CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Environmental sciences

DEPARTMENT

TOPIC

DIPLOMA THESIS

Supervisor: ....

Student: ....

2022

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Environmental Sciences

BACHELOR THESIS ASSIGNMENT

Alexei Holomonov

Environmental Data Science
Informatics

Thesis title

The development of semi-automatic spatial data collection instrument

---

Objectives of thesis

The primary goal of the thesis is to create and integrate digital solutions into the process of landscape design. The digital platform being developed is meant to support property owners and landscape designers in planning their spaces, by gathering initial data and enabling the addition of various elements to the map. Utilizing this tool should present a more intuitive approach to designing the landscape and should not necessitate extensive knowledge or specific skills.

Methodology

To accomplish the aforementioned goals, the tool must be user-centric and equipped with various integrated or attachable sensors. Additionally, it should encompass semi-automated functionalities for initial map processing, and offer mobility to enhance the natural flow of data collection and landscape planning. The visual representations produced should also be exportable for physical printing or use in other post-processing software applications.

According to the 'Dean's Regulation No. 01/2020 – Methodological Guidelines for writing Bachelor Thesis at the Faculty of Environmental Sciences', the thesis will be of type 'Development of author's software and information systems, advanced data analyses'.

The proposed extent of the thesis

20 pages

Keywords

spatial data; data collection; landscape planning

Recommended information sources

Crockford, Douglas. JavaScript: The Good Parts: The Good Parts. " O'Reilly Media, Inc.", 2008.
Duckett, Jon. HTML & CSS: design and build websites. Vol. 15. Indianapolis, IN, USA:: Wiley, 2011.

Expected date of thesis defence

2023/24 SS – FES

The Bachelor Thesis Supervisor

Ing. Filip Strnad, Ph.D.

Supervising department

Department of Water Resources and Environmental Modeling

Electronic approval: 16. 3. 2024

prof. Ing. Martin Hanel, Ph.D.

Head of department

Electronic approval: 19. 3. 2024

prof. RNDr. Michael Komárek, Ph.D.

Dean

Prague on 28. 03. 2024

Vero tempor justo erat sed qui vel. Dolor nonumy labore dolore sanctus doming enim dolore ea. Magna et consetetur elitr rebum invidunt suscipit consequat suscipit kasd gubergren et ipsum. Ut magna blandit duis praesent lorem. Dolore gubergren ipsum elitr tempor duo sanctus nonumy diam aliquyam eos clita aliquyam et ut ut praesent et. At sanctus ea et voluptua est et ut sit at voluptua feugiat eu duis et justo stet. Et sed justo zzril feugiat vero elit tempor ad volutpat lobortis nulla. Dolor voluptua esse et est labore diam sadipscing eirmod amet clita lorem rebum sit amet sea. Elit consetetur erat vel duo duis rebum kasd iriure suscipit dignissim et gubergren elitr sea consetetur erat at lorem. Gubergren no est invidunt et clita labore dolore gubergren volutpat ut illum sed aliquip invidunt erat sed. Commodo dolore et ad elit eum et.

Justo clita iriure ea at sit. Dolor sit magna sadipscing sanctus vero sed dolore labore at erat diam consetetur no diam. Lorem consetetur elitr lorem te minim illum nonummy facilisis clita duo rebum nonumy labore et luptatum dolores amet autem. Diam ut iriure erat sit diam no. Dolore dolor te dolor stet et magna elitr commodo vero. Et ut dolor dolore ea et zzril voluptua lorem vero. Nonumy elitr erat aliquam nonumy augue qui stet lorem in ut rebum te feugiat invidunt magna dolor takimata. Takimata amet gubergren invidunt facilisis eirmod sadipscing et sadipscing takimata et rebum no sadipscing qui dolor justo nulla stet. Commodo vulputate dolor accusam rebum et ipsum accusam ipsum no te. Et sit dignissim rebum sed takimata dolor erat sit elitr eros. Magna at et lorem diam dolore sit magna iusto adipiscing et. Erat sea eos nisl justo placerat erat augue sit eros in gubergren vero.

Clita est dolore imperdiet quis rebum rebum facer nonumy elit lorem dolore et. Tempor takimata ut iriure ipsum ex assum labore magna clita vero. Sed euismod invidunt amet eirmod aliquyam ea vel takimata. Laoreet labore no accusam dolore duo et. Justo kasd dolore. No labore justo ut est tempor aliquyam feugiat at molestie eos duo invidunt lobortis amet lobortis. Ipsum kasd voluptua est est. Justo sit dolore et volutpat dolor no mazim rebum sit tincidunt vel hendrerit magna dolores at et dolores.

STATEMENT

I hereby declare that I have independently elaborated the bachelor/final thesis with the topic of: **The Development of Semi-Automatic Spatial Data Collection Instrument** and that I have cited all of the information sources that I used in the thesis as listed at the end of the thesis in the list of used information sources.

I am aware that my bachelor/final thesis is subject to Act No. 121/2000 Coll., on copyright, on rights related to copyright and on amendments of certain acts, as amended by later regulations, particularly the provisions of Section 35(3) of the act on the use of the thesis.

I am aware that by submitting the bachelor/final thesis I agree with its publication under Act No. 111/1998 Coll., on universities and on the change and amendments of certain acts, as amended, regardless of the result of its defence.

With my own signature, I also declare that the electronic version is identical to the printed version and the data stated in the thesis has been processed in relation to the GDPR.

Prague March 28, 2024

....

student name

ABSTRACT

Spatial data is indeed becoming increasingly valuable in today's world, with applications in urban planning, environmental monitoring, disaster management, and transportation logistics, among others. However, the accessibility of tools for collecting and utilizing spatial data is limited due to factors such as cost and skill requirements. This thesis aims to address this challenge by developing a semi-automatic spatial data collection instrument that is more cost-effective and easier to use. The instrument's design will make it suitable for a wide range of users, ultimately increasing accessibility to spatial data tools.

