Augmented Reality Application for Flood Visualization

A Report

by

Vijit Rathi

January 21, 2021

Abstract

The report present the detailed overview of the work done in summer internship on topic 'Augmented Reality Application for Flood Visualisation'. It starts with a brief 'Introduction to Augmented Reality' explaining what is AR, its history, applications and tools required to use this technique. Then problem statement and its need has been explained. Then App development process has been explained in chapter 2 with a little introduction to some terminologies. Chapter 3 describes some of the algorithms working in background to make AR possible. Last chapter gives a detailed work flow of the App. Its working and supported device are also explained in the last chapter. The code for the app can be found on Github profile: https://github.com/vijitrahi
The project has been made for a subject

Contents

List of Figures

Abbreviations

3D	3 Dimensional
API	Application Programming Interface
AR	Augmented Reality
BRIEF	Binary Robust Independent Elementary Features
COM	Concurrent Odometry and Mapping
CSS	Cascading Style Sheets
CV	Computer Vision
DL	Deep Learning
FAST	Features from Accelerated Segment Test
FRM	Flood Risk Management
HMD	Head Mounted Display
HUD	Head-Up Display
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
JDK	Java Development Kit
ML	Machine Learning
OS	Operating System
OpenGL	Open Graphics Library
SDK	Software Development Kit
SLAM	Simultaneous Localization And Mapping
UI	User Interface
UX	User Experience
VR	Virtal Reality
XML	Extensible Markup Language

Chapter 1

Introduction to Augmented Reality

Augmented Reality, in short AR, is adding virtual 3D objects to real world. If we go with Wikipedia, then 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, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory. It is a technology that augments a person's visual or auditory experience of their surroundings. It superimposes the digital data or information as interactive 3D virtual objects over the real world. It is a technology that fulfil these three basic criteria:

- A blend of real world and virtual objects.
- A real-time interaction with the virtual objects.
- Precise 3D mapping of real and virtual objects.

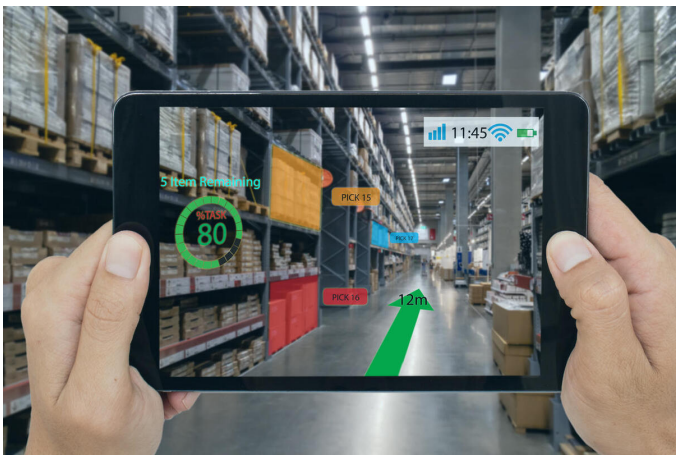

Figure 1.1: Augmented Reality

It is also called mixed reality due to the fact that it involves mixing of virtual and real worlds. It is not just addition of virtual objects to real world but also giving an immersive experience so as to make the virtual object feel like a real object. Now a days, augmented reality is being used in a lot of domains such as education, training, medical, shopping, fashion, gaming, social media, healthcare and entertainment industries.

1.1 History

The term Augmented Reality was introduced by a researcher Tom Caudell from aircraft company Boeing in 1990. The first Head Mounted Display (HMD) for AR was developed by Ivan Sutherland in 1968. He named it 'The Sword of Damocles'. It was the starting point in history for AR. The first fully immersive AR system was developed by Louis Rosenberg at US Air Force Research laboratory. In 1998, AR system was first used by NASA for navigation for their spacecraft X-38.

Figure 1.2: Pokemon game: a famous example of AR

The first AR game was launched in year 2000, by AR Quake. To play the game, people had to wear HMD, and they have to carry computer and gyroscopes in their backpack. The first AR game in smartphone was developed by Nokia in 2005, called AR Tennis. In 2009, Esquire became first company to publish AR enabled magazines, which was able to show Robert Downey Jr., Iron Man lead actor, to come out of magazine when his image is scanned. Pokemon Go, a very popular AR based game was released in 2016. This was a landmark game that introduced the AR technology to many people.

The history of AR is not very long owing to high computational speed and accurate sensors which this technology requires, which constrained it inside the research labs. But now as the technology is advancing, this technology is reaching to the hands of people.

1.2 Features of Augmented Reality

Simply adding the virtual 3D objects to real world environment is not the sole purpose of AR. The main purpose of this technology is to provide an immersive experience of virtual objects in real world, i.e. the virtual objects should behave like real objects. Obviously, they can not become real object (at least with current technology), but they should act like real objects. Following are the properties which are required to make virtual object behave like real objects:

- **Shadow:** There should be a shadow of the virtual object in real world ground, that too according to the lighting condition and pattern of real world. Even the shadow should move if the lighting direction is changing. Apart from shadow, it's texture, colour and shading should also follow the lighting condition, i.e. for a simple object, the face towards light appear brighter than the face opposite to light.

- **Fixed:** The virtual objects should be fixed at one point relative to real world. If the camera or the device is moved, virtual object should not move with it rather it should be fixed to real world and behave like a real world object is placed static.

- **Occlusion:** Like in real world, an object blocks other object present behind it, same should happen for virtual objects. And not only that but occlusion should also occur between a real object and virtual object. No need to worry of occlusion between two real objects.

- **Solid:** The virtual objects should behave solid, i.e. they should not overlap with each other or with real objects. Furthermore they should follow physics principles.

- **Context Awareness:** The AR system should know everything around it. It need to know the length of each object in real world in real time. It should know lighting pattern. It should track any movement happening.

The more the above-mentioned goals can be achieved, the more realistic a virtual object will look. Another important and attractive feature of AR is that virtual 3D objects can be scaled, rotated, moved or animated with just tapping or pinching of fingers on the screen.

