An Exploratory Study on Augmented Reality (AR) and Tools

Abstract- *Augmented Reality (AR) is a technology that blurs the line between what is real and what is computer generated. It creates an immersive environment by enhancing what we feel, see, or hear. This report deals with AR from scratch discussing the very essence of this technology. Various AR systems are explored with interesting examples. Furthermore, we see how it has been widely used in numerous fields making things more interesting and easier for both producer and consumer of the services. Software development kits (SDK's) like UNITY and VUFORIA are also discussed in the view of Augmented Reality. Finally, a short experiment on a Marker based AR is performed. It has been seen that unlike Virtual Reality; AR has successfully intensified the visualization method of the user.*

Table of Contents

Introduction

AR has come all the way from being a sci-fi notion to science-based reality and is now the dream of every workplace. It is by far the most interactive way to render the digital data onto the real environment. This technology basically works on computer vision-based recognition algorithms to overlay different data types like speech, graphics, video etc. onto the real world using camera of the device in use [1].

Augmented reality explores two main fields namely **Computer Vision** and **Computer Graphics**. Computer vision when applied to AR includes cluster of image processing techniques like image analysis, object detection, feature detection and tracking, and thereby constructing an environment including different types of sensors. On the other hand, research on Computer Graphics includes the computer generated stereo graphics and animations. Thus, a simple augmented reality system consists of a camera, a computational unit and a display [2].

Birth of AR

The term "augmented reality" has been in existence since 1990. From the moment man started making gadgets and were able to relate to the environment supplying the users with their information, AR was there.

Even though its existence has been in some form or the other, the term "Augmented Reality" is supposed to have been coined by Professor Tom Caudell while working in Boeing's Computer Services' Adaptive Neural Systems Research and Development project in Seattle. While in search of finding an easier way to help aviation Company's manufacturing and engineering process, he began applying virtual reality concepts and eventually landed up devising a complex software that could overlay the positions of certain cables in the building. This meant that the mechanics didn't have to ask or look at the complex diagrams of manuals.

At the same time in 1992, two other teams made a big step into the same technology.LB Rosenberg created what is widely recognised as the first functioning AR system for the US Air Force known as VIRTUAL FIXTURES where the fixtures were described as cues to help guide the user in their task that too in big letters. A second group made up of- Steven Feiner, Blair MacIntyre and Doree Seligmann, submitted a paper on a prototpye system they called KARMA (Knowledge-based Augmented Reality for Maintenance Assistance).The paper went really well and was enormously cited within the scientific community.

Until 1999, what remained a toy for the scientists was then made available to the open source community as an AR tool namely ARToolkit released by Hirokazu Kato. Once ARToolkit was ported to Adobe Flash, the journey began where today AR is possible through the desktop browser and the webcam as well.

Augmented Reality Systems

Marker-based AR system (Recognition based):

AR displays the data in the real-world context. To achieve this, the system needs to know where exactly the user is and what he/she is gaping at. As the environment is not known to the system, the system selects the coordinate axis orientation in a manner that is random and unknown to the user**.**

Figure 1. Marker based Augmented Reality

To overcome this issue, an easily detectable pre-defined sign in the environment is required that enables the system to detect the region where the data is to be augmented. A **marker** is such a sign that enables system to detect it using image processing and computer vision techniques. After detection, both correct scale and camera pose is specified [2]**.** This is the approach of Marker-based AR. Some of the toolkits that form the basis of developing such marker-based AR applications include: ARToolKit, ARTag and ALVAR.

Procedure for Marker Detection:

The main aim is to determine the outlines of a potential marker and then deduce location of marker's corners in the image [2]. It is required that the system verifies if it really is a marker and correctly perceive its identity.

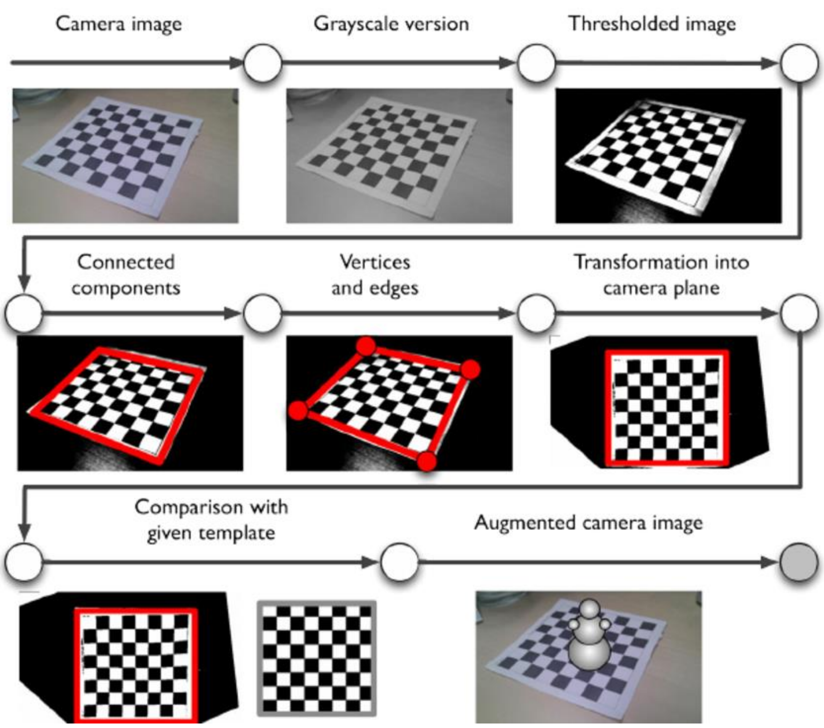

Figure 2. General steps for marker detection

Following are the steps for marker detection:

1. **Image Acquisition-** It is the action of retrieving an image from some source and is completely unprocessed (Hardware-generated mostly) [3]**.** Several tools that are available for this purpose include MATLAB, Simulink and for AR, tools like DSVideoLib and HighGui are widely used.

