VISUALIZATIONS OF HISTORIC AND COASTAL FLOODING IMPACTS AND SOLUTIONS

by

Mark Halle, B.S.

THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE

in Software Engineering

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

MAY, 2020

VISUALIZATIONS OF HISTORIC AND COASTAL FLOODING IMPACTS AND SOLUTIONS

by

Mark Halle

APPROVED BY

____

Soma Datta, Ph.D., Chair

____

Marc Mokrech,, Ph.D., Committee Member

____

Anne Henry, MFA, Committee Member

RECEIVED/APPROVED BY THE COLLEGE OF SCIENCE AND ENGINEERING

____

David Garrison, Ph.D., Associate Dean

____

Miguel A Gonzalez, Ph.D., Dean

ABSTRACT

VISULIZATIONS OF HISTORIC AND COASTAL FLOODING IMPACTS AND SOLUTIONS

Mark Halle
University of Houston-Clear Lake, 2020

Thesis Chair: Soma Datta, Ph.D.

Flooding is one of the most deadly and costly natural disasters known to humans. While much is done to study the impacts of flooding, not enough is done to inform the public about the dangers of flooding. In a society where information is becoming more readily accessible, there needs to be more ways of informing individuals of the everyday dangers they face. Simple maps and charts, while accurate, do not display enough information at the level needed for people to act. The biggest factor in influencing individuals' behavior on risk is fear. In recent years, new systems have been developed in specific locations to better inform the community members about the dangers of flooding. Using mobile augmented reality and 3D mapping, a new application was constructed that can be readily used by anyone anywhere. The application was developed using Unity 3D, Vuforia, and Mapbox technologies. The application used historic as well as predictive flooding data to give an individual a greater perspective on their local flooding situations. With the ability to switch between augmented reality and standard mapping an individual can understand the severity of past and future flooding scenarios. The application gave greater insight into the possibility of different flood modeling and visualization systems. It is hoped that with new technologies developed more individuals will be educated and saved from rising flood waters.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I:

GROWING FLOOD CONCERNS

Flooding is one of the worst natural disasters known to humankind, with trillions of dollars wasted each year. A study done on the global population in 2010 showed that long term global population totals are exposed to flooding of 992 million people with a value of 46 trillion USD [1]. This is expected to increase from 2010 to 2050 as global population grows and new costal population centers emerge [1]. The most at risk are individuals who are in less developed countries such as Asia and Africa [2]. While there are many different factors at play in future coastal flooding scenarios, some may be able to be controlled while others are just a natural expansion of the global population.

The most significant components affecting global coastal flooding hazards are tides, storm surge, wave run-up, vertical land movement and sea level rise [3]. Some events are rapid and spontaneous while other events are just a natural progression of the environment. Global sea level rise has been a news topic in recent years, with different scenarios showing various estimates of sea level increase. Flood elevation is predicted to rise on average by 0.3 meters by 2030 which would result in widespread flood inundation and more economic damages [4]. Even if sea level did not rise, there would still be a rise in flooding and economic risk. In a study done on the coastal flooding and wetland loss in the 21st century, different scenarios were evaluated, and impacts were compared for the next century. In 2004 it was noted that in a situation of even no climate-induced-sea-level rise that all scenarios would still experience an increase in the frequency of flooding through 2020 [5]. Even without sea level rise, coastal property is slowly becoming ocean property as coastal fronts recede.

As such new methods are needed to show the individuals most at risk of danger to their property as well as themselves face. In the last ten years new flood modelling and visualization techniques have created new powerful mechanisms for communicating about flood risk [6]. It has become easier, with the advent of computing technology, to predict and visualize how violently an area will react to flooding. While these tools that use geographic information systems (GIS) modeling are more used by professionals, there has been a move towards more usable information systems. Most notably several different web applications have been developed in the last few years that allow for communities to see flood hazard scenarios. These flood hazard visualizations have brought together a variety of stakeholders and experts for the betterment of a community [7].

CHAPTER II:
RESEARCH QUESTIONS

The purpose of this proposal is to explain the need for flood modeling systems in real time. Furthermore, it will explain the need for simplistic, lightweight systems that can be widely used by the public. It is hoped that with more widely available applications an individual will be able to determine their perceived risks due to flooding. Overall, this study and solution propose to answer the following questions:

Is there a viable market for simulations of historic flooding events and what visualization systems could be best applied?

- How can individuals perceive risks of flooding dangers?
- Why is there a need for easy to use flood visualization applications?
- When can one make trade off in modeling complexity for easy of use and understanding?
- Is it possible for realistic flooding to be simulated in augmented reality space?

It is suggested that new technologies be used to demonstrate the flooding risks individual face. Augmented and virtual reality are two important visualization methods that can be utilized to further visualize flooding risks in an immediate area. These technologies need to be explored to further see if they are successful in demonstrating flooding hazards to local individual. It is hoped with the application developed that further research and applications will be developed to greater inform the public of flooding hazards.

CHAPTER III:

TYPES OF RISK

Defining risk is important to understand why people live and work where they do. There are four variables that shape rationality in the choices made concerning natural hazards [8].

- Existence of a limited range of alternatives
- Misperceptions of risks
- Individuals and their crisis orientation
- Tension between individual versus collective management

While most of the variables involving risk are outside of one's control, misperceptions of risks can easily be addressed. There is a role of risk perceptions in improving the resilience of individuals and communities to natural hazards [8]. Furthermore, when faced with images of hazards and events, mental representations people have can carry an affect tag [8]. While this affect may not always have a positive effect, it will generally lead to more personal risk awareness. It was found that flooding animations and visualizations did contribute to an expressed desire to become more politically and socially active in response to flooding [9].

It is important to realize that not all individuals are affected by the same risks. A study done in the area of Águeda, Portugal showed that while major flooding happened, not all areas of Águeda were affected equally. There were variations of intensity and frequency of the damage that is generated [10]. As such each individual and community would have a different perception of the amount of flood risk in their area. Additionally, it is important to look at where people perceive the risk is coming from. Overall the study found that social perception of the flood risk is that the flooding is caused due to global climate change and global warming [10]. The source of the flood risks is important because it will affect how people react. It was found that most respondents even when affected by floods reoccupied the buildings and restarted old activities [10]. The source of the flooding risks could also affect who assumes the responsibility. In Águeda the community thought that the local administrative and policy entities should assume greatest responsibility for flood risk mitigation and management, while, significantly, only 6% considered that the population should be principally responsible in this matter [10]. This shows the need for flood risk visualization systems to inform the public about the hazards. A high level of disaster risk-reduction education, social cohesion and solidarity coupled with trust between government authorities, community leaders and vulnerable communities can help to increase communities' resilience to coastal hazards and tropical cyclones [11]. It is important that a person is fully informed because, person who feels he is insufficiently informed about a hazard, believes it is not his responsibility to inform himself about it [12].

If people are not active in taking the responsibility to inform themselves, then how does one communicate risk? As such, risk communication should focus on raising risk perception using a fear appeal together with messages to increase response [12]. Using a primal emotion of fear can instill new perceptions of reality on a person that they may not have had before. Therefore, there should be better methods of communicating risk than what already exist for the common public to use.

Figure 1: Flood viewer of University of Houston – Clear Lake (UHCL) campus. Common flooding areas are made apparent and an individual can pan and zoom from various areas. From just a glance it can be hard to make out the specific areas where it might flood ("FEMA Flood Map" 2017).

Looking at currently employed flood risk communication methods on the Federal Emergency Management Agency's (FEMA) website one can quickly get overwhelmed with the amount of information present. While the map does accurately communicate the risks and dangers of flooding, it lacks the "shock" factor needed to raise awareness. A study done on flood hazard visualization of potential home purchases showed that participants are less likely to select lower hazard houses when flood hazard information is only presented in map format [13]. This shows that the information, while there, is not quick and easy to understand by most individuals. New tools are needed to effectively communicate flood risk other than two dimensional numbers on paper. It has been shown that communities are mainly interested in scenarios that they can relate to from observational experience [14]. In the last few years technology has greatly improved to the point where three-dimensional rendering is now more possible. New tools such as the FloodView have been developed for the specific purpose of visualizing the social, economic, cultural, and environmental risk and impacts [15]. As time progresses there needs to be similar tool to allow for more individuals to identify their own risks and allow them to better prepare for an extreme scenario.