1.3 Hardware and Software required for AR Systems

Tracking is an important task to be done in most of the AR related systems. Tracking is scanning, segmenting, recognizing and analyzing the environment around us and using it as an information for our purpose. There are two types of tracking possible in AR systems:

- **Inside-Out Tracking:** This type of tracking is done when the camera and sensors are all present inside the system and there is no other sensor from outside which is supporting the tracking task. These type of systems need to have all sensors inside the device and hence their size, weight as well as complexity increases and they should support higher computational complexity too. Ex: Microsoft Hololens.
- **Outside-In Tracking:** In this tracking system, the camera is outside the AR device. The tracking of device is done outside and data after most of the computation is fed to the AR device. Other than camera, infrared sensors or ultrasonic sensors placed at different positions can also be used for the same purpose. It has an advantage of low weight and size of device, but the drawback is that it requires a good network connection between device and sensors. The other drawback is that it can not be used everywhere but only at the place where camera and other sensors are fixed, or for using it anywhere else the whole setup has to be moved.

1.3.1 Hardware

Following are the hardware or sensors required in AR systems:

1. **Camera:** Camera is used to take the real time images or streaming of the surrounding environment. Apart from that it is also used to locate the device position with respect to real world and to find the coordinates of camera. The images from camera are also used for feature extraction of surrounding using some pre-coded algorithms described in later chapters. The camera is the backbone of AR systems.

2. **IMUs:** Inertial Measurement Units including Accelerometer, Gyroscope and Magnetometer are used for motion tracking of the device, which is required for mapping of virtual objects in real world correctly. Any rotation or movement of the device would result in a change in the display of virtual object on screen and hence they are needed to be recorded.

3. **GPS:** It is not a necessity in all AR systems but is needed for location based AR systems. Ex: Pokemon Go

4. **Display:** Any AR system would require a display screen to show AR. It can be the smartphone's screen, HMD, monitor or even car windshield.

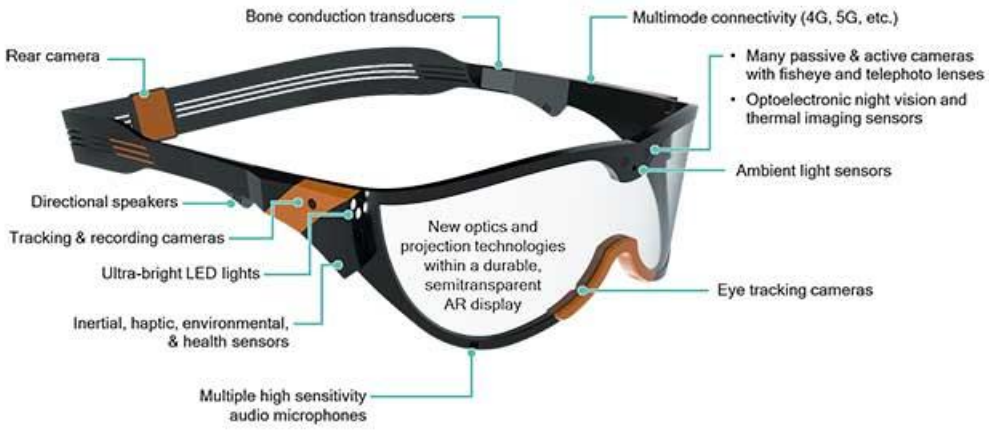

Figure 1.3: A protoype of HMD from Qualcomm showing all sensors

1.3.2 Software support needed

The AR device should have good computational power, because in background it needs to do a lot of calculations including algorithms involving ML (Machine Learning), DL (Deep Learning), CV (Computer Vision) and a lot of image processing as well as signal processing for the data received from above mentioned sensors i.e. IMUs, etc. Rest of the support required also depends on the application being used. For example if an AR app is made using ARCore library then the device should support ARCore as well as some certain API level decided by developer.

1.4 Application

Now a days AR is being used in a wide variety of fields from architecture to astronomy, education to gaming, shopping to fashion and surgery to welding. Below are few examples demonstrating how this technology is helping a lot in certain fields.

1.4.1 Education

Remember how teaching shifted from blackboard to projector and computer. Now it is slowly incorporating AR also in the domain. Schools and colleges can show models of complex machines which are not possible to show in real world either due to their high cost or size. Not only they can be shown but can be labelled, zoomed, disassembled and animated to give student a good grasp on the subject. Some example for such models are rockets, engines, reactors etc. Also it can be used in field of art or medical domain. Medical students can be shown detailed models of internal body parts example, digestive system, respiratory system etc or even interactive visualizations of brain and nervous system.

Figure 1.4: AR in architecture

1.4.2 Architecture

The actual sized model of the building can be seen at the real site of the building even before a brick is placed through AR. In 2004, this technology was demonstrated by Trimble Navigation. It is not only helpful for architects but also to customers. The 2D plans of buildings can be converted to 3D designs and different designs and colors can be tested and chosen. It can also be used to visualise and plan for the deployment of rigid structures, pipes and cables. With increasing accuracy of GPS, Ar being used to visualise geo-spatial models of different terrains and areas.

1.4.3 Transportation

Recently in CES 2021, Panasonic Automative demonstrated AR Head-Up Displays (HUDs) which can project 3D and AI driven information, example speedometer, route guidance, signboards, pedestrian detection, signals, lanes etc, directly on the windshield of car. So, the information is now directly in driver's line of sight. It reduces driver's distraction and increases the safety. This has become possible due to the advancement in optics, but not much in use due to high cost.

Figure 1.5: AR in shopping furniture

1.4.4 Archaeology

Historically important sites, buildings and landscapes can be constructed back as 3d virtual models and can be augmented using AR. AR is being used in archaeological research. Also, some of the museums and historical monuments are using AR to show augmented models of old ruins, things and also to show some events as animation, and attracting tourists.

