GeoScene
Augmented Reality and POI Geolocation Scenery viewer Android mobile application

Itay Bouganim, Sahar Vaya, Lior Hassan

Software Engineering final project

Ben-Gurion University of the Negev

Contents

Chapter 1
Introduction

Overview

This project goal is to create an Android mobile devices application which will use Augmented Reality technology in order to provide the user spatial information. The provided information will include the mobile device user surrounding points of interest by using the mobile device camera and sensors and wrap the viewed world with location markers indicating the location of the said points of interest.

The application will also enable the user to take only the points of interests that are in their physical line of sight into consideration. Additional information that will be provided regarding each location marker and additional supported functionality will be elaborated in this document.

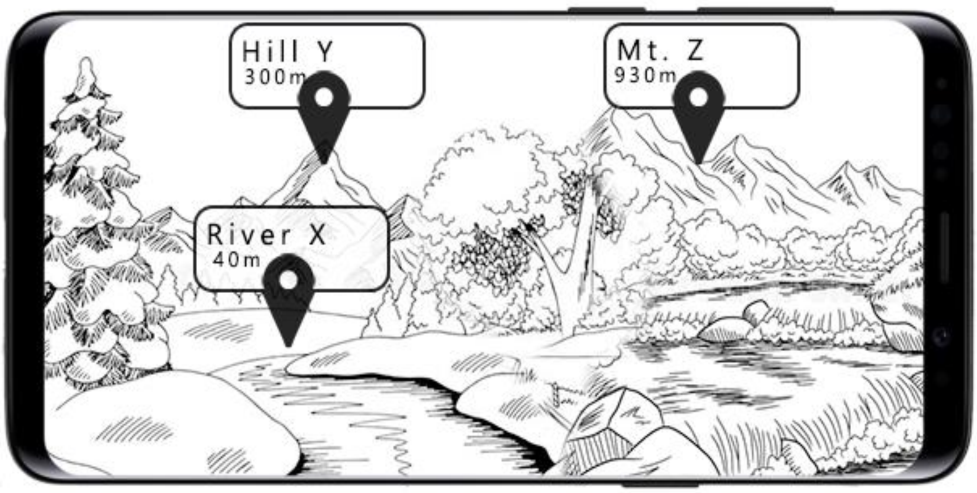


Illustration of the application observing in a nature reserve

Illustration of the application near an urban area

Background

In recent years Augmented Reality (AR) technology became very popular and limitations of marker-based augmented reality applications, which would require scanning a real-world marker in order to display a model at its location, can now be solved by using marker-less augmented reality.
Both video games and cellphones are driving the development of augmented reality. Everyone from tourists, to soldiers, to someone looking for the closest subway stop can now benefit from the ability to place computer-generated graphics in their field of vision. In this project, we will develop an Android mobile application that uses marker-less AR which relies on location data instead of physical markers in order to display users their surrounding Points of interest (POI) and help them explore the scenery. POIs will be gathered from open-source mapping services and from internal backend source having predefined locations saved by their latitude and longitude coordinates. The application will integrate Google's Augmented Reality SDK called ARCore to render objects at various points of interest in the real world.

1.1 The Problem Domain

This section will primarily focus on presenting and describing the main domains and difficulties that this project will address during the development phase.

Open-Source Usage

The project will mainly use open-source data, SDKs and APIs in order to achieve the desired project goal, as most geospatial tools and APIs online are pre-paid or paid after a monthly/annual requests limit is met. One of the main challenges of using mainly open-source sources is the lower quality, speed reduction and network overhead that may be caused by using those type of sources. The application will need to overcome those obstacles and befit those services to the applications usage scenarios and requirements.

Mobile development

The application will be developed for mobile devices running Android OS. The development will be done using both React Native and Android SDK.

Augmented Reality

Augmented Reality (AR) is an enhanced version of reality where live direct or indirect views of physical real-world environments are augmented with superimposed computer-generated images over a user's view of the real-world, thus enhancing one's current perception of reality.
The application will use AR as the main platform in which POI markers will be displayed and integrate with the users surrounding environment.

Geo-Location

Geolocation refers to the identification of the geographic location of a user or computing device via a variety of data collection mechanisms. Typically, most geolocation services use network routing addresses or internal GPS devices to determine this location. As such, an internet connection and GPS-enabled mobile phone will be required for the application usage.
The application will rely on the network provider's internet and the mobile device GPS to identify the user's current location. Location managers and listeners are will be used constantly to update the user's position and re-calculate its relative position to the AR environment.