   Image can be retrieved via: Camera, Image Acquisition Hardware like Desktop scanner and a massive optical telescope etc.

2. **Pre-processing-** This step involves converting image from any given format (for eg. RGB) into greyscale image. Tools available for this include Photoshop tools, MATLAB, OpenCV. Furthermore, it involves the following steps:
   - Low level processing – involves noise reduction, image enhancement and image sharpening. Several filters like Gaussian filter, median filter, Normalized box filter etc. are used depending upon the type of noise encountered (salt-pepper noise, Gaussian noise, Speckle noise). Image Enhancement involves contrast stretching, brightness, thresholding etc. Region of interest (RoI) is separated out by thresholding functions. This separation is based on the intensity variations between background and object pixels. Thus, a comparison of each pixel value with the threshold value is carried out and when separated properly a determined value is set to identify them; say 0(white) and 255(black) or any value that suits the user [4]**.**

- Restoration- This involves applying inverse process to recover the original image from the degraded model. Techniques like inverse filtering, minimum mean square errors (Weiner filtering) are used for this using filter like Mean filters and adaptive filters [5].

- Segmentation- deals with the detection of edges, lines and objects within the image. Several approaches for this include Water-shed Algorithm, Region Growth Approach and edge-based methods which use Sobel Filters, Canny edge filters and Laplace filters.

3. **Detection of Potential Markers-** Markers that fail to be rich in details are rejected by studying various image attributes.
   Characteristics of a Good Marker [6]**:**
   - The detector performs best when image is rich and highly textured, thus it is required that the image be detailed rather than including blocky designs or bold lines.
   - Markers with self-similar sections or repeating sequences are not preferred. Similar patterns at different places cause confusion in detecting the actual region of interest.
   - Visibility of sufficient details at a range of scale is necessary to ensure proper pose estimation and avoiding blurred image detection during fast motion of camera.
   - For an image to be a good trackable image, its details should not be concentrated on one side of the image for the pose to be more stable.
   - Good contrast in the image intensities. High contrast helps to detect image under different illumination conditions.

4. **Identification and Decoding–** This involves identifying the type of marker used for augmentation and its respective decoding.

   Types of Markers:

   - **Template Markers-** These are black and white markers with basic (uncomplicated) image inside a black border. Typically, these markers are identified by comparing them with the marker templates. Marker Templates represent sample images of the marker stored in database [7]**.** This method may result in matching errors or detection of undesired marker similarity.

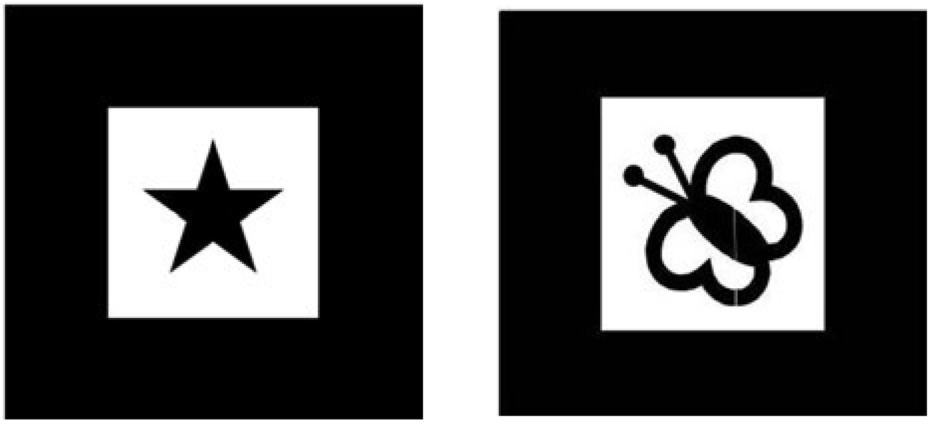

Figure 3. Examples of Template Markers [2].

- **Circular Markers-** These types of markers are conventionally used in Photogrammetry. As the circles are often identical, they require manual or semiautomatic mapping from frame to frame. Thus, such markers are rarely used. Even though the circular code has high accuracy, but a single centre coordinate is not enough to derive camera pose making usage of square shaped markers more practical [7].

