Enhancing Search and Rescue Operations: A Pragmatic Application of User-Centered Development

Tyler Choi

A Capstone Project Report
Submitted in partial fulfillment of the
Requirements of the degree of

Master of Science in Computer Science & Software Engineering

University of Washington

May, 30th, 2023

Project Committee:

Kelvin Sung, Committee Chair

Michael Stiber, Committee Member

Yusuf Pisan, Committee Member

Abstract

This capstone paper investigates the development of a software solution tailored for search and rescue (SAR) operations, with a particular emphasis on evaluating the implementation and effectiveness of user-centered development (UCD) principles. Initially, the project aimed to create a Virtual Reality (VR) Interactive Topographical Mapping System. This phase resulted in the research and development of a sophisticated VR prototype, incorporating a comprehensive suite of features that facilitated live, interactive topographical mapping within a 3D virtual environment.

The objectives of UCD involve placing users' needs and requirements at the forefront of the design process, ensuring that solutions not only possess technical prowess but also deliver value and impact for the target audience. However, despite the numerous technical accomplishments of the VR project, end-user feedback from stakeholders, such as forest firefighters, revealed the necessity for a solution that better aligned with their real-world requirements. These users required direct observation of ground and vegetation conditions to make informed decisions about mission trajectories, a capability unattainable with the VR application. This insight led to a pivotal shift in our approach, redirecting the project towards the development of a targeted desktop application explicitly designed to address the operational needs of Search and Rescue (SAR) personnel.

The resulting product is a desktop application accessible through both a Graphical User Interface (GUI) and a Command Line Interface (CLI), with development centered on continuous end-user engagement and feedback. This solution offers two distinct interfaces catering to different end-users, prioritizing a concise UI and output while avoiding unnecessary complexity and irrelevant details.

In evaluating the implementation of UCD, the project demonstrates that adopting a user-centric approach can enhance the efficiency and effectiveness of SAR operations, emphasizing users' preference for utility over visual and graphical elements. Furthermore, the project's evolution from a cutting-edge VR system to a specialized desktop application provides insights into the broader fields of computer science and emergency response.

In future work, this report investigates potential enhancements, illustrating a sustained commitment to continuous improvement and alignment with user requirements. The accomplishments of the VR project, despite the pivot, attest to the importance of innovation and exploration in software development. Additionally, the project underscores the vital role of UCD in crafting solutions that combine technical utility with a focus on addressing real-world challenges.

Table of Contents

Chapter 1: Introduction

The initiation of this Capstone project was driven by the objective to create a tool that could optimize the operations of forest firefighters and emergency service responders. The initial intent was to develop a Virtual Reality (VR) Interactive Topographical Mapping System that would provide a vivid, immersive, and interactive visualization of topographical landscapes.

1.1 The First Project -- VR Topographical System

The cornerstone of our initial Virtual Reality (VR) project was to create an Interactive Topographical Mapping (ITM) system using Digital Elevation Models (DEMs). This technology has numerous applications, including surface process modeling [1] and hazard assessments [2], making it an attractive option for our initial goal: to develop an immersive and interactive visualization tool for emergency service responders and forest firefighters in rural terrains. Despite its advanced functionality, such as offering a detailed view of terrain and its features, the VR system had inherent limitations. The most significant of these was the necessity for live, in-person viewing. This requirement highlighted the need for a more flexible and adaptable solution appropriate for real-life missions.

1.2 The Second Project -- GUI/CLI

The challenges encountered with the VR system led us to reassess our approach. Instead of creating a solution first and then looking for potential users, we decided to pivot towards a more user-centered approach. This involved understanding the specifications and requirements from the end-users before developing a solution. This shift in perspective brought us to focus on understanding user needs and constraints, followed by solution design.

Our second project, therefore, was focused on designing a solution that addressed the specific needs of Search and Rescue (SAR) personnel. Through discussions with stakeholders, we identified several challenges they faced in their operations, particularly with the use of SarTopo [3], a web map software extensively used in SAR missions. SarTopo lacked a feature to compute route and area intersections and the traversed lengths. This deficiency resulted in SAR personnel having to manually trace and calculate intersection lengths, a time-consuming process that often stretched into the late hours of the night during multi-day missions, impacting the efficiency of next-day mission planning.

Based on this feedback, we developed a desktop application designed to enhance the functionality of SarTopo by incorporating the missing features. This application was developed in two formats - a Graphical User Interface (GUI) for non-technical users and a Command Line Interface (CLI) for more tech-savvy users.

1.3 Summary

Our journey began with the development of a highly interactive VR application incorporating real-world data, an educational process that resulted in an impressively engaging system. However, we learned that creating a visually stunning application does not necessarily equate to creating a practical or effective tool. The importance of user-driven development became evident, leading to our shift in focus towards solutions that were more closely aligned with end-user needs.

Our experiences underscored that there are numerous practical problems that can be significantly improved through thoughtful software design, leading to enhanced efficiency and effectiveness for individuals undertaking crucial missions. In conclusion, our journey highlighted the importance of flexibility, adaptability, and a steadfast commitment to aligning with user needs in software development. As we look ahead, these lessons will continue to guide our efforts to create impactful solutions for emergency response operations.

Chapter 2: Literature Review and Related Work of the VR Project

This chapter provides a comprehensive review of the technologies, methodologies, and systems integral to the first part of this capstone project.

2.1 Digital Elevation Models (DEM)

DEMs are raster-based representations of the Earth's surface, generated from elevation data collected through various remote sensing techniques [4].

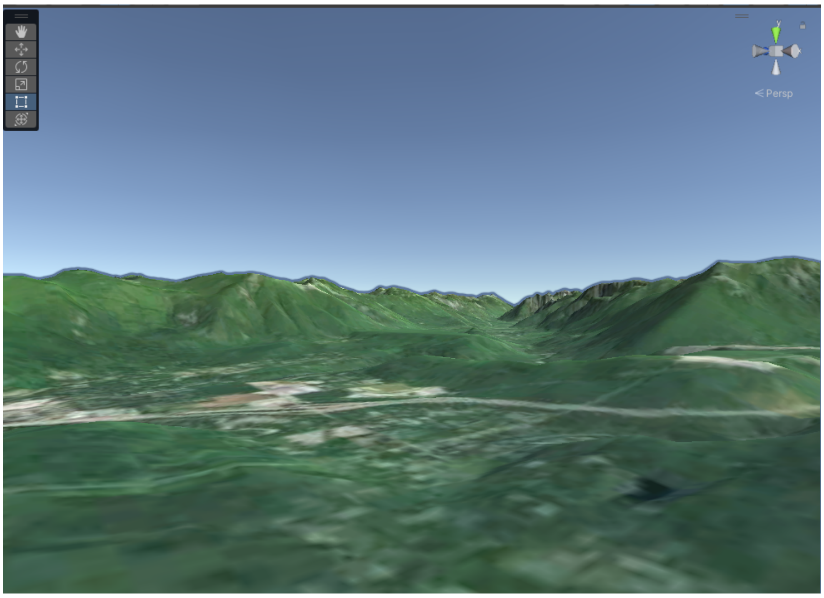


Figure 1. Visualized Digital Elevation Model (DEM)

As depicted in Figure 1, DEMs are a visual representation of the Earth's surface, including information about the shape and elevation of the terrain. This 3-dimensional visualization showcases the detailed topographical information that can be derived from a DEM, including contouring, gradients, and relative height information. This granularity of data makes DEMs crucial in geographic information systems (GIS), environmental modeling [5], hazard assessments, and terrain visualization [6].

2.2 Light Detection and Ranging (LiDAR)

LiDAR, an active remote sensing technology, uses laser light to measure distances and generate high-resolution elevation data [7]. It is capable of producing extremely accurate and detailed elevation models, capturing both ground surface and aboveground features.

2.3 DEMs vs. LiDAR: Differences, Applications, and Challenges

The primary differences between DEMs and LiDAR pertain to the data collection methods, resolution, and accuracy. DEMs, being raster-based, typically have relatively lower resolution and accuracy compared to the higher-resolution elevation models generated by LiDAR [8]. Furthermore, LiDAR, being an active sensing technology, provides finer details and more precise elevation measurements, making it suitable for applications requiring high-resolution terrain models such as urban planning, forestry management, and flood modeling [9].

However, it is important to note that the adoption of LiDAR has been slower than that of DEMs, primarily due to two factors. First, LiDAR, being a relatively newer technology, has not yet achieved the same level of accessibility as DEMs. Second, LiDAR datasets are significantly larger in size due to the fine-grained detail of images, necessitating more overhead when integrating into applications [10].

Despite these challenges, the continuous advancements in remote sensing technologies and data storage capabilities are expected to progressively mitigate these issues, making LiDAR datasets more accessible and manageable in the future. As these improvements occur, the adoption of LiDAR is expected to increase, leading to more accurate and detailed terrain models [11].

2.4 Existing Terrain Visualization Systems

Early systems like GeoVR [12] and GoogleVR [13] highlighted the potential of virtual terrain visualization using DEMs and LiDAR data. However, these systems often face limitations in real-time data processing, user-specific information augmentation, and cross-reality collaboration.

2.5 Immersive Terrain Modeling in Tactical Operations

The successful use of Immersive Terrain Modeling (ITM) systems in tactical military operations underscores the benefits of virtual terrain simulation [14]. The demand for such systems in wildfire response and management already exists, as demonstrated by the Wildland Fire Coordination Association's (WFCA) Fire Map [15].

2.6 Collaborative Efforts and Open-Source Software

Collaborative efforts, such as the Shuttle Radar Topography Mission (SRTM) [16], have led to the creation of large-scale DEMs. Additionally, open-source GIS software like QGIS [17] and GRASS GIS [18] has facilitated the integration of DEMs and LiDAR data into terrain visualization systems.