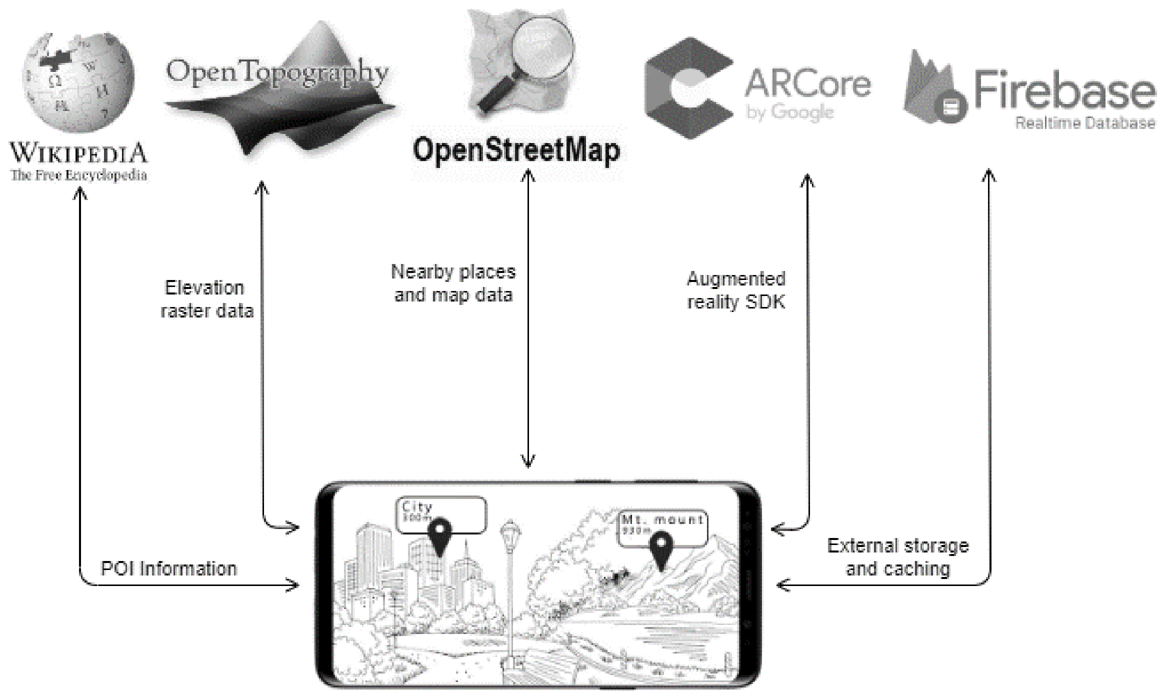
Geospatial-Analysis

Geospatial analysis includes any of the formal techniques which studies entities using their topological, geometric, or geographic properties. In order to inform the user only about POIs which are currently present in the users FoV (Field of View), The application will use a Viewshed computation algorithm to determine the raster surface locations visible to the user in the observer FoV (Field of View).

1.2 Context

The application will rely on several external services to perform its goal.
There are a lot of AR SDKs like Wikitude, Vuforia that integrate location to create Geo-location AR apps, but all of these AR engines are licensed and do not integrate well with native app development. Therefore, the applications main external service that will be used to display an augmented reality view will be Google's ARCore and Scenform Android libraries As ARCore is available for all users to view and use freely, projects developed with it can be made open-source which address one of the main problem domains mentioned before.

In order to analyze the topographic scenery around the user, the application main algorithm (Viewshed algorithm) will be used according to the topographic elevation raster data that will be provided by the OpenTopography open-source API.
In addition, the application will use Nearby Places and map data provided by OpenStreetMap SDK/API services. A more elaborate description about the POIs that are present in the users FoV will be provided by Wikipedia's open-source API.



1.3 Vision

The basic concept of the application is to help travelers, tourists and other users to gain spatial orientation. The application will identify and present information about the POIs currently in the users FoV using marker-less geo-location based AR environment and the mobile device camera. When the user points their camera at a POI, a location marker will be augmented onto their camera view showing the name and distance of the POI from the user. The user can view the location marker object from any angle and interact with it, receive additional information and insights about the location. Users will also have the option the view the map surrounding their current location, from within the augmented reality view and as a separate feature. Additionally, users will have the ability to save areal data locally on their mobile device in order to use the application to navigate in that area in offline mode.
Users will also be able to login to an external map provider and use its services through the applications interface. As part of the services logged in users will be able to add new location POIs in two main methods: via pinpointing a location on a map and by determining their current geographic location as the POI location. In addition, the application will have predefined organization administrators that will be able to add users to special groups that will enable those added users to add location via triangulation method that will be elaborated later in this document. Users will also have the ability to remove and edit places previously added by them.

1.4 Stakeholders

- Travelers and Tourists – Will be the main expected type of users, as they are usually interested in the surrounding places and scenery they encounter while traveling. Those type of users would possibly like to be able to identify and receive additional information about their viewed locations with the comfort of using their mobile device.

- Israel Nature and National Parks Protection Authority – Will like the people traveling in the nature reserves to have access to information about the scenery and important POIs in the reserve and around it without the need of signs or human directions.

- Prof. Ohad Ben-Shahar – As the founding and acting director of the Interdisciplinary Computational Vision Laboratory would like to have a proof of concept of such a projects feasibility and have a working product to demonstrate it.

  Except of those main expected users and clients the purpose of the application is to be open for free download via the Google Play Store and everyone will be able to gain spatial information by using it.