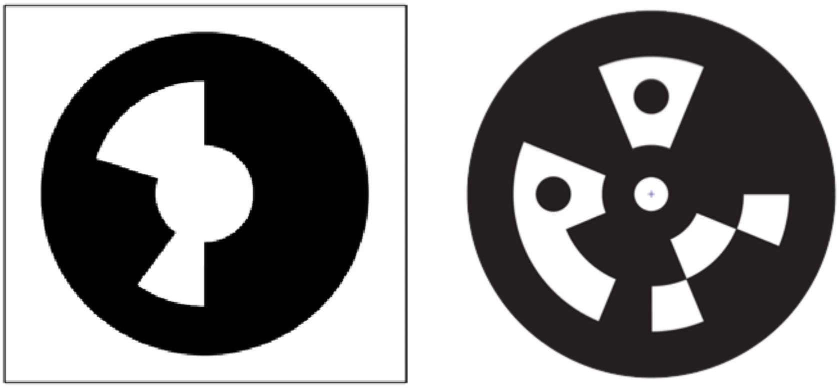

Figure 4. Examples of Circular Markers

- **2D Barcode Markers-** They contain black and white data cells with perhaps a boundary. They can further be defined by identity (ID marker) or containing more data (Data marker). While ID markers are associated with just one piece of information i.e, ID number the data markers contain more data and has error detection and correction. ARTag is one such example [2]. Error detection and correction is done using algorithms like Hamming codes and Reed-Solomon codes. Examples: Aztec code, Data Matrix, QR code etc. Such markers have data coded as "0" or"1" by dividing the marker into multiple regions of black and white squares. For the purpose of decoding, it is required that the centre and the size of each cell be calculated. Then each cell is assigned a binary value so that the whole data of marker is represented as a binary number (marker ID) or a series of binary values.

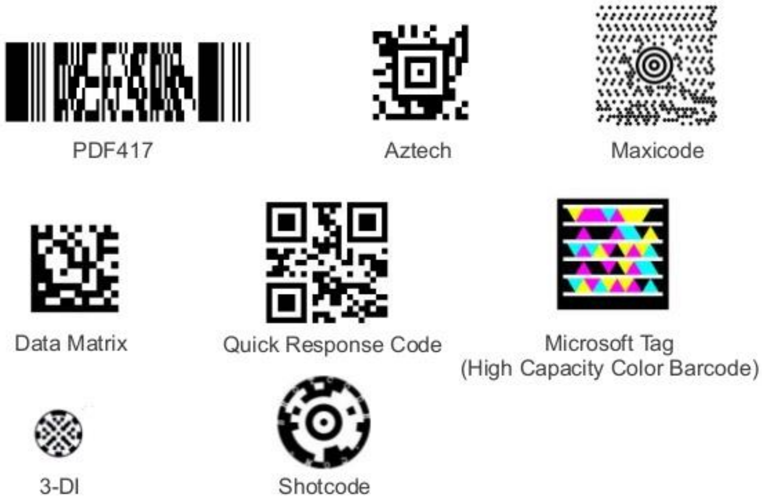

Figure 5. Examples of 2D barcode Markers

- **Imperceptible Markers** [2] [7] **-** If having visible markers is undesired, invisible markers that can only be detected by machine can be made use of. This can be done by making use of markers and detectors operating at wavelengths other than the visible light eg: Infrared markers. The system can detect IR markers with special IR camera**.** Another possibility is to make use of markers that are too small to be detected by human eye.eg: Miniature markers.

Figure 6. Examples of Imperceptible Infrared Markers

5. **Pose Estimation and Camera Calibration-**

Camera pose refers to its location and orientation in world coordinates. Orientations can be expressed with rotation angles (α, β, γ) around the coordinate axes and position can be expressed with translation coordinates (X, Y, Z) along each coordinate axis. The pose of the calibrated camera can be uniquely determined from a minimum of four coplanar but non-collinear points [2]. Thus using the four corner points of the marker in image coordinates, the system can evaluated the marker's pose (relative to the camera).

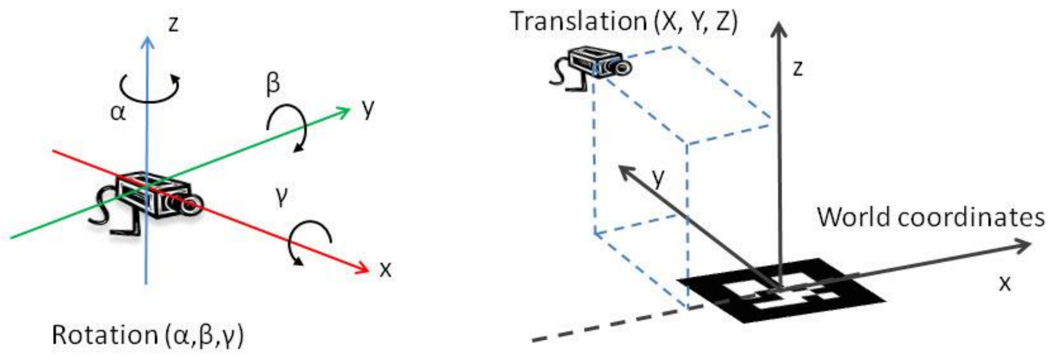

Figure 7. Illustration of rotation angles (on the left) and translation coordinates (on the right) [2]

In an ideal pinhole camera system, all rays pass through an infinitely small optical centre of camera and thereby register the object on image plane. This image is called an *ideal image.*

In digital cameras, these ideal coordinates differ due to variations in focal length, image orientation and size (camera image) and thus we require an additional camera coordinate conversion.

5.1 Camera Transformation

The transformation between camera and marker is given by

$$x = \mathbf{TX}$$

where **X** are points in world coordinates and x is its projection in ideal image coordinates and **T** is the *camera transformation matrix or extrinsic camera matrix*.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Now **T** can be expressed as **T= [R|t]**, a 3×4 matrix that corresponds to the Euclidean transformation. Here **R** represents a 3×3 rotation matrix and **t** a translation.

