IE685: MSc-PhD Research Project 1

Topic: Application of Augmented Reality in Industry

Aakash Roy | 21i190007

Supervisor: Prof. Jayendran Venkateswaran

Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

Autumn, 2022-2023

Acknowledgement

I would like to thank Prof. Jayendran Venkateswaran for his guidance and support in this project. His encouragement had enabled me to enter into the world of the evolving technology of the future. His advice and guidance have been remarkable in my learning.

Contents

Chapter 1

Introduction

Industrial Engineering and Operations Research is a field that covers the problems faced by the industries. From modelling a problem to solving, requires many heuristics as well as many robust methods. The very initial step requires understanding the problem which sometimes becomes a bit difficult and for finding the scope of improvements, we need to analyse them in a more playful way. What if we can visualize the problem in no time? It gives us an extra edge to see things one dimension ahead.

Augmented Reality (or AR) in simpler terms means intensifying the reality of real-time objects which we see through our eyes or gadgets like smartphones. AR can impactfully offer an unforgettable experience either of learning, measuring the three-dimensional surfaces, or reviewing the medical statistics of emergency situations consisting of the utmost complications.

So, we started by thinking about the possibilities offered by Augmented reality. Also, anyone with a smartphone can see through AR, which makes it affordable (almost free) for students. Here, we discussed some ideas related to Augmented Reality, which have some major applications in the field of Industrial Engineering and Operations Research.

Chapter 2

Literature Overview

Augmented Reality mainly has two components those are **Software** and **Hardware**. Nowadays, there are many software packages and tools that allow for the creation of AR and VR applications. There are three possible kinds of combinations of realities: Augmented Reality, Augmented Virtually (AV), Mixed Reality, and Virtual Reality. Virtual Reality is a fully digitalized world, where the observer stands in the first person in a completely virtual environment populated by digital objects and scenes. VR requires the use of immersive devices, such as HMDs or Oculus Rift and PlayStation VR. In this framework, the definition of “Mixed Reality” deals with the relationship between real and virtual.
The bridge connecting real and virtual is populated by Mixed Reality technologies that are capable of blending virtual content into the real world or vice versa. A major contribution to this topic is represented by the seminal work by Paul Milgram et al. describing the “Reality-Virtuality Continuum”. This continuum spans from a Real Environment to a fully Virtual Environment. Augmented Reality and Augmented Virtuality can be considered intermediate steps between the outer limits. Shortening this bridge means obtaining the best immersive experience so that optimal results will be achieved when no differences between real and virtual will be perceived by the end-user.

2.1 Software

There are many libraries dedicated to Virtual and Augmented reality. Marker-based tools like ARToolkit are available on the market. They exploit white and black markers [1,2] to retrieve the orientation of the camera used to frame the external world and to refer correctly to the several reference systems (camera, marker, object) necessary to implement AR. The market offers other programs, such as Vuforia, and Unity which allow for using more advanced technologies such as the marker-less AR strategy.
But in our case, we will try to develop things from scratch and by creating each file one by one. Which will require a website for deployment and rendering the scenes.

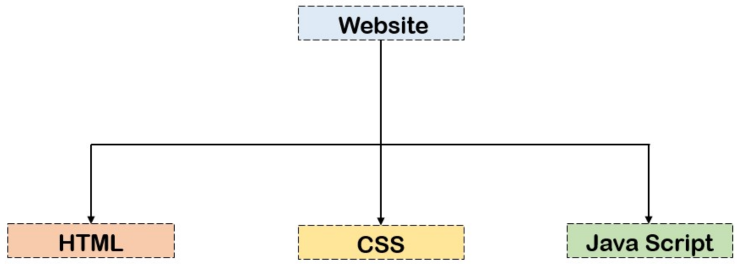

Figure 2.1: Framework of Website

Together with HTML, CSS and JavaScript is one of the three core components of the internet. Any website is bound to use a combination of these three programming languages, each with its own particular purpose. Let's have a look at them in more detail.

2.1.1 HTML

HTML is the language for describing the structure of Web pages. Few of the characteristics of HTML are as follows:

- HTML gives the flexibility to Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

2.1.2 Cascading Style Sheets(CSS)

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts.

- It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers.
- CSS is independent of HTML and can be used with any XML-based markup language.

- The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments.

2.1.3 Java Script

Last but not least, JavaScript allows you to make your website interactive. It allows for all the content you created with HTML and styled with CSS to become more engaging. From clickable buttons to log-in options, JavaScript is what turns your website from a wall of text into an experience. JavaScript wouldn't be a cornerstone of the programming world if it was just about pressing buttons. The practical applications of JavaScript are almost endless, but we've highlighted some of the most common uses down below.