1.5 Software Context

The application operates on several external input sources, mobile device sensors input and user interaction with the mobile device.
The application workflow can be divided by the responsible entity in the process and can be generally described as follows:

- *Mobile device* - On initial application start the user location and orientation will be measured using the mobile device sensors
- *OpenTopography API* - A call to an external topographic raster resource will be made.
- Mobile device - After receiving the answer, the returned raster information will be used as input for a viewshed calculation algorithm that outputs the visible raster cells according to the area's topography.
- *OpenStreetMap API* - Simultaneously a call will be sent by nearby places module to an external map provider API which will result in the users nearby locations and their respective bounding boxes/bounding geometry.
- *Mobile device* - Succeeding to both of this API calls and after the needed calculation output is available the places data will be crossed with the viewshed visible area to determine what places are visible in the users FoV.
- *Mobile device* - Subsequently after the determination of the visible surrounding POIs and when the users chooses to launch the application AR capability (Supported by Google's Sceneform and ARCore Android libraries), place markers will be displayed above the visible POIs outputted in the previous process mentioned.
- *Wikipedia API* - By interacting with those place markers a call with the appropriate place name will be made to the Wikipedia API to retrieve the place description and additional information which will then be presented to the user.

Furthermore, users belonging to the triangulation feature permitted group will have the additional functionality of adding new places to be marked by using the triangulation method.

Chapter 2
Usage Scenarios

2.1 User Profiles – The Actors

The actors in our system are mostly end users that are not logged nor registered to the external map provider, they may have several different characteristics but the use cases for all of the users are similar. Users that are already registered and logged in to the external map provider using the platform the application provides, will be able to use additional functionality provided by the external map source. Therefore, the system actors that are logged in to the external map provider will have specific unique use-cases.

Human Actors

- Users that are not logged in to the external map provider – will be the main expected audience of the application and are predicted to consist mainly of travelers and tourists that would like to gain spatial orientation and get information about their surroundings.
- Users that are logged in to the external map provider – will be users that were previously registered to the external map provider and can use its services through the application interface in order to perform mapping operations (adding, editing or removing locations).
- Users that are associated with a group that can identify locations using triangulation – will be users that can use the application AR triangulation feature (will be elaborated later in the document) in order to identify non-marked locations without pin-pointing their exact location on a map.
- Predefined Organization Administrators – will be predefined users that will be able to add new users to the user group allowed to use the application triangulation feature.

2.2 Use-Cases

2.2.1 Viewing nearby places using AR

Primary actors:
All application users

Preconditions:
- Mobile device active internet connection
- Available mobile device GPS signal
- Available mobile device camera
- Given the application permission to use mobile device sensors and camera.
- The applications elevation usage settings are on/off.

Postconditions (success guaranty):
- A location marker is shown in AR view using the mobile device camera above every place considering:
  1. Elevation settings on - The user can physically see the location (Currently in the users FoV).
  2. Elevation settings off – All the places that are surrounding the user (in line of sight or not).
- Every location marker shown contains the correct name and additional details of the real-world location located in the location markers orientation.

Main success scenario (Basic flow):
1. The user chooses to open the AR view feature of the application.
2. The AR view of the application has initialized.
3. The user looks around with the mobile device camera.
4. Depending on the elevation settings:
   - i. Elevation settings on - All the users surrounding places that are currently in his FoV are presented to the user using location markers.
   - ii. Elevation settings off - All the users surrounding places are presented to the user using location markers.

Extensions (Alternate flows):
1. The user opens the GeoScene application.
2. The user loses GPS signal/Internet connectivity while the application is loading.
3. The application prompts the user to retry and relaunch the application.

Abstract Example:
The application is being used by the user inside a crater named "Crater" with 20 other locations surrounding the crater. Since the crater's elevation is 100m below sea level,
considering the elevation settings:
1. Elevation settings on - The user will not be able to see any location markers outside the crater.
2. Elevation settings off – Since the elevation settings of the application is turned off, the user can see 20 location markers, one for each surrounding POI around the crater.

2.2.2 Assigning a user to the triangulation group

Primary actors:
Predefined Organization Administrator

Preconditions:
- Internet connectivity
- The administrator is logged in to the GeoScene web application using global administrator credentials.
- The user the administrator wishes to assign to the triangulation group is registered to the external map provider service.

Postconditions (success guaranty):
- The assigned user will be able to use the triangulation feature that is now available for him in the application.

Main success scenario (Basic flow):
1. The organization administrator opens the GeoScene web application.
2. The organization administrator chooses to option to assign a user to the triangulation group.
3. The organization administrator inserts the user credentials (i.e. email).
4. The organization administrator approves the assignment of the user to the group.
5. The organization administrator is prompted with a message indicating that the user was assigned successfully.

Extensions (Alternate flows):
1. The organization administrator opens the GeoScene web application.
2. The organization administrator chooses to option to assign a user to the triangulation group.
3. The organization administrator inserts the user credentials (i.e. email).
4. The organization administrator is prompted with a message indicating that the user is already in the triangulation group.
5. The organization administrator is re-routed to the assignment page of the web application.

2.2.3.a Adding locations to the external map provider – using map

Primary actors:
Users that are logged in to the external map provider

Preconditions:
- Mobile device active internet connection
- Available mobile device GPS signal
- Available mobile device camera
- Given the application permission to use mobile device sensors and camera.
- The user is registered to the external map provider.
- The user is logged in to the external map provider.