1.4.5 Shopping and Fashion

Before buying anything, one can see how the furniture will look in house, not even by its look but also by dimensions, through AR. One can see how shoes, clothes etc will look in real space. Lenskart, which is a spectacles selling company, gives an option of trying on glasses before buying. The website augments the glasses on person's face and person can see if he/she likes it and then can decide which one to buy.

1.5 Challenges in Augmented Reality

Following challanges which persist with AR technology are as follows:

- **Size:** Since a lot of sensors: camera, accelerometer, gyroscope, GPS all have to be incorporated in a single device. Hence, it is generally having large size. Small size comes with an increase in cost.
- **Power:** Firstly, the technology requires a high performance processor to perform a lot of calculation running in background. Second, since lot of sensors are working at same time, the power requirement of device is high and to incorporate that in a small size is a big challange.
- **Heat:** Due to large number of calculations in background, it produces a lot of heat and due to small size of device, heat syncing becomes a big difficulty to resolve.
- **Dim Light:** Some of the running background algorithms for signal processing do not work in dim light conditions due to unable to process the feature extraction and hence subsequent tasks.
- **Complex 3D models:** To achieve realism in 3D models, their number of tris counts or vertices has to be increased. But it is harder to process more complex 3D models and to augment them in real world due to a direct increase in number of calculations. Hence, one has to trade-off between realism and effective rendering.

1.6 Summary

AR is an expanding technology. It is increasing its reach to hands of people rapidly. It is being used commercially and in fields such as Education, architecture, medical etc. It uses sensors such as camera, accelerometer and gyroscope as well as a lot of calculations in background which require good computational power processors. This also provides a lot of challenges such as size, power and heat. But with the expansion of technology, it is overcoming the challenges one by one rapidly.

Chapter 2

App Development

In this chapter, the app development process has been described. The parts of the app has been described as little snippets of the code. Along with that certain terminologies and libraries related to application development are also defined.

2.1 Problem Statement: Flood Visualization

The main motive of the internship is to develop an AR application which can be used to visualise flood like situations. The area of environmental design and planning using AR is being done but at very few places compared to the growth of AR. With advancement in technology, now AR has come out of research labs and is available in mobile devices such as smartphones or laptops. The idea was to develop a real- time complex, immersive and interactive AR application for flood visualisation and reduce the implementation complexity using technology. Along with flood visualisation, some text, which can tell about flood history, evacuation routes as well as live sensor readings of river height, dam height or weather prediction, can also be shown as AR augmented in the real world.

Now, most of the part of Flood Risk Management is done by hands, offline using flood graphs, geo-spatial maps, inundation mapping etc. AR can provide not only FRM experts, policy makers or volunteers but also common citizens, an opportunity to take part in FRM process using interactive AR visualisation tools. Every year floods cause a lot of damage to life, property, crops, livestock and health. Damage to roads, bridges and power plants create a lot of problems. Most of the people affected are the ones living near rivers, canals or lakes. And a lot of money is spent in overcoming the damage created by floods. If using AR, FRM process can reach to common people, then more ideas to deal with flood will come and it is evident that the best solution to a problem is provided by people facing it.

Along with flood visualisations, since we are incorporating other information also like

Figure 2.1: Flood: Devastating Natural Disaster

live sensor readings such as river height, weather predictions, one can also work upon predicting flood for near future using the data available. Also, information like flood history, evacuation routes and other minute details of a place can be used in case of real flood events. And since the application is being made for an android enabled smartphone, it will be accessible to a lot of people.

2.2 Terminology

Below are some of the terms described which will be repeatedly used in subsequent sections:

- **API:** API stands for Application Programming Interface. It is an interface that defines communication between two applications. It defines what calls and requests has to be made as well as which data formats or extensions and protocols have to be used. Some APIs run in background while some have to be included by the developer if there is a need of communication between two different applications. One of the API used here is ARCore developed by google. It communicates between the sensors, camera and the app and do all the computations for coordinate transformations required to augment 3D model on real world, example feature extraction, plane detection and homography estimation.

- **SDK:** SDK stands for Software Development Kit. It includes software development tools in one package. Some SDKs are necessarily required for a specific platform

based application (eg. OS). Example: Development of an android application requires a SDK called Java Development Kit (JDK). It helps application creation by providing compiler, debugger or full software framework. One of the SDK used in my app is Sceneform SDK. It is a 3D framework with a physically based renderer that is optimized for mobile devices. It makes easy for developer to build AR apps without requiring to code for OpenGL, which is an API used to render 2D and 3D vector graphics. It was developed by Google in 2008 fully in Java language.

- **MaterialDrawer:** It is a library developed by Mike Penz for android application development. It is a very flexible and easy to use library to create a menu drawer in the application. It's been used in the app for creating a left side menu which is having options to enter any information as text and deploy it as augmented text box in real world.

- **jsoup:** jsoup is an open-source Java library designed to parse, extract, and manipulate data stored in HTML documents. It is used to scrape some file or information from a website. The information is extracted by first requesting the page from server using internet and then navigating the web page using CSS selectors such as classes. In my app, it is used to extract live river height sensor reading from a website: `http://www.mysuwanneeriver.org/realtime/river-levels.php`. The website provides up-to-date river height readings for some stations in US.

- **Android Studio:** It is an official IDE for specifically developing application for Google's Android OS. It was released in 2013. Most of the application is coded in Java or Kotlin, along with the user interface part or front-end in XML. The app developed using Android Studio can be published on Google Play Store.