- Interactive Websites
- Video games
- Web Mobile apps

2.1.4 ARCore

Google's ARKit, ARCore is a developer platform for Augmented Reality design and deployment. With this SDK users' phones are able to sense the external environment. What's stunning about this AR SDK is that is supports both Android and iOS devices.
ARCore comes with 3 core capabilities to merge virtual and real worlds:

- Motion tracking – to track phone's position relative to the surroundings.
- Environmental understanding – to detect the size/location of surfaces, from horizontal and vertical, to even angled surfaces.
- Light estimation – to estimate the real-life lighting conditions.

In essence, ARCore is based around 2 elements – real-time position tracking and integration of virtual and real objects. It lets us place objects, texts, etc. withing physical surroundings, and other people then can discover it with Android or iOS phones.

2.1.5 WebXR

WebXR, with the WebXR Device API at its core, provides the functionality needed to bring both augmented and virtual reality (AR and VR) to the web. Together, these technologies are referred to as mixed reality (MR) or cross reality (XR). Mixed reality is a large and complex subject, with much to learn and many other APIs to bring together to create an engaging experience for users. Before getting into too much detail, let's consider some basic concepts of how to develop XR code.

Field of view

The term field of view (FOV) is one which applies to any visual technology, from old film cameras to modern digital video cameras, including the cameras in computers and mobile devices.

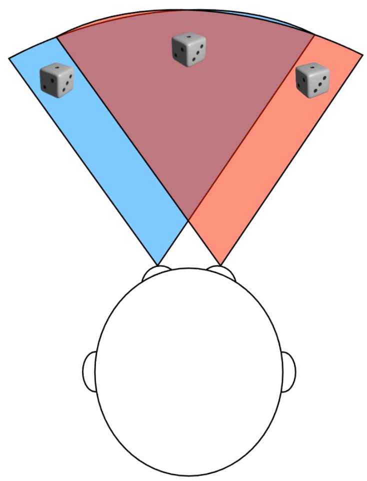

Figure 2.2: Field of View

The field of view is the extent to which we are able to see the environment. The width of the field of view, specified in either degrees or radians, is measured as the angle defining the arc from the far left edge of our field of view to the far right edge.

A human eye is able to take in a FOV of around 135°. Assuming a person has two healthy eyes, the total field of view ends up being about 200° to 220° wide. Why is the FOV wider with two eyes, but not double the single-eye FOV? It's because the two eyes' FOVs overlap a lot. That overlap gives us depth perception, which is around 115° wide. Outside the overlap area, our vision falls back to monocular.
The drawing shown here demonstrates the concept of FOV: blue wedge for the left eye, red wedge for the right eye. The light brown overlapping area is where the viewer has binocular vision and can perceive depth.

Field of view and mixed reality devices

To achieve a wide enough field of view that the user's eyes are tricked into believing that the virtual world completely surrounds them, the FOV needs to at least approach the width of the binocular vision area. Basic headsets typically start around 90° or so, while the best headsets generally have a field of view of around 150°. Because the FOV is a matter of the size of the lenses and how close they are to the user's eyes, there are limitations on how wide the FOV can get without installing lenses into the user's eyeballs.

Degrees of freedom

The term degrees of freedom is an indication of how much freedom of movement the user has within the virtual world. This is directly related to how many types of movement the WebXR hardware configuration is capable of recognizing and reproducing into the virtual scene.

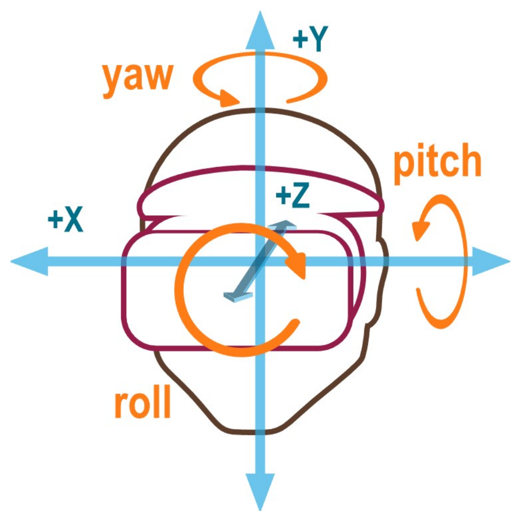

Figure 2.3: Diagram showing the movements possible with 3 degree of freedom hardware: yaw, roll, and pitch

Freedom of translational movement

The other three degrees of freedom are translational, providing the ability to sense movement through space: forward and backward, left and right, up and down. Support for all six degrees of freedom is referred to as 6DoF.