CHAPTER IV:
FLOOD MODELING

As technology advances more and more tools of representing flooding risks have become available. These tools are flooding simulations tools that only domain experts can really understand and use. Most of these tools involve simulations of processes that are complex and rely on heavy computational and data-intensive models [16]. While advanced tools will always be needed, there are significant advantages of a simpler approach. Simple tools allow for significantly less CPU time, fewer parameters, and simpler model structures [17]. A simply designed system could be more easily deployed and used by a variety of people. Tools need to be in the hands of the general population so that they can understand the individual risks they take, and how to best act upon them. Recently there has been a great drive to identify and develop tools that are accessible for practitioners and domain experts of flood management [16]. The level of accessibility of these tools varies from online web interaction to mobile phone applications. Each brings the idea of informing the general public about specific flood risks in their area.

Interactive Flood Hazard Visualizations

There have been several web applications developed recently that visualize flooding scenarios. Most notably, the recently developed flood mapping web application was deployed and tested over the area of the Gippsland Lakes in Victoria, Australia. Here a model was developed using Adobe Flash to construct dynamic simulation models on knowledge of flood -hazard characteristics and future scenarios. GIS mapping was used to build the Gippsland Lakes area in a virtual space. It was noted that fine-grained spatial resolutions and other dimensions such as time can only be handled on a much coarser level [18]. The methods used for digital elevation modeling (DEM) with GIS do matter and can radically affect performance and accuracy of mapping [19]. This was taken into consideration on the Gippsland Lakes project. A model for a design was created that would encourage interactivity from the end users, but not be used as predictive modeling.

The visualization tool provides an effective, inexpensive and interactive spatial communication method that allowed the residents to see at a resolution of individual land parcels [18]. In the visualization tool developed, water level could easily be changed by the end user. Different locations and layers could simply be selected and turned on and off to see the varying degree of flooding risks. Images from past events were integrated into the flood viewer to give a realistic scale of the amount of flood risk a specific area was in. Overall, it was found that the flood viewer application was a good tool for stakeholders to explore the nature of flooding and flooding related risks.

Communicating in 3D Video

A more traditional way of representing flood hazard scenarios has been to use pre-rendered videos constructed with GIS data. In the Exeter Area of the UK a different approach to representing flooding dangers was taken. In the past, one- and two-dimensional models were built that used a high-resolution digital terrain model. While these are useful in displaying accurate flooding modeling and risk area, it was found harder for people to understand the risk associated with higher profile floods [20]. New software was created and designed in three steps:

- Data Collections
- Hydrodynamic modelling
- Creating of three-dimensional models and visualizations videos

Three-dimensional models and interactive videos were important because it was found that not only the dimensionality of modeling is important, but how the uncertainties are displayed and understood study by participants [21]. As such, different visuals were added to the water and notable locations were shown as water slowly overtook them. One can clearly see the level of detail as the muddy water can be seen overtaking the road which is clearly marked.

The overall reception to the Exeter modeling project was taken as a "wake-up call" to most who saw it [20]. Notably it was able to convince the local government council to combine a total of 6 million euros to be made toward future flood upgrades. It allowed for the public to be able to easier to interpret a simple video than decipher a complex map. Overall, most people realized how the generated flood model affected them, so they put forth a greater effort in protecting the community.

Augmented Reality in Flooding Visualizations

As technology increases, AR technologies will serve as enablers of healthier and safer living for individuals and communities [22]. Recently at the University of Sheffield, UK a real time visualization application using natural feature tracking (NFT) was developed to show past and current flooding levels. Built on Android systems using Java and Vuforia the flood visualization application was developed for anywhere augmentation with simultaneous localization and mapping [23]. This was slightly different from traditional visualization methods as it used natural features tracking instead of specific digital elevation models. Features can be built in real time with basic objects available. Flood plain information is then rendered onto the environment and displayed at the appropriate level. The left view shows how the application is able to build primitive shapes to depict realistic structures along with natural feature tracking in the right view. While this visualization may not be as accurate as others, it does allow for more people, specifically those who live in flood prone areas, to see the dangers of flooding.

No Need for Big Data

Recent 3D visualization applications have been turning towards high resolution digital elevation models for simulation. While detailed, digital elevation data is not always needed to illustrate the risk. There are some challenging issues in handling the large volume of data needed for GIS calculations. This traditionally leads to long load and processing times [24]. One can reduce the scope of said data, but only to an extent. A one-meter resolution is recommended to accurately represent a detailed elevation model [25]. If an application is to be widely used it needs to be able to run on the least amount of hardware available. This is currently not feasible on mobile technology just due to the shear amount of data needed. As such, new applications need to emerge that will track the world a different way.

CHAPTER V:

METHODOLOGIES

Flood risks information systems are lacking in the clarity of the information presented. There is a need for a clear and easy to use modeling system that can show several historic flooding scenarios. A different approach to flood modeling was taken in order to complement the currently available FEMA flood mapping systems. A flood risk information system was proposed to consider the historic impacts rather than the predicted impacts of flooding. As such, a flood risk information and visualization system were constructed for easily accessible mobile use. The system was made to be adapted to multiple mobile devices as well as desktop computers. A two-pronged approach was taken to accurately display flooding information; A combination of both augmented reality and high-level flood mapping systems were developed. It was hoped that with a combination of both augmented reality and mapping that the individual would be better informed about local flooding risks. Augmented reality shows flood water in an immediate area to the user. It considers the different objects and furniture around them, mapping them accordingly in virtual space. Several different techniques were used to map the area around a user in AR space. A mapping system shows the flood waters over the local area, with notable landmarks and elevation made apparent. Both visualization systems can be toggled in real time, to allow the user to change their perspective on the

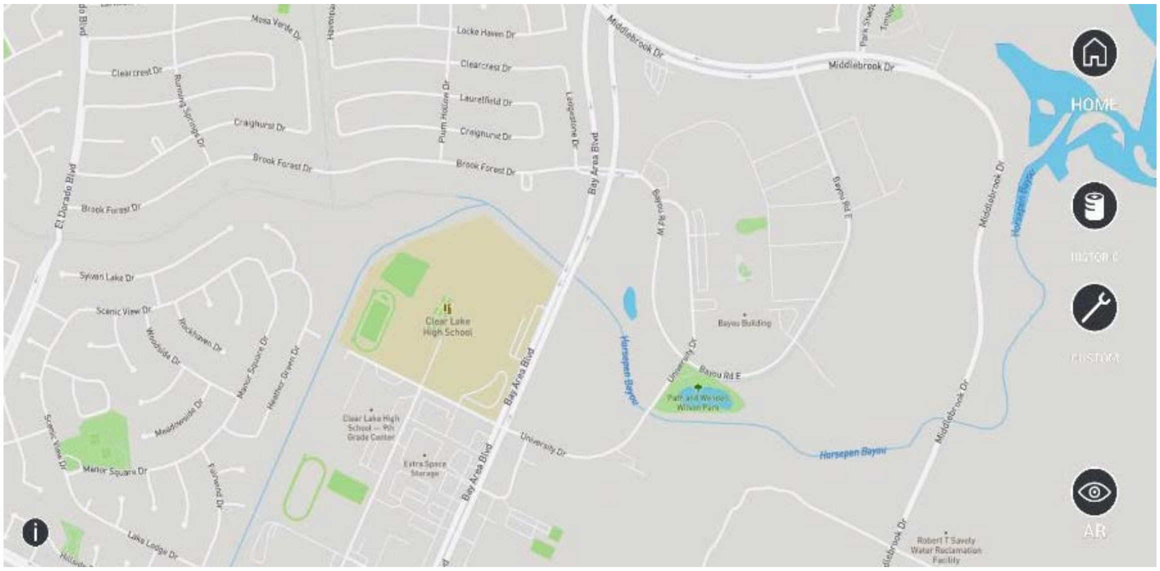

Figure 2: Main view of the developed flood mapping application. Various features can be selected from the right. The user can toggle from AR to Map view on the bottom right of the main screen. Selected home reset the entire application to default.

flooding. Historic river data is available to simulate the past flooding scenarios. Additionally, custom controls allow the user to adjust the flood waters to see theoretical flooding in their area. Coastal sea level rise scenarios were also incorporated into the design. A user, if on a coast can see the various scenarios where sea level continues to rise into 2100. The mobile application was designed in Unity 3D using Vuforia for AR, Mapbox for mapping, and Firebase for databased management support. The mobile application is available for custom IOS systems for easy accessibility and can be easily ported to Android system thanks to Unity.