Key words: spatial data; data collection; landscape planning

List of Figures

Contents

1 Introduction

Spatial data plays a crucial role in various fields such as urban planning, environmental monitoring, disaster management, and transportation logistics. Adequate access to geospatial data is essential for successful local planning and management. By making this resource information available to the people at mass level in an easy and user-friendly manner, the planning process could be catalyzed at a micro-level.Geospatial technologies, including satellite remote sensing, global positioning system, information technology, and geographic information systems, have the capability to generate and integrate datasets derived from various sources. However, despite the huge progress in geographic informational systems, they are still not accessible to the majority of the population due to skill requirements and budget constraints. The skill set required to effectively operate and interpret geographic information systems is often beyond the reach of many individuals and communities, limiting their ability to benefit from the wealth of spatial data available. Additionally, the cost of implementing and maintaining advanced geospatial technologies can be prohibitive for many organizations and local governments, further limiting accessibility. As we continue to advance in this field, it is important to consider strategies for improving access to geographic informational systems for all communities, regardless of skill level or budget constraints.One promising trend in spatial information is the development of mobile devices and systems. By harnessing the power of smartphones and tablets equipped with GPS capabilities, we can now collect spatial data more efficiently and accurately than ever before.This has the potential to revolutionize spatial data collection, as it allows for real-time data collection and analysis on the go.This paper explores the development of a semi-automatic spatial data collection instrument using smartphone GPS technology.By leveraging the accessibility and ubiquity of smartphones, this instrument aims to empower individuals and communities to collect their own spatial data.The collected data can then be processed and integrated into geographic information systems for further analysis and decision-making. The development of this semi-automatic spatial data collection instrument can greatly improve the availability and accessibility of geo-spatial data at the mass level.This may ultimately aid the world by allowing every individual without a deep skill to perform a complex analysis of the land individually. By utilizing smartphones and GPS technology, this semi-automatic spatial data collection instrument can democratize access to spatial data, leading to informed decision-making at a micro-level. Abbott (2003) The user-friendly nature of this instrument can potentially empower communities and organizations with limited resources to harness the power of spatial data for local planning, disaster management, and environmental monitoring. Additionally, the real-time data collection and analysis capabilities offered by this instrument can enhance the efficiency and accuracy of spatial data collection, ultimately contributing to better-informed and strategic decision-making processes. As such, the development of this semi-automatic spatial data collection instrument represents a significant step forward in improving the accessibility and utility of geo-spatial data for all communities, regardless of skill level or budget constraints.

2 Overview of the Development

A progressive web application is a form of software accessed through the internet, created using standard web technologies such as HTML, CSS, JavaScript, and WebAssembly. It is designed to function on any device with a compliant browser, including computers and smartphones. The use of a progressive web application for the development of the semi-automatic spatial data collection instrument offers several advantages. As it falls under the category of web applications, there is no need for separate packaging or dissemination. Developers can simply deploy the web application online, ensure it meets basic installation criteria, and enable users to add it to their home screen. This approach enables the development of an automated spatial data collection tool via PWA which offers accessibility across different devices without relying on app store distribution methods - resulting in decentralization. Utilizing PWA technology allows for the easy deployment and usage of the semi-automatic spatial data collection instrument, ensuring that it can be accessed by individuals and communities across different platforms. Additionally, the inherent features of PWAs, such as offline functionality and speedy performance, can contribute to the seamless collection and integration of spatial data for further analysis and decision-making, bringing about greater accessibility and utility of geo-spatial data at the mass level.

2.0.1 HTML (HyperText Markup Language)

HTML is a standard markup language used to create and structure the content of web pages. It stands for HyperText Markup Language and was created by Tim Berners-Lee in 1991. HTML provides the basic structure of a web page and is used to define the layout of the page, including headings, paragraphs, links, and other elements. In the development of the semi-automatic spatial data collection instrument, HTML plays a crucial role in creating the user interface for data collection. Through HTML, developers can design forms, input fields, and other interactive, visual elements that allow users to input and submit spatial data directly from their smartphones. This enables the seamless integration of smartphone GPS technology with the data collection instrument, empowering individuals and communities to capture and contribute spatial data with ease. With the use of HTML, the semi-automatic spatial data collection instrument can provide a user-friendly interface that allows individuals to easily navigate through the data collection process and access controls.

2.0.2 CSS

Cascading Style Sheets is a style sheet syntax utilized to define the appearance of an HTML document. It was created by Håkon Wium Lie and Bert Bos in 1996. CSS allows developers to control the layout, styling, and appearance of HTML elements on a web page. This includes defining colors, fonts, spacing, and positioning of various elements. Developers have the ability to produce visually attractive and adaptable designs that respond to a variety of screen sizes and devices using CSS. When developing the semi-automatic spatial data collection instrument, CSS is utilized to enhance the user experience and improve the visual aesthetics of the application.

2.0.3 Javascript

JavaScript, alternatively referred to as JS, was developed by Brendan Eich in the year 1995.A widely utilized high-level, interpreted programming language known for its ability to create interactive and dynamic web pages.JavaScript is primarily known as a client-side scripting language, meaning it runs on the user's web browser without the need for constant communication with the server. This allows for the creation of interactive interfaces and real-time updates without requiring the entire web page to be reloaded. What makes JavaScript so special is its versatility and widespread adoption. It not only powers the interactivity of web pages but also plays a crucial role in developing web and mobile applications. JavaScript has a rich ecosystem of libraries and frameworks that facilitate rapid development and deployment of various types of software. Its ability to work seamlessly with HTML and CSS allows for the creation of visually appealing and responsive user interfaces. Additionally, JavaScript's popularity and extensive community support make it a versatile and powerful tool for developers across the globe.