Postconditions (success guaranty):
- A request to add new location with the user current real-world coordinates or map coordinates has been sent to the external map provider.
- The new location inserted contains the additional details provided by the adding user.

Main success scenario (Basic flow):
1. The user is presented with a menu option that allows him to add a new location at his current real-world position/option to mark a location on the map.
2. After choosing the Add location functionality the user is automatically navigated to the correct screen.
3. The user inserts the location name and additional information in the appropriate text boxes presented in the screen.
4. The user is prompted that the new location add request was sent successfully.

Extensions (Alternate flows):
1. The user opens the GeoScene application.
2. The user navigates to the applications login screen.
3. The user inserts his credentials in order to login.
4. The user is presented with a menu option that allows him to add a new location at his current real-world position/option to mark a location on the map.
5. After choosing the Add location functionality the user is automatically navigated to the correct screen.
6. The user inserts the location name and additional information in the appropriate text boxes presented in the screen.
7. The user is prompted that the new location add request was not sent due to network connectivity issues.

2.2.3.b Adding locations to the external map provider – using triangulation

Primary actors:
Users that are associated with a group that can identify locations using triangulation

Preconditions:
- Mobile device active internet connection
- Available mobile device GPS signal
- Available mobile device camera
- Given the application permission to use mobile device sensors and camera.
- The user is registered to the external map provider.
- The user is logged in to the external map provider.
- The user is associated with a group of users that can perform the triangulation functionality.

Postconditions (success guaranty):
- If the user approved an existing triangulation (The second user to triangulate that point) A request to add new location with the triangulated coordinates has been sent to the external map provider. If the user starts a triangulation process, then the information regarding his phone sensors bearings and location is saved to the applications caching service.

Main success scenario (Basic flow):
1. The user chooses the triangulation feature menu option.
2. An AR view is opened for the user.
3. The user identified the location he wishes to mark, if it was previously marked the user will be presented with the option to approve this location by an existing triangulation request, else, the user will be presented with the option to apply for a new triangulation request.
4. The user chooses to apply/approve according to the request status.
5. The user is prompted that the new location add request was sent successfully if he currently approved it, or the user is prompted that a new triangulation request has been applied.

Extensions (Alternate flows):
1. The user opens the GeoScene application.
2. The user navigates to the applications login screen.
3. The user inserts his credentials in order to login.
4. The user chooses the triangulation feature menu option.
5. An AR view is opened for the user.
6. The user identified the location he wishes to mark, if it was previously marked the user will be presented with the option to approve this location by an existing triangulation request, else, the user will be presented with the option to apply for a new triangulation request.
7. The user chooses to apply/approve according to the request status.
8. The user gets an error message due to connectivity issues.

2.2.4 Editing previously added locations

Primary actors:
Users that are logged in to the external map provider

Preconditions:
- Mobile device active internet connection
- Available mobile device GPS signal
- Available mobile device camera
- Given the application permission to use mobile device sensors and camera.
- The user has previously added the location he is trying to edit.

Postconditions (success guaranty):
- A previously added location update request is sent with the new information.

Main success scenario (Basic flow):
1. The user opens the GeoScene application.
2. The user navigates to the applications login screen.
3. The user inserts his credentials in order to login.
4. The user is presented with a menu option that allows him to view and edit the location previously added by him.
5. The user chooses the location he wished to edit information for from the presented locations.
6. After choosing the Edit location functionality the user is automatically navigated to the correct screen.
7. The user edits the location information in the appropriate text boxes presented in the screen.
8. The user is prompted that the location edit request was sent successfully.

Extensions (Alternate flows):
9. The user opens the GeoScene application.
10. The user navigates to the applications login screen.
11. The user inserts his credentials in order to login.
12. The user is presented with a menu option that allows him to view and edit the location previously added by him.
13. The user chooses the location he wished to edit information for from the presented locations.
14. After choosing the Edit location functionality the user is automatically navigated to the correct screen.
15. The user edits the location information in the appropriate text boxes presented in the screen.
16. The user gets an error message due to connectivity issues.

Chapter 3
Functional Requirements

3.1 Fetching and analyzing data

No.	*Description*	*Priority*	*Risk*
1	The application will use the mobile phone sensors (GPS, magnetic) to determine the user location and device orientation.	MH	Low
2	The application will support filtering surrounding locations according to the user's field of view (FoV).	MH	High
2.1	The application will be able to analyze topographic elevation data and determining the users viewshed (FoV).	MH	High
2.2	The application will enable the user to specify the visible areas circular radius to display POIs in.	NTH	Low
3.1	The application will enable users setting their ground relative height to increase their FoV depending on their non-topographic elevation.	NTH	Low
3.2	The application will use the mobile device GPS sensor altitude and the pre-loaded raster elevation data to estimate the users relative height above ground.	MH	Low
4	The application will enable the user to download and store user-predefined areal POIs and elevation data to be used offline instead of the live online fetching.	NTH	Med
5	The application will use a caching service in order to store recent data (POIs, POI information, elevation) and will re-load that data from cache when requested.	NTH	Med