Augmented Reality

With the use of augmented reality, the application can show flood waters in the user's immediate area. The immediate area is mapped out with augmented reality in a simple grid pattern. Simple obstructions can be recognized, such as tables, chairs, and boxes. The elevation and altitude of the user is used to calculate the difference of location to the flood water sensors. Once the flood water is recognized to be above the currently mapped object, the water is then rendered into AR space. For historic scenarios, the water is only rendered if there was actual flooding in the area. Additionally, coastal sea level rise scenarios are available. If near the coast, the user can see the predicted sea level rise over the next 100 years.

Mapping larger areas in augmented space has always been a challenge. As discussed earlier there are several different methods that can be used. The purpose of this application was to be able to render anywhere at any time. This requirement made it difficult as several objects have complex geometry. Several different techniques were attempted in Vuforia to map the local area. At first an attempt was made to combine different detected ground planes and stich them together to generate a virtual ground. This proved problematic as certain geometry wouldn't be detected as a plane. This would leave to gaps in the virtualized floor or misinterpreted ground planes. Through Vuforia one can access the point cloud from the AR engine. The point cloud from Vuforia can provide the developer with numerous points in 3D space relative to real world space. Various prototypes of the application were tested using Vuforias point cloud to stitch together thousands of points to generate a coarse ground map. Several attempts were made using the Delaunay triangulation method to be able generate the needed triangles for the ground plane. This ultimately proved to be too resource-intensive for the mobile device being used as the testing rig. The amount of data that needed to be stored and process was far greater than the available memory. Finally, an approach was used that took Vuforia identified points and placed them into "buckets", these buckets were laid across the surface in a grid. As a point was detected within the grid, it was average into the overall position of said grid. This approached allowed for an easier way to deal with the large amount of points that can be generated from Vuforias point cloud and not overload the mobile device. The grid can be adjusted for a finer resolution of the ground and is able to be expanded as one move in the real environment. However, problems may arise when moving as the GPS tracking can get out of sync with the ground grid being rendered. This is most likely due to differences in the accuracy of GPS and the ground plane being visualized. This problem seemed to get worse as one moved indoors, leading to less GPS accuracy. As such each “bucket” is constantly updating with new points to attempt to keep and accurate reading.

The water was displayed using a flat plane structure within Unity with a modified shader to give the appearance of water. This plane could be easily raised or lowered based on the historic water level at the current location. The plane was centered at the location of the user and is then scaled based on the amount of area that can be mapped. As the historic data was taken from nearby river sensors a user would have to be relatively close in order to get an accurate representation of the historic flooding. To account for slight differences in elevation, the deviation in elevation and altitude was calculated from the original river sensor to the current location of the user. It was accounted for when calculating the total water rise. Within the augmented reality space, the ground plane visualizer was able to hide the plane if the area was mapped above. Additionally, making it appear as if the water was rising from the ground. This was used to help give the illusion of flood waters interacting with the different object in an area.

Overall, an area can be mapped out in augmented reality with a certain level of detail. The application is set up to start out with a grid of five by five meters with a resolution of one meter. The floodplain can grow as the user moves around in real world space. The elevation and altitude of the user is used to calculate a deviation with the river sensor elevation. There are drawbacks, however with the current implementation. It should be able to be used anywhere and anytime, but as the distance from the sensor increases the error of water rise will also increase. AR mapping has problems with GPS sway, leading to drifting in the ground plane mapping system. The distance that can be represented in AR space is also limited. However, the mapping component can give a higher-level view of the flooding and its impact on the larger area.

Local Mapping

A mapping feature was added along with augmented reality. The map can show the local area around the user consisting of notable landmarks such as streets, rivers, and parks. The augmented reality is toggleable with the mapping feature in real time. As the simulated flood water rises one can see the water move up out of the rivers and onto the local roads. Elevation is taken from Mapbox Terrain-RGB raster tile sets to give a rough representation of local terrain elevation. As the water plane rises, it crests over the various bumps and crevasses in the map terrain. Local landmarks are made clear on the map. Mapping is locked to the local area due to the historic data from local rivers. However, it is possible to zoom in and out of the map to a limited degree to get a closer view of the flooding scenario. Detailed GIS elevation data was considered at a point but was thrown out due to the shear amount of data that would be used on mobile devices.

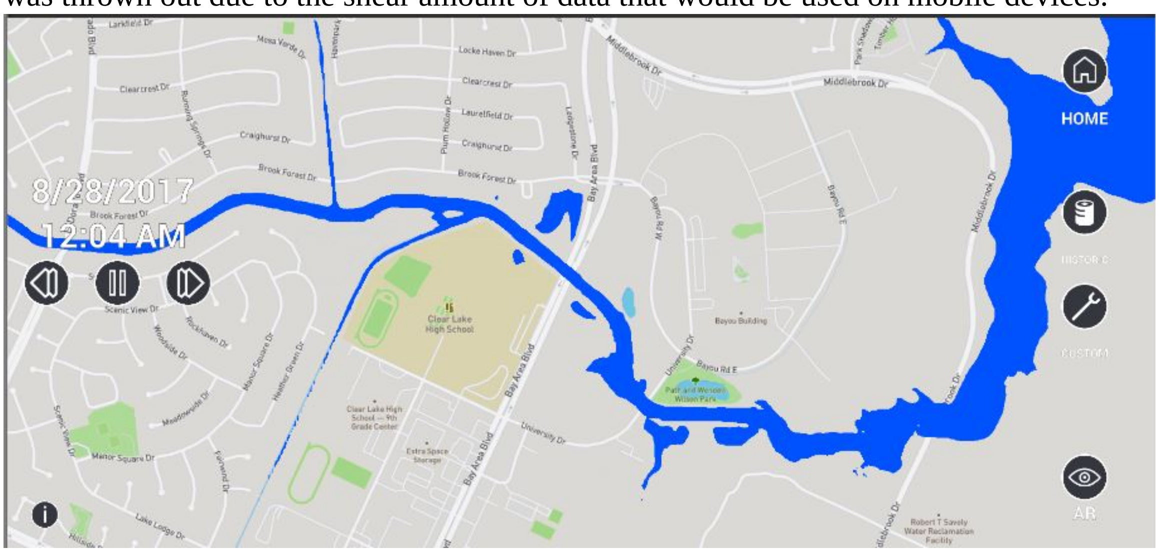

Figure 3: Map view can show the various elevation view of the current location. The water is shown to slowly crest over the various feature and landmarks in the area. River and streams are made apparent to the user.

These maps are comparable to the FEMA flood maps from Figure 1 but can show the historic and local flooding data. Overall, with a combination of mapping and augmented reality, it is intended that end users gain a better understanding of local flooding scenarios.

Historic Data

Historic river data was gathered from the Harris County flood warning system as well from USGS water information system. Data was stored remotely in Google's Firebase database management system. A separate application was developed for desktop applications to quickly upload and edit the historic data in Firebase. This application can take in selected data from the flood information systems, parse it, and upload it to Firebase for the mobile application's use. Firebase was used for its easy integration with Unity as well as possible integration, in the future with the Microsoft HoloLens. Coastal flooding scenarios are also stored within the Firebase system and can be expanded upon through the external application. Data needed to be taken from scenarios that were close to river sensors. As stated, the deviation from the flood sensor to the current user location was recorded, but as distance increased the accuracy of the flood levels becomes distorted. Overall, the Firebase management system provides a reliable and organized structure for storing large amounts of data recorded in historic flooding scenarios.

CHAPTER VI:

SURVEYS AND STATISTIC

For testing, the application was built on an IOS device as well as deployed as a desktop application. A general survey was conducted on the various features of the flood mapping application. For IOS surveys it included the AR portion of the application while desktop surveys were limited to the mapping aspect of the application. The survey was conducted with the help of the new software engineering students at the University of Houston – Clear Lake. Most of the respondents were from out of town or international, which provided a good sample to test as they had not experienced Houston flooding before. Responses to the survey were ranked from least beneficial to most beneficial. The following questions were asked:

1. Do the simulations accurately display flooding hazard relevant to you?
2. Do the simulations show the risk of flooding in your specific area?
3. Were you able to understand how fast flood waters can rise?
4. Do you now have a better understanding of the past flooding events in your area?
5. Would more simulations like this be helpful in determining flood risks in your area?