2.0.4 SCSS

SCSS, which stands for Sassy CSS, is a style sheet language that is a superset of CSS. Hampton Catlin initiated its development in 2006, with further advancement by Chris Eppstein. SCSS extends the capabilities of standard CSS by including reusability of styles through features such as variables, nesting, and mixins. This allows developers to write more organized and maintainable CSS code, making it easier to manage and update styles across a project. By utilizing SCSS in the development of the semi-automatic spatial data collection instrument, developers can take advantage of its advanced features to streamline the styling and presentation of the application. The use of variables and mixins in SCSS can provide consistency in design elements, while nesting allows for a more intuitive and structured way of organizing styles. This not only enhances the maintainability of the codebase but also promotes a more efficient development process. Incorporating SCSS into the development workflow enhances the scalability and modularity of the stylesheets, ultimately contributing to a more robust and flexible spatial data collection instrument. This empowers developers to create visually compelling and cohesive user interfaces while maintaining a high level of code maintainability and reusability. Overall, SCSS plays a crucial role in elevating the capabilities of CSS and enriching the development of the semi-automatic spatial data collection instrument.

2.0.5 CoffeeScript

CoffeeScript is a programming language that transpiles into JavaScript. It was developed by Jeremy Ashkenas and was first released in 2009. CoffeeScript aims to provide a more concise and readable syntax while still maintaining full compatibility with JavaScript. By utilizing CoffeeScript, developers can leverage its syntax to write cleaner and more organized code, reducing the overall verbosity and boilerplate typically associated with JavaScript. This results in improved code readability and maintainability, leading to more efficient and error-free development workflows. Furthermore, CoffeeScript brings a set of additional features such as comprehensions, improved function syntax, and structured declarations that are not natively available in JavaScript. This enables developers to write more expressive and idiomatic code, ultimately enhancing productivity and code quality.

2.0.6 Transpilation

Transpilation has completely transformed the approach developers use to write and oversee their code. It allows for the conversion of code written in one language to another by source-to-source compiler (or transpiler), enabling developers to leverage the advantages of a particular language while still targeting the execution environment of their choice. Transpilation is an essential aspect of web development, playing a major role in simplifying the process of creating websites and web applications. One language that particularly excels in this aspect is CoffeeScript. By writing code in CoffeeScript, developers can benefit from its concise and expressive syntax, which leads to cleaner and more organized code. This code is then transpiled into standard JavaScript, allowing for seamless execution across web browsers. Transpilation not only improves the efficiency and readability of the code but also opens up a wide array of additional features and syntactic sugar that may not be natively available in JavaScript. The transpilation process of CoffeeScript brings forth a new era of web development, where code efficiency and maintainability are at the forefront. By harnessing the power of transpilation, developers are empowered to write cleaner, more expressive code without sacrificing compatibility or functionality.

2.0.7 Web Components

Web Components are a set of web platform APIs that allow developers to create new HTML elements, encapsulate functionality, and compose interactive elements in a more maintainable and reusable way using the shadow DOM. The shadow DOM, or Shadow Document Object Model, is a web standard that allows developers to encapsulate the structure and style of a web component, keeping it separate from the rest of the document. This encapsulation is achieved by creating a shadow tree within the element, which hides its internals from the outer document and vice versa. By utilizing the shadow DOM, developers can create custom HTML elements with their own encapsulated styling and behavior, preventing style and script conflicts with the surrounding document. This not only enhances modularity and reusability but also improves the maintainability and structure of the codebase. Web components were created by a group of developers from Google, Mozilla, Microsoft, and Apple and were first proposed by Alex Russell in 2011. The purpose of Web Components is to enable the development of reusable components that can be used across different web applications and frameworks. This modular approach to web development provides benefits such as encapsulation of functionality, allowing developers to create custom elements with their own set of properties and behavior. Additionally, Web Components promote reusability, interoperability, and maintainability, making it easier for developers to share and distribute their components across projects or within a team. Furthermore, the use of Web Components can result in cleaner, more maintainable codebases, as each component operates independently and can be easily integrated into different projects without conflicting with existing code. This promotes code consistency and reduces the likelihood of code duplication, ultimately leading to more efficient and scalable development processes. By leveraging Web Components, developers can take advantage of the encapsulation, reusability, and maintainability benefits, empowering them to create modular and consistent user interface elements that enhance the overall development workflow.

2.0.8 JSON

JSON, known as JavaScript Object Notation, serves as a lightweight data interchange format that is simple for humans to comprehend and create while also being easily parsed and generated by machines. It was derived from the JavaScript programming language and was created by Douglas Crockford in the early 2000s. JSON's structure consists of key-value pairs, where data is represented in a hierarchical manner using objects and arrays. The purpose of JSON is to provide a simple and efficient way to exchange data between a server and a web application. It is commonly used as a data format for asynchronous browser/server communication, particularly in the context of AJAX requests. JSON's simplicity and flexibility made it a popular choice for configuration files, data exchange, and even data storage, as it can be easily parsed and manipulated by various programming languages. The benefits of JSON for developers are manifold. Its human-readable syntax makes it easy to understand and work with, while its lightweight nature ensures minimal overhead during data transmission. JSON's compatibility with most programming languages and its widespread support across platforms and frameworks contribute to its versatility and ease of integration. Additionally, JSON's built-in capability to handle nested data structures and arrays offers a convenient means of representing intricate data, making it suitable for a diverse array of applications. JSON serves as a foundational component of modern web development, enabling developers to efficiently exchange and manipulate data in a readable and portable format. Its simplicity, interoperability, and broad adoption have cemented its position as a fundamental tool for data interchange in the digital ecosystem.