WebXR session modes

WebXR offers support for both augmented reality (AR) and virtual reality (VR) sessions, using the same API. Which type of session you want to create is specified when creating the session. This is done by specifying the appropriate session mode string for the kind of session you want to create.

Augmented reality

In augmented reality (AR), the user sees the imagery you render presented on top of the physical, real-world environment around them. Because AR is always an immersive experience, in which the scene is the entire world around the user (rather than being enclosed in a box on a screen), the only AR session mode is immersive-ar.

2.1.6 WebGL

WebGL (Web Graphics Library) is a JavaScript API for rendering high-performance interactive 3D and 2D graphics within any compatible web browser without the use of plug-ins. WebGL does so by introducing an API that closely conforms to OpenGL ES 2.0 that can be used in HTML ¡canvas¿ elements. This conformance makes it possible for the API to take advantage of hardware graphics acceleration provided by the user's device.

2.1.7 Three.js

Three.js is an open source JavaScript library that is used to display the graphics, 3D and 2D objects on the web browser. It uses WebGL API behind the scene. Three.js allow you to use your GPU(Graphics Processing Unit) to render the Graphics and 3D objects on a canvas in the web browser. since we are using JavaScript so we can also interact with other HTML elements.
Since Three.js is open source so we can easily watch the source code and understand the functionality of the code(functions). When we use WebGL for Graphics then it doesn't support most of the browser but Three.js supports most of the browsers. It doesn't required any third party plugin to run the code. We just need to work on only one programming language JavaScript and off course HTML.

2.2 Hardware

The scope of the final application is the main driver in the selection of hardware for AR applications. If we look at hardware specifically developed to tailor AR needs, Head Mounted Displays (HMDs) play a crucial role. HMDs can be classified into two main kinds: Video-See-Through (VST) and Optical-See-Through (OST). The VST technology is based upon the use of a camera that frames upon the external frames. In the following, the virtual symbols or models are added in real-time to the video streaming. The OST devices work in a different way, projecting synthetic virtual models onto semi-transparent lenses: in this case, the user sees the real external world, added by a virtual model. On the other hand, using VST hardware, the user sees the real world in the display of the device. Smartphone applications are typical examples of VST technology. A further classification of AR hardware can be carried out by splitting hardware into two categories based on the processing unit, stand-alone devices include the electronic boards and processors capable of framing the external world, superimposing models, and visualizing the final streaming; other devices require additional hardware to work properly, which must be connected through cables, Bluetooth or Wi-Fi connection.

The simplest XR presentation involves rendering the scene directly to the user's screen, either in the context of a web document, or in full screen mode. This is most common when the user either doesn't have a dedicated XR device, or when the user is viewing the AR or VR app on a phone or other handheld device.

Simpler and lower-priced XR devices typically use an integrated computer or connect to a smartphone, essentially using the mobile CPU and GPU to run apps, render images, and display them to the user. Higher-powered solutions typically offload application execution and graphics processing to an external device such as a desktop computer, and are either tethered to the computer using a cable or use a wireless network to receive the imagery to display to the user.

2.2.1 Gyroscope sensors and Accelerometer

Gyro sensors, also known as angular rate sensors or angular velocity sensors, are devices that sense angular velocity. Angular velocity. In simple terms, angular velocity is the change in rotational angle per unit of time. They are used to track the rotational movements of pitch, yaw, and roll which is an integral part of AR.

2.2.2 Magnetic Sensor

AR apps rely a lot on accurate direction based data. Without a Compass, an Augmented Reality app cannot display information accurately, as it won't know which direction you are pointing to. This may hamper your AR experience. Most AR apps usually don't work at all if the compass is missing on your device.

2.2.3 GPS

Apps that support Augmented Reality generally display results based on location based searches. Without accurate location tracking tools, such as GPS, your augmented reality experience would be quite rough. Most AR apps don't function without turning on location services.