When a marker is detected, the camera matrix for each frame is calculated by the marker tracking system.

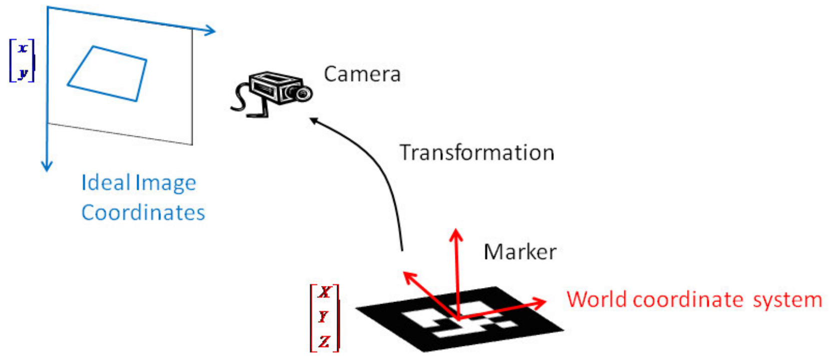

Figure 8. Euclidean transformation from world to camera coordinates [2]

5.2 Camera Calibration

Camera calibration matrix **K** is given by:

calibration matrix

$$K = \begin{pmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

where

- f_x and f_y are the scale factors and are proportional to the focal length, f, of the camera.
- p_x and p_y are the horizontal and vertical displacements of the image from optical axis.
- $c=[p_x\ p_y]^T$ is called the principal point.
- s is the skew factor. Also s=0 & $f_x= f_y$, as present day cameras have square pixels and the columns and rows are straight.

  The camera is thus calibrated when these intrinsic parameters of camera are known.

5.3 Pose Calculation

Assuming no distortion in camera model, points of camera coordinates are labelled as **x** and world coordinates are labelled as **X.**

For four corner points **x** is taken to be $\mathbf{x}_i$ where i = 1, 2, 3, 4. Implies:

$\mathbf{x}_i = \mathbf{KTX}_i$ **.** The product **KT** gives the projection matrix **M**.

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Thus, both intrinsic and extrinsic parameters of camera are calculated. However, errors in pose estimation do occur and this requires a continuous marker tracking system. The stability of such a tracking system for AR can be improved with methods like Kalman filtering (KF), extended Kalman filtering(EKF) and single constraint at a time (SCAAT) method that can predict marker or camera movements and stabilise the tracking results [2].

5.4 Rendering Graphics

The main part of Augmented Reality is the rendering of virtual elements onto the real world as if they are a part of it. For this purpose camera pose is used to determine the right scale and perspective. The virtual camera is then moved to the same pose as that of the real camera and virtual objects are rendered. If the system wants to augment the object in a different pose, it needs to add object transformation T_{object} in the rendering pipeline [2]. Since both the marker and camera are not in static position, the tracking system is required to determine the relative pose, but absolute position is unknown. The system can do so via accelerometer which is an in-built sensor in new smartphones.

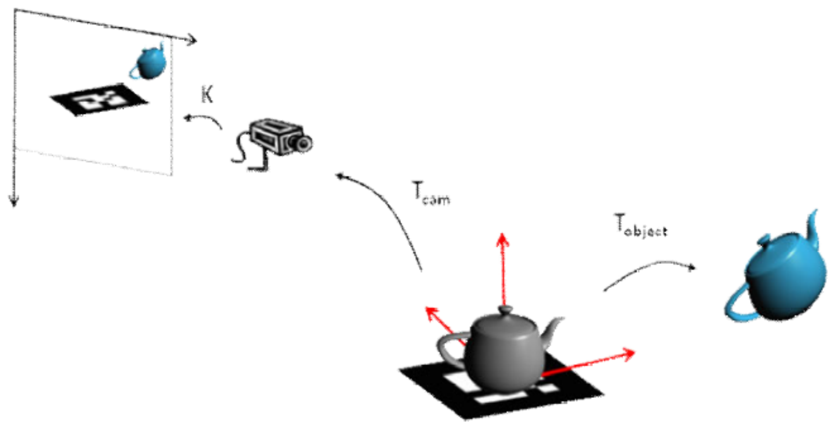

Figure 9. Rendering of Virtual object [2]

Marker-less AR system (Location based):

As the name suggests, Marker-less AR applications deal with augmenting 3D objects in scene without any pre-knowledge of the environment on which the virtual element is to be augmented. Let us take the example of Wikitude and understand how this technology works.

Wikitude is one of the world's leading augmented reality SDK (Software Development kit) that combines 3D tracking technology (SLAM), Image Recognition and Tracking as well as Geo-location AR for various AR applications [8]. Using the camera and device sensors, Wikitude AR browser scans the user surrounding for geo-referenced content. The information about that particular object is then displayed right in front of the camera.

Wikitude enables us to discover all the places of our interest. Each place (or POI, Point of Interest) belongs to provider, a so called “Worlds”. Local Worlds are listed on strt-up, each one providing information of places around you. E.g., check “Wikipedia” to see all Wikipedia information around you. You can then watch it in a List, on a Map or in the Cam (AR-View). Wikitude requires: Compass, Accelerometer, GPS and Camera and is available on almost all devices with the requested hardware [8].