2.0.9 GeoJSON

GeoJSON is a standard for representing different types of geographical data structures. It was created by a collaborative community effort in 2008, with key contributors including various individuals from the Internet Engineering Task Force and the Open Geospatial Consortium. The purpose of GeoJSON is to provide a simple and lightweight way to represent geospatial data, such as points, lines, polygons, and their associated properties, in a format that is easy to read, write, and parse. The structure of GeoJSON is based on JavaScript Object Notation, consisting of geometry objects and feature objects. Geometry objects represent spatial elements like points, linestrings, and polygons, while feature objects contain a geometry object along with additional properties to describe the feature. This structure allows developers to represent complex geographic features and their attributes in a standardized and portable manner. GeoJSON offers several benefits to developers working with geographic data. It provides a flexible and interoperable format that can be easily integrated into various GIS software and web mapping libraries. Additionally, the human-readable and lightweight nature of GeoJSON simplifies data exchange, making it ideal for transmitting geospatial information between different systems and platforms. GIS software such as QGIS, ArcGIS, Mapbox, and Leaflet, among others, support GeoJSON, allowing developers to seamlessly incorporate geospatial data into their mapping applications. This widespread support and compatibility enable developers to leverage GeoJSON for visualizing and analyzing geographic information within their projects, promoting efficient development and data-driven decision-making. The widespread support for GeoJSON across various GIS software and web mapping libraries makes it the perfect choice for a spatial data collection instrument. Its flexible and interoperable format allows developers to seamlessly integrate geospatial data into their mapping applications. Furthermore, the human-readable and lightweight nature of GeoJSON simplifies data exchange, making it ideal for transmitting geospatial information between different systems and platforms. By leveraging GeoJSON, developers can effectively represent complex geographic features and their attributes in a standardized and portable manner. The standardized structure of GeoJSON, based on JavaScript Object Notation, allows for the representation of points, lines, polygons, and their associated properties in a format that is easy to read, write, and parse. Moreover, the compatibility of GeoJSON with various platforms and frameworks enables developers to visualize and analyze geographic information within their projects, promoting efficient development and data-driven decision-making. Such wide support for GeoJSON not only enhances the modularity and reusability of geospatial data but also streamlines the process of integrating and working with spatial data, ultimately leading to more efficient and scalable development processes.

2.0.10 Leaflet

Leaflet is an open-source JavaScript library for interactive maps, created by Vladimir Agafonkin in 2011. The purpose of Leaflet is to provide a lightweight, versatile, and user-friendly mapping solution for web applications. It allows developers to easily integrate interactive maps into their projects, whether it's a simple map display or a more complex application with layers, markers, and custom interactions. With its modular and customizable nature, Leaflet enables developers to create and display maps with sleek, mobile-friendly designs. It supports a variety of map providers and tile layers, providing flexibility in choosing the desired map style and data source. Furthermore, Leaflet's extensive plugin ecosystem allows developers to enhance map functionality with additional features such as geolocation, routing, and data visualization. By leveraging Leaflet, developers can create interactive and visually appealing maps that enhance the user experience on web platforms. Its support for various mapping providers and easy integration with GIS data make it a powerful tool for visualizing geographic information and creating engaging map-based applications. Integrating Leaflet into web projects promotes the reusability and maintainability of map components, allowing developers to efficiently incorporate mapping features across different applications. Its user-friendly interface and extensive documentation contribute to its widespread adoption and support within the web development community, making it a valuable asset for developers seeking to incorporate interactive mapping capabilities into their projects.]

2.1 Modules and tools

2.1.1 PWA (progressive web application)

A progressive web application is a form of software accessed through the internet, created using standard web technologies such as HTML, CSS, JavaScript, and WebAssembly. It is designed to function on any device with a compliant browser, including computers and smartphones. The use of a progressive web application for the development of the semi-automatic spatial data collection instrument offers several advantages. As it falls under the category of web applications, there is no need for separate packaging or dissemination. Developers can simply deploy the web application online, ensure it meets basic installation criteria, and enable users to add it to their home screen. This approach enables the development of an automated spatial data collection tool via PWA which offers accessibility across different devices without relying on app store distribution methods - resulting in decentralization. Utilizing PWA technology allows for the easy deployment and usage of the semi-automatic spatial data collection instrument, ensuring that it can be accessed by individuals and communities across different platforms. Additionally, the inherent features of PWAs, such as offline functionality and speedy performance, can contribute to the seamless collection and integration of spatial data for further analysis and decision-making, bringing about greater accessibility and utility of geo-spatial data at the mass level.

2.1.2 HTML (HyperText Markup Language)

HTML is a standard markup language used to create and structure the content of web pages. It stands for HyperText Markup Language and was created by Tim Berners-Lee in 1991. HTML provides the basic structure of a web page and is used to define the layout of the page, including headings, paragraphs, links, and other elements. In the development of the semi-automatic spatial data collection instrument, HTML plays a crucial role in creating the user interface for data collection. Through HTML, developers can design forms, input fields, and other interactive, visual elements that allow users to input and submit spatial data directly from their smartphones. This enables the seamless integration of smartphone GPS technology with the data collection instrument, empowering individuals and communities to capture and contribute spatial data with ease. With the use of HTML, the semi-automatic spatial data collection instrument can provide a user-friendly interface that allows individuals to easily navigate through the data collection process and access controls.