3.2 Displaying information

No.	*Description*	*Priority*	*Risk*
6	The application will support displaying nearby locations data over real-world view using the mobile device camera.	MH	High
6.1	The application will display the name, distance, and additional information regarding the user surrounding POI (points of interest).	MH	High
6.2	The application should handle overlapping location markers by flattening them horizontally to avoid location markers being concealed.	MH	Med
7	The application will identify when a user is in a closed space by determining surrounding vertical surfaces and will have the option to not display POIs in that scenario.	NTH	High
8	The application should enable the user to minimize or maximize location marker information view.	NTH	Low
9	The application will display the location markers sized relative to their real-world distance from the user in order to distinguish closer location.	NTH	Med
10	The application will support displaying a map of the users surrounding area.	NTH	High
10.1	The application should support showing the map view side-by-side to the real-world view as well as independently.	NTH	Low
10.2	The application will support showing the current visible POIs on the map.	NTH	Med
11	The application will support displaying a compass showing the users current real-world orientation.	NTH	Low

3.3 Adding/Editing features

No.	***Description***	***Priority***	***Risk***
12	The application will use an external map provider logging system in order to identify users.	MH	Low
12.1	The application will provide an interface for users that are logged in to the external map provider to add new two-dimensional points of interest to be viewed by other users determined by their current location.	MH	High
12.2	The application will provide an interface for users that are logged in to the external map provider to add new two-dimensional points of interest to be viewed by other users determined by a chosen map location.	NTH	High
12.3	The application will provide an interface for users that are logged in to the external map provider to identify and tag unknown coordinate locations using the mobile device bearing, the identified locations will be cached until approved.	MH	High
12.3.1	The application will not add new two-dimensional points of interest for places that were previously identified until approved from a different location (by other user or the same user at a different time).	MH	High
12.3.2	The application will provide an interface for users that are logged in to the external map provider to approve previously identified locations from a different location and add a new two-dimensional point using the external map provider.	MH	High
12.3.3	The application will use bearing triangulation techniques in order to determine the estimated real-world coordinates of a location identified and approved.	MH	High
12.4	The application will provide an interface to users that are logged in to the external map provider to add additional information regarding their added places.	NTH	Low
12.5	The application will provide an interface to users that are logged in to the external map provider to edit or remove previously locations that were added by them.	MH	Low
13	The project will contain an external resource (i.e. web application service) where defined organization administrators will be able to assign users to the triangulation enabled group.	MH	Low

Chapter 4
Non-Functional Requirements

4.1 Implementation constraints

Performance

The following performance measurements were measured for a high usage case including total data size (elevation raster data and POIs geometry and features) of 12MB (2MB for $50km^2$ raster elevation data and 10MB for ~1500 POIs including their respective geometric features) and for average real world speeds for mobile 3G, 4G and 5G technologies measured by cellular service providers (ex. Reference: https://www.tigermobiles.com/faq/mobile-download-speed-guide/, 3G: 1 MB/s, 4G: 7.5 MB/s, approx. 5G: 18.75 MB/s):

- The application data fetching time (web-requests including elevation raster and POIs for the measurements detailed above) will not exceed:
    - 20 seconds on places with 3G reception areas (12 seconds measured).
    - 5 seconds on places with 4G reception areas (1.6 seconds measured).
    - 2 seconds on places with 5G reception areas (0.64 seconds measured).
- The application topographic analyzing process on $50km^2$ elevation data with total of $1{,}000^2$ elevation data points (SRTM GL3 90m resolution) will take no more than 10 seconds as (7.2 seconds were measured on mid-tier supported mobile device, Samsung Galaxy S8).
- The application will load all the necessary information in offline mode in not more than 3 seconds (0.8 seconds measured on mid-tier supported mobile device, Samsung Galaxy S8).
- The application will be responsive enough to refresh the AR view at-most one second from pointing the mobile device at a location marker until it appears on screen.
- The application will make and handle independent unrelated web-requests and computations concurrently.
- The application will make requests to retrieve at least $30km^2$ raster data per request in order to lower network overhead (less requests over a period of time will be needed).

Reliability and Stability

- The application will not crash except as the result of operating system error.
- The app will handle exceptions gracefully and inform about error occurrence and will allow for repair/restore if it is possible.
- The application will retry to recover from external web-requests error by resending the request and informing the user of the ongoing issue.
- The application will cache data from latest requests or operation to be recovered in the case of a failure or sudden operation abort by system or the user.

Security

- The application will not make use of user's location data outside of the applications scope.
- The application will only store encrypted user data.
- The application will enforce user's being part of the triangulation group so that users that are not part of that group will not be able to use the triangulation feature.

Environmental and Portability

- The application will use AR (Augmented reality) technology provided by Google's ARCore using Sceneform https://developers.google.com/ar/discover.
- The mobile device used will be one of the ARCore's supported device models: https://developers.google.com/ar/discover/supported-devices
- The user must have access to download the app from Google play store.
- The mobile device that uses the application must have an active camera.
- The mobile device that uses the application must have a GPS.
- The mobile device that uses the application must have and network connectivity when using in online mode.