2.7 Summary and Observations

Terrain visualization systems are continuously evolving. The ongoing advancements in remote sensing, data storage, and processing capabilities, as well as the development of machine learning algorithms, are expected to contribute to the growth and diversification of terrain visualization applications [19]. By addressing current limitations and harnessing emerging technologies, terrain visualization can become an indispensable tool in various real-world scenarios, including emergency response operations, urban planning, environmental management, and military training.

Chapter 3: Project Goals and Objectives

The initial objective of this capstone project was to develop a Virtual Reality (VR) Interactive Topographical Mapping (ITM) system aimed at providing an immersive and interactive visualization of topographical landscapes for emergency service responders and forest firefighters. This ambitious endeavor was planned to be realized through the application of Digital Elevation Models (DEMs) and the utilization of the Unity Engine [20] for VR development. The primary goal was to endow users with a comprehensive understanding of terrain and its features, an element of significant importance for strategic planning in emergency response operations.

3.1 Requirements

As we ventured into the development of the Interactive Topographical Mapping (ITM) system, our focus was to design a platform resembling a modern GPS-based 2D map navigation system. However, the planned ITM system was envisioned to surpass the conventional single-user interaction by facilitating real-time collaboration among multiple users on 3D terrains. The system would empower users to jointly create, annotate, and analyze various pathways traversing the landscape. The necessary requirements for such a system included:

- *Runtime map creation capabilities*: The system is required to dynamically generate and examine 3D terrains at any specified location, utilizing Digital Elevation Models (DEMs).

- *Efficient map rendering*: The system is required to present terrains in a way that offers a clear sense of depth and scale, along with a comprehensive overview of the entire terrain.
- *Interactive map and pathway augmentation:* In line with modern GPS-based navigation systems, the system is required to enable users to create, refine, and analyze pathways through the terrain. Essential features included pathway markers, real-time user position and orientation indicators, and distance information feedback.
- *Remote cross-reality collaboration*: To facilitate collaboration among geographically diverse participants, the system is required to support different reality settings. This involved the inclusion of user-specific avatars and information, crucial for immersive examination of terrains and environments by frontline personnel.

3.2 System Design Specifications

In response to the outlined system requirements, we established the following design specifications for our ITM system:

- *Flexible Map Generation*: The system will be designed to handle DEMs of various resolutions and formats, and transform them into 3D terrains in real-time.
- *High-Quality Rendering*: The system will demonstrate high-performance graphics rendering capabilities. It is crucial to ensure that the rendering process does not compromise system performance and user experience.
- *Interactive Pathway Creation and Analysis*: The design will accommodate an intuitive and user-friendly interface for pathway creation and analysis. This includes visually distinct markers, real-time indicators for user position and orientation, and clear distance information feedback.

- *Real-time Collaborative Features*: To enable real-time collaboration, the system will be designed to support the simultaneous presence of multiple users.
- *Cross-platform Compatibility:* The system will be designed to be compatible with various devices and platforms, including desktops and VR headsets.

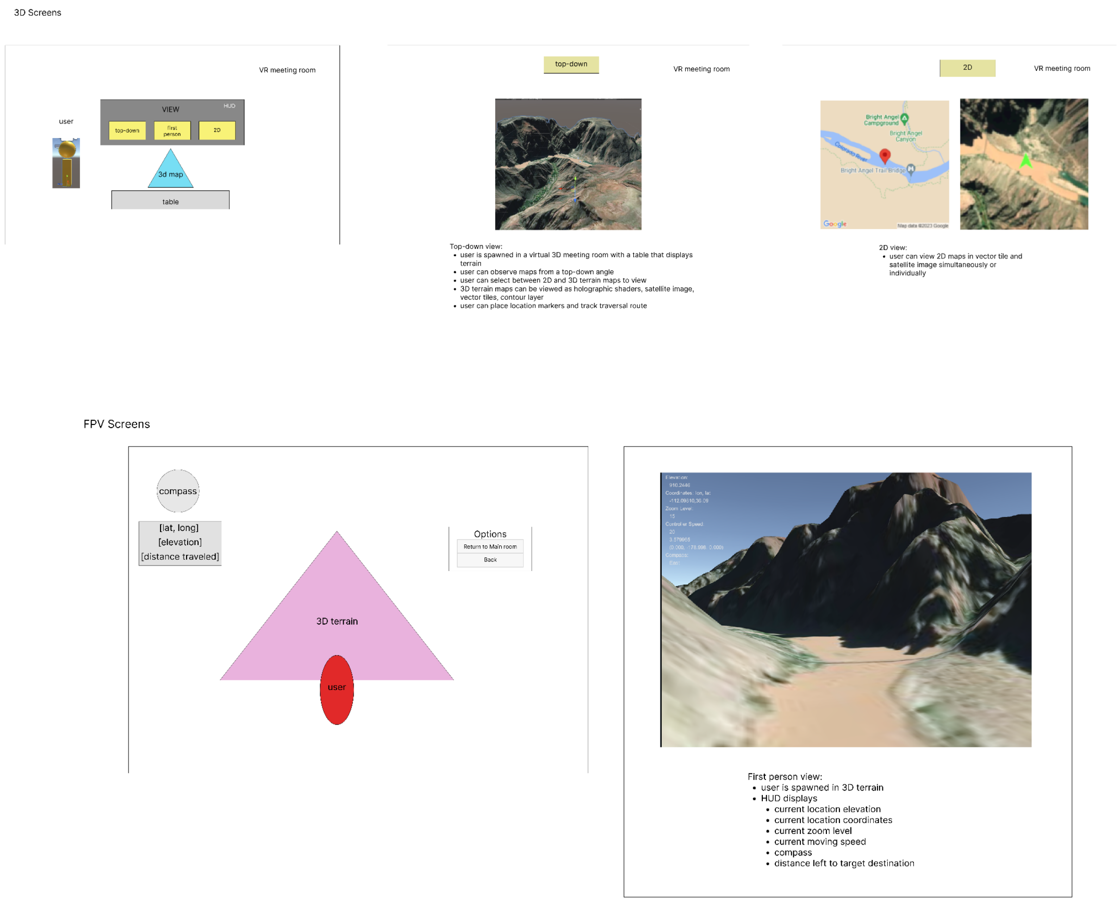


Figure 2. VR UI Screens

As seen in Figure 2, these preliminary UI designs depict the envisioned features and functionality of the ITM system. Each screen represents a key user interaction point, providing an intuitive and user-friendly environment for users to navigate and perform their tasks. These design specifications served as a blueprint for developing the ITM system, ensuring that it would fulfill the identified requirements and effectively meet the needs of the end-users.

3.3 Delivered System

In this project, we delivered a functional Interactive Topographical Mapping (ITM) system, designed to assist in navigation and pre-mission planning for emergency service members and researchers. The development process involved extensive research, system specification and design, and final implementation of the VR ITM system. For the system development, we utilized the Maps SDK [21] to generate Digital Elevation Models (DEMs) and leveraged the Unity platform for VR development.

The developed system presented an automated pipeline for visualizing 3D terrains based on specific areas of interest. It incorporated features analogous to a Global Positioning System (GPS), offering fundamental navigational support such as interactive selection of a destination, pathway recommendations, and estimations of travel time and distance to the selected destination. It also provided real-time updates on the user's travel speed and cardinal direction.

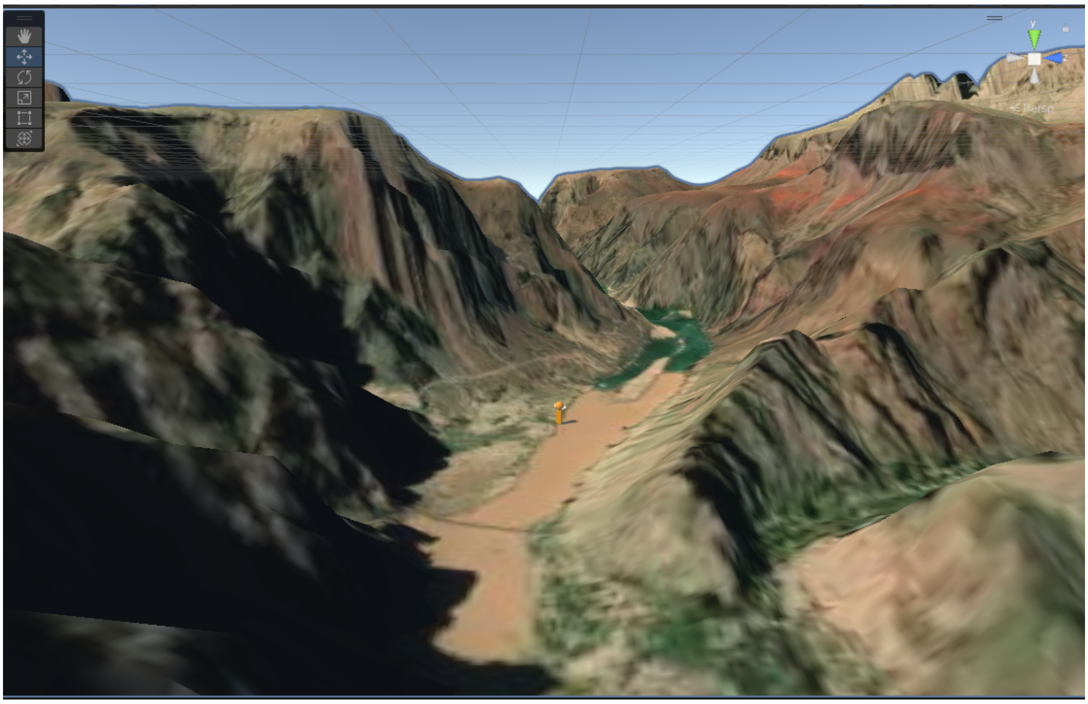


Figure 3. User Avatar Placed in 3D Scene

Figure 3 provides a visual depiction of the system's interface with the user avatar, depicted in yellow, placed within a 3D terrain scene. This offers a direct, intuitive method for users to visualize their virtual location and the surrounding topographical features.