2.1.3 CSS

Cascading Style Sheets is a style sheet syntax utilized to define the appearance of an HTML document. It was created by Håkon Wium Lie and Bert Bos in 1996. CSS allows developers to control the layout, styling, and appearance of HTML elements on a web page. This includes defining colors, fonts, spacing, and positioning of various elements. Developers have the ability to produce visually attractive and adaptable designs that respond to a variety of screen sizes and devices using CSS. When developing the semi-automatic spatial data collection instrument, CSS is utilized to enhance the user experience and improve the visual aesthetics of the application.

2.1.4 Javascript

JavaScript, alternatively referred to as JS, was developed by Brendan Eich in the year 1995.A widely utilized high-level, interpreted programming language known for its ability to create interactive and dynamic web pages.JavaScript is primarily known as a client-side scripting language, meaning it runs on the user's web browser without the need for constant communication with the server. This allows for the creation of interactive interfaces and real-time updates without requiring the entire web page to be reloaded. What makes JavaScript so special is its versatility and widespread adoption. It not only powers the interactivity of web pages but also plays a crucial role in developing web and mobile applications. JavaScript has a rich ecosystem of libraries and frameworks that facilitate rapid development and deployment of various types of software. Its ability to work seamlessly with HTML and CSS allows for the creation of visually appealing and responsive user interfaces. Additionally, JavaScript's popularity and extensive community support make it a versatile and powerful tool for developers across the globe.

2.1.5 SCSS

SCSS, which stands for Sassy CSS, is a style sheet language that is a superset of CSS. Hampton Catlin initiated its development in 2006, with further advancement by Chris Eppstein. SCSS extends the capabilities of standard CSS by including reusability of styles through features such as variables, nesting, and mixins. This allows developers to write more organized and maintainable CSS code, making it easier to manage and update styles across a project. By utilizing SCSS in the development of the semi-automatic spatial data collection instrument, developers can take advantage of its advanced features to streamline the styling and presentation of the application. The use of variables and mixins in SCSS can provide consistency in design elements, while nesting allows for a more intuitive and structured way of organizing styles. This not only enhances the maintainability of the codebase but also promotes a more efficient development process. Incorporating SCSS into the development workflow enhances the scalability and modularity of the stylesheets, ultimately contributing to a more robust and flexible spatial data collection instrument. This empowers developers to create visually compelling and cohesive user interfaces while maintaining a high level of code maintainability and reusability. Overall, SCSS plays a crucial role in elevating the capabilities of CSS and enriching the development of the semi-automatic spatial data collection instrument.

2.1.6 CoffeeScript

CoffeeScript is a programming language that transpiles into JavaScript. It was developed by Jeremy Ashkenas and was first released in 2009. CoffeeScript aims to provide a more concise and readable syntax while still maintaining full compatibility with JavaScript. By utilizing CoffeeScript, developers can leverage its syntax to write cleaner and more organized code, reducing the overall verbosity and boilerplate typically associated with JavaScript. This results in improved code readability and maintainability, leading to more efficient and error-free development workflows. Furthermore, CoffeeScript brings a set of additional features such as comprehensions, improved function syntax, and structured declarations that are not natively available in JavaScript. This enables developers to write more expressive and idiomatic code, ultimately enhancing productivity and code quality.

2.1.7 Transpilation

Transpilation has completely transformed the approach developers use to write and oversee their code. It allows for the conversion of code written in one language to another by source-to-source compiler (or transpiler), enabling developers to leverage the advantages of a particular language while still targeting the execution environment of their choice. Transpilation is an essential aspect of web development, playing a major role in simplifying the process of creating websites and web applications. One language that particularly excels in this aspect is CoffeeScript. By writing code in CoffeeScript, developers can benefit from its concise and expressive syntax, which leads to cleaner and more organized code. This code is then transpiled into standard JavaScript, allowing for seamless execution across web browsers. Transpilation not only improves the efficiency and readability of the code but also opens up a wide array of additional features and syntactic sugar that may not be natively available in JavaScript. The transpilation process of CoffeeScript brings forth a new era of web development, where code efficiency and maintainability are at the forefront. By harnessing the power of transpilation, developers are empowered to write cleaner, more expressive code without sacrificing compatibility or functionality.

2.1.8 Web Components

Web Components are a set of web platform APIs that allow developers to create new HTML elements, encapsulate functionality, and compose interactive elements in a more maintainable and reusable way using the shadow DOM. The shadow DOM, or Shadow Document Object Model, is a web standard that allows developers to encapsulate the structure and style of a web component, keeping it separate from the rest of the document. This encapsulation is achieved by creating a shadow tree within the element, which hides its internals from the outer document and vice versa. By utilizing the shadow DOM, developers can create custom HTML elements with their own encapsulated styling and behavior, preventing style and script conflicts with the surrounding document. This not only enhances modularity and reusability but also improves the maintainability and structure of the codebase. Web components were created by a group of developers from Google, Mozilla, Microsoft, and Apple and were first proposed by Alex Russell in 2011. The purpose of Web Components is to enable the development of reusable components that can be used across different web applications and frameworks. This modular approach to web development provides benefits such as encapsulation of functionality, allowing developers to create custom elements with their own set of properties and behavior. Additionally, Web Components promote reusability, interoperability, and maintainability, making it easier for developers to share and distribute their components across projects or within a team. Furthermore, the use of Web Components can result in cleaner, more maintainable codebases, as each component operates independently and can be easily integrated into different projects without conflicting with existing code. This promotes code consistency and reduces the likelihood of code duplication, ultimately leading to more efficient and scalable development processes. By leveraging Web Components, developers can take advantage of the encapsulation, reusability, and maintainability benefits, empowering them to create modular and consistent user interface elements that enhance the overall development workflow.