Chapter 3

Problem Statement

We start with the most basic application that AR can do. We want a program which scans QR code provided on a machinery and shows all the necessary stats related to that. This will have implications in most of the industrial sectors because, it's a very simple way to see through the world. It'll also help professionals working in the production industry to find out the bottle necks, to more efficiently plan the production layout. It'll play a very big role when it comes to the maintenance of industry devices or household devices. And since, the future is based on cloud services, it'll be easier to retrieve data from the cloud. The framework works as follows:

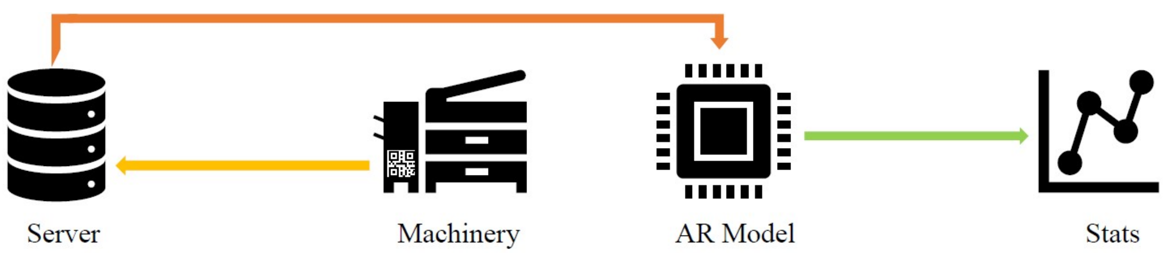

Figure 3.1: AR Informer Framework

Here, as the user scans the QR code on the machinery, it redirects it to the data that has been stored into the server and retrieve the AR Statistics/details of that and shows that as AR. It actually helps to know about details of the machines as well their state details easily and decisions can be taken based on that. If things can be polished bit more, everything can be shown in real-time. In our case, we created two different types of models.
Scan to view any 3d object on the ground:
This can be used for running any simulation in AR to get a better perspective or a real world view of the process.
Scan to view any 3d object floating:
This can be used to visualize data in a more interactive way. Which will help in data analytics to understand the data in 3 dimension.

Chapter 4

Application Development & Deployment

4.1 Development

The fundamental parts of the framework are as follows:

4.1.1 Developing and Importing 3d models

For developing a 3d scene, we first need to construct the light scene using **THREE.Scene** which contains lights that case shadows, and a mesh that will receive shadows. As the materials will render as a black mesh without lights in our scenes, we add an ambient light so our material can be visible, as well as a directional light for the shadow. We need light to cast shadow also, made a large plane to receive our shadows. Created a mesh with a shadow material, resulting in a mesh that only renders shadows once we flip the 'receiveShadow' property using 'THREE.Mesh'. At the end we added lights and shadow material to scene.
After adding the light elements, we created a 3d cube scene where, we wanted to place 3d cube boxes randomly. Where we started by constructing a scene, constructing the materials and constructing 3d boxes using 'THREE.Mesh' with 0.2 unit of length, breadth and height. At the end we added this to the scene we just constructed.
Coming to importing 3d models, we can import many different file types but the majorly used 3d model file types are .glb, .gltf, .obj, .fbx, .stp, .stl etc. Here we imported the .gltf file type using 'THREE.GLTFLoader()'.
All the above mentioned work can be found in the 'utils.js' file.

4.1.2 Adding CSS

'app.css' contains the css file required for the website which contains all the fonts, colors and other basic elements needed.

4.1.3 Developing the application

Here comes the tricky part. We already created the 3d models, light scene and css files required for developing the AR. Now we need to add logic to the whole part.
The idea as a whole is as follows:

- Check for WebXR support because without this the browser will not be able to render the scenes.
- Creating a container class to manage connecting to the WebXR Device API and handle rendering on every frame.
- Initialize a WebXR session using "immersive-ar".
- Create the canvas that will contain our camera's background and our virtual scene.
- With everything set up, start the app.
- Add a canvas element and initialize a WebGL context that is compatible with WebXR.
- All the above mentioned elements will be called when the XRSession begins.
- We set up our three.js renderer, scene, and camera and attach our XRWebGLLayer to the XRSession and kick off the render loop.
- Then we create the function 'onXRFrame' which is called on the XRSession's requestAnimationFrame and with the time and XRPresentationFrame.
- At the end, we initialize three.js specific rendering code, including a WebGLRenderer, a scene, and a camera for viewing the 3D content.

The file containing the codes described above can be found in 'app.js'.

4.1.4 Developing the website

Here comes the final part of developing the website through which we'll be able to access all the above created files and get an immersive Augmented Reality experience. We linked the above files, loaded the required libraries, and created a basic page with a button to enter.
The file containing this is 'index.html'.

