency on faulty telemetry.

5.2 Future Work

The next step in Sfm+ICP research is to test it in real time with a predefined flight pattern and an accurate way to assess the algorithm's georeferencing accuracy. This endeavour would also be invaluable to other georeferencing algorithms, as all of them would be uniformly and accurately evaluated using a dataset especially made for just that.

Moreover, the Sfm+ICP algorithm could be improved by implementing a series of modifications such as:

- Coupling the KLT feature tracker with other feature extractors, thereby increasing the algorithm's ability to reconstruct and georefer more complex environments, and allowing it to create denser and more accurate reconstructions.
- Performing several different ICP variations per algorithm run, as having several ICPs would fix several types of sensor noise. Applying several ICPs with different parameters could also allow the algorithm to provide a better estimate of the scale parameter, which would improve its high altitude accuracy.
- Integrating the Sfm+ICP method with an optic-ray georeferencing algorithm such as the one proposed by Santana [5]. The optic-ray-based algorithm would refine the scale factor estimate, while Sfm+ICP would reduce the sensor noise's impact.
- Integrating this algorithm with a fire segmentation algorithm also developed by the FIREFRONT team, which could feed SfM+ICP automatically with the image pixels corresponding to the firefront and smoke. This would streamline the user's interaction and reduce he number of feature outliers.
- Applying the Sfm+ICP technique in a Simultaneous localization and mapping (SLAM) process. While SLAM's main goal is not to perform georeferencing, it can be used to perform longer reconstructions, thereby incorporating more measurements and images in each georeferencing estimate. Even though using SLAM is an highly unorthodox approach to the georeferencing problem, it can be useful in forest fire monitoring, since most missions perform circular flights around the targets, meaning the SLAM algorithm would be able to perform several loop closures.
- Incorporating several image sources in each Sfm run, for example: images from several UAVs and from several ground control points with a good view of the fire. This could help solve the scale factor problem, and massively increase the algorithm's georeferencing accuracy.
- Implementing the algorithm in a more optimal way, run-time-wise. Improving Sfm+ICP's time efficiency would also open the doors to a more profound study of this approach's limits, in terms of having denser reconstructions, multi-threaded Sfm+ICP methods and being able to process longer image sequences.

Bibliography

[1] M. Alonzo, H. E. Andersen, D. C. Morton, and B. D. Cook. Quantifying boreal forest structure and composition using UAV structure from motion. *Forests*, 9(3):1–15, 2018. ISSN 19994907. doi: 10.3390/f9030119.

[2] F. Mancini, M. Dubbini, M. Gattelli, F. Stecchi, S. Fabbri, and G. Gabbianelli. Using unmanned aerial vehicles (UAV) for high-resolution reconstruction of topography: The structure from motion approach on coastal environments. *Remote Sensing*, 5(12):6880–6898, 2013. ISSN 20724292. doi: 10.3390/rs5126880.

[3] P. Miller, J. Mills, S. Edwards, P. Bryan, S. Marsh, H. Mitchell, and P. Hobbs. A robust surface matching technique for coastal geohazard assessment and management. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(5):529–542, 2008. ISSN 09242716. doi: 10.1016/j.isprsjprs.2008.02.003.

[4] C. Yuan, Y. Zhang, and Z. Liu. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian Journal of Forest Research*, 45(7):783–792, 2015. ISSN 12086037. doi: 10.1139/cjfr-2014-0347.

[5] B. Santana, A. Bernardino, and R. Ribeiro. Direct Georeferencing of Forest Fire Aerial Images using Iterative Ray-Tracing and a Bearings-Range Extended Kalman Filter. Master's thesis, Tecnico Lisboa, 2020.

[6] M. Beighley and A. C. Hyde. Portugal Wildfire Management in a New Era: Assessing Fire Risks, Resources and Reforms. *Independent report*, 9(1):52, 2018. URL `https://www.isa.ulisboa.pt/files/cef/pub/articles/2018-04/2018_Portugal_Wildfire_Management_in_a_New_Era_Engish.pdf`.