The system was designed with a particular emphasis on user-centered interactivity and versatility. Its key features included:

- *Interactive Selection and Display of Terrain*: Users could select and dynamically view the 3D representation of any terrain generated from DEMs.
- *Multiple Camera Angles*: The system supported first-person, third-person, and bird's eye views, providing diverse perspectives of the 3D landscape.
- *Flexible Navigation Options*: Users could opt to walk or fly over the generated terrain for a comprehensive understanding of the area.
- *Diverse Mini Map Styles:* The system incorporated various mini map styles, including 3D, 2D, satellite image, and contour, enhancing the user's visual experience.
- *Compass for Orientation*: A built-in compass was provided to facilitate easy navigation and orientation within the terrain.
- *Optimized Performance*: The system maintained a consistent performance of 30 frames per second (fps) from all camera angles, ensuring smooth interactivity.
- *Versatile Usage*: The system could be operated with or without a VR headset on a Windows PC, providing flexibility to the users based on their preferences and available resources.

Through this development, we successfully achieved our initial goals and delivered a system that effectively catered to the potential needs of end-users.

3.4 Evaluating Project Deliverables

Our project set out to create an Interactive Topographical Mapping (ITM) system that would serve as an invaluable tool for navigation and pre-mission planning in rural or uncharted territories. After extensive research, system design, and implementation leveraging the Maps SDK and Unity platform, we successfully delivered a system conducive to our initial goals and expectations.

The developed system, by offering a real-time, interactive 3D visualization of terrains based on Digital Elevation Models (DEMs), addressed our primary requirement of runtime map creation. The system's ability to render these terrains efficiently, with a clear sense of depth and scale, fulfilled our second requirement of effective map rendering.

- *Interactive Map Augmentation*: Our system successfully facilitated the creation, refinement, and analysis of pathways. This was achieved by incorporating features such as pathway markers, user position and orientation indicators, and distance information feedback. The system thus provided a user-friendly, intuitive interface reminiscent of modern GPS-based navigation systems.
- *Remote Cross-Reality Collaboration*: The system was designed to support participants from different geographical locations. This feature allowed for a shared, immersive examination of the terrains.
- *Consistent Performance:* The system met our performance requirements by maintaining a consistent rate of 30 frames per second (fps) from all camera angles. This ensured smooth interactivity, significantly enhancing the user experience.
- *Versatile Usage:* In line with our design specifications, the system was compatible with or without a VR headset on a Windows PC. This provided users with the flexibility to operate the system based on their preferences and available resources.

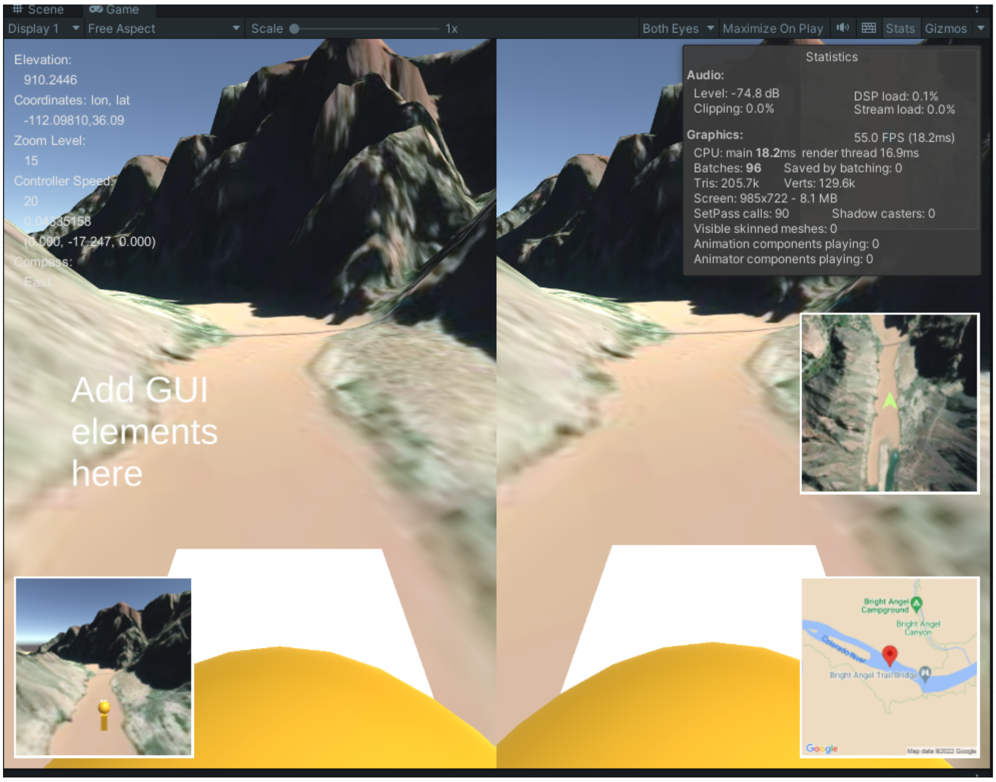


Figure 4a. Scene in VR Mode

Figure 4b. Scene in WebGL Mode

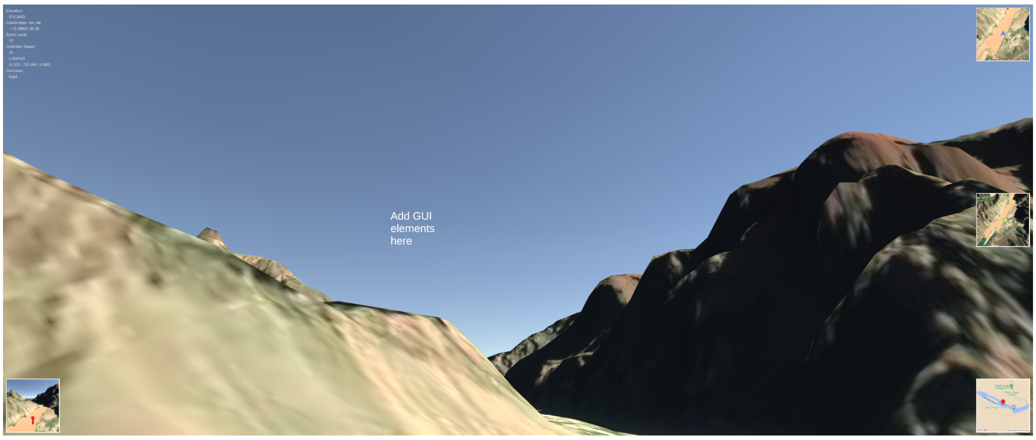


Figure 4c. Scene in Desktop Mode

As shown in Figures 4a - 4c, the system also successfully demonstrated compatibility with VR headsets, WebGL, and Desktop respectively. The screenshots showcase the various functionalities implemented such as the various viewing map styles, compass, distance and elevation tracking, and situational 3D environments. The images present an example of the system's response in its various interface modes, offering users a highly immersive experience.

By meeting these requirements and design specifications, our project succeeded in delivering an effective and efficient VR ITM system.

3.5 End-User Feedback and Reflection

Upon completion of the prototype and minimum viable product, we met with potential end-users, including forest fire researchers and search and rescue (SAR) members. These meetings were essential as they provided us with critical end-user information and highlighted the significance of user-centered design. During these interactions, we learned that we had developed a solution prior to accurately identifying a genuine problem. Although the VR ITM system was technically sound, it lacked real-world utility. For real-life forest fire or search and rescue missions, the ability to observe the actual terrain with their own eyes was a critical component for mission success.

This realization served as an important lesson for us, highlighting that a successful project must begin with the identification of a specific need followed by the design of a solution that addresses that need. Although the VR ITM system did not meet the needs of forest fire researchers and SAR members, it demonstrated our capacity to develop a fully functional VR ITM system using DEMs and the Unity Engine for VR development. It was clear however, that a pivot was needed in the direction of our project.

Chapter 4: Second Project - Intro and Spec Requirements

The original project direction, centered around the creation of a Virtual Reality (VR) Interactive Topographical Mapping System, embodied a technological breakthrough with a focus on innovation. However, it fell short in meeting the pragmatic expectations of Search and Rescue (SAR) personnel. These end-users underscored the paramount importance of having a tool that addressed their operational needs effectively, shedding light on the limitations of the initial VR system.

4.1 Realization of Challenges and Decision to Pivot

Upon acknowledging the disparity between the novel VR solution and its practical utility, we were prompted to reevaluate our approach. This transition was induced by feedback from SAR personnel, which highlighted the need for a solution that was more in tune with their real-world requirements. The strategic refocus of the project leaned towards a greater alignment with end-user needs, marking the inception of a user-centric design methodology. This shift was not merely a change in technical direction, but a paradigm shift in our project philosophy, highlighting the importance of a solution's applicability over its technological novelty.

4.2 User-Centered Design Approach: Overview and Project Goals

User-Centered Design (UCD), also known as human-centered design, is an iterative design process that extensively involves end-users at each stage of the design and development process [22]. The primary objective of UCD is to create a solution that resonates with the users' needs, preferences, and contexts, thereby ensuring the product's usability and relevance.