The responses to the survey questions were generally positive. The application was able to simulate the past flooding hazards in the local area as well as show the speed of the flood's development. Most respondents were able to get a better understanding of the past flooding events in the area. They found the simulation helpful to understanding the general flood risks inherit to the area. Those who were surveyed on IOS, felt that the mapping feature of the application provided significantly better information than the augmented reality feature. This is most likely to the loss of scale and impact in the local augmented reality visualizations. The visualizations seem to lack the necessary detail needed to give the user a better risk indication of the flood waters.

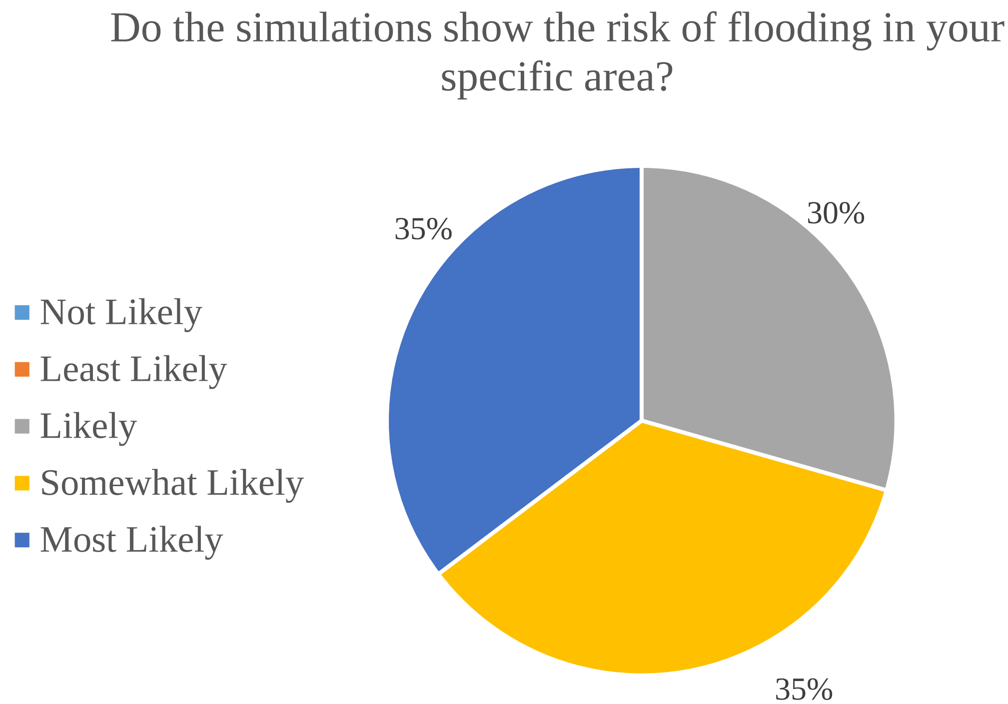

Figure 4: Do the simulations show the risk of flooding in your specific area? From those surveyed, the application developed was somewhat able to give an accurate representation of the risk in the local area.

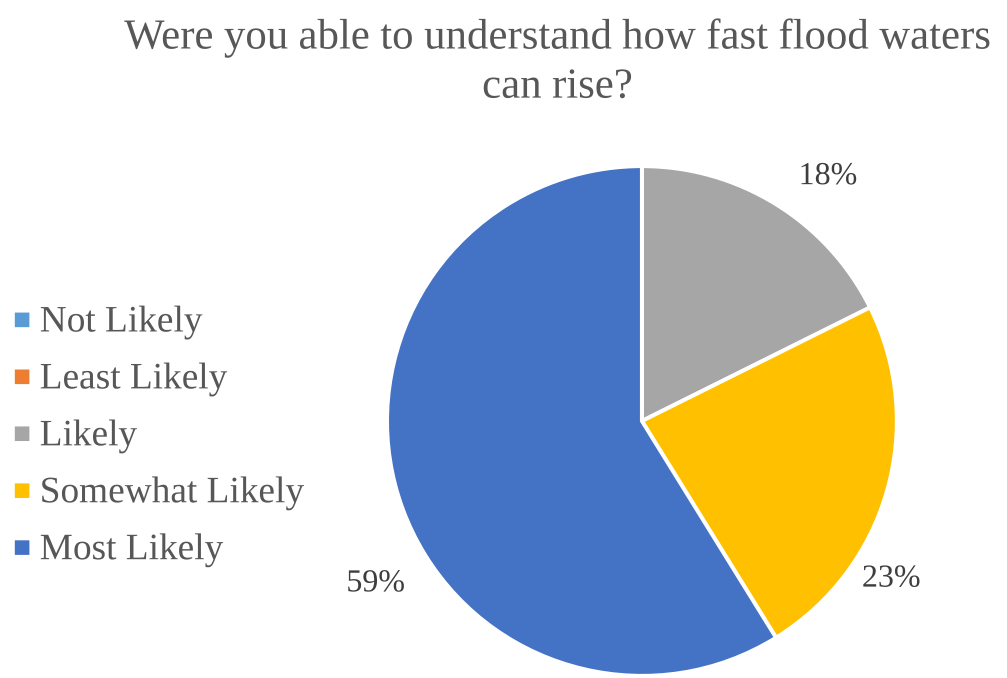

Figure 5: Where you able to understand how fast flood waters can rise? Most surveyed understood that the flood waters can rise at accelerated rates.

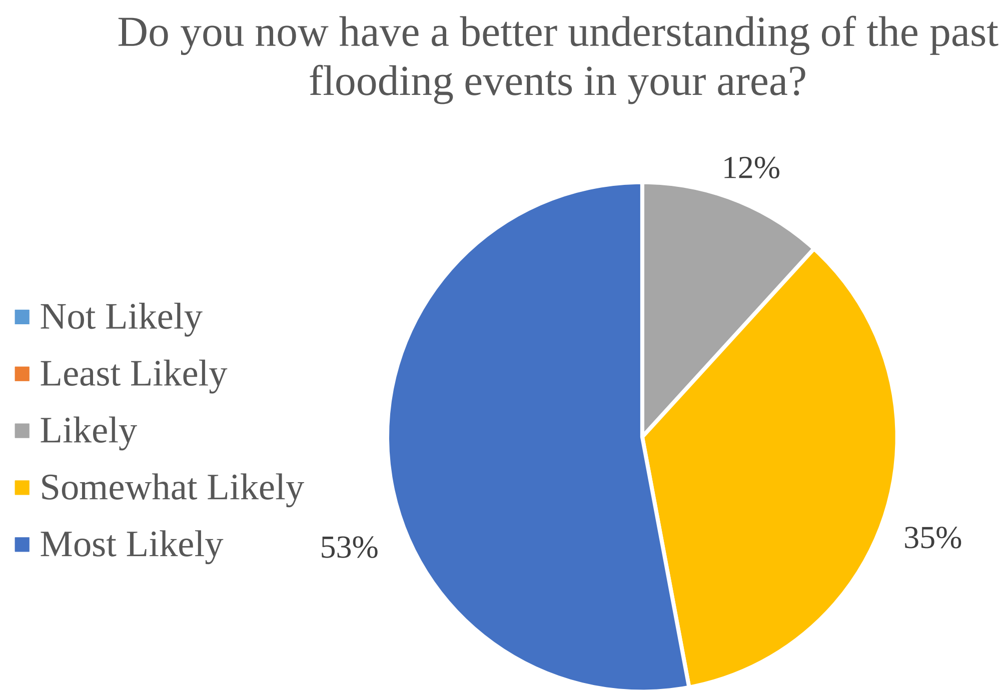

Figure 6: Do you now have a better understanding of the past flooding events in your area? A high percentage of students surveyed were understanding of the flood risks.