Figure 10. Wikitude - Location Based AR

Applications of Augmented Reality

Without any doubt the spread of AR technology has been phenomenal, and this is indicated by its widespread applications in various fields like medicine and healthcare, education, commerce, advertising, entertainment and Tourism etc.

Medicine

With the help of AR, the entire world of healthcare and sciences can be presented to the user in an AR way. For example, the healthcare provider can install an app on his/her mobile. The app then provides with various medical measures to be selected by the user. Once the healthcare provider selects from the options, the first screen displays where the tracking pattern is to be situated in the patient’s body. Following this pattern application, the training model begins. The training program then shows animated simulations in 3D, indicating precisely when, where, and in what the various manoeuvres should be performed [9]. Examples: AccuVein helps in locating the patient’s vein for injections, Brain Power works on autism spectrum teaching them with life skills etc.

(a) 

(b) 

Figure 11. (a) AR in Medical Field, (b) AccuVein app

Education

To capture the undivided attention of the audience, AR is the best way to enhance the teaching environment. Students will be able to access models via Augment's app and they can later refer to these models when they wish to do so. This will help them to gain better understanding of the concepts. Moreover, they will not have to invest on physical models. With this exciting technology the student will be able to retain the information for longer periods. Examples: Elements 4D is an exciting app that enables students to combine various elements and see chemistry in action. The tutor can print images of the elements and then the student uses AR experience to see the formation of compounds and their characteristics, Math alive is another AR software that the tutor uses, placing the trigger cards in front of the camera for counting and basic numerical skills.

(a)

(b)

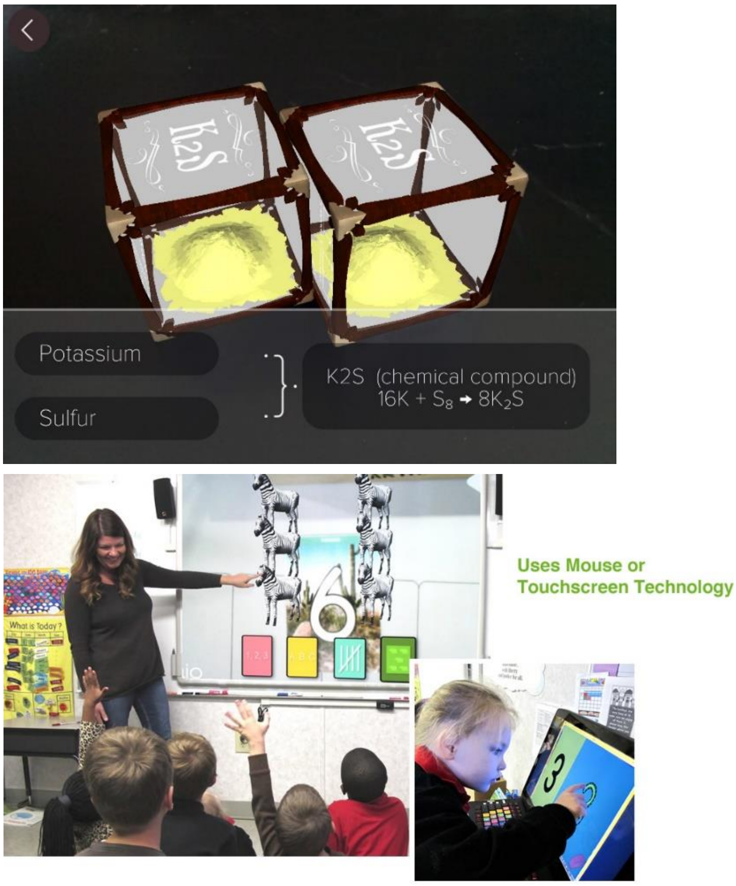

Figure 12. (a) Elements 4D app, (b) Math alive app

Commerce

E-commerce is another industry that has positively taken up AR technology to benefit to great extent. Specifically with online shopping the user is unable to predict if the cloth fits him or not. Hence, individuals start to connect AR with electronic shopping, e.g., clothing shopping [9]. Also, product shopping is another domain taking up AR for its benefits. The customer can try on different types of products say Spectacles and take better shopping decisions.

Figure 13. AR in e-Commerce

Travel and Tourism

Planning a perfect trip requires careful planning to ensure that your travel, stay and entertainment are all reserved way ahead of time. Such planning activities can be tedious and confusing. With the help of AR the tourist can take virtual tours of the hotels and check out amenities like pool, restaurants and bars. In this way the user experience is enhanced. Examples: ViewRanger is one such app that helps user to navigate outdoor adventures. For those who are interested in exploring mountain peaks or simply have a day out in the countryside, this app is perfect. Hudway is another app that allows driver to focus on road by guiding them through the streets.

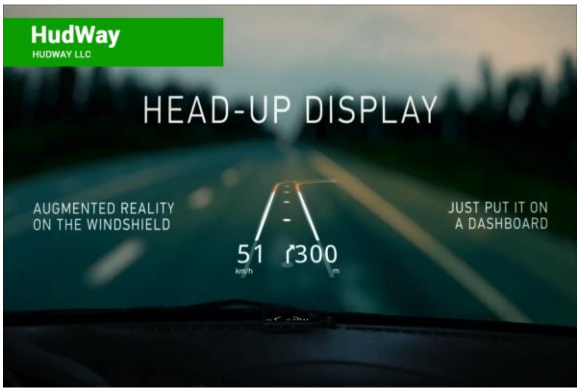

Figure 14. Examples of AR in Travel and Tourism

An Overview on Unity and Vuforia SDK's for AR

Unity

The question that first comes to our mind is that what is unity? Why do we need to know about it? How does it work? And the search goes on. In simple words, Unity is an ultimate game development platform extremely famous for the depth and quality optimizations that it provides to the user. It is feature rich and highly flexible editor. What is more interesting to know is how is this unity engine implemented?