Availability

- The application will always require the user to have an active network connectivity during application use in online mode.
- The application will always require the user to have an enabled GPS sensor on the operating mobile phone during application use.
- The application will require the user to give mobile device camera permissions.
- The application will require the user to give the application location permissions.
- The application should always be available for the users.
- The application will require users to login only in order to use features that are related to adding/editing/deleting location data.

Useability

- The application will contain an attractive and performant UI (user interface) that is non experienced user-friendly and easy to navigate, 85% satisfied users according to a UI/UX survey that will be held pre-launch.
- The application will use Wikipedia open-source service in order to retrieve places information for POIs.
- The application will enable the user to navigate to Wikipedia page for specific information about the POI's.
- The application will provide the option to perform Google search on the viewed locations.
- The application is intended to use in both portrait and landscape mobile device orientation.
- The application AR view location markers occupy at most 50% of the screen to avoid clutter and so that the real-world view will not be concealed.
- The application will support English and Hebrew languages.

4.2 Platform constraints

The native modules of the application (i.e. the AR view and threaded calculation and requests) will be developed using the Android Studio development environment due to the restriction of using the Android SDK which is best supported on that platform. The native modules will be packaged and build using the Gradle build tool to better support new versions of libraries that the application rely on (i.e. ARCore and Sceneform).

The Augmented library that will be used in the project is Googles ARCore combined with the Sceneform AR API. ARCore is Google's platform for building augmented reality experiences. Using different APIs, ARCore enables the mobile device to sense its environment, understand the world and interact with information. Sceneform makes it straightforward to render realistic 3D scenes in AR and non-AR apps. The AR libraries were chosen due to the fact that they are the leading resource to render AR in Android based environments.

The main restriction regarding the external platform that will be used in this project were the requirement of the platform to be open-source yet robust enough to support our application required features and requirements. The platform choices for topographic raster data was chosen to be OpenTopography API due to the non-limited web requests that can be made to the API, in daily web-request count and in the API's response size.

The nearby locations platform the application will use is OpenStreetMap API/SDK due to the need of the application to have the bounding box area of each retrieved location, a limitation that OpenStreetMap API/SDK solves.
OpenStreetMap will also be used as the external mapping provider and for user-identification by using OAuth (OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords) feature of the platform.

The application main source for additional information and insights will be Wikipedia by using the MediaWiki API (The MediaWiki API is a mature and stable interface that is actively supported and improved) to retrieve information regarding locations. The application will also use Google search resources to gain additional information when requested by the user.

Although it is not a constraint, the application main UI and navigation modules will be written with the React-Native framework by using TypeScript (React Native is an open-source mobile application framework used to develop applications for Android).

4.2.1 Software engineering project constraints

- Since the application will be developed for Android, the project team members personal Android mobile devices will be used for developing and testing purposes.
- During development fabricated location coordinates will be used in order to simulate the mobile device GPS placements with the goal of exploring places that may have edge cases that the application will need to be able to deal with. Real testing data will also be used by outdoor testing that will be held in nature reserves and other known scenery locations sometime pre-launch.
- During project presentation, we expect that the application will be available for download via the Google Play Store and therefore any mobile device that meets the application requirements will be able to be used as the demonstrating device.
- During project presentation we will use live location data along side with pre-prepared edge cases data that will be used for demonstration purposes (by fabricating the mobile device location).
- A short video displaying the applications capabilities and a pre-prepared application usage footage will also be presented at the final presentation.

4.3 Special restrictions and limitations

- The application will assume the correctness of data retrieved from external resources (i.e. elevation data, nearby places, etc.).
- The application will conform to the Material UI design language.
- The application will rely on internal and external caching services(i.e Firebase) will will assume the reliability of the platform for caching purposes.

Chapter 5
Risk assessment and Proof of concept plans

5.1 Risks assessments

- The open-source libraries, APIs and SDKs that will be used in the project may be too limited for the expected requests that the application will need to make to perform its intended functionality.
- When trying to determine the users FoV, precision is especially important because a too sensitive error tolerance can cause visible areas to not be marked and the opposite, place markers can be accidently placed for places that can't be seen by the user.
- Since the application does not take external sources height that are not represented in the areas topographic information (i.e. construction, trees, entities, etc.) into account, error regarding the shown place markers can be made and an elegant way to solve it needs to be formed.
- The algorithms that will be used to compute the users viewshed can be computationally expensive and can potentially be too resource-hungry for mobile device usage.
- Because the application is intended to support adding custom locations by users with permissions, there is a difficulty validating the correctness, reliability, and precision of the inserted information. Some sort of revision is required on the inserted data.
- Including an offline usage feature extends the complexity of the application and requires it to handle large amounts of data and store them locally in an efficient way so that they will be able to reload quickly and reliably. Additionally, the application will require to switch between online mode and offline mode in a seamless fashion.