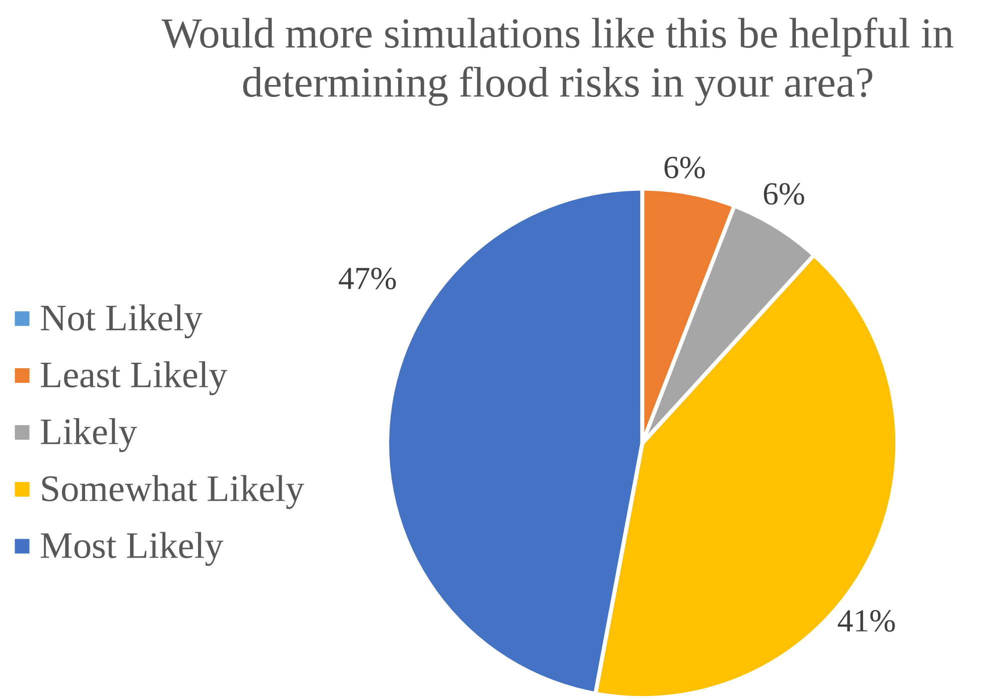

Figure 7: Would more simulations like this be helpful in determining flood risks in your area? While most said more simulations would be helpful some wanted a combination of both the FEMA flood plains as well as the developed application.

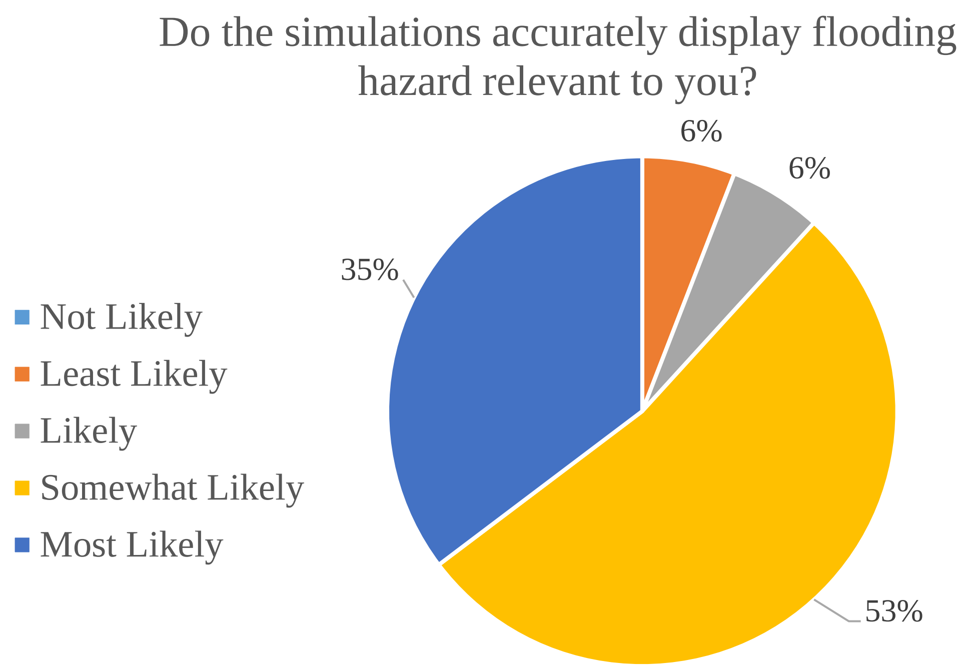

Figure 8: Do the simulations accurately display flooding hazards relevant to you? Most suggested that more information would be needed on the mapping system. It would be helpful for more datasets as well as better elevation mapping.

General response questions were also asked to be able to compare the flooding application to the currently available flood maps on the FEMA website. A brief response was requested for the following questions:

1. Comparing to the currently available flood mapping systems, which visualizations gave you a better understating of the flood risks?
2. What improvements could be made to the application to help you better understand the risks of the flooding?

Overall, most responses were positive towards the application. Table 1 shows the full extent of the responses. The simulations accurately displayed the flooding information while also showing the risks involved. When compared to the FEMA flood mapping system, a lot of respondents indicated that they would like to see features from both the flood mapping systems incorporated into a single application. The mapping feature of the application performed significantly better than the augmented reality feature. The mapping feature was able to give those surveyed a better look at the current flooding extents and how it was interacting with the various landmarks in the area. The augmented reality feature unfortunately was limited on the size and scale that was able to be mapped leading to distorted flood waters. The general response was that the AR feature inaccurately represented flooding scenarios. Due to field of view problems over large areas the flood waters would appear distorted over large distances. Going forward it is suggested that the application be fully developed by using the mapping feature and abandoning augmented reality development due to inaccuracies and lack of realistic flood waters.

Overall, the mapping feature of the historic flood application was a success, and it is suggested that a similar application be developed in the future. If used along with the currently implemented FEMA flood hazard maps, the application allows users to better understand the risks of flooding. The augmented reality feature of the application could be improved upon, but it is suggested that for large scale flooding that augmented reality not be used. It would be interesting to consider using a more virtual reality approach as it would have more of a graphical impact for the user. If a continuation is to be done, it would be suggested that focus be given to a larger scale mapping system with historical and future flooding scenarios.

CHAPTER VII:

DISCUSSION

The current flood information tools are too limited in their design. It is difficult to accurately show understandable information about the local flooding hazards. The visual risk of flooding is lost on many commonly available flood mapping systems. As such, newer more interactive systems have been proposed to increase flooding awareness. Several different mapping systems were explored with various levels of complexity. Each had different levels of abstraction of flooding visualizations. Most in depth flood mapping systems have been limited to specific local areas, with preplanned, restructured, prerendered scenarios that are only applicable to specific locations. The mobile application that has been developed attempts to rectify some of these static problems. It attempts to provide an anywhere augmented reality mapping solution that can be used in any location, indoors or out. Additionally, the local area is available mapped to a wider view. With this application that was developed it is hoped that future projects will be initiated that involve historic flooding scenarios.

A mobile application was developed that was able to display historic flooding scenarios in augmented reality as well overlay on a local map. The application can be used anywhere and anytime with the ability to pull from historic as well as custom flooding scenarios. Each historic scenario is taken from local river level data and converted for the applications use. A separate application is available to input custom or additional historic data where needed. While the AR flooding simulation runs, users can see the flooding water slowly rise around different object and eventually overtake them. This is done to invoke a feeling of dread in the user, with the hopes of them better understanding the risks flood waters pose. Additionally, a higher-level abstraction of the nearby area can be seen that allows users to see local landmarks and the impact flood waters have on them. Users can easily toggle between augmented reality view and the map to get a better understanding of scale in real time. Overall, the application can provide a decent simulation of past flooding scenarios in both augmented and local space.

The survey conducted indicated that an application of this type would be useful for illustrating flooding hazards of the past. The results of the survey showed that the simulations were likely to illustrate the risks of the flooding in the immediate area to the user. The application was able to teach about the various flood risks in the local area as well. Overall, the flooding application was successful in illustrating past flooding events to new individuals to the area. While the mapping component of the application was successful the augmented reality feature of the application was not. The need for an augmented reality display of flooding information is not high. One would need to be in a location that had previously flooded as well as simulate a past event when flooding occurred. Overall, from the survey it was seen that there is a need for historic flooding representation through mapping, while an augmented reality approach is not as desirable. When compared to the currently available flood mapping systems there could be a future where features from both the FEMA flood prediction and the developed historic flood viewer could be used. There was a strong interest in a combined application that could show you accurate past events as well as show the flood plains and future risks factors of flooding in the area.

With using historic river flood data as the source of flooding, several issues can occur. As one gets further away from the local sensors, there will be a greater margin of error in displaying accurate flooding. With the limited number of sensors in an area it can be hard to accurately represent flooding a few miles away. This application does not consider soil types, runoff, etc. into flooding calculations. The application takes historic river levels and overlays them in augmented reality space. As such, the application should in no way be taken as a predictor or an accurate historical representation of flood water.