Once you get the feel of how amazing the tool is you want to know how they deliver such graphics with great performance. The core of Unity engine itself is written in C\C++. All the graphics, physics, sound is coded mostly in C++. Another question that now arises into our mind is why then we program in C# when all the coding is done in C++ under the hood. The answer is: Wrappers. Wrapper function is a subroutine in a computer program or a software library. Over here it provides us with a layer that enables cross language interoperability. Thus you can call a C\C++ code from C#, Java, Python or Perl programs. Unity Editor is one such user interface (UI) that helps us to build games using Unity engine.

Unity Editor

The Unity Editor provides an ASSET folder that is the representation of any item to be used in the game or the project. An asset may come from a file created outside of Unity, such as a 3D model, an audio file, an image, or any of the other types of files that Unity supports. There are also some asset types that can be created within Unity, such as an Animator Controller, an Audio Mixer or a Render Texture [10]. All that you want in your Project Window must be included within the Assets Folder. Also, the user can import 3D models, animations, sounds, textures and much more from the ASSET STORE. These are downloaded and directly imported into your project.

The Unity has several Main Windows. The project Window is the cluster of all windows of project. The Scene window is the platform where the user creates his/her own world. The Inspector window displays detailed information about the currently selected GameObject, including all attached components and their properties, and allows you to modify the functionality of GameObjects in your Scene [10].

While creating your game you will require Scene that includes all the objects of your game. Within each scene different environments, obstacles etc. can be placed. Rotations in 3D objects are represented in either Euler or Quaternions angles. Internally it uses Quaternion angles but displays values as Euler angles for the ease of user in editing.

While creating any new project the user can specify to operate the unity editor in either 2D mode or 3D mode. This switch can also be done at any point of time. In order to customize the behaviour of unity editor, a number of PEFERENCE SETTINGS are available. BUILD SETTINGS enables the user to decide the target platform on which they want to run their application be it PC, Mac& Linux standalone, Android or an iOS. SETTINGS MANAGERS influence the overall functionality of the editor such as Graphics, Audio, and Physics etc. The ASSET STORE of Unity editor provides the user with varieties of 3D objects and characters with animations for augmenting on the target. Thus, unity engine is most loved in gaming industry.

Vuforia

This is yet another AR SDK that helps in developing AR applications. It uses Computer Vision technology to decipher and track planar image targets and simple 3D objects. It also contains the prefabs and scripts required to bring augmented reality to your Unity application.

Image Targets represent those images that the Vuforia SDK detects and tracks. The Image Targets need not be traditional black and white regions or codes like Data matrix or QR codes. The SDK detects and tracks the features that are naturally found in the image itself by comparing these natural features against a known target resource database [11]. As long as the image is within the camera's field of view it is tracked by the SDK. Common uses of Image Targets include augmentation of printed media and recognition. Image Targets can be created using Vuforia Target Manager using JPG or PNG images (RGB or Grayscale) and the upper limit to the size of image is 2MB. The features extracted from these images are then stored in the database which can then be downloaded and packaged with the application like Unity Editor Engine.

Working with these Target images includes two phases: First is to design your image target and second is to upload them to the Vuforia portal for further tracking and processing. The image is then rated accordingly depending upon how rich features it possesses. When talking about User-defined targets, they are target images created at the runtime from camera frame selected by the user. As a result, the user can experience AR anywhere at any time without having to carry any pre-defined targets. The process of capturing, building, and tracking a User Defined Target is managed by your app logic using the Vuforia API [11].

Short Experiment working with Unity and Vuforia AR SDK's

Marker based AR Experiment

I. Firstly, I decided Image Target on which my 3D object would be augmented. For this I selected an image of a maze (black and white) and this would act as the origin.

II. To get started, an account on Vuforia was created (Vuforia's Developer Portal). Once logged in, I created a License Key for my application (Develop ≫License Manager). After selecting "Add License Key", the application name was entered as "VuforiaBasics2". On selecting "VuforiaBasics2", the license key was generated which was later used in Unity (see fig.15). Database and Image Target were created (see fig.16).

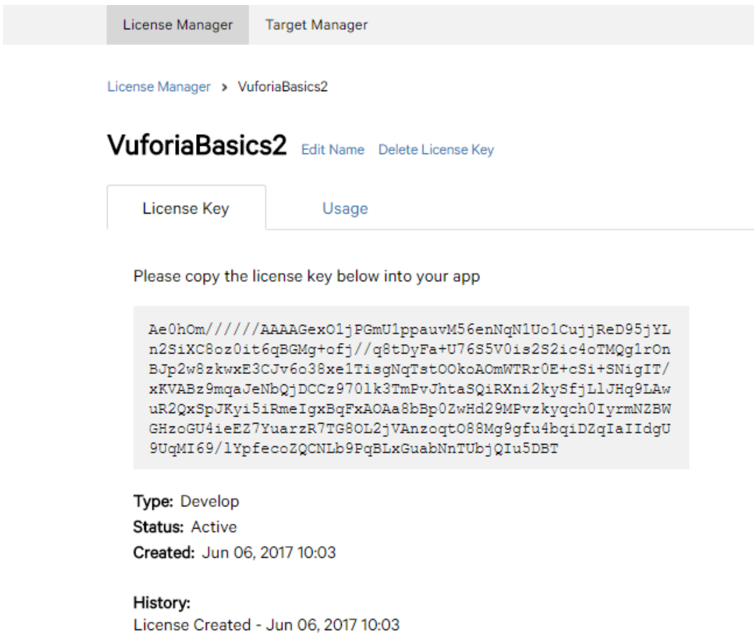

Figure 15.