2.3 Explanation of Code

Since, the code is of almost 650 lines. Full code can not be included in the report. So, some important parts of the code have been explained in next few pages:

2.3.1 UI

UI or user Interface of an app is the part of the app which is seen on screen by the user. Also it include all the placement of texts, buttons, input segments, images which are to be shown to the user. It is mostly saved as file name $activity\_ux.xml$, where UX stands

for User Experience. Since this app is made for AR, what is shown to the user is mostly surrounding environment feed from camera augmented by some 3D virtual object. Hence, only an AR fragment is included in the code:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".HelloSceneformActivity">

    <fragment
        android:id="@+id/ux_fragment"
        android:name="com.google.ar.sceneform.ux.ArFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <SeekBar
        android:id="@+id/seekBar"
        android:layout_width="match_parent"
        android:layout_height="36dp"
        android:thumb="@drawable/thumb" />

</FrameLayout>
```

The fragment part above is used to add an AR fragment in the UI. And a seekBar also has been added on the top which will serve as a height changer for 3D assets. The menu drawer which appears when swiped from left on screen is not included in UX file, but in main Java file.

2.3.2 Variable and Variable types

Some of the variables used in the application are very common like 'float' integer etc. But sum of other variables are specifically used for the purpose of AR. Here . I am describing variables one by one and what is their use:

```
public Anchor anchor;
```

The first variable is anchor. Anchor is a point in real space to which a virtual 3D object is fixed. In my application, a total of six anchors are used one For The flood plain, five other for information texts.

```
public AnchorNode anchorNode;
```

The second variable is AnchorNode. An anchor node can act as a parent node to other 3D virtual objects which can be parented to the corresponding anchor. For each Anchor an anchor node is defined. Hence there total six anchor nodes used.

```
public TransformableNode transformableNode;
```

Third variable is TransformableNode. Transformable node is used for an anchor how to make it do following transformations like rotation, translation or scaling.

```
private ModelRenderable waterSurfaceRenderable;
```

One variable is ModelRenderable. It is used for the 3D object model which is being used in the app to render. For example in our app we are rendering a floodplain. Hence the model of the flood plain will be saved in the variable model renderable.

Other simple variables are string for saving information, 'float' for saving height of the 3D asset from the plane, water level height etc.

2.3.3 Rendering Model

The following code is used to render the 3D model in the real world. First then anchor is created by using Hit Testing. When we tap on screen, the app will perform a hit testing It will draw a line outward until it hits the surface. And by intersection of the plane and ray, we will get a point which is called Anchor. Then, an anchor node is created and it is parented to the AR scene view which is the feed from camera or real world environment, which act as a parent to the node. Then transformable node is declared to make the node able do operations like scaling, rotation and translation. Then water surface model, which is flood plane, is rendered. In the end, the shadow has been turned off for the flood plane according to the feedback observed from testing the app. The same kind of procedure is used for rendering text information in form of 2D text box augmented in real world feed.

```
anchor = hitResult.createAnchor();
anchorNode = new AnchorNode(anchor);
anchorNode.setParent(arFragment.getArSceneView().getScene());
transformableNode = new TransformableNode(arFragment.
    getTransformationSystem());
transformableNode.setParent(anchorNode);
transformableNode.setRenderable(waterSurfaceRenderable);
waterSurfaceRenderable.setShadowCaster(false);
```

2.3.4 Web Scraping

The following code has been used to scrape the height of level in the river from a website. First the HTML document is requested using the library jsoup. Then after getting the HTML document, we navigate to the point where the water level is stored. Then we copy the water level in a string. As well as the time when the water level was last updated is stored in a string named time.

```
public String doInBackground(Void... params) {
            String title2 ="";
            Document doc;
            try {