Overall, it is hoped that the demonstration of the developed flood mapping application illustrates the need for additional historic and future flooding resources. Future development is suggested to iterate on the mapping feature. The public must be informed in a more impactful and visual manner in order to fully illustrate the dangers of flooding. Without proper flood mapping systems individuals can be blind to the potential dangers lurking in the waters.

CHAPTER VIII:

FUTURE WORK

As stated, future work should be considered with the mapping aspect of the flood visualization system. From those survey the general free response was one of wanting both the FEMA flood mapping system as well as the historic flood visualization application developed. The augmented reality application seemed to preform much worse than the simple mapping, as such it is suggested to avoid augmented reality for large area flood mapping. This is largely due to needing to be in an immediate area that had flooded in order to see flood waters in AR. Going forward, future work should be considered in a mobile mapping application that can show both past, present, and future flood risks.

REFERENCES

[1] B. Jongman, P. J. Ward, and J. C. J. H. Aerts, “Global exposure to river and coastal flooding: Long term trends and changes,” *Glob. Environ. Change*, vol. 22, no. 4, pp. 823–835, Oct. 2012, doi: 10.1016/j.gloenvcha.2012.07.004.

[2] B. Neumann, A. T. Vafeidis, J. Zimmermann, and R. J. Nicholls, “Future Coastal Population Growth and Exposure to Sea-Level Rise and Coastal Flooding - A Global Assessment,” *PLOS ONE*, vol. 10, no. 3, p. e0118571, Mar. 2015, doi: 10.1371/journal.pone.0118571.

[3] A. Toimil, I. J. Losada, P. Díaz-Simal, C. Izaguirre, and P. Camus, “Multi-sectoral, high-resolution assessment of climate change consequences of coastal flooding,” *Clim. Change*, vol. 145, no. 3–4, pp. 431–444, Dec. 2017, doi: 10.1007/s10584-017-2104-z.

[4] M. E. Mousavi, J. L. Irish, A. E. Frey, F. Olivera, and B. L. Edge, “Global warming and hurricanes: the potential impact of hurricane intensification and sea level rise on coastal flooding,” *Clim. Change*, vol. 104, no. 3–4, pp. 575–597, Feb. 2011, doi: 10.1007/s10584-009-9790-0.

[5] R. J. Nicholls, “Coastal flooding and wetland loss in the 21st century: changes under the SRES climate and socio-economic scenarios,” *Glob. Environ. Change*, vol. 14, no. 1, pp. 69–86, Apr. 2004, doi: 10.1016/j.gloenvcha.2003.10.007.

[6] S. Tunstall, S. McCarthy, and H. Faulkner, “Flood risk management and planning policy in a time of policy transition: the case of the Wapshott Road Planning Inquiry,

Surrey, England," *J. Flood Risk Manag.*, vol. 2, no. 3, pp. 159–169, Sep. 2009, doi: 10.1111/j.1753-318X.2009.01030.x.

[7] S. Burch, S. R. J. Sheppard, A. Shaw, and D. Flanders, "Planning for climate change in a flood-prone community: municipal barriers to policy action and the use of visualizations as decision-support tools: Planning for climate change in a flood-prone community," *J. Flood Risk Manag.*, vol. 3, no. 2, pp. 126–139, Mar. 2010, doi: 10.1111/j.1753-318X.2010.01062.x.

[8] S. Birkholz, M. Muro, P. Jeffrey, and H. M. Smith, "Rethinking the relationship between flood risk perception and flood management," *Sci. Total Environ.*, vol. 478, pp. 12–20, Apr. 2014, doi: 10.1016/j.scitotenv.2014.01.061.

[9] D. J. Lieske, T. Wade, and L. A. Roness, "Climate change awareness and strategies for communicating the risk of coastal flooding: A Canadian Maritime case example," *Estuar. Coast. Shelf Sci.*, vol. 140, pp. 83–94, Mar. 2014, doi: 10.1016/j.ecss.2013.04.017.

[10] E. Figueiredo, S. Valente, C. Coelho, and L. Pinho, "Coping with risk: analysis on the importance of integrating social perceptions on flood risk into management mechanisms – the case of the municipality of Águeda, Portugal," *J. Risk Res.*, vol. 12, no. 5, pp. 581–602, Jul. 2009, doi: 10.1080/13669870802511155.

[11] D. M. Lumbroso, N. R. Suckall, R. J. Nicholls, and K. D. White, "Enhancing resilience to coastal flooding from severe storms in the USA: international lessons," *Nat. Hazards Earth Syst. Sci.*, vol. 17, no. 8, pp. 1357–1373, Aug. 2017, doi: 10.5194/nhess-17-1357-2017.

[12] W. Kellens, R. Zaalberg, and P. De Maeyer, “The Informed Society: An Analysis of the Public’s Information-Seeking Behavior Regarding Coastal Flood Risks: Analysis of the Public’s Information-Seeking Behavior Regarding Coastal Flood Risks,” *Risk Anal.*, vol. 32, no. 8, pp. 1369–1381, Aug. 2012, doi: 10.1111/j.1539-6924.2011.01743.x.

[13] B. A. Dobson, J. J. Miles-Wilson, I. D. Gilchrist, D. S. Leslie, and T. Wagener, “Effects of flood hazard visualization format on house purchasing decisions,” *Urban Water J.*, vol. 15, no. 7, pp. 671–681, Aug. 2018, doi: 10.1080/1573062X.2018.1537370.

[14] M. P. Wadey, S. N. Cope, R. J. Nicholls, K. McHugh, G. Grewcock, and T. Mason, “Coastal flood analysis and visualisation for a small town,” *Ocean Coast. Manag.*, vol. 116, pp. 237–247, Nov. 2015, doi: 10.1016/j.ocecoaman.2015.07.028.

[15] J. Clarke, J. Wicks, L. Lovell, D. Han, and J. Davis, “Keep it simple: user-focused flood risk visualisation,” *Proc. Inst. Civ. Eng. - Civ. Eng.*, vol. 167, no. 5, pp. 41–48, May 2014, doi: 10.1680/cien.13.00016.

[16] J. G. Leskens *et al.*, “An interactive simulation and visualization tool for flood analysis usable for practitioners,” *Mitig. Adapt. Strateg. Glob. Change*, vol. 22, no. 2, pp. 307–324, Feb. 2017, doi: 10.1007/s11027-015-9651-2.

[17] H. Apel, A. H. Thieken, B. Merz, and G. Blöschl, “Flood risk assessment and associated uncertainty,” *Nat. Hazards Earth Syst. Sci.*, vol. 4, no. 2, pp. 295–308, Apr. 2004, doi: 10.5194/nhess-4-295-2004.

[18] M. L. F. Coller, P. Wheeler, J. Kunapo, and J. Peterson, “Interactive flood hazard visualisation in Adobe Flash: Interactive flood hazard visualisation,” *J. Flood Risk Manag.*, vol. 11, pp. S134–S146, Jan. 2018, doi: 10.1111/jfr3.12188.

[19] B. Ayeni and K. J. Samuel, "An evaluation of digital elevation modeling in GIS and Cartography," *Geo-Spat. Inf. Sci.*, vol. 17, no. 2, pp. 139–144, Apr. 2014, doi: 10.1080/10095020.2013.772808.

[20] S. Y. Evans, M. Todd, I. Baines, T. Hunt, and G. Morrison, "Communicating flood risk through three-dimensional visualisation," *Proc. Inst. Civ. Eng. - Civ. Eng.*, vol. 167, no. 5, pp. 48–55, May 2014, doi: 10.1680/cien.13.00020.

[21] N. J. Lim, S. A. Brandt, and S. Seipel, "Visualisation and evaluation of flood uncertainties based on ensemble modelling," *Int. J. Geogr. Inf. Sci.*, vol. 30, no. 2, pp. 240–262, Feb. 2016, doi: 10.1080/13658816.2015.1085539.

[22] M. N. Kamel Boulos, Z. Lu, P. Guerrero, C. Jennett, and A. Steed, "From urban planning and emergency training to Pokémon Go: applications of virtual reality GIS (VRGIS) and augmented reality GIS (ARGIS) in personal, public and environmental health," *Int. J. Health Geogr.*, vol. 16, no. 1, Dec. 2017, doi: 10.1186/s12942-017-0081-0.

[23] P. Haynes, S. Hehl-Lange, and E. Lange, "Mobile Augmented Reality for Flood Visualisation," *Environ. Model. Softw.*, vol. 109, pp. 380–389, Nov. 2018, doi: 10.1016/j.envsoft.2018.05.012.