[7] Forest fires — European Environment Agency. `https://www.eea.europa.eu/data-and-maps/data/external/peseta-iii-projections-of-fire`, 2019. Accessed: 2021-05-14.

[8] Projections of Fire Weather Index (PESETA III) — European Environment Agency. `https://www.eea.europa.eu/data-and-maps/data/external/peseta-iii-projections-of-fire`. Accessed: 2021-05-14.

[9] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017. ISSN 14740508. doi: 10.1017/S096249291700006X.

[10] J. L.Carrivick. *Structure from motion in the geosciences*. 1377. ISBN 9781118895849.

[11] M. Beyeler. *OpenCV with Python Blueprints Design and develop advanced computer vision projects using OpenCV with Python*. 2015. ISBN 9781785282690.

[12] R. Hartley and A. Zisserman. Multiple View Geometry in Computer vision. *Computer-Aided Design*, 16(2):672, 2003. URL http://journals.cambridge.org/production/action/cjoGetFulltext?fulltextid=289189.

[13] J. Iglhaut, C. Cabo, S. Puliti, L. Piermattei, J. O'Connor, and J. Rosette. Structure from Motion Photogrammetry in Forestry: a Review. *Current Forestry Reports*, 5(3):155–168, 2019. ISSN 21986436. doi: 10.1007/s40725-019-00094-3.

[14] Y. Jinxi, H. Qingqing, S. Meixian, W. Linjun, L. Hui, L. Ouya, and W. Haitao. Bundle Adjustment —A Modern Synthesis. *Zhonghua Wei Zhong Bing Ji Jiu Yi Xue*, 28(6):563–566, 2016. ISSN 20954352. doi: 10.3760/cma.j.issn.2095-4352.2016.06.018.

[15] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. S. Generic, and R. T. Structure. Generic and Real Time Structure from Motion using Local Bundle Adjustment To cite this version : HAL Id : hal-01635634 Generic and Real Time Structure from Motion using Local Bundle Adjustment 1. 2017.

[16] S. Jiang, C. Jiang, and W. Jiang. Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167(September 2019):230–251, 2020. ISSN 09242716. doi: 10.1016/j.isprsjprs.2020.04.016. URL https://doi.org/10.1016/j.isprsjprs.2020.04.016.

[17] M. Lhuillier. Fusion of GPS and structure-from-motion using constrained bundle adjustments. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3025–3032, 2011. ISSN 10636919. doi: 10.1109/CVPR.2011.5995456.

[18] M. Ravanbakhsh and C. S. Fraser. A comparative study of DEM registration approaches. *Journal of Spatial Science*, 58(1):79–89, 2013. ISSN 14498596. doi: 10.1080/14498596.2012.759091.

[19] EU-DEMv1.1-flyer. https://land.copernicus.eu/user-corner/publications/eu-dem-flyer/at_download/file. Accessed: 2021-05-14.

[20] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. ISSN 01628828. doi: 10.1109/34.121791.

[21] F. Pomerleau, F. Colas, and R. Siegwart. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015. ISSN 1935-8253. doi: 10.1561/2300000035.

[22] Y. Sheng. Comparative evaluation of iterative and non-iterative methods to ground coordinate determination from single aerial images. *Computers and Geosciences*, 30(3):267–279, 2004. ISSN 00983004. doi: 10.1016/j.cageo.2003.11.003.

[23] H. Xiang and L. Tian. Method for automatic georeferencing aerial remote sensing (RS) images from an unmanned aerial vehicle (UAV) platform. *Biosystems Engineering*, 108(2):104–113, 2011. ISSN 15375110. doi: 10.1016/j.biosystemseng.2010.11.003. URL `http://dx.doi.org/10.1016/j.biosystemseng.2010.11.003`.

[24] C. Xu, D. Huang, and J. Liu. Target location of unmanned aerial vehicles based on the electro-optical stabilization and tracking platform. *Measurement: Journal of the International Measurement Confederation*, 147:106848, 2019. ISSN 02632241. doi: 10.1016/j.measurement.2019.07.076. URL `https://doi.org/10.1016/j.measurement.2019.07.076`.