                doc = Jsoup.connect("http://www.mysuwanneeriver.org/
   realtime/river-levels.php").get();
                title2 = doc.title();
                Elements masthead = doc.select("td.val");
                Elements masthead2 = doc.select("td.valEM");
                time = masthead.get(16).text();
                waterLevel = masthead2.get(2).text();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return title2;
        }
```

These were some of the important code snippets used in the application. At many places, try and error, exceptions are also included, in case any error occurs, so as to stop app getting crashed.

2.4 Summary

This chapter included the problem statement of the internship which is to create an app for flood visualisation and its importance as to become a complimentary tool for FRM process. Then some of the common terminologies related to app development are described. Then some important parts of the whole code are explained as to include whole 650 lines of code does not make sense.

Chapter 3

Algorithms used in AR

Some of the most prominent Algoritms used in background to do feature extraction, description and matching, to estimate homography, to locate the position of camera, etc are explained in this chapter.

3.1 FAST

FAST is an algorithm developed by Edward Rosten and Tom Drummond in 2006. FAST stands for Features from Accelerated Segment Test. It is used for feature detection in an image. It is also used in SLAM described below. It is a corner detection algorithm. The feature points extracted are later used to track and map objects in real world while using AR. It is a very computationally effective algorithm. Hence it can be used in real time. The algorithm is mentioned below:

The basic idea is to consider a pixel and check the brightness of pixel surrounding it. Based on a threshold to determine if it is a corner or not and hence it can be used as a feature point or not.

1. A pixel P is selected from the image . The intensity of the pixel is I.

2. We need to know if this pixel can be used as a feature point or not. Threshold value t is set for comparing the pixel values it can be 20% of the I.

3. Now we have to consider a circle of 16 pixel surrounding p.

4. First we compare pixel 1, 5, 9 and 13 of circle with I. If three of these pixel hello value less than I-t or greater than I+t, then we proceed to next point and p is called an interesting point, else we stop the algorithm and move to next pixel.

5. So if P is an interesting point can we check for valie all the 16 pixels around P twelve of these pixels are below I-t or above I+t, then P will act as a feature point.

6. We have to repeat this procedure for all the pixels in the image.