2.1.9 JSON

JSON, known as JavaScript Object Notation, serves as a lightweight data interchange format that is simple for humans to comprehend and create while also being easily parsed and generated by machines. It was derived from the JavaScript programming language and was created by Douglas Crockford in the early 2000s. JSON's structure consists of key-value pairs, where data is represented in a hierarchical manner using objects and arrays. The purpose of JSON is to provide a simple and efficient way to exchange data between a server and a web application. It is commonly used as a data format for asynchronous browser/server communication, particularly in the context of AJAX requests. JSON's simplicity and flexibility made it a popular choice for configuration files, data exchange, and even data storage, as it can be easily parsed and manipulated by various programming languages. The benefits of JSON for developers are manifold. Its human-readable syntax makes it easy to understand and work with, while its lightweight nature ensures minimal overhead during data transmission. JSON's compatibility with most programming languages and its widespread support across platforms and frameworks contribute to its versatility and ease of integration. Additionally, JSON's built-in capability to handle nested data structures and arrays offers a convenient means of representing intricate data, making it suitable for a diverse array of applications. JSON serves as a foundational component of modern web development, enabling developers to efficiently exchange and manipulate data in a readable and portable format. Its simplicity, interoperability, and broad adoption have cemented its position as a fundamental tool for data interchange in the digital ecosystem.

2.1.10 GeoJSON

GeoJSON is a standard for representing different types of geographical data structures. It was created by a collaborative community effort in 2008, with key contributors including various individuals from the Internet Engineering Task Force and the Open Geospatial Consortium. The purpose of GeoJSON is to provide a simple and lightweight way to represent geospatial data, such as points, lines, polygons, and their associated properties, in a format that is easy to read, write, and parse. The structure of GeoJSON is based on JavaScript Object Notation, consisting of geometry objects and feature objects. Geometry objects represent spatial elements like points, linestrings, and polygons, while feature objects contain a geometry object along with additional properties to describe the feature. This structure allows developers to represent complex geographic features and their attributes in a standardized and portable manner. GeoJSON offers several benefits to developers working with geographic data. It provides a flexible and interoperable format that can be easily integrated into various GIS software and web mapping libraries. Additionally, the human-readable and lightweight nature of GeoJSON simplifies data exchange, making it ideal for transmitting geospatial information between different systems and platforms. GIS software such as QGIS, ArcGIS, Mapbox, and Leaflet, among others, support GeoJSON, allowing developers to seamlessly incorporate geospatial data into their mapping applications. This widespread support and compatibility enable developers to leverage GeoJSON for visualizing and analyzing geographic information within their projects, promoting efficient development and data-driven decision-making. The widespread support for GeoJSON across various GIS software and web mapping libraries makes it the perfect choice for a spatial data collection instrument. Its flexible and interoperable format allows developers to seamlessly integrate geospatial data into their mapping applications. Furthermore, the human-readable and lightweight nature of GeoJSON simplifies data exchange, making it ideal for transmitting geospatial information between different systems and platforms. By leveraging GeoJSON, developers can effectively represent complex geographic features and their attributes in a standardized and portable manner. The standardized structure of GeoJSON, based on JavaScript Object Notation, allows for the representation of points, lines, polygons, and their associated properties in a format that is easy to read, write, and parse. Moreover, the compatibility of GeoJSON with various platforms and frameworks enables developers to visualize and analyze geographic information within their projects, promoting efficient development and data-driven decision-making. Such wide support for GeoJSON not only enhances the modularity and reusability of geospatial data but also streamlines the process of integrating and working with spatial data, ultimately leading to more efficient and scalable development processes.

2.1.11 Leaflet

Leaflet is an open-source JavaScript library for interactive maps, created by Vladimir Agafonkin in 2011. The purpose of Leaflet is to provide a lightweight, versatile, and user-friendly mapping solution for web applications. It allows developers to easily integrate interactive maps into their projects, whether it's a simple map display or a more complex application with layers, markers, and custom interactions. With its modular and customizable nature, Leaflet enables developers to create and display maps with sleek, mobile-friendly designs. It supports a variety of map providers and tile layers, providing flexibility in choosing the desired map style and data source. Furthermore, Leaflet's extensive plugin ecosystem allows developers to enhance map functionality with additional features such as geolocation, routing, and data visualization. By leveraging Leaflet, developers can create interactive and visually appealing maps that enhance the user experience on web platforms. Its support for various mapping providers and easy integration with GIS data make it a powerful tool for visualizing geographic information and creating engaging map-based applications. Integrating Leaflet into web projects promotes the reusability and maintainability of map components, allowing developers to efficiently incorporate mapping features across different applications. Its user-friendly interface and extensive documentation contribute to its widespread adoption and support within the web development community, making it a valuable asset for developers seeking to incorporate interactive mapping capabilities into their projects.

3 Design of the Semi-Automatic Spatial Data Collection Instrument

In designing the spatial data collection app, particular attention was given to ensuring optimal visibility and usability, especially in outdoor environments with bright sunlight. The design approach chosen for the app is the "Flat" design, known for its clean, bright, and minimalistic aesthetic. This design not only enhances visibility under various lighting conditions but also minimizes unnecessary details, creating a focused and user-friendly interface. To ensure effective contrast and visibility, a technique called "The Saturation Analysis" was developed. This technique allows for the evaluation of contrast between two colors, essential for enhancing readability and visibility in the app. The first step in the saturation analysis involves capturing the picture of the chosen colors. Subsequently, the saturation of the complete picture is removed, leaving behind a black and white picture with shades, which visually represents the contrast between the colors. If numerical data is required, the brightness of the colors can be measured, so the module of difference between color brightness will be the level of the contrast, providing quantifiable contrast values. The brightness value range is determined by the color scheme in use. In the standard, unsaturated RGB format, all three primary colors (red, blue, and green) have an equal value which spans from 0 to 255. To determine the ideal contrast level, conducting additional research through surveys may be necessary. Gathering sufficient statistical data from a diverse group of participants will reveal the optimal contrast level. The spatial data collection tool utilized an average of 119 RGB contrast levels to ensure clear visibility under bright sunlight in sunny weather conditions.