5.2 Plans for the proof of concept

In the next phase of the project, in order to achieve our goals, we will take several steps that will be concluded with a limited functionality working prototype.
We can divide the needed steps to three main processes that will take place sequentially:

i. *Research*
  - Learning and understanding important geographical and topographical terms used frequently in written professional documentation.
  - Better understanding the Android development paradigm using both Android SDK and React-Native.
  - Better understanding of Augmented Reality, especially Google's ARCore and Sceneform libraries, and how we can use it the realize our project goals.
  - Researching for open-source libraries that supply our needed demands and that we will be able to use in the development phase.

ii. *Preliminary work*
  - Developing a custom viewshed algorithm (Identify the visible raster tiles from the users FoV) that will be sufficiently accurate and fast enough for the application usages. The algorithm will be tailored for the application needs and resources.
  - Testing the most suitable way to use the AR capabilities for the application needs in a closed testbed (i.e. location markers placement capabilities, compass and GPS calibration, identifying surfaces).
  - Implementing utilities that will support the application with topographical and geographical calculations and conversions (i.e. bounding box calculations, latitude and longitude conversions to xy plane, Web Mercator projections, triangulation process testing)

iii. *Proof of concept implementation*

  Implementing a working version with basic capabilities while focusing on the must have and highest risk functional requirements and the intermediate steps to achieve them:

  - Fetching the users nearby POIs and their respective bounding boxes.
  - Calibrating the mobile device sensors to be able to convert real-world placements to device-relative placements while keeping an accurate sense of orientation.
  - Fetching and analysing topographic elevation data to be used as input for the viewshed algorithm that will be used to determine the users FoV.
  - Developing a basic AR environment with location markers placement capability while taking only the users FoV into account by applying the viewshed algorithm on the users location and crossing it with the nearby places bounds.
  - Implementing a basic UI with the option to execute the AR environment capability and some basic UI enabled settings modifications that will also contribute to debugging the application at its initial phases.
  - Supporting some optional functionalities that will be present in the released version of the project (i.e. maps integration, compass integration, basic UI menu system, custom viewing distance settings etc.)

Appendices

Resources

- GeoScene github page https://github.com/itaybou/GeoScene-App

Development Tools references

- React Native official website https://reactnative.dev/
- TypeScript official website https://www.typescriptlang.org/
- Android SDK API website https://developer.android.com/reference
- Google Firebase website https://firebase.google.com/
- Google ARCore platform website https://developers.google.com/ar
- Google Sceneform API website https://developers.google.com/sceneform/develop
- Retrofit android HTTP client website https://square.github.io/retrofit/

External resources references

- OpenTopography website https://opentopography.org/
- OpenStreetMap website https://www.openstreetmap.org/
- osmdroid Android Maps SDK https://github.com/osmdroid/osmdroid
- Overpass API – OpenStreetMap query language https://overpass-api.de
- MediaWiki API https://www.mediawiki.org/wiki/API:Main_page

UI design standards

- Material UI design https://material.io/

File Formats

- Esri Grid raster GIS file format https://en.wikipedia.org/wiki/Esri_grid

Additional Materials

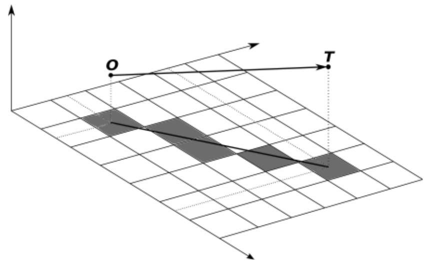
- Web Mercator projection https://en.wikipedia.org/wiki/Web_Mercator_projection
- Bresenham's line rasterization algorithm https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm
- Midpoint circle algorithm https://en.wikipedia.org/wiki/Midpoint_circle_algorithm
- Computer Vision Triangulation - https://en.wikipedia.org/wiki/Triangulation_(computer_vision)
- Bounding Box Finder http://bboxfinder.com/

Geo-Analysis Viewshed Problem Overview

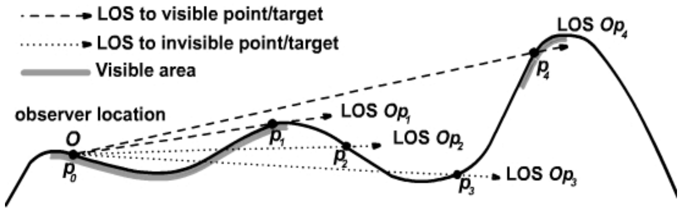
Most of GIS problems related to visibility involve the viewshed computation and in general. The applications main problems involve the area of visibility queries. The visibility queries consist in checking if a given point is visible or not from an observer (another point) on the terrain. In this section we will present the main ideas behind the viewshed computation algorithms.

Given point O where O is the observer's WGS84 datum latitude/longitude coordinates, raster grid T, line of sight radius r and $p_{i,j}$ $\forall$ $0 \leq i,j \leq |T|$ POIs we can say that $p_{i,j}$ is in O's viewshed if
$p_{i,j} \in viewshed(O,r) = \{q \in T \mid dist(O,q) \leq r \text{ } and \text{ } q \text{ } is \text{ } visible \text{ } from \text{ } O\}$.

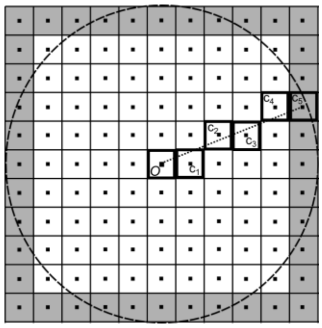
In order to determine O's entire viewshed we need to rasterize a circle with radius r on the raster T and from each point of the rasterized circle, rasterize a line to determine the visible cells from T on that line by using the slopes from O to every $p_{i,j}$ that intersects with the rasterized line.