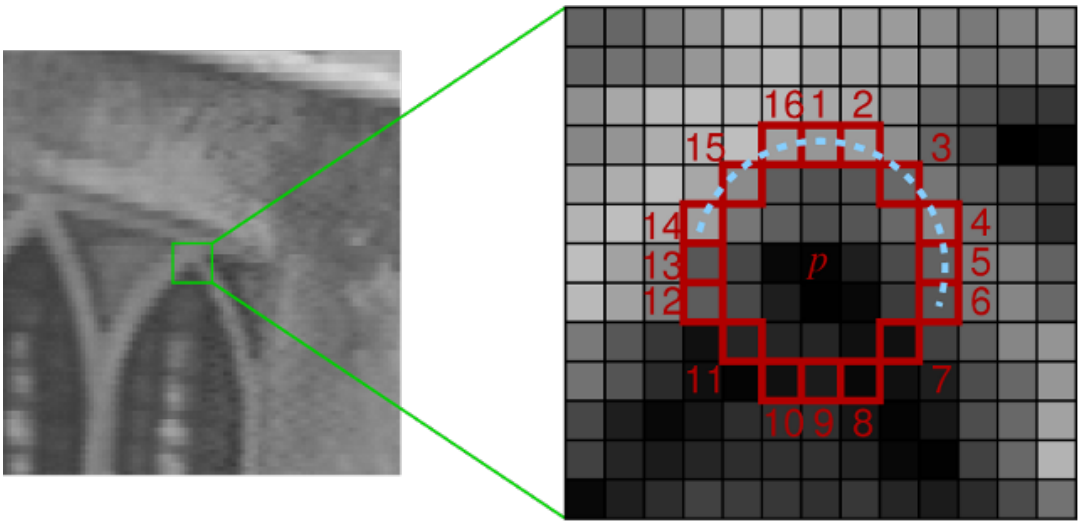

Figure 3.1: FAST Algorithm: a circle of 16 pixels around centre pixel P

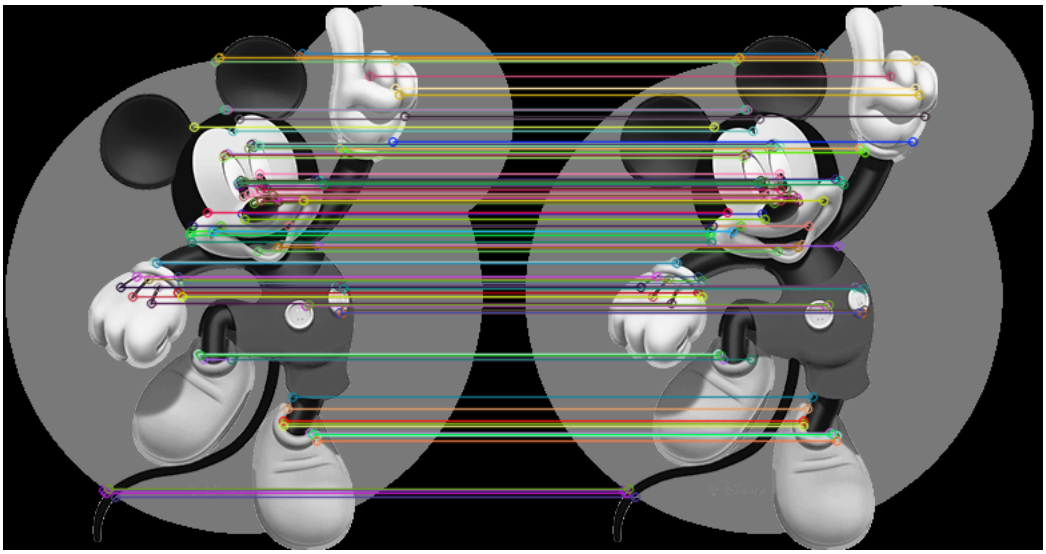

Figure 3.2: FAST Algorithm: Feature points detection and matching

3.2 BRIEF

BRIEF stands for Binary Robust Independent Elementary Features. After applying FAST algorithm, BRIEF is used as a feature descriptor. Brief convert image patches into a binary feature vector. Binary features vector also know as binary feature descriptor is a feature vector that only contains 1 and 0. In brief, each feature point is described by a feature vector which is 128–512 bits string containing 0s and 1s. Brief is very sensitive to noise and hence it uses gaussian kernals for smoothening of the image. Below is algorithm used for creating the feature vector:

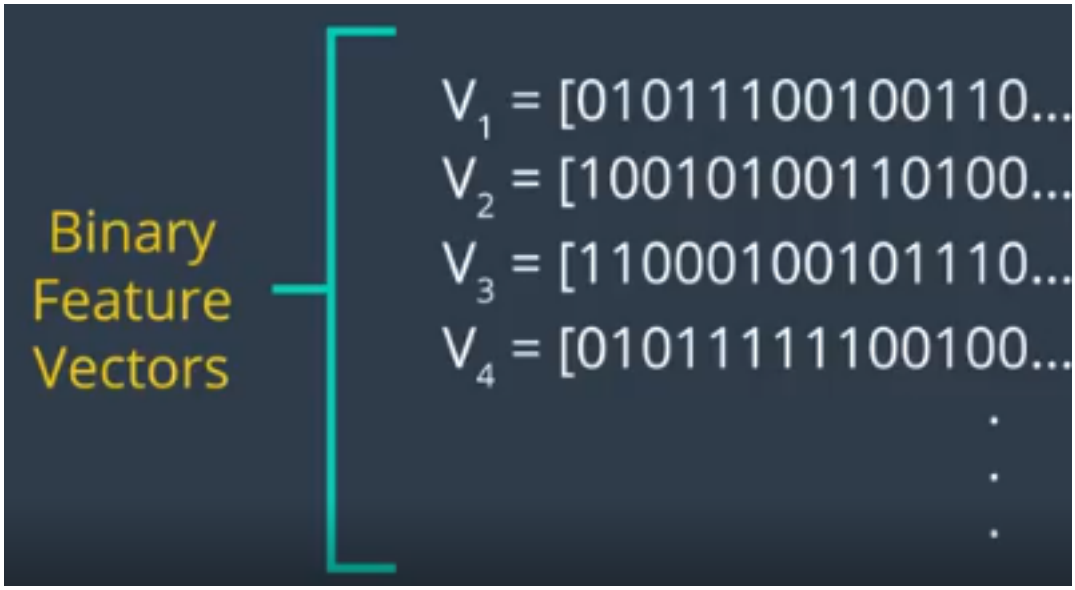

Figure 3.3: BRIEF Algorithm: Feature Vectors

1. For a feature point pixel P, two neighbourhood pixel x and y are considered, and their intensity p(x) and p(y) are compared.
2. A value 1 is stored in τ(p,x,y) if p(x) < p(y) and 0 else.
3. The above procedure is repeated for n number of (x,y) pairs where n represent number of bits in feature vector
4. n (x,y) pairs are uniquely defined set from a number of distributions, of which one is selected: Uniform, Gaussian or coarse polar grid.

In end, Brief descriptor is:

$$f(p) = \sum_{i=1}^{n} 2^{i-1} \tau(p, x_i, y_i)$$

3.3 SLAM

SLAM stands for simultaneous localisation and mapping. The main aim of the algorithm is to fulfill this two condition: First it has to build map of the environment and second is to locate the device within that environment. In case of beacons present in the environment at different positions, location can be found easily by triangulation. But this is not the case always if the device is smartphone then what we have is camera frames and sensor readings. And every sensor, be it camera or accelerometer, have errors which will accumulate over time known as drift. So to provided drift free results is also a big challenge. The Slam algorithm is divided into four parts:

- First is data getting from camera, accelerometer, and gyroscope. There can be additional sensors also like GPS or depth sensors.
- Second is the feature points which are given by the previous algorithms FAST and BRIEF. These are also called map points.
- Third is back end which tries to find correlation between different frames from camera and handle overall geometrical reconstruction.
- The last one is the result containing features, their locations and camera position in the world.