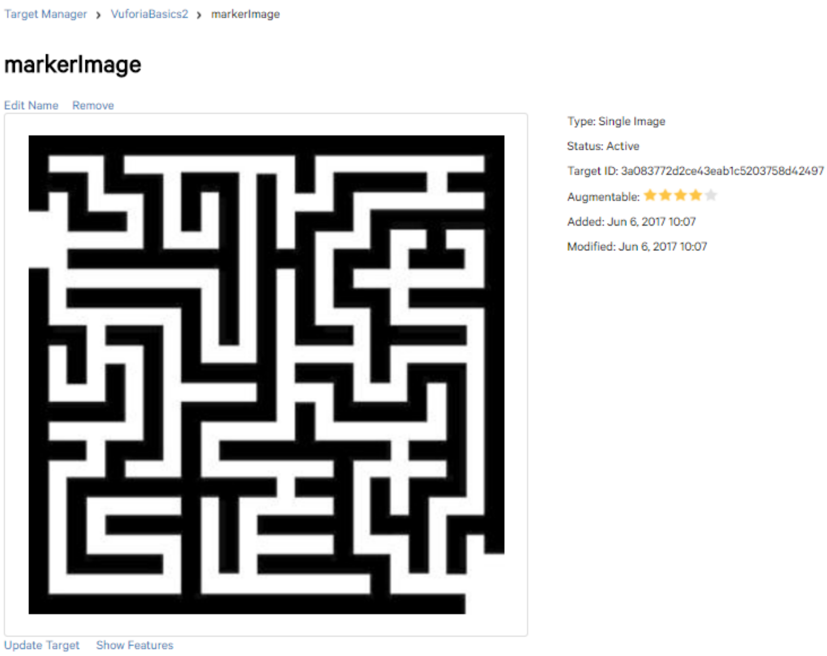

Figure 16.

III. Further the dataset was downloaded into Unity (Download Dataset ≫ Unity)

IV. Integration with Unity required creating a new project and importing the Vuforia Unity packages (Assets ≫ Import Package ≫ Custom Package). Two packages one corresponding to the dataset and the other corresponding to the Vuforia SDK for Unity were imported.

V. Now the scene had to be created. ARCamera prefab from Vuforia was dragged into the scene (Main Camera deleted). For this camera to work, App license Key was added. In the inspector panel, the unique key generated using Vuforia was pasted. At this point, I was able to see my WebCam feed in the Game View by pressing on Play button in Unity Editor.

VI. Now the Target image was added into the scene (Assets≫ Vuforia≫ Prefabs). In the inspector Panel, under the “Image Target Behaviour (Script)”,changes were made to add the image.

VII. As the base was set, it was time to add the 3D object. A simple 3D sphere was chosen for this purpose (GameObject≫ 3D Object≫ Sphere). It was made a child of the Image Target (Parent).