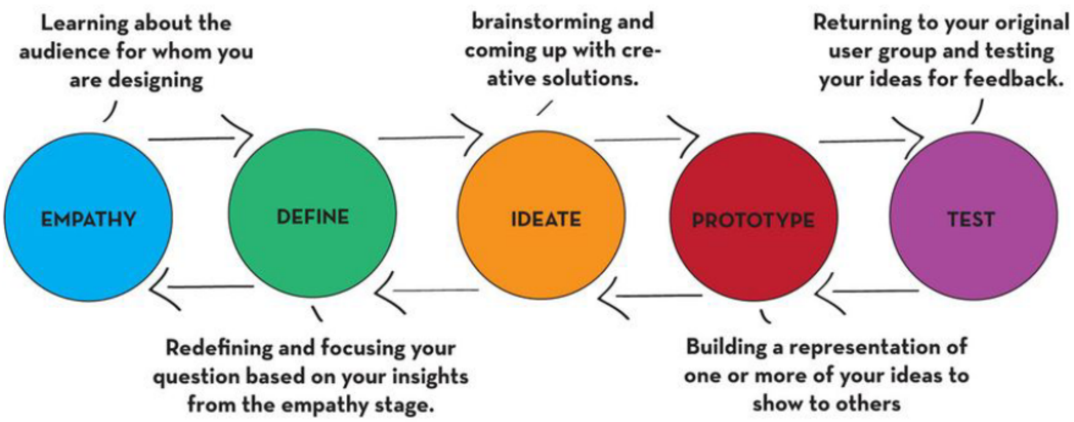


Figure 5. User Centered Design Process

As depicted in Figure 5, the UCD process is a cycle that starts with empathizing with the users to understand their needs, followed by defining the problem, ideating solutions, creating prototypes, and testing them. The insights gained from each testing phase are then used to refine the solution, and the process is repeated until a satisfactory product is achieved. In the context of this project, the implementation of UCD has been underpinned by several key goals:

- *Practice Collecting*: This goal focused on effectively gathering valuable insights from end-users through interviews, surveys, and observation of their workflows.
- *UI Evolution:* The project aimed at developing an interface that is intuitive, user-friendly, and tailored to the needs and preferences of SAR personnel.
- *Software Usability*: The usability of the software was a primary concern. The aim was to develop a solution that was not just technologically advanced but also easy to use and understand, thus enhancing user engagement and experience.

- *Software Testing*: This goal emphasized the importance of rigorous testing of the software at each stage of development to ensure its functionality, reliability, and efficiency.
- *Software Delivery*: The project aimed to ensure timely and efficient delivery of the software, while allowing for iterative improvements based on continuous user feedback.

4.3 Project Goals and Alignment with End-User Needs

In adopting a UCD approach, it was imperative to ensure that our project objectives were seamlessly aligned with the end-users' needs. This entailed setting goals that were directly informed by end-user feedback and that served to enhance the overall user experience. Active engagement with SAR personnel, encompassing feedback collection and workflow observation, served as the cornerstone of this alignment.

This user interaction shaped the project's objectives and provided valuable insights into the unique needs and challenges faced by SAR personnel. This data-driven approach, in turn, informed the evolution of the software's User Interface (UI), its usability, and the delivery and testing protocols, thus ensuring a solution that was not only technically robust but also met the practical requirements of the end-users.

4.4 Specification Requirements

Establishing the technical and functional specification requirements was a collaborative effort that involved close interaction with key stakeholders, primarily SAR personnel. Detailed discussions and interviews were instrumental in gathering invaluable end-user input, which significantly contributed to defining the project's scope and objectives. This engagement with end-users ensured that the resulting software solution was robust, flexible, and tailored to meet the specific needs of SAR personnel.

4.5 Summary

In summary, this chapter chronicles the project's transition from a technology-centric approach to a user-centric design methodology. This strategic pivot, prompted by end-user feedback, underscores the importance of ensuring that project goals align with end-user needs. This transformation, which permeates the entire second phase of the project, culminates in the creation of a software solution that is not only technologically versatile but also provides a practical and beneficial tool for SAR personnel.

Chapter 5: Problem Derivation

The purpose of this chapter is to provide an in-depth understanding of the fundamental problem our project aimed to address, a problem deeply rooted in the complexities of Search and Rescue (SAR) operations. These challenges encompassed limitations in existing technological tools, difficulties in visualizing search areas, and systemic inefficiencies in processing and integrating critical data.

5.1 Understanding User Needs and Pain Points

Through structured interviews and observational studies of SAR workflows, we identified specific pain points and needs of SAR personnel. According to a key personnel at the Washington State Search and Rescue Planning Unit, a consistent issue that emerged was the labor-intensive and time-consuming process of manually processing GPS tracks, particularly in calculating the total track line length (TLL) within search segments. This TLL is essential for determining Coverage, Probability of Detection (POD), and Probability of Success (POS), all of which are crucial components of Search Theory (see Appendix A).

In the absence of a simple, low-overhead, offline tool capable of quickly calculating the TLL for each segment, the existing process was not only slow but also presented risks of inefficient resource allocation and prolonged search times. These inefficiencies often led to increased risks for the missing individuals and stressed the need for a more efficient solution.

5.2 Deriving the Problem Statement


```
            [
              -121.28541469573976,
              48.663218382270564
            ],
            [
              -121.27970695495607,
              48.66663382581094
            ],
            [
              -121.27966403961182,
              48.66666216877235
            ]
          ],
          "type":"LineString"
        },
        "id":"d4ed10ab-3476-4ec2-af19-c7c818803462",
        "type":"Feature",
        "properties":{
          "creator":"1SE1FN",
          "pattern":"solid",
          "description":"",
          "fill":"#FF0000",
          "title":"Segment 2",
          "stroke":"#FF0000",
          "folderId":"bec56835-4809-408d-a883-14ca37b5114a",
          "gpstype":"TRACK",
          "stroke-opacity":1,
          "stroke-width":2,
          "fill-opacity":0.1,
          "class":"Shape",
          "updated":1612112726000
        }
      },
      {
```


Figure 6. GeoJSON data sample

As SarTopo offers downloadable GeoJSON files that contain comprehensive data of map project files, understanding these files and the ability to work with them was crucial to our solution development. In Figure 6, we display a sample of the GeoJSON data which primarily includes LineString and Polygon objects.

These objects represent live GPS tracks derived from SAR operations. A careful examination of these datasets revealed frequent instances of idle or exceedingly close latitude-longitude coordinate points within the tracks. This observation highlighted the need for a data cleaning and filtering algorithm, critical for efficiently managing such irregularities in the raw data.

Armed with these insights, we defined our problem statement: The primary challenge involves developing an application capable of expediently processing GPS tracks, inclusive of data cleaning and filtering, even under offline conditions typical of SAR mission Command Posts. The proposed solution should empower on-scene search planners to compute probabilities quickly when planning the next Operational Period by processing and computing accurate total track line lengths within search segments.. This, in turn, could improve the efficiency of search missions, potentially hasten the discovery of lost subjects, and ultimately, save lives.

5.3 Conceptualizing the Solution and Identifying Constraints

Recognizing this challenge, we proposed a specialized tool designed to automate this laborious workflow. Our concept was to create a software pipeline that would process GeoJSON files downloaded from SarTopo and output desired numerical results in a few seconds, transforming a manual, time-consuming task into an automated, efficient process.

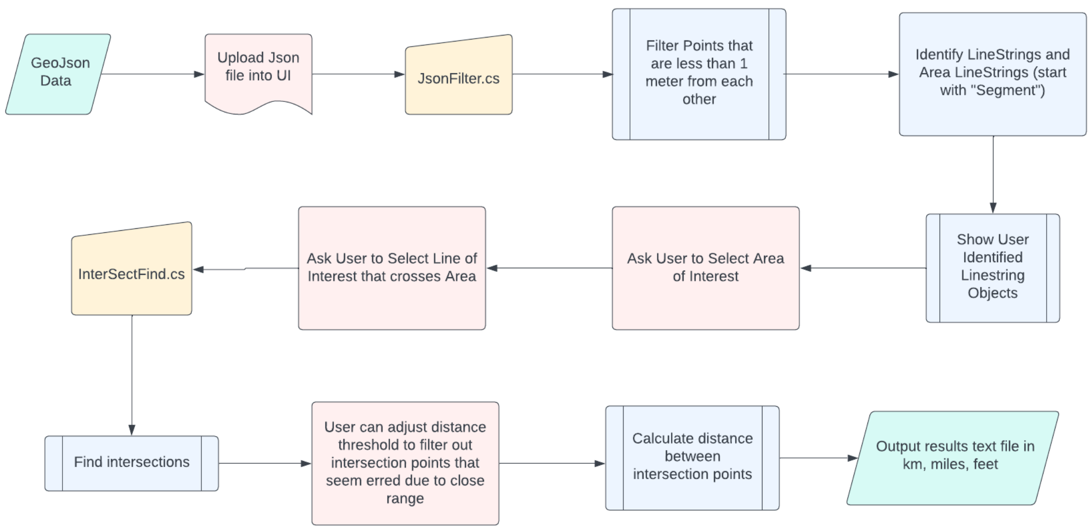


Figure 7. Proposed Solution Pipeline

As depicted in Figure 7, the proposed solution pipeline begins with the acquisition of GeoJSON data from SarTopo. This data, containing information about the terrain and search areas, serves as the primary input for our proposed system. The next stage of the pipeline involves the use of an algorithm designed to analyze the GeoJSON data, identify intersections, and compute the necessary information for Search and Rescue operations. The output of the pipeline is a set of processed results, useful for planning and updating SAR missions.

Though this was a conceptual design, it served as a roadmap for the development of the actual solution. It also helped us identify potential constraints and challenges that we would need to address during the development process.

In devising a practical solution for SAR operations, specific criteria had to be met:

- *Simplicity*: The solution should have a simple and intuitive user interface to facilitate easy usage even under stressful and time-sensitive situations.
- *Offline Availability*: The tool must work offline, as SAR command posts often operate in areas with limited or no internet connectivity.
- *Low Overhead:* The system should have a low overhead to not burden the limited computational resources available during SAR operations.
- *Speed*: The tool must be fast and capable of processing data quickly to facilitate real-time decision-making.
- *Utility-focused*: The solution should primarily focus on utility and efficiency, providing valuable insights and aiding in decision-making for SAR personnel.