Initial camera position is determined by the key-points between the subsequent camera frames. The camera pose is refined by using additional sensor data. In starting matching is done only between two frames but later it is done in many frames. One important part of SLAM algorithm is loop detection and loop closing. It checks if feature point from current frame matches with some feature points detected previously. By setting a threshold, if feature points matching is above certain level then user has returned to a previous place. This is called loop closing. Now since we are at a previous position where we already have been we can know what are the errors that have propagated along the whole path. Hence by removing those error the map is updated. Google ARCore uses another method called COM (Concurrent Odometry and Mapping), which is similar to SLAM, for mapping.

3.4 Summary

Three of the important algorithms are discussed used in AR: FAST, BRIEF and SLAM. FAST is used to find feature points. BRIEF converts feature points to 128-512 bits array. And SLAM is used to map the real environment around AR device.

Chapter 4

Result: Overview of the App

This chapter deals with a detailed description of the app made, how to use the app. Also the work flow of the app has been presented.

4.1 App

4.1.1 Start

Figure 4.1: Start screen

On start, application shows a hand rotating to see the environment around, mark the

features points (which are mainly high contrast points, like corners) and to detect the planes. So, the user has to move the smartphone a bit around to get started. The plane is shown as dots:

4.1.2 Flood Plane Positioning

Once the planes are detected, user can tap at any point on the plane. At that point an anchor will be created and a flood plane will appear. This flood plane can be scaled by usual pinch in or out on the screen. Also, it can be move around by holding tap and moving on the screen.

4.1.3 Side Panel

On sliding from the left of the screen, a window with many options appears as shown below:

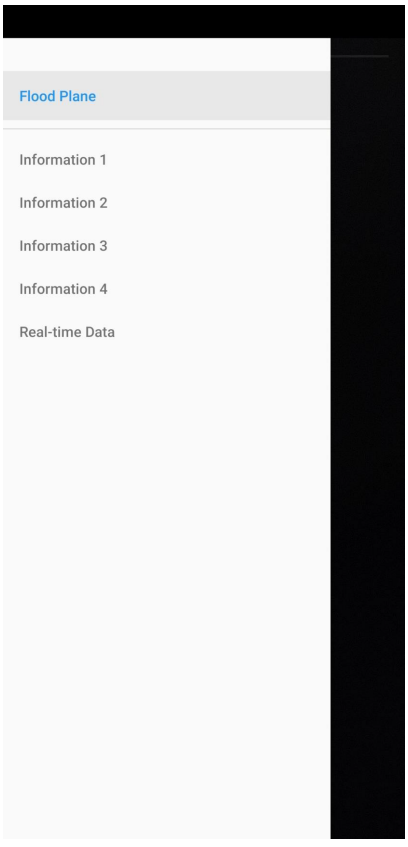

Figure 4.2: Side Panel

1. **Flood Plane:** On selecting 'Flood Plane' in the options, the flood plane will get selected. So, any operation done next like scaling, changing height will happen for the flood plane.

2. **Information:** On selecting one of the boxes out of Information 1 to 4, a pop-up window will appear as shown below:

   Here any information can be written, for example "The highest water level here was on date DD/MM/YYYY and the height was H metres.", "The warning level is X metres and the danger level is Y metres" or "Evacuation route is . . . ", etc. On pressing 'OKAY', the text will get stored in the corresponding information. Next when we will tap anywhere on the plane in the scene, a 2D textbox will appear at that position having text what was entered. It looks like:

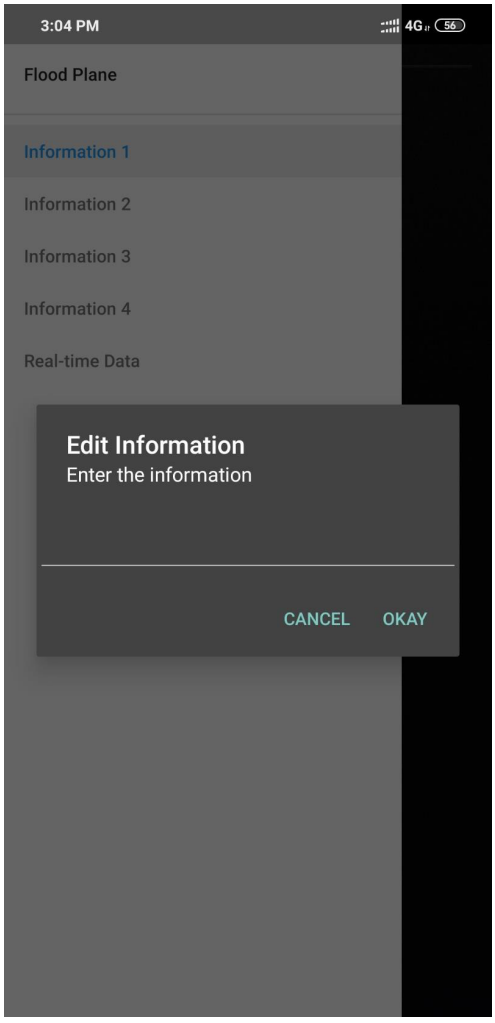