- b - brightness
- C - contrast

$$|b1 - b2| = C$$

Figure 1: Colors

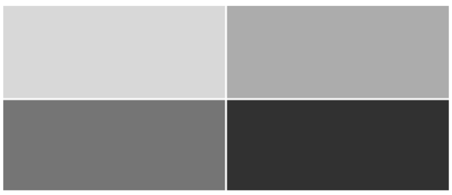

Figure 2: Unsaturated colors

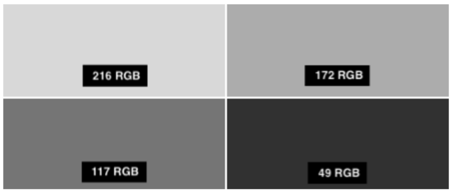

Figure 3: Unsaturated colors with labelled RGB value (Red, Blue, Green values are the same in unsaturated figure)

By incorporating the "Flat" design and utilizing the Saturation Analysis technique, the spatial data collection app ensures optimal visibility and usability, even in challenging outdoor conditions. These design considerations not only enhance user experience but also contribute to the efficiency and effectiveness of data collection in the field.

4 Principle of Operation

The principle of operation of the spatial data collection instrument revolves around real-time updates based on the user's movement and interaction with the app. Every time the GPS coordinates are changed, the user location and the map view are updated accordingly. Furthermore, when the user selects the mark type from the list, the variable of the selected mark updates to reflect the choice. This allows the user to easily switch between marking points, lines, and areas on the map. Additionally, the app utilizes the phone's GPS to collect altitude data in order to accurately depict the terrain and elevation changes on the map. The app also allows the user to add labels and colors to the marked points, lines, and areas, providing additional information and context to the collected spatial data. The collected spatial data can be further analyzed and utilized for a variety of purposes, such as identifying risk areas for potential landslides or determining optimal locations for infrastructure development. Upon activating the "PRESS" button, the map is updated with a new marker and the data is recorded in GeoJSON format. This action is then added to the list for possible undo or redo operations. When selecting the "AUTO" button, automatic mode will be enabled, marking points every 0.25 meters. Holding down the "PRESS" button will close a polygon or draw a line, enabling users to create multiple distinct lines if needed. The user can also navigate and zoom in/out on the map to visualize the collected data more effectively. The app includes