Our aim was to develop a tool that effectively addressed these constraints, meeting the needs of the SAR personnel, and contributing to the success of their missions.

Chapter 6: Solution Process and User Results

This chapter outlines the methodologies implemented and the outcomes of our project, aimed at creating a desktop application to augment the efficacy and efficiency of Search and Rescue (SAR) operations. Our approach was fundamentally grounded in User-Centered Design (UCD) principles, resulting in a software solution that met SAR personnel's unique needs.

6.1 Problem Analysis and Technical Requirements Investigation

Our solution development initiated with an exhaustive analysis of the problem and an understanding of the user's needs. We identified crucial challenges including the laborious and time-intensive process of manually calculating intersection lengths and traversed distances within search areas. Our software solution sought to address these problems, making the process more efficient.

Technical requirements were distilled from these user needs. The requirements highlighted the need for an accurate, efficient, and user-friendly software solution, adaptable to the dynamic needs of SAR operations. A significant aspect of the software was to handle GeoJSON data objects and to programmatically calculate intersection lengths and traversed distances.

6.2 Initial Algorithm Development

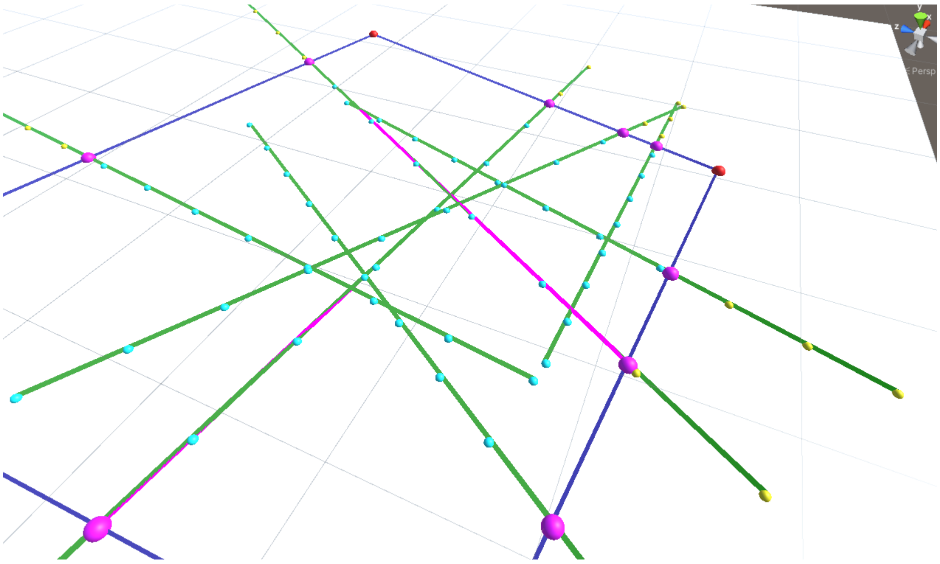


Figure 8. Initial algorithm development on a 2D plane

The preliminary development phase of the core algorithm, as shown in Figure 8, was conducted using Unity. This platform offered us the advantage of visual interface features and simplified scripting. We constructed a rudimentary model that allowed the user to manually place spheres on a 2D plane, forming a concave polygon intersected by a line. The program was designed to compute the intersection points and the length of the intersection line within the polygon on a 2D surface, represented by the cyan colored spheres that lie on the magenta intersection line, and the yellow spheres that exist outside of the polygon boundary. (Note: Due to the camera's viewing angle when the artifact image was taken, not all of the intersection lines show the magenta line layered above the green user-drawn lines.) This 2D design and development enabled us to validate the fundamental design of the algorithm and make necessary adjustments based on the visual results.

As we continued refining the algorithm, we incorporated functionalities to handle edge cases such as scenarios where the line begins or ends within the polygon. These enhancements were critical in ensuring precise intersection length computations under all circumstances, as we were informed that it is critical to consider the events where a SAR personnel would start or end their recorded track mission inside a search segment. Through the iterative development and refinement process, it became apparent that our final solution should prioritize portability and low overhead, prompting a transition from the Unity platform for the final algorithm design to Windows .NET application development [23].

6.3 Data Cleaning and Filtering Algorithm

Our algorithm was developed as a solution to the challenges identified in our problem statement (Section 5.2). Specifically, it was designed to measure the distance between sequential points in a LineString and filter out those points that were idle or overlapping based on their proximity. By refining the raw data in this way, our algorithm facilitated more efficient data processing and significantly streamlined overall data handling.

In addition to the filtering mechanism, we implemented an error-handling protocol as part of our algorithm. This feature validates the GeoJSON file and ensures the data is parsed accurately, despite any inconsistencies. This aspect of our solution not only enhances the robustness of our algorithm but also demonstrates its capability to effectively handle real-world data with a variety of irregularities.

6.4 User Interface Development

Our User Interface (UI) played a central role in shaping the solution, reflecting our dedication to User-Centered Design (UCD) principles. We implemented the UI in an iterative manner, relying heavily on feedback from the end-users to fine-tune and enhance the design, with the overarching goal of providing a simple, intuitive, and user-friendly experience.

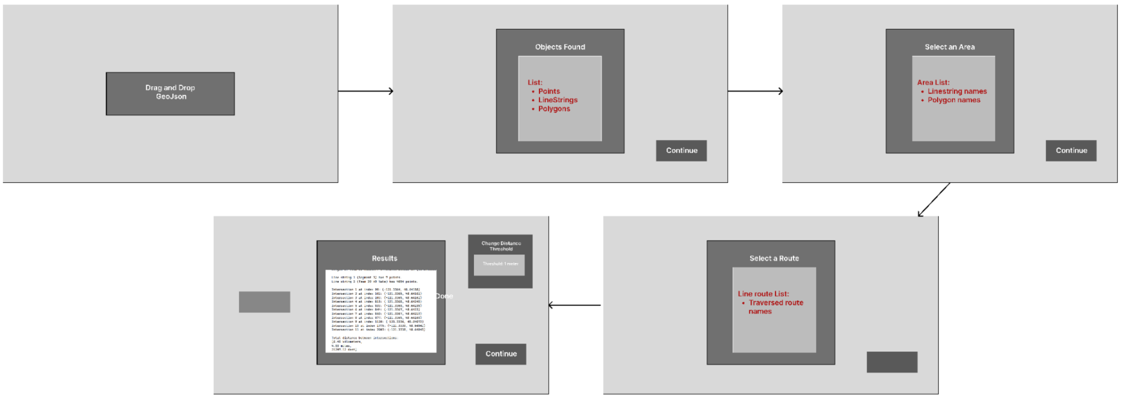


Figure 9a. Initial UI design for GUI

```
Length of 13: 17.2859520026156 km (10.7409925460815 miles)

Length of Team 13 K9 Oso : 20.3442709656803 km (12.6413440704346 miles)

Length of Segment 1: 2.72184787703654 km (1.69127786159515 miles)

Length of Team 29 K9 Nate: 24.9994144294196 km (15.5339164733887 miles)

Length of Segment 3: 0.91579190952718 km (0.56904673576355 miles)

Length of Team 22 K9 R2 PM: 10.0456302359058 km (6.2420654296875 miles)

Length of Segment 2: 4.12037489347169 km (2.5602822303772 miles)

Length of Team 22 K9 R2 AM: 7.03656052421536 km (4.37231636047363 miles)

Length of Team 13 Handler : 18.217173514999 km (11.3196268081665 miles)

Length of Team 29 Handler: 31.0479505297481 km (19.2923030853271 miles)

Length of Team 22 Handler: 17.1148353694377 km (10.6346664428711 miles)

Line string 1 (Segment 2) has 32 points.
Line string 2 (Team 29 K9 Nate) has 6187 points.

Intersection 1: (-121.2856, 48.66688)
Intersection 2: (-121.2856, 48.66687)
Intersection 3: (-121.2833, 48.66451)
Intersection 4: (-121.2833, 48.66446)
Distance between intersection points 1 and 3: 0.3744089 miles
Distance between intersection points 2 and 4: 0.3756388 miles
```

Figure 9b. Initial UI design for CLI application

Our initial design, shown in Figures 9a and 9b, included detailed data such as latitude and longitude coordinates of all intersections, collected GeoJSON data objects, and distance metrics in both miles and feet. However, following multiple rounds of user feedback, it became evident that this level of detail was excessive for our end-users. They were primarily concerned with quickly and easily identifying the intersection areas and their lengths.

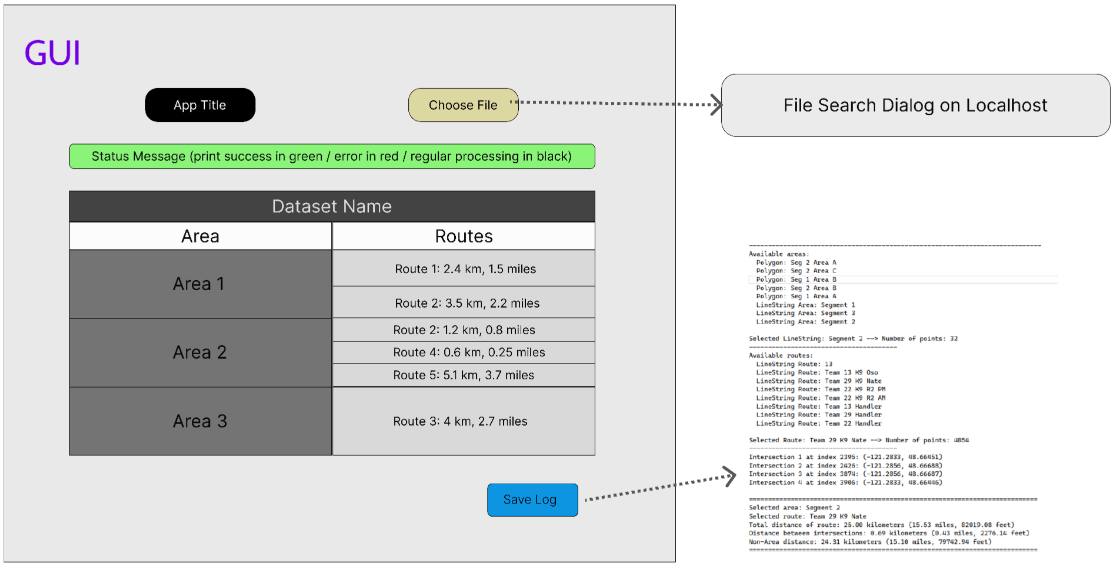


Figure 10. Iterated GUI UI design

The feedback-driven iterative process led to significant modifications in the UI design for the Graphical User Interface (GUI), as shown in Figure 10. The revised design provides a streamlined and focused display, emphasizing only the necessary data - intersection areas and lengths, thereby increasing the utility and user-friendliness of the GUI application.