Figure 4.3: Information Menu

This information can be edited again by doing same process, i.e. slide window from left Click on the 'Information' option, write the text and click 'OKAY'.

Figure 4.4: Augmented Information

3. **Real-time Data:** On choosing this option from side window, some real-time data of a river can be displayed. Here, after selecting this option and tapping on the plane, a 2D textbox will appear which will contain the text "The water level at place is X feet at HH:MM am EST.". It is taking data from a website using library Jsoup: `http://www.mysuwanneeriver.org/realtime/river-levels.php`

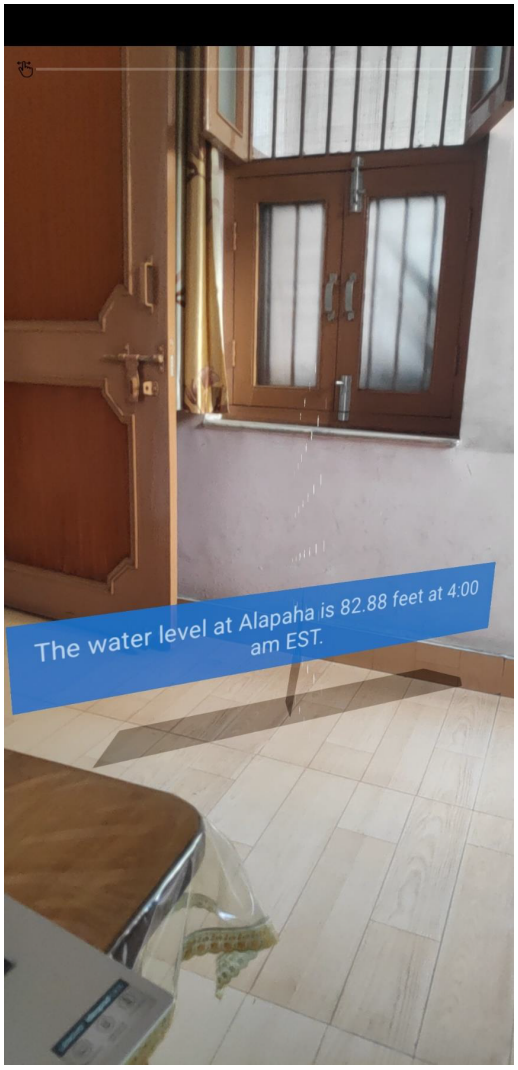

Figure 4.5: Real Time Sensor Data

4.1.4 SeekBar

The seekBar on the top can be used to change the height of any rendered object in the scene by moving the slider. The seekBar looks like this: Here, the seekBar is at minimum and

Figure 4.6: Seek Bar

maximum value. The heights can be tuned:

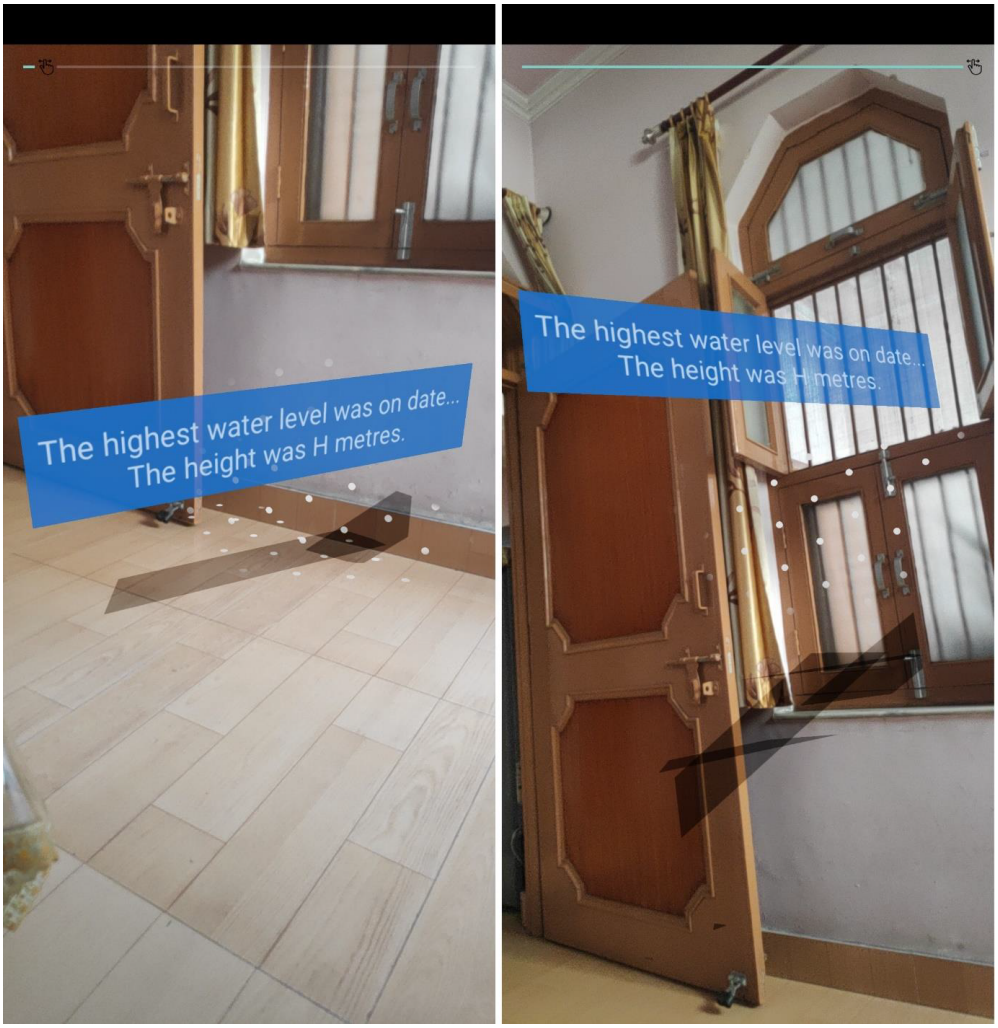

Figure 4.7: Different height of Augmented Text

4.2 Supported Devices

Minimum Android Oreo 8.1 (API level 27) required.
Also, app works only on devices which support ARCore listed here: `https://developers.google.com/ar/discover/supported-devices`

4.3 Summary

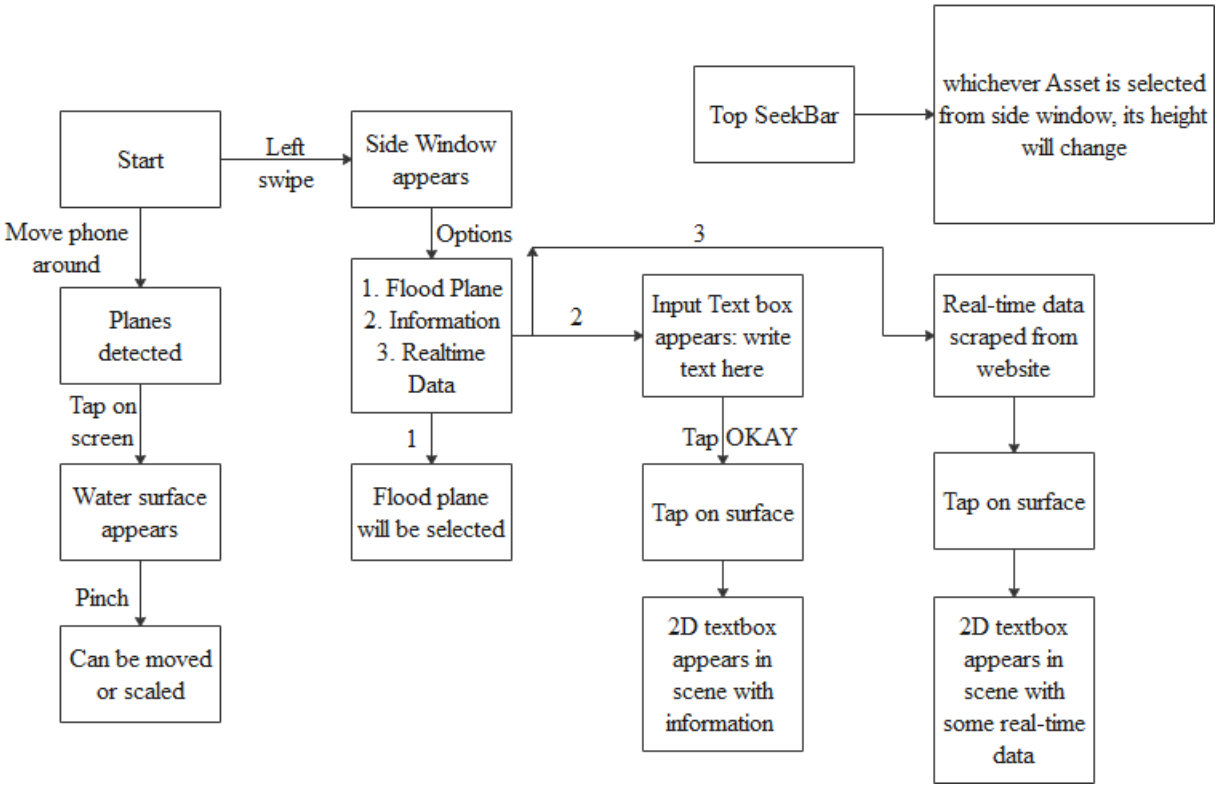

Figure 4.8: App Workflow

Bibliography

- Haynes, Paul Hehl-Lange, Sigrid Lange, Eckart. (2018). Mobile Augmented Reality for Flood Visualisation. Environmental Modelling and Software.
- Haynes, Paul Lange, Eckart. (2016). In-situ flood visualisation using mobile AR. 243-244. 10.1109/3DUI.2016.7460061.
- Bandrova, Temenoujka Kouteva-Guentcheva, Mihaela Pashova, Lyubka Savova, Denitsa Marinova, Silvia. (2015). CONCEPTUAL FRAMEWORK FOR EDUCATIONAL DISASTER CENTRE "SAVE THE CHILDREN LIFE". ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XL-3/W3. 225-234. 10.5194/isprsarchives-XL-3-W3-225-2015.
- Puertas, J.; Hernández-Ibáñez, L.; Cea, L.; Regueiro-Picallo, M.; Barneche-Naya, V.; Varela-García, F.-A. An Augmented Reality Facility to Run Hybrid Physical-Numerical Flood Models. Water 2020, 12, 3290.
- Daniel Winkler, Jonatan Zischg, Wolfgang Rauch; Virtual reality in urban water management: communicating urban flooding with particle-based CFD simulations. Water Sci Technol 29 January 2018; 77 (2): 518–524.
- https://www.andreasjakl.com/basics-of-ar-anchors-keypoints-feature-detection/
- https://www.andreasjakl.com/basics-of-ar-slam-simultaneous-localization-and-mapping/
- https://medium.com/data-breach/introduction-to-brief-binary-robust-independent-elementary-features-436f4a31a0e6
- https://medium.com/data-breach/introduction-to-fast-features-from-accelerated-segment-test-4ed33dde6d65
- http://developers.google.com/
- https://developer.android.com/