[25] M. Lhuillier. Incremental Fusion of Structure-from-Motion and GPS using Constrained Bundle Adjustment. 34(12):1–20, 2017.

[26] G. Forlani, F. Diotri, U. M. di Cella, and R. Roncella. Indirect UAV strip georeferencing by on-board GNSS data under poor satellite coverage. *Remote Sensing*, 11(15), 2019. ISSN 20724292. doi: 10.3390/rs11151765.

[27] A. Matese, S. F. Di Gennaro, and A. Berton. Assessment of a canopy height model (CHM) in a vineyard using UAV-based multispectral imaging. *International Journal of Remote Sensing*, 38 (8-10):2150–2160, 2017. ISSN 13665901. doi: 10.1080/01431161.2016.1226002. URL `http://dx.doi.org/10.1080/01431161.2016.1226002`.

[28] G. Conte, M. Hempel, P. Rudol, D. Lundström, S. Duranti, M. Wzorek, and P. Doherty. High accuracy ground target geo-location using autonomous micro aerial vehicle platforms. *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 1–14, 2008. doi: 10.2514/6.2008-6668.

[29] X. Zhuo, T. Koch, F. Kurz, F. Fraundorfer, and P. Reinartz. Automatic UAV image geo-registration by matching UAV images to georeferenced image data. *Remote Sensing*, 9(4):1–25, 2017. ISSN 20724292. doi: 10.3390/rs9040376.

[30] F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. B. Schön, and F. Gustafsson. Geo-referencing for UAV navigation using environmental classification. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1420–1425, 2010. ISSN 10504729. doi: 10.1109/ROBOT.2010.5509424.

[31] L. H; and D. W. X. Image Registration and Fusion of Visible and Infrared Integrated Camera for Medium-Altitude Unmanned Aerial Vehicle Remote Sensing. *Remote Sensing*, 2017. doi: 10.3390/rs9050441.

[32] M. Hamidi and F. Samadzadegan. Precise 3D geo-location of UAV images using geo-referenced data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Informa-*

tion Sciences - ISPRS Archives, 40(1W5):269–275, 2015. ISSN 16821750. doi: 10.5194/isprsarchives-XL-1-W5-269-2015.

[33] D. g. Sim and R. h. Park. Localization Based on DEM Matching Using Multiple Aerial Image Pairs. *EEE Transactions on Image Processing*, 11(1):52–55, 2002.

[34] J. J. Rodríguez and J. K. Aggarwal. Matching Aerial Images to 3-D Terrain Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1138–1149, 1990. ISSN 01628828. doi: 10.1109/34.62603.

[35] S. Carreño, P. Wilson, P. Ridao, and Y. Petillot. A survey on terrain based navigation for AUVs. *MTS/IEEE Seattle, OCEANS 2010*, 2010. doi: 10.1109/OCEANS.2010.5664372.

[36] Set or query heading angle of camera for geographic globe - MATLAB camheading. URL `https://www.mathworks.com/help/map/ref/globe.graphics.geographicglobe.camheading.html`.

[37] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. *Computer Vision–ECCV 2006*, pages 404–417, 2006. URL `http://link.springer.com/chapter/10.1007/11744023_32`.

[38] B. D. Lucas and T. Kanade. Iterative Image Registration Technique With an Application To Stereo Vision. 2(September):674–679, 1981.

[39] S. S. NN Brushlinsky, M Ahrens. World Fire Statistics. `https://ctif.org/sites/default/files/2018-06/CTIF_Report23_World_Fire_Statistics_2018_vs_2_0.pdf`, 2018. Accessed: 2021-11-1.

[40] A. Mouratidis and D. Ampatzidis. European digital elevation model validation against extensive global navigation satellite systems data and comparison with SRTM DEM and ASTER GDEM in Central Macedonia (Greece). *ISPRS International Journal of Geo-Information*, 8(3), 2019. ISSN 22209964. doi: 10.3390/ijgi8030108.