[24] H. Singh and R. D. Garg, "Web 3D GIS Application for Flood Simulation and Querying Through Open Source Technology," *J. Indian Soc. Remote Sens.*, vol. 44, no. 4, pp. 485–494, Aug. 2016, doi: 10.1007/s12524-015-0498-5.

[25] C. Krolik-Root, D. L. Stansbury, and N. G. Burnside, "Effective LiDAR-based modelling and visualisation of managed retreat scenarios for coastal planning: An

example from the southern UK," *Ocean Coast. Manag.*, vol. 114, pp. 164–174, Sep. 2015, doi: 10.1016/j.ocecoaman.2015.06.013.

APPENDIX A:

IMPORTANT CODE AND GITHUB

The following is the important code that is used within the ground plane visualization system. Additional controller and UI scripts can be found at the git.

It can also be cloned from github at: https://github.com/MarkHalle/Coast-Flooding-AR

```
public class GroundPlaneVisualizar : MonoBehaviour
{
    [SerializeField]
    private float startingCubeHeight = 5f;
    [SerializeField]
    public GameObject cubeBasicPrefab;
    [SerializeField]
    public int minimumNumberOfPointsToUpdateCubes = 5;
    [SerializeField]
    private static float globalGroundMin = 0;

    private ARInterface.PointCloud m_PointCloud;
    private bool isCalculatingExtremes = false;

    public static event EventHandler<PointCloudProperties>
PointExtremesUpdatedEvent;

    private void Start()
    {
        RunBasicARGround();
    }

    private void RunBasicARGround()
    {
        // Square size must be 1
        CreateBaseCubes();
        RefreshPointCloudExtremes();
    }

    private void CreateBaseCubes()
    {
        for (int x = -5; x <= 5; x++)
        {
            for (int z = -5; z <= 5; z++)
            {
                Instantiate(cubeBasicPrefab, new Vector3(x, startingCubeHeight,
z), Quaternion.identity, transform);
            }
        }
    }

    private void RefreshPointCloudExtremes()
    {
```

```
        try
        {
            if (ARInterface.GetInterface().TryGetPointCloud(ref m_PointCloud))
            {
                var pointCloudPoints = new List<Vector3>(m_PointCloud.points);
                if (pointCloudPoints.Count() > minimumNumberOfPointsToUpdateCubes
&& !isCalculatingExtremes)
                {
                    isCalculatingExtremes = true;

                    var pointCloudProperties = new
PointCloudProperties(pointCloudPoints.Min(x => x.x), pointCloudPoints.Max(x =>
x.x),
                        pointCloudPoints.Min(z => z.z), pointCloudPoints.Max(z =>
z.z), pointCloudPoints);
                    GlobalGroundMin = CalculateIfNewMinimum(pointCloudPoints);

                    PointExtremesUpdatedEvent?.Invoke(this, pointCloudProperties);

                    isCalculatingExtremes = false;
                }
            }
        }
        catch(Exception)
        {
            Invoke("RefreshPointCloudExtremes", 1f);
        }
        Invoke("RefreshPointCloudExtremes", 1f);
    }

    private float CalculateIfNewMinimum(List<Vector3> pointCloudPoints)
    {
        float tempMin = pointCloudPoints.Min(y => y.y);
        if (tempMin < GlobalGroundMin)
        {
            return tempMin;
        }
        return GlobalGroundMin;
    }

    public static float GlobalGroundMin { get => globalGroundMin; private set =>
globalGroundMin = value; }

}


// Mark Halle - Software Engineering - UHCL
// Master's Thesis - Coastal Flooding in AR
// 29/01/2020
// Cube controller =>  Where the actual ground is adjusted based on the given
extremes
// Determines from the point cloud how many points are within the cube bounds
// Determines the type of world object based on the number of points given
```

```
public class Cube : MonoBehaviour
{
    private Vector3[] vertexesOfCube = new Vector3[4];
    private CubeType cubeType = CubeType.Unknown;

    [SerializeField]
    private float verticalLocationY = -1;

    [SerializeField]
    private long countOfPointsWithinCube = 0;

    [SerializeField]
    private GameObject cubeGameObject;

    private bool UpdatingPoints = false;

    private void Start()
    {
        SetVertexes();
        GroundPlaneVisualizar.PointExtremesUpdatedEvent +=
GroundPlaneVisualizar_PointExtremesUpdatedEvent;
    }

    private void GroundPlaneVisualizar_PointExtremesUpdatedEvent(object sender,
PointCloudProperties pointCloudProperties)
    {
        if (!UpdatingPoints)
        {
            UpdatingPoints = true;
            if (CheckIfCubeIsInPointCloud(vertexesOfCube.Length - 1,
pointCloudProperties))
            {
                UpdatedVerticalLocationWithPointCloud(pointCloudProperties);
                cubeType = UpdatePredictedCubeType();
                MoveCubeToDeterminedLocation(pointCloudProperties);
            }
            UpdatingPoints = false;
        }
    }

    private void SetVertexes()
    {
        float tempDistance = (cubeGameObject.transform.localScale.x / 2);
        vertexesOfCube[0] = new Vector3(transform.position.x - tempDistance,
transform.position.y, transform.position.x - tempDistance); //-x -z
        vertexesOfCube[1] = new Vector3(transform.position.x - tempDistance,
transform.position.y, transform.position.x + tempDistance); //-x +z
        vertexesOfCube[2] = new Vector3(transform.position.x + tempDistance,
transform.position.y, transform.position.x - tempDistance); //+x -z
        vertexesOfCube[3] = new Vector3(transform.position.x + tempDistance,
transform.position.y, transform.position.x + tempDistance); //+x +z
    }

    private bool CheckIfCubeIsInPointCloud(int indexOfVertexLength,
PointCloudProperties pointCloudProperties)
```

```
    {
        if (indexOfVertexLength >= 0)
        {
            if (vertexesOfCube[indexOfVertexLength].x <=
pointCloudProperties.MaxXPosition && vertexesOfCube[indexOfVertexLength].x >=
pointCloudProperties.MinXPosition &&
                vertexesOfCube[indexOfVertexLength].z <=
pointCloudProperties.MaxZPosition && vertexesOfCube[indexOfVertexLength].z >=
pointCloudProperties.MinZPosition)
            {
                CheckIfCubeIsInPointCloud(--indexOfVertexLength,
pointCloudProperties);
            }
            else
            {
                return false;
            }
        }
        return true;
    }

    private bool CheckIfPointInCubeBounds(Vector3 point)
    {
        if (vertexesOfCube[0].x <= point.x && vertexesOfCube[0].z <= point.z) //-x
-z
        {
            if (vertexesOfCube[1].x <= point.x && vertexesOfCube[1].z >= point.z)
//-x +z
            {
                if (vertexesOfCube[2].x >= point.x && vertexesOfCube[2].z <=
point.z) //+x -z
                {
                    if (vertexesOfCube[3].x >= point.x && vertexesOfCube[3].z >=
point.z) //+x +z
                    {
                        return true;
                    }
                }
            }
        }
        return false;
    }

    private void UpdatedVerticalLocationWithPointCloud(PointCloudProperties
pointCloudProperties)
    {
        var pointsWithinCube = pointCloudProperties.PointCloudPoints.Where(point
=> CheckIfPointInCubeBounds(point)).Select(point => point.y);
        if (pointsWithinCube.Count() > 0)
        {
            countOfPointsWithinCube += pointsWithinCube.Count();
            verticalLocationY = (pointsWithinCube.Average() +
verticalLocationY)/2;
        }
    }
```

```
    private CubeType UpdatePredictedCubeType()
    {
        if (countOfPointsWithinCube > 25)
        {
            return CubeType.Ground;
        }
        if (countOfPointsWithinCube <= 0)
        {
            return CubeType.Wall;
        }
        return CubeType.Furniture;
    }

    private void MoveCubeToDeterminedLocation(PointCloudProperties
pointCloudProperties)
    {
        if (cubeType == CubeType.Ground)
        {
            transform.position = new Vector3(transform.position.x,
GroundPlaneVisualizar.GlobalGroundMin, transform.position.z);
        }
        else if (cubeType == CubeType.Furniture)
        {
            transform.position = new Vector3(transform.position.x,
verticalLocationY, transform.position.z);
        }
        else if (cubeType == CubeType.Wall)
        {
            transform.position = new Vector3(transform.position.x,
cubeGameObject.transform.localScale.y / 2, transform.position.z);
        }
    }

}


// Mark Halle - Software Engineering - UHCL