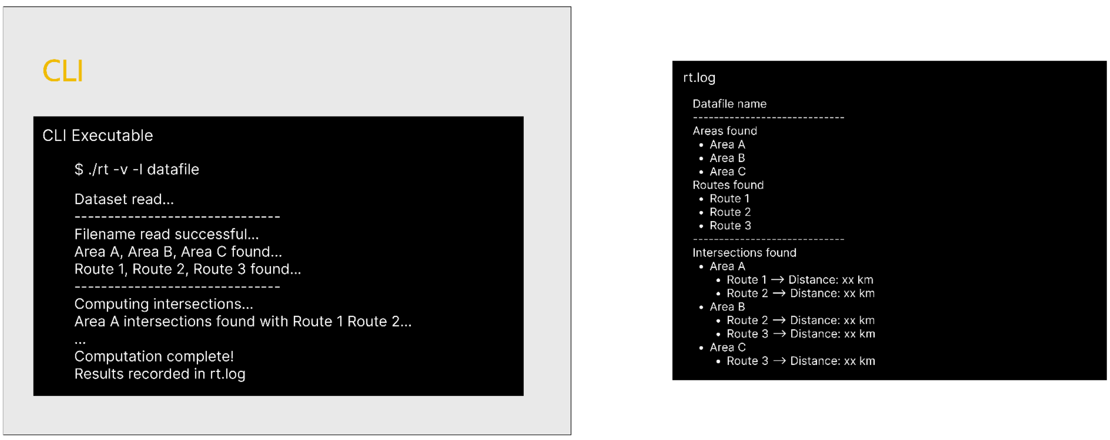
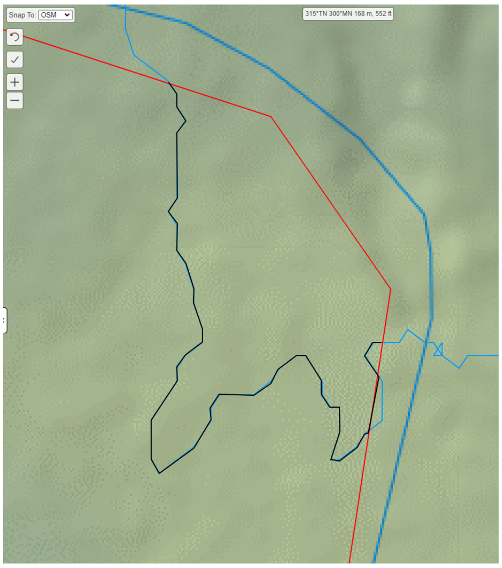


Figure 11. Iterated CLI UI design

Likewise, our Command Line Interface (CLI) application underwent a similar iterative design process based on end-user feedback. The revised CLI UI, shown in Figure 11, offers a streamlined experience focused on delivering relevant information. This feedback-driven, iterative process ensured that our solution was truly user-centered and matched the needs and preferences of SAR personnel.

6.5 Algorithm Verification

Our algorithm verification process was robust and meticulous, designed to ensure the accuracy and reliability of our software solution. This process entailed a comprehensive visual examination, coupled with manual measurements of the data outputted by our algorithm.



```
Line string 1 (Segment 2) has 32 points.
Line string 2 (Team 22 K9 R2 AM) has 2156 points.

Intersection 1 at index 794: (-121.2741, 48.67358)
Intersection 2 at index 798: (-121.2742, 48.67352)
Intersection 3 at index 801: (-121.2742, 48.67344)
Intersection 4 at index 847: (-121.2747, 48.674)

Total distance between intersections:
[0.16 kilometers,
0.10 miles,
520.45 feet]
```

Figure 12. Verification of Algorithm Output

As shown in Figure 12, we validated the effectiveness of our algorithm by visually and numerically comparing its outputs against data from SarTopo. We confirmed the correct identification of intersection points and the accuracy of the computed intersection lengths.

The left image in Figure 12 represents 4 intersection points from SarTopo with a measured length of 0.16 kilometers. This is verified in the output of the right image, which showcased the 4 intersection coordinate points found between the area segment and linestring route object, with a measured length of 0.16 kilometers as well. This careful and detailed verification process reinforced the reliability of our software solution and confirmed its precision, ultimately enhancing the overall value and utility of our tool.

6.6 Final Deliverable

The final deliverable was a standalone, portable application that could efficiently process GeoJSON files downloaded from SarTopo. The software was designed with a simple and user-friendly interface, facilitating ease of operation by SAR personnel, even under demanding circumstances. Moreover, it was designed to work offline, proving reliable in remote locations with limited or no internet connectivity.

The technical architecture of the solution was structured around three primary classes: IntersectionSupport, WaSARxGUI, and WaSARxCLI, which emphasized modularity and flexibility.

6.6.1 IntersectionSupport

IntersectionSupport, a central class in our software solution, holds the JsonParsing and IntersectionCalculation code that forms the foundation of the software's algorithmic capabilities. It processes GeoJSON data objects and performs computations essential for identifying intersection areas and lengths.

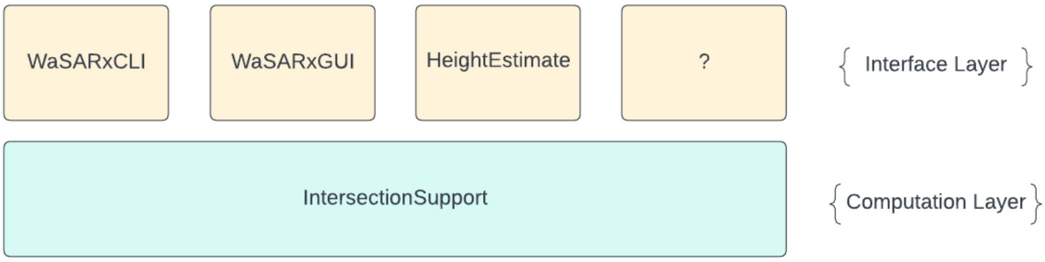


Figure 13. Modular Architecture Design

As seen in Figure 13, IntersectionSupport sits within the computation layer of our modular architecture design. This strategic positioning enables the construction of additional interface-based applications on top of it, located within the Interface Layer. These applications, which include the CLI and GUI apps, are designed to interact seamlessly with IntersectionSupport, leveraging its computational functionalities to deliver necessary information to end-users in a convenient, intuitive manner.

In addition to the GUI and CLI applications, we also developed a third application that queries the elevation values of latitude and longitude coordinate points using the Open Topo Data API [24]. This application adds another dimension to our software, allowing for the analysis of topographic elevations, which can increase the accuracy and applicability of intersection data, particularly in hilly or mountainous regions.

By designing IntersectionSupport as a modular and scalable computation layer, we ensured that the software could be easily expanded and adapted for future developments and enhancements, in line with the evolving needs of SAR operations

6.6.2 WaSARxGUI

WaSARxGUI is the class that forms the basis of the Graphical User Interface (GUI) of our software solution. It has been designed to offer an intuitive and straightforward user interaction model, making it easier for SAR personnel to navigate and use the software effectively.

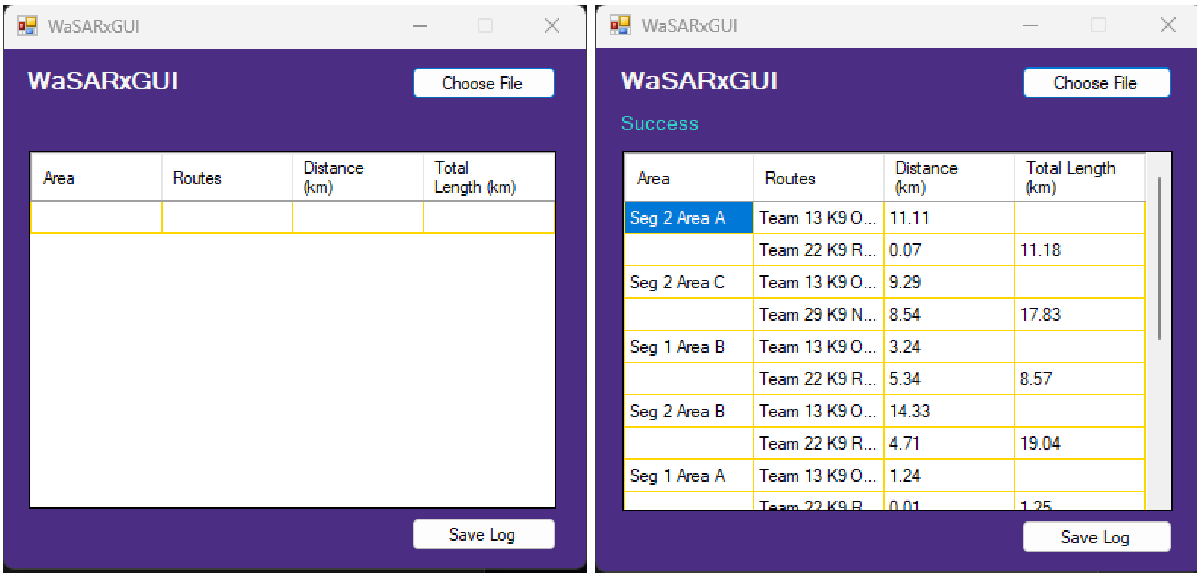


Figure 14. GUI application UI

As displayed in Figure 14, the GUI allows users to effortlessly select and load a GeoJSON data file from their local machine. Upon selection, the software parses the file, processing the data and highlighting any errors if the file is incorrectly formatted or unreadable.