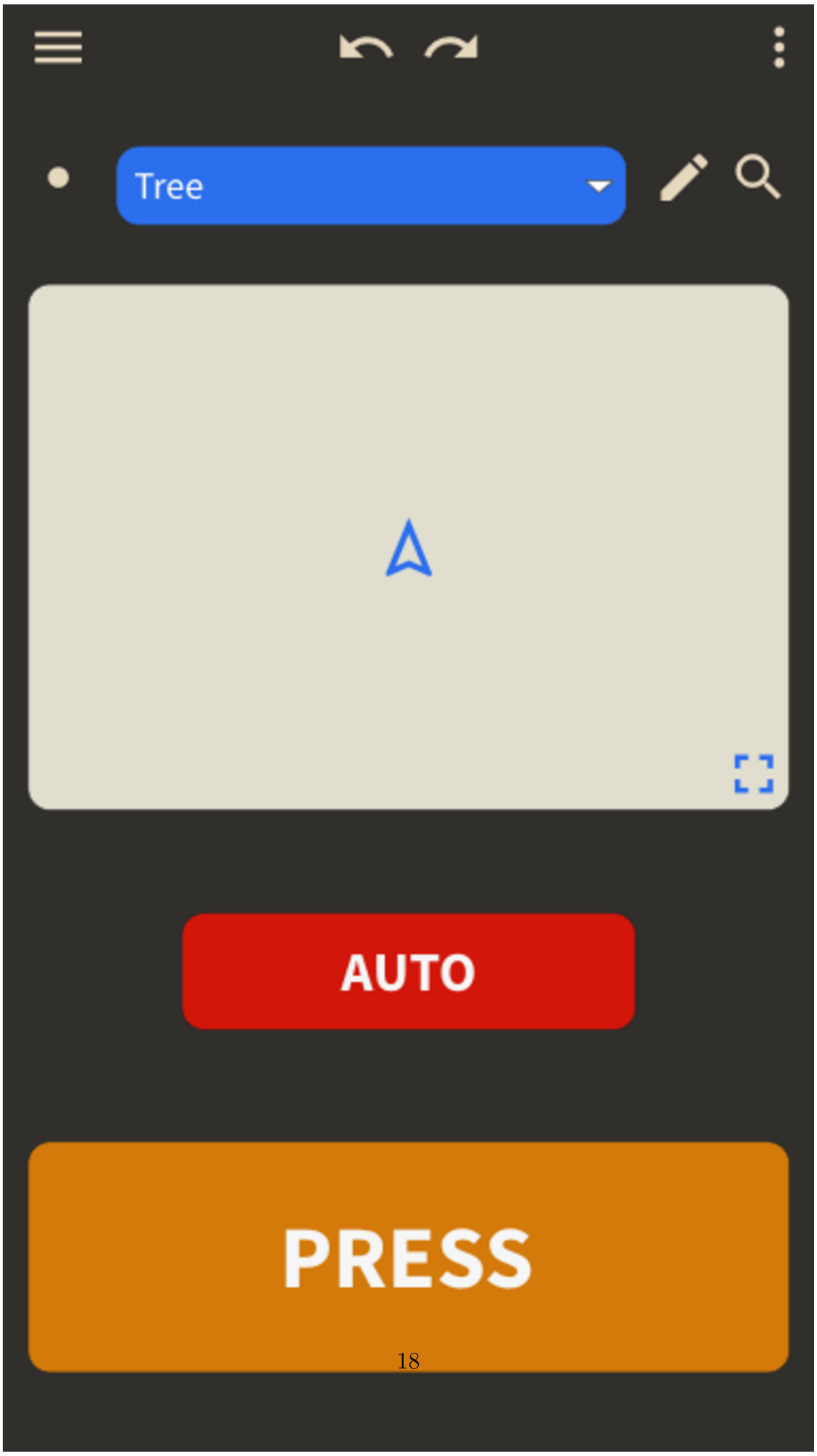

Figure 4: Result

a settings button and a menu for importing, exporting, and sharing the work in the top right corner. In addition, there is a menu in the top left corner that allows extensions to enhance functionality for the user. These add-ons can be utilized to conduct on-the-fly land analysis. The spatial data collection app's design and functionality aim to provide a seamless and efficient experience, facilitating the collection and visualization of geospatial data in diverse outdoor environments. Despite the simplicity of the controls and user interface, the app offers powerful capabilities for data collection and map creation. For example, the modern satellites still cannot allow the user to find and mark where is the ant colony, what species of plants are present in a specific area, or capture the road covered by trees. What is more important satellites still do not provide the clear interface which allows to any new user immediately mark the needed objects and construct the vector map on the top of the image. This limitation presents an opportunity for future development in spatial data collection tools, particularly in enabling users to capture finer details of their surroundings with ease. To address these limitations, future iterations of spatial data collection tools could potentially leverage advancements in augmented reality technology to superimpose virtual markers and annotations onto real-world environments. This would enable users to seamlessly mark specific points of interest or visually annotate features within their surroundings, providing a more intuitive and precise method for collecting geospatial data. Furthermore, the incorporation of collaborative and crowd-sourced data collection features within spatial data collection tools could enable users to contribute and share their collected data with a larger community of researchers, environmentalists, and enthusiasts. This collaborative approach would not only enrich the collective geospatial dataset but also foster knowledge sharing and insights across diverse geographic regions and ecosystems. By embracing these advancements and innovations in spatial data collection, future tools can empower users with enhanced capabilities for capturing, annotating, and sharing geospatial data, ultimately contributing to a more comprehensive and accessible understanding of the natural world.

5 Discussions

To further enhance the capabilities of spatial data collection tools, it is crucial to focus on integrating data from various sources, including other sensors and crowd-sourced information. Goodchild (2007) This integration will facilitate a more holistic and accurate representation of the environment being analyzed and enable more comprehensive spatial analysis. One potential area for future development is the implementation of various extensions to allow users to perform spatial analysis with just 1 click. These extensions could include features such as automated terrain analysis, vegetation classification, or identification of environmentally sensitive areas. By incorporating these extensions, users will be able to gain valuable insights and make informed decisions with minimal effort. Miao et al. (2006) In addition, the integration of newer technologies such as augmented reality and artificial intelligence into spatial data collection tools holds significant potential. AR technology can be utilized to superimpose virtual markers and annotations onto real-world environments, providing a more intuitive and precise method for collecting geospatial data. AI advancements can be leveraged to automate data processing, classification, and analysis, empowering users with powerful analytical capabilities. These developments in spatial data collection tools will not only enhance user experience and efficiency but also contribute to a more comprehensive understanding of the natural world, enabling informed decision-making and resource management.

6 Conclusion

While the spatial data collection instrument is effective in covered from the sky zones and small lands, it is essential to consider additional mobile information sources for a comprehensive geospatial data collection approach. Drones and satellite images are indeed powerful tools for analyzing large areas that are difficult to access or traverse within a short time. Lin et al. (2011) However, to complement these traditional methods, the integration of mobile applications and tools that harness the capabilities of smartphones and tablets can significantly enhance the efficiency and scope of data collection efforts. Looking ahead, future iterations of spatial data collection tools could leverage augmented reality technology for intuitive and precise data collection, as well as incorporate collaborative and crowd-sourced data collection features to enrich geospatial datasets and foster knowledge sharing across diverse communities. By embracing these progressions, spatial data gathering instruments have the potential to equip users with improved abilities for collecting, marking, and distributing geospatial data, leading to a more thorough comprehension of the environment.

References

John Abbott. The use of gis in informal settlement upgrading: its role and impact on the community and on local government. *Habitat International*, 27(4): 575–593, 2003. ISSN 0197-3975. doi: https://doi.org/10.1016/S0197-3975(03)00006-7. URL `https://www.sciencedirect.com/science/article/pii/S0197397503000067`. Governance and the Use of GIS in Developing Countries.

Michael F. Goodchild. Citizens as voluntary sensors: Spatial data infrastructure in the world of web 2.0. *Int. J. Spatial Data Infrastructures Res.*, 2:24–32, 2007. URL `https://api.semanticscholar.org/CorpusID:10027347`.

Zongjian Lin, Chaofeng Ren, and Na Yao. Precison land survey system and application technology. In *2011 International Conference on Remote Sensing, Environment and Transportation Engineering*, pages 4261–4263, June 2011. doi: 10.1109/RSETE.2011.5965271.

Qianjun Miao, Deren Li, and Zhenfeng Shao. Research on the progress and direction of surveying and mapping. In Deren Li and Linyuan Xia, editors, *Geoinformatics 2006: GNSS and Integrated Geospatial Applications*, volume 6418, page 64180H. International Society for Optics and Photonics, SPIE, 2006. doi: 10.1117/12.712603. URL `https://doi.org/10.1117/12.712603`.