// Master's Thesis - Coastal Flooding in AR
// 02/04/2020
// Historic Scenario Controller =>  handles the overtime of historic flood planes
// Steps up the plane overtime from a given data historic set
// Able to adjust speed through UI

public class HistoricScenarioController : MonoBehaviour
{
    [SerializeField]
    private HistoricScenarioPanel historicScenarioUI;
    [SerializeField]
    private GameObject floodPlaneGameObject;
    [SerializeField]
    private FirebaseController FirebaseController;
    [SerializeField]
    private List<HistoricScenario> historicScenariosLoaded;
```

```
    [SerializeField]
    private HistoricScenario historicEventsSelected;
    [SerializeField]
    private AbstractMap ARAlignedMap;
    [SerializeField]
    private float timePeriod = 5f;
    private int savedIndexForPause;

    [SerializeField]
    public LocationController locationController;

    [SerializeField]
    public GameObject ARCamera;

    [SerializeField]
    public GameObject mapCamera;

    [SerializeField]
    private float startingMapDistance = 4f;
    [SerializeField]
    private float startingARDistance = 10f;


    [SerializeField]
    private CurrentScenarioController scenarioController;

    public void RunHistoricScenario()
    {
        if (historicScenariosLoaded == null)
        {
            return;
        }
        historicEventsSelected = GetHistoricScenarioFromUI();
        if (historicEventsSelected == null)
        {
            return;
        }
        scenarioController.ActivateHistoricMode();

        if (ARCamera.activeSelf)
        {
            startingARDistance = (float)locationController.Location.Elevation +
(float)locationController.Location.Altitude;  //have to update twice check later
            RunHistoricScenarioSequence(historicEventsSelected,
startingARDistance, 1);
        }
        else
        {
            RunHistoricScenarioSequence(historicEventsSelected,
startingMapDistance, 1);

        }
    }
```

```
    private void RunHistoricScenarioSequence(HistoricScenario
currentHistoricScenarioSelected, float startingDistance, int
counterForCurrentTimePeriodIndex)
    {
        float elapsedTime = 0;
        float endPoint = -startingDistance +
currentHistoricScenarioSelected.historicEvents[counterForCurrentTimePeriodIndex -
1].waterLevel;
        DateTime currentSelectedDate =
currentHistoricScenarioSelected.historicEvents[counterForCurrentTimePeriodIndex -
1].dateTime;

        UpdateDateText(currentSelectedDate);
        StartCoroutine(MoveOverSeconds(floodPlaneGameObject, new Vector3(0.0f,
endPoint, 0f), timePeriod, elapsedTime, counterForCurrentTimePeriodIndex,
currentSelectedDate));
    }

    private HistoricScenario GetHistoricScenarioFromUI()
    {
        var selectedDateFromDropdown =
historicScenarioUI.GetSelectedDropdownValue();
        foreach (var loadedScenario in historicScenariosLoaded)
        {
            Debug.Log(loadedScenario.scenarioDate.ToShortDateString());
            Debug.Log(selectedDateFromDropdown);
            if (loadedScenario.scenarioDate.ToShortDateString() ==
selectedDateFromDropdown)
            {
                return loadedScenario;
            }
        }
        return null;
    }

    public async void LoadHistoricLocationDataFromFirebaseAsync()
    {
        var tempLocation = historicScenarioUI.ReadLocationFromInputField();
        tempLocation = "Horsepen Bayou - Bay Area Boulevard"; //overrides
locationfield for now
        historicScenariosLoaded = await
FirebaseController.LoadHistoricLocationFromFirebaseAsync(tempLocation);
        UpdateDropdownWithDates(historicScenariosLoaded);
    }

    private void UpdateDropdownWithDates(List<HistoricScenario> historicScenarios)
    {
        var tempListOfDates = new List<string>();
        foreach (var day in historicScenarios)
        {
            tempListOfDates.Add(day.scenarioDate.ToShortDateString());
        }
        historicScenarioUI.UpdateInlandDropDown(tempListOfDates);
    }
```

```
    public void ResetFloodController()
    {
        floodPlaneGameObject.transform.position = new
Vector3(floodPlaneGameObject.transform.position.x, -2f,
floodPlaneGameObject.transform.position.z);
        floodPlaneGameObject.SetActive(false);
    }

    public void TurnOnFloodPlane()
    {
        floodPlaneGameObject.SetActive(true);
    }

    private IEnumerator MoveOverSeconds(GameObject objectToMove, Vector3 end,
float seconds, float elapsedTime, int counterForCurrentTimePeriodIndex, DateTime
currentDate)
    {
        Vector3 startingPos = objectToMove.transform.position;
        Debug.Log($"Starting position {startingPos}");
        Debug.Log($"EndingPosition position {UpdateDataWithMapScale(end)}");

        while (elapsedTime < seconds)
        {
            ActivateFloodPlaneForAR(objectToMove);
            objectToMove.transform.position = Vector3.Lerp((startingPos),
UpdateDataWithMapScale(end), (elapsedTime / seconds));
            elapsedTime += Time.deltaTime;
            yield return new WaitForEndOfFrame();
        }
        counterForCurrentTimePeriodIndex++;
        savedIndexForPause = counterForCurrentTimePeriodIndex;

        objectToMove.transform.position = UpdateDataWithMapScale(end);

        UpdateDateText(currentDate);
        RunNextHistoricEvent(counterForCurrentTimePeriodIndex);
    }


    private void ActivateFloodPlaneForAR(GameObject objectToMove)
    {
        if(ARCamera.activeSelf && objectToMove.transform.position.y <
GroundPlaneVisualizar.GlobalGroundMin)
        {
            objectToMove.SetActive(false);
        }
        else
        {
            objectToMove.SetActive(true);
        }
    }

    private void RunNextHistoricEvent(int counterForCurrentTimePeriodIndex)
    {
```

```
        if (counterForCurrentTimePeriodIndex >
historicEventsSelected.historicEvents.Count)
        {
            return;
        }

        if (ARCamera.activeSelf)
        {
            RunHistoricScenarioSequence(historicEventsSelected,
startingARDistance, counterForCurrentTimePeriodIndex);
        }
        else
        {
            RunHistoricScenarioSequence(historicEventsSelected,
startingMapDistance, counterForCurrentTimePeriodIndex);
        }
    }

    private void UpdateDateText(DateTime currentDate)
    {
        string dateTimeToText = $"{currentDate.ToShortDateString()}
{currentDate.ToShortTimeString()}";
        historicScenarioUI.UpdateDateText(dateTimeToText);
    }

    public void IncreaseSpeedButton()
    {
        if (timePeriod == 1)
        {
            return;
        }
        timePeriod--;
        //historicScenarioUI.UpdateSpeedText(timePeriod.ToString());
    }

    public void DecreaseSpeedButton()
    {
        if (timePeriod > 10)
        {
            return;
        }
        timePeriod++;
        //historicScenarioUI.UpdateSpeedText(timePeriod.ToString());
    }

    public void PauseButton()
    {
        StopAllCoroutines();
    }

    public void ResumeButton()
    {
        RunNextHistoricEvent(savedIndexForPause);
    }
```

```
    private Vector3 UpdateDataWithMapScale(Vector3 distance)
    {
        if (mapCamera.activeSelf)
        {
            return new Vector3(distance.x, ARAlignedMap.WorldRelativeScale *
distance.y, distance.z);
        }
        return distance;
    }

    public void ChangeCameraHistoric()
    {
        if (historicEventsSelected != null && scenarioController.ScenarioState ==
ScenarioState.HISTORIC)
        {
            PauseButton();
            if (ARCamera.activeSelf)
            {
                var adjustValue = (floodPlaneGameObject.transform.position.y +
startingMapDistance) - startingARDistance;
                floodPlaneGameObject.transform.position = new
Vector3(floodPlaneGameObject.transform.position.x, adjustValue,
floodPlaneGameObject.transform.position.z);
                ResumeButton();

            }
            else
            {
                var adjustValue = (floodPlaneGameObject.transform.position.y +
startingARDistance) - startingMapDistance;
                floodPlaneGameObject.transform.position = new
Vector3(floodPlaneGameObject.transform.position.x, adjustValue,
floodPlaneGameObject.transform.position.z);
                ResumeButton();
            }
        }
    }

}
```