VIII. In the Inspector Panel, under the Datasets, the “Load VuforiaBasics2” checkbox was selected to activate it.

IX. The Scene was then saved and ready for augmentation.

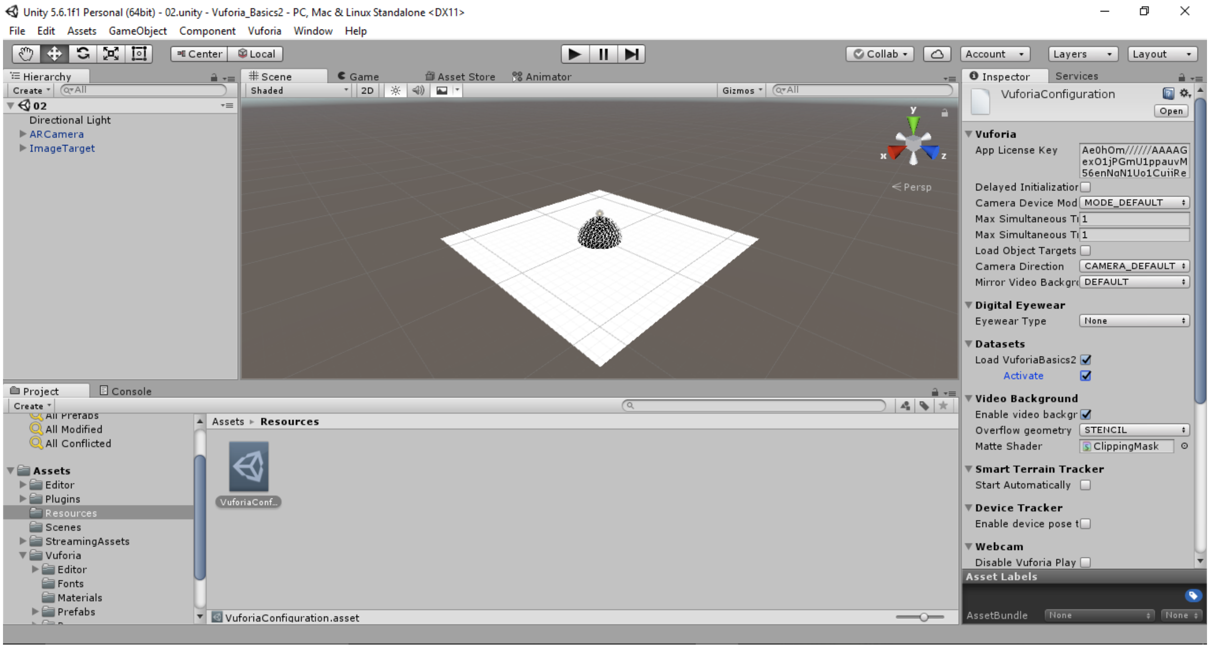

Figure 17. Final Scene ready for Augmentation

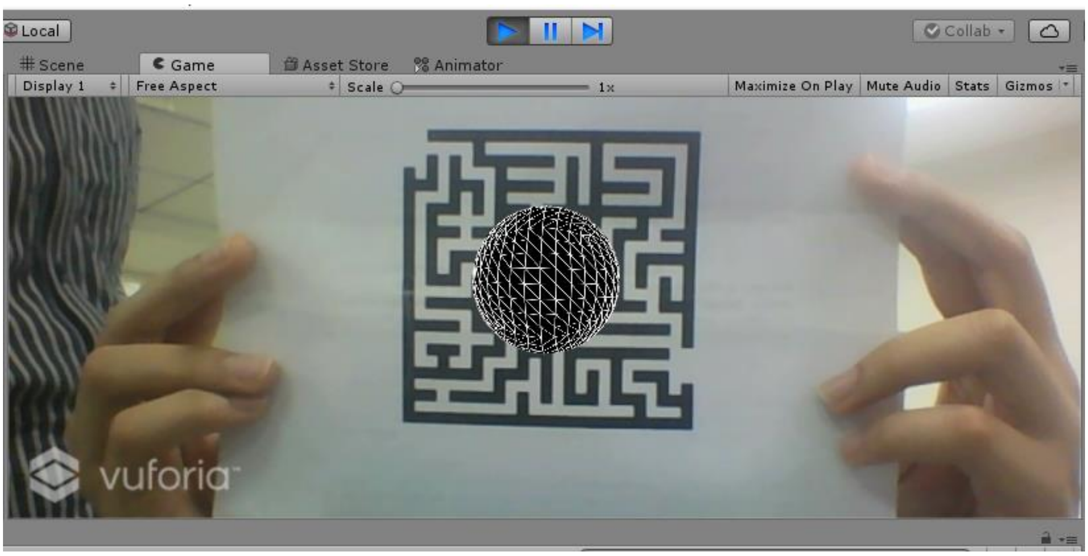

Figure 18. Simple Marker Based Augmentation of a Sphere

Conclusion

Augmented reality has been in existence for quite a long time, but it has gained immense popularity in the recent past. There seems to be no industry who is not interested in exploiting this new technology for the human good. The idea itself is fascinating not only to the AR app developers but also to its users. Planning for success is very important while dealing with emerging technologies and this technology is looking for increasing customer engagement, reduction in cost and boosting sales. Clearly, we are entering into a new reality, a reality that is mutable, malleable, and highly personalized and is completely designed and operated by us. From wearable computers, Smart Browsers to intelligent systems this technology is extending human capabilities and giving us superpowers. Thus, it is the technology of future interest and will soon be a complimentary part of our lives.

References

[1] https://www.3pillarglobal.com/insights/augmented-reality-introduction-and-its- real-world-uses

[2] Theory and applications of marker-based augmented reality - VTT http://www.vtt.fi/inf/pdf/science/2012/S3.pdf

[3] http://www.wisegeek.com/what-is-image-acquisition-in-image-processing.htm

[4] OpenCV library -http://docs.opencv.org/master/db/d8e/tutorial_threshold.html

[5] https://www.slideshare.net/kalyanacharjya/image-restoration-40589017

[6] Augmented Reality Fundamentals: Markers | Kudan https://www.kudan.eu/kudan-news/augmented-reality-fundamentals-markers/

[7] IJESRT JOURNAL - AUGMENTED REALITY MARKERS, IT'S DIFFERENT TYPE , CRITERION FOR BEST FIDUCIALLY MARKER AND NECESSARY REQUIREMENTS TO SELECTING APPLICATION ORIENTED MARKERS -Vipulkumar P.Chauhan , Dr.Manish M.Kayasth Department of Computer Science, Veer Narmad *South Gujarat University*, India

[8] https://www.wikitude.com

[9] Abrar Omar Alkhamisi, Muhammad Mostafa Monowar "Rise of Augmented Reality: Current and Future Application Areas" International Journal of Internet and Distributed Systems, 2013, 1, 25-34 Published Online November 2013 (http://www.scirp.org/journal/ijids) http://dx.doi.org/10.4236/ijids.2013.14005

[10] https://docs.unity3d.com

[11] Vuforia Developer Portal - https://library.vuforia.com