The core algorithm then takes over, computing the intersections, routes, and traversed distances based on the loaded data. The results of these computations are displayed in a tabulated format, featuring columns for Area, Routes, Distance, and Total Length traversed. This table view presents the data clearly and concisely, ensuring easy comprehension for users.

In addition, the GUI offers a filter feature that allows users to refine the table view based on specific parameters such as Area or Mission personnel. The user then has the ability to save the table as a log on their local machine as well. This versatility in viewing options ensures that end-users can view and understand the intersection data in a way that is most meaningful and useful for them.

6.6.3 WaSARxCLI

WaSARxCLI represents the Command Line Interface (CLI) aspect of our software solution. This class is designed specifically for those users who prefer to interact with the software through text-based commands, adding an extra layer of flexibility to our solution.

```
C:\Users\Tyler\source\repos\WaSARxCLI\bin\Release>rt ../../foramttedMarbleMt.txt
GeoJSON file has been read successfully.
Area x Route intersections processing...
Processing complete. Check the output file for results.
Execution time: 2485 ms
```

Figure 15. CLI application UI

As depicted in Figure 15, the CLI application has been designed for easy execution on Windows Command Prompt. The user is guided through the operation by clear instructions provided both within the software's prompts and the README documentation [25], enabling an experienced CLI user to run and debug their GeoJSON data with ease.

```
Results:
Processing file: MarbleMt.txt
--------------------------------------------------------------------------------------
Areas:
  Seg 2 Area A
  Seg 2 Area C
  Seg 1 Area B
  Seg 2 Area B
  Seg 1 Area A
  Segment 1
  Segment 3
  Segment 2

Routes:
  13
  Team 13 K9 Oso
  Team 29 K9 Nate
  Team 22 K9 R2 PM
  Team 22 K9 R2 AM
  Team 13 Handler
  Team 29 Handler
  Team 22 Handler

Intersection Results:
--------------------------------------------------------------------------------------
Area: Seg 2 Area A
--> Route: Team 13 K9 Oso
  Total distance of route: 11.11 kilometers
--> Route: Team 22 K9 R2 AM
  Total distance of route: 0.07 kilometers
Total traversed in Area Seg 2 Area A --> 11.18 kilometers
--------------------------------------------------------------------------------------
Area: Seg 2 Area C
--> Route: Team 13 K9 Oso
  Total distance of route: 9.29 kilometers
--> Route: Team 29 K9 Nate
  Total distance of route: 8.54 kilometers
Total traversed in Area Seg 2 Area C --> 17.83 kilometers
--------------------------------------------------------------------------------------
Area: Seg 1 Area B
--> Route: Team 13 K9 Oso
  Total distance of route: 3.24 kilometers
--> Route: Team 22 K9 R2 AM
  Total distance of route: 5.34 kilometers
Total traversed in Area Seg 1 Area B --> 8.57 kilometers
--------------------------------------------------------------------------------------
```

Figure 16. CLI output log

The CLI interface not only computes and presents the required data but also maintains a detailed log of the processed data and results, as shown in Figure 16. This log is automatically saved, providing users with a comprehensive record of their operations, which can be examined post-hoc for analysis or debugging purposes.

6.7 Results

The principal accomplishment of our software solution is its notable reduction of manual computation time during SAR missions (See Appendix A). This achievement aligns seamlessly with our key goals, outlined in Chapters 4 and 5, thereby manifesting the successful application of our User-Centered Design (UCD) approach.

In accordance with our '*Practice Collecting*' goal, our development process was informed by valuable insights gathered from end-users through interviews, surveys, and workflow observations. This input directly influenced the design of the software, ensuring it meets the practical needs of SAR personnel.

Our '*UI Evolution*' goal aimed to develop an intuitive, user-friendly interface, customized to the needs and preferences of SAR personnel. The delivered software solution features a straightforward interface, providing the users with a choice between a graphical user interface (GUI) and a command-line interface (CLI), thus catering to users' varying technical proficiency levels.

The '*Software Usability*' goal underscored the need for an easy-to-use and understandable solution. Our delivered software, with its intuitive interface and clear instructions, is user-friendly and promotes user engagement. It also enhances user experience by providing valuable error messages for troubleshooting, thereby streamlining the process of identifying and resolving potential issues.

Our '*Software Testing*' goal stressed the importance of rigorous testing at each development stage. The application underwent comprehensive testing during its development, ensuring functionality, reliability, and efficiency. The software successfully parses GeoJSON files, calculates intersection lengths, and produces concise, valuable output data, fulfilling the requirement of functionality.

Finally, our '*Software Delivery*' goal emphasized the importance of efficient software delivery. Initially, there were hiccups during the delivery of the application, as end-users encountered difficulty running the software due to lack of file organization. This setback underlined the importance of an easy-to-use interface and the necessity for detailed, comprehensible instructions and organization.

In response to this feedback, we made iterative improvements to enhance the user experience (UX) and added comprehensive instructions for accessing and successfully running the app. This experience served as a valuable lesson in the nuances of user-centered design, reiterating the importance of considering all user touchpoints in the software delivery process.

Consequently, we delivered an improved version of the application on time, closely adhering to the requirements and specifications defined by end-users, thereby effectively fulfilling our 'Software Delivery' goal.

Regarding our criteria defined in Section 5.3, our software solution fulfills each criterion:

- *Simplicity*: The software has a simple and intuitive interface, ensuring easy usage under stressful, time-sensitive situations typical of SAR missions.
- *Offline Availability*: The software works offline, accommodating the need for functionality in areas with limited or no internet connectivity - a common challenge for SAR command posts.
- *Low Overhead:* Our software solution presents a low overhead, ensuring it does not burden the limited computational resources available during SAR operations.
- *Speed*: The software processes data swiftly, facilitating real-time decision-making and enhancing the effectiveness of SAR operations.
- *Utility-focused*: Our software solution provides valuable insights and aids decision-making for SAR personnel, with a primary focus on utility and efficiency.

End-user feedback has been instrumental in validating the practical utility of our software. One user described its impact as such: "*This is pretty much exactly what we envisioned! Simple, easy-to-use tools that we can use offline at Command Post on real* *missions. It could be a HUGE time saver for us. It would allow us to be much more effective in calculating coverage and planning for the next day's operations. It would be a tremendous help*!" (See Appendix A).

In summary, our software's design and functionality effectively meet the objectives and specifications defined by our UCD approach. The results underscore the efficacy of our user-centered development strategy and demonstrate its success in addressing the practical needs of SAR operations.

6.8 Limitations

Despite the demonstrated robustness and functionality of our solution, it's important to acknowledge that our development and testing were based on a statistically insignificant number of datasets. While the algorithm performed successfully with the few datasets available, it's critical for it to be evaluated with multiple and diverse datasets to establish wider validity and applicability.

In the spirit of User-Centered Design (UCD), we believe that continuing to iterate on the software, incorporating a broader range of test data, and continually seeking end-user feedback will enable us to further refine our solution and enhance its value. Future work should consider these aspects, always striving for the most effective, efficient, and user-friendly software that can assist SAR personnel in their vital work.

Chapter 7: Conclusion

The trajectory of this capstone project underscores the importance of a user-centered development strategy in designing effective software solutions. The project's initial focus was the development of a Virtual Reality (VR) Interactive Topographical Mapping System. This phase led to notable technical accomplishments, like the creation of a prototype facilitating real-time, interactive 3D topographical mapping. However, it also highlighted a vital insight: the need for software solutions to be squarely aligned with end-users' specific, practical requirements.

This revelation prompted a shift in our approach, steering us towards a more user-centric software development philosophy. The second phase pivoted to creating a desktop application, specially designed to meet the needs of Search and Rescue (SAR) operations. This strategic reorientation drew from the lessons learned during the VR project, emphasizing that even technically sophisticated solutions may fail to serve their purpose if they do not address users' real-world problems.

Throughout the GeoJSON processing application's development, we engaged key stakeholders at every stage, including SAR personnel, researchers, and directors. Their invaluable insights and feedback significantly shaped the application, underlining the impact of a user-centered approach. The positive feedback from end-users further affirms the effectiveness of our methodology and the value of the delivered solution.

7.1 Future Work

The end-result of this Capstone project and our continued discussions with our end-user present promising opportunities for future research and development:

- *Land Classification Model & Current Terrain Situation*: We plan to integrate a model that analyzes the current condition of the terrain. This feature could provide valuable insights into terrain accessibility, aiding SAR teams in their mission planning.
- *Computation of Average Speed*: We aim to develop a function to compute the average speed a SAR person can travel based on the terrain type and current conditions. This information would increase the software's calculation accuracy, leading to more precise SAR mission planning.
- *Automatic Separation of Region by Land Classification*: Our goal is to introduce a feature that automatically separates regions based on their land classification, enabling more efficient resource and personnel allocation during SAR missions.
- *Searchable Regions:* To augment the software's capabilities, we plan to add a feature allowing users to search for specific regions, simplifying the process of locating and focusing on mission-critical areas.