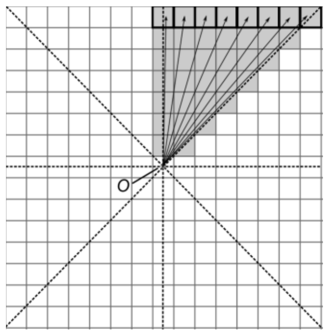
Line rasterization



Viewshed calculation along a line



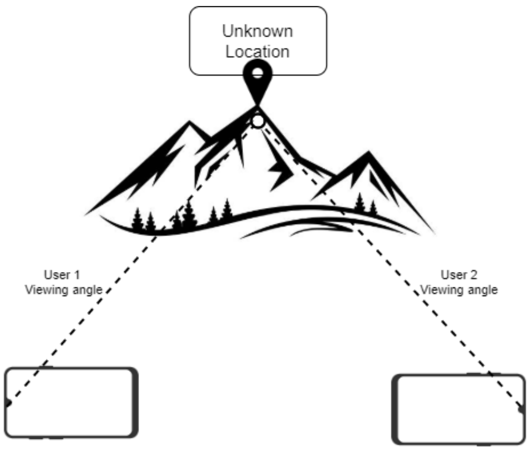
Circle rasterization



Viewshed computation along raster

Triangulation process

In computer vision triangulation refers to the process of determining a point in 3D space given its projections onto two, or more, images. In order to solve this problem, it is necessary to know the parameters of the camera projection function from 3D to 2D for the cameras involved, in the simplest case represented by the camera matrices. In the applications context, the triangulation process of pin-pointing the location of an object will be by taking bearings to it from two remote points. In order to perform triangulation of a POI we will need two users or the same user at different location in another point of time that will point their mobile devices at the currently non-marked location that they wish to tag. Then, the compass bearing of each of the mobile devices carried by the users will form a straight line to that location that we will then cross in order to determine the location approximated real-world coordinates.



Triangulation process illustration

Glossary

User - A user is a person who utilizes a computer or network service. In the context of the application, the majority of users are expected to be travellers, tourists and workers of the Israel Nature and National Parks Protection Authority.
Israel Nature and National Parks Protection Authority - The Israel Nature and Parks Authority is an Israeli government organization that manages nature reserves and national parks in Israel.
Mobile Device - A mobile device is a computer small enough to hold and operate in the hand.
Android - Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices.
Open-Source - Open source is a source code that is made freely available for possible modification and redistribution. Products include permission to use the source code, design documents, or content of the product.
OpenGL - OpenGL (Open Graphics Library) is a cross-language, cross-platform application programming interface (API) for rendering 2D and 3D vector graphics.
UI – User Interface, In the industrial design field of human-computer interaction, a user interface (UI) is the space where interactions between humans and machines occur.
UX - User experience (UX or UE) is a person's emotions and attitudes about using a particular product, system or service.
API - An application programming interface (API) is a computing interface which defines interactions between multiple software intermediaries.
SDK - A software development kit (SDK) is a collection of software development tools in one installable package.
Augmented Reality (AR)– Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information.
Markerless AR - Markerless Augmented Reality is used to denote an AR application that doesn't need prior knowledge of a user's environment to overlay 3D content into a scene and hold it to a fixed point in space.
Field Of View (FoV)– The field of view (FoV) is the extent of the observable world that is seen at any given moment.
Point of interest (POI) – A point of interest, or POI, is a specific point location that someone may find useful or interesting.
GPS - The Global Positioning System (GPS), provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. Obstacles such as mountains and buildings block the relatively weak GPS signals.
Topography - Topography is the study of the forms and features of land surfaces. The topography of an area could refer to the surface forms and features themselves, or a description (especially map depiction).
Geospatial - geospatial data and information is data and information having an implicit or explicit association with a location relative to Earth (a geographic location or geographic position).
Elevation - The elevation of a geographic location is its height above or below a fixed reference point.
Raster - raster graphics or bitmap image is a dot matrix data structure that represents a generally rectangular grid of pixels.
Rasterization - Rasterization is the task of taking an image described in a vector graphics format (shapes) and converting it into a raster image.
Viewshed - A viewshed is the geographical area that is visible from a location. It includes all surrounding points that are in line-of-sight with that location and excludes points that are beyond the horizon or obstructed by terrain.
WGS84 (World Geodetic System) - The World Geodetic System (WGS) is a standard for use in cartography, geodesy, and satellite navigation including GPS.
Web Mercator Projection - is a variant of the Mercator projection and is the de facto standard for Web mapping applications.
Latitude - In geography, latitude is a geographic coordinate that specifies the north–south position of a point on the Earth's surface.
Longitude - Longitude is a geographic coordinate that specifies the east–west position of a point on the Earth's surface.
Bounding Box - A bounding box is an imaginary rectangle that serves as a point of reference for object detection and creates a collision box for that object.
Triangulation - In trigonometry and geometry, triangulation is the process of determining the location of a point by forming triangles to it from known points.