4.2 Deployment

For deploying them, we need to host the files we created in the local machine. For hosting from the local machine, we used an application named 'XAMPP' where we selected the path of the file and changed the connection as a secure connection because, other than that WebXr was not loading and was showing unsupported browser.
After finally hosting the website, we created a python file to convert the web address into a QR code so that, we can simply scan the QR code and access the site.

Chapter 5

Results and Discussion

5.1 Results

When we scan the QR code, we land on the following page:

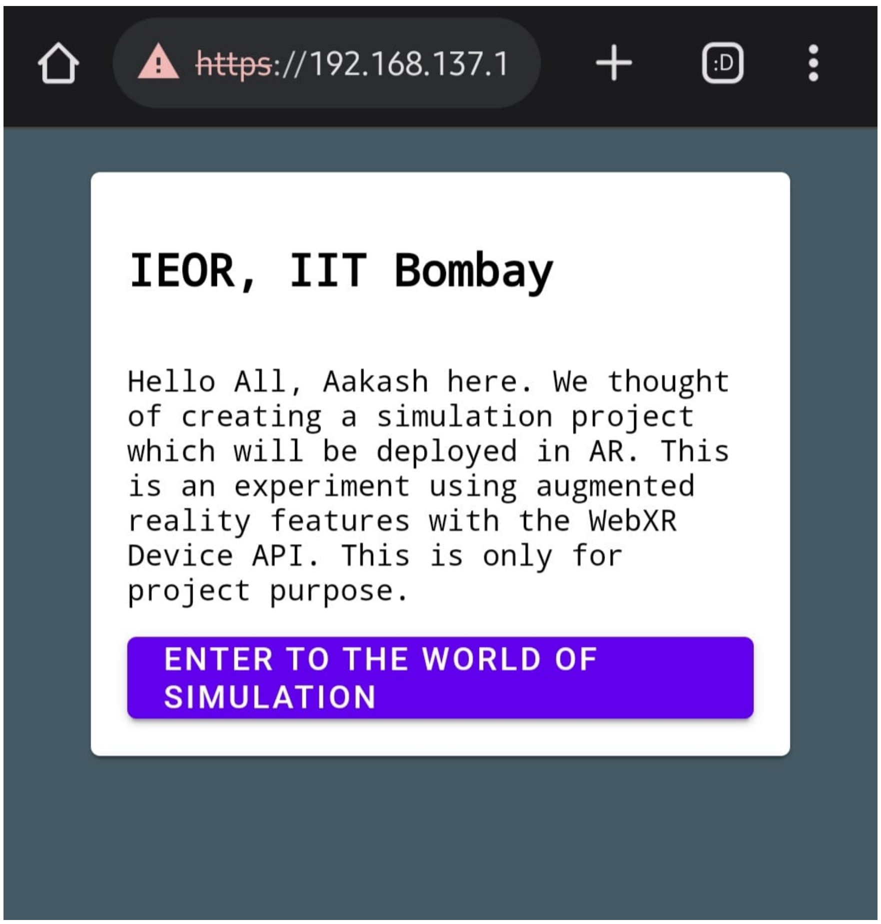

Figure 5.1: Homepage

After this we click on 'ENTER TO THE WORLD OF SIMULATION'.

5.1.1 Generated Cube Scene

Figure 5.2: AR Informer Framework

5.1.2 Loaded .gltf File Scene

Figure 5.3: AR Informer Framework

5.2 Discussion

In the Cube Scene, as we move the camera around, the cubes do not move and they seemed to be at their place as if in the real life. So, we can have it's view from all directions(figure 5.2).
This same was in the case of the imported 3D object in figure 5.3.
Here We have created the framework to work with Augmented reality. We can either work by creating our own scene for example we want plot a 3D scatter plot, view data analytics dash board or by importing already created 3d objects to view them and add extra animations to them. This framework can also be implemented to the server so that, people with internet can access this. This in future will give an extra edge to teaching students or conveying any idea to anyone.

Bibliography

[1] Siltanen, S. Texture generation over the marker area. In Proceedings of the ISMAR 2006: Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality, Santa Barbara, CA, USA, 22–25 October 2006; pp. 253–254.

[2] Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans. Robot. 2015, 31, 1147–1163

[3] 'Augmented Reality in Industry 4.0 and Future Innovation Programs', Gian Maria Santi, Alessandro Ceruti, Alfredo Liverani and Francesco Osti

[4] https://threejs.org/

[5] https://arvr.google.com/arcore/

[6] https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API

[7] https://www.javatpoint.com/em-algorithm-in-machine-learning

[8] https://www.geeksforgeeks.org