We recommend that future researchers or development teams embarking on similar projects adhere to a user-centric approach. Prioritizing stakeholder engagement, iterative development, rigorous evaluation, and responsiveness to end-user feedback will ensure the development of not only technically robust solutions, but also tools that effectively address end-users' real-world challenges.

In conclusion, this capstone project has compellingly illustrated the importance of user-centered development strategies in creating impactful software solutions. The transition from the initial VR project to the final desktop application stands as a testament to the efficacy of developing software solutions that align with end-user needs. As we advance with future iterations of this project, our commitment to refining and expanding the software, guided by the feedback and needs of the SAR community, will remain steadfast.

7.2 Learnings

This project presented a remarkable opportunity for personal and professional growth. The responsibility to research and comprehend new technologies critical to the project's success was a constant throughout each week. The continual learning process, although challenging, offered invaluable chances to explore and experiment with new areas of knowledge and technical expertise.

During the project, I engaged with several technologies for the first time, including Unity for VR 3D visualization, .NET for Windows application development, JSON for data handling, and principles of linear algebra and vector calculus for core algorithm development. Each of these areas opened a new frontier of understanding, making the learning and application process within the project's context rewarding.

The experience underscored the versatility of software development and the importance of continuous learning in this swiftly evolving field. More importantly, it emphasized the significance of user-centric design and the profound impact it can have when software solutions are designed to meet the specific, practical needs of end-users. This project not only resulted in a valuable software solution for SAR operations but also facilitated significant personal growth and professional development.

References

Source code: https://github.com/UWB-ARSandbox/2022.6.Tyler-ITM/tree/master .

[1] T.-X. Yue, Z.-P. Du, D.-J. Song, and Y. Gong, “A new method of surface modeling and its application to DEM construction,” *Geomorphology*, vol. 91, no. 1, pp. 161–172, Oct. 2007, doi: 10.1016/j.geomorph.2007.02.006.

[2] K. Jafarzadegan, D. F. Muñoz, H. Moftakhari, J. L. Gutenson, G. Savant, and H. Moradkhani, “Real-time coastal flood hazard assessment using DEM-based hydrogeomorphic classifiers,” *Nat. Hazards Earth Syst. Sci.*, vol. 22, no. 4, pp. 1419–1435, Apr. 2022, doi: 10.5194/nhess-22-1419-2022.

[3] “Team and Organization Account Info - SARTopo.” https://caltopo.com/about/teams/

[4] “What is a digital elevation model (DEM)? | U.S. Geological Survey.” https://www.usgs.gov/faqs/what-digital-elevation-model-dem

[5] Md. S. Chowdhury, “Modelling hydrological factors from DEM using GIS,” *MethodsX*, vol. 10, p. 102062, Feb. 2023, doi: 10.1016/j.mex.2023.102062.

[6] J. L. Mesa-Mingorance and F. J. Ariza-López, “Accuracy Assessment of Digital Elevation Models (DEMs): A Critical Review of Practices of the Past Three Decades,” *Remote Sens.*, vol. 12, no. 16, Art. no. 16, Jan. 2020, doi: 10.3390/rs12162630.

[7] N. O. and A. A. US Department of Commerce, “What is LIDAR.” https://oceanservice.noaa.gov/facts/lidar.html

[8] X. Shi, L. Girod, R. Long, R. DeKett, J. Philippe, and T. Burke, “A comparison of LiDAR-based DEMs and USGS-sourced DEMs in terrain analysis for knowledge-based digital soil mapping,” *Geoderma*, vol. 170, pp. 217–226, Nov. 2011, doi: 10.1016/j.geoderma.2011.11.020.

[9] “What is the difference between lidar data and a digital elevation model (DEM)? | U.S. Geological Survey.” https://www.usgs.gov/faqs/what-difference-between-lidar-data-and-digital-elevation-model-dem.

[10] X. Liu, “Airborne LiDAR for DEM generation: some critical issues,” *Prog. Phys. Geogr. Earth Environ.*, vol. 32, no. 1, pp. 31–49, Feb. 2008, doi: 10.1177/0309133308089496.

[11] Z. Wang and M. Menenti, “Challenges and Opportunities in Lidar Remote Sensing,” *Front. Remote Sens.*, vol. 2, 2021, [Online]. Available: https://www.frontiersin.org/articles/10.3389/frsen.2021.641723

[12] B. Huang and H. Lin, “GeoVR: a web-based tool for virtual reality presentation from 2D GIS data,” *Comput. Geosci.*, vol. 25, no. 10, pp. 1167–1175, Dec. 1999, doi: 10.1016/S0098-3004(99)00073-4.

[13] J. Crampton, “Keyhole, Google Earth, and 3D Worlds: An Interview with Avi Bar-Zeev,” *Cartographica*, vol. 43, pp. 85–93, Jun. 2008, doi: 10.3138/carto.43.2.85.

[14] “The Tactical Assault Kit Meets Virtual Reality (Sponsored),” *Defense Media Network*. https://www.defensemedianetwork.com/stories/tactical-assault-kit-meets-virtual-reality-sponsored/

[15] “WFCA Fire Map: Tracking Current Wildfires in the US.” https://wfca.com/fire-map

[16] N. Earth Science Data Systems, “SRTM,” *Earthdata*. https://www.earthdata.nasa.gov/sensors/srtm

[17] “Welcome to the QGIS project!” https://qgis.org/en/site/.
[18] “GRASS GIS - Bringing advanced geospatial technologies to the world,” Apr. 07, 2023. https://grass.osgeo.org/
[19] J. Y. Huang, C. H. Tsai, and S. T. Huang, “The next generation of GPS navigation systems,” *Commun. ACM*, vol. 55, no. 3, pp. 84–93, Mar. 2012, doi: 10.1145/2093548.2093570.
[20] “Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine,” *Unity*. https://unity.com.
[21] “Maps SDK for Unity: 3D worlds, AR, & POIs | Mapbox.” https://www.mapbox.com/unity
[22] A. Chammas, M. Quaresma, and C. Mont’Alvão, “A Closer Look on the User Centred Design,” *Procedia Manuf.*, vol. 3, pp. 5397–5404, Dec. 2015, doi: 10.1016/j.promfg.2015.07.656.
[23]. “NET desktop apps | Free tools and frameworks,” *Microsoft*. https://dotnet.microsoft.com/en-us/apps/desktop
[24] “Open Topo Data.” https://www.opentopodata.org/
[25] “UWB-ARSandbox/2022.6.Tyler-ITM at master.” https://github.com/UWB-ARSandbox/2022.6.Tyler-ITM/tree/master.

Appendix A

From Eric R. Director | Washington State SAR Planning Unit:

Search Theory, in the Search and Rescue world, has to do with gathering all the relevant incident data, converting that information into probabilities, and mapping those probabilities so you can allocate resources (search teams) in the most efficient way possible to find the subject in the shortest amount of time. In short, Search Theory helps determine where to assign your search teams to get the biggest bang for the buck.

After the initial planning effort, applied Search Theory relies on data generated from the field (primarily search team GPS tracks). Search missions which require applied Search Theory (most missions don't need applied Search Theory because they are resolved quickly in the initial phase of the mission) are usually relatively large missions with dozens or hundreds of search resources in the field. The field teams generate a lot of data (tracks), which usually comes near the end of an Operational Period. That data needs to be processed quickly enough to inform the search plan for the next Operational Period. The impact of not processing that data in a timely fashion is either idling resources (search teams waiting around in Command Post for an assignment) or the inefficient deployment of resources (giving teams assignments that are not optimized from a Search Theory point of view). So it is very important that the tracks the field teams generate are processed quickly.

The main pain point with processing tracks for use with applied Search Theory is measuring the length of all the tracks within a search segment. The total track line length (TLL) of all GPS tracks within a search segment is required in order to calculate Coverage, Probability of Detection (POD), and Probability of Success (POS). It is a slow, time-consuming process to manually trim each GPS track within a search segment so the TLL can be used with Search Theory in a timely manner. Currently we don't have a simple, low-overhead, offline tool that we can supply search segments (georeferenced polygons) and GPS tracks and it will produce the TLL in meters for each segment.

The application developed by Tyler Choi at the University of Washington will greatly reduce the amount of time it takes to process GSP tracks during active search and rescue missions, when it is needed most. It works offline, which is important because often there is no internet connectivity at the Command Post of a SAR mission, and it is quick and easy to use, saving a lot of time and effort. Using this application will allow search planners on scene at command post to calculate probabilities in time for planning the next Operational Period, which is something very challenging to do now, and ultimately could help find the lost subject quicker and potentially save lives.

Acknowledgements

I would like to extend my heartfelt gratitude to those who played pivotal roles in the development and completion of this project. Firstly, I owe a tremendous debt of gratitude to my advisor, Professor Kelvin Sung, for his invaluable guidance, patience, and unwavering support throughout the course of this project.

I am equally grateful to Eric R., our end-user, whose precious time and insights significantly contributed to the development of a specialized application for SAR teams. Your feedback and collaboration were instrumental in ensuring the practical utility of our software solution.

A special thank you goes to the University of Washington Bothell's Master of Science in Computer Science & Software Engineering department. UWB’s robust academic program provided me with an unparalleled opportunity to delve into a fascinating field of study and emerge with a wealth of knowledge and professional development experience.

I also express my sincere gratitude to my committee members, Professor Michael Stiber and Professor Yusuf Pisan, for their support, advice, and guidance throughout this journey. Your expertise and dedication played an integral part in the successful realization of this project.

Last but not least, my utmost gratitude to my family and friends for their unwavering faith and support throughout the entirety of this graduate endeavor the past few years. I would not have made it through without them.

In conclusion, this project would not have been possible without the collective effort, wisdom, and support of everyone involved. I extend my profound thanks to all for being part of this significant accomplishment in my educational career.