A LAYERED FRAMEWORK FOR VIRTUAL GUIDANCE TO NETWORK MAINTENANCE BASED ON AUGMENTED REALITY

Prof. Dr. Syed Akhter Hossain[1], Istiaque Ahmed[2], Rishad Islam[3], M Shahir Rahman[4], Shahriar N

Abstract—**With the growing interest in Augmented Reality (AR) researchers and developers are engaged in developing systems with AR technologies. AR provides a unique experience in visualizing the solutions and provides an exciting way of interacting with the real world. The network management system is a complicated process that can be addressed efficiently by AR technologies. The visualization of network topology information of network devices decreases the burden of the network administrator to a great extent. In this paper, we proposed frameworks for both Platform Independent Model and Platform Specific Model which will be acting as a guide for the development of an application based on AR. We further developed a prototype for visualizing the network device information using the proposed framework. The prototype is acting as Proof of Principle (PoP) of our proposed framework. This prototype can be fully developed into a working application and installed in any organization with a network system as a guide for network device management and maintenance. The framework can also be modified to support various types of AR in different systems as per the requirement of the developer.**

I. INTRODUCTION

Nowadays, AR applications are becoming the backbone of the education industry. Apps are being developed which embed text, images, and videos, as well as real-world curriculum. Printing and advertising industries are developing apps to display digital content on top of real world magazines. Augmented reality displays superimposed information in our field of view and can take us into a new world where the real and virtual worlds are tightly coupled. It is not only limited to desktop or mobile devices, different wearable computer devices like Google Glass, Microsoft HoloLens with an optical head-mounted display, are perfect examples of using AR technologies [1].

Network management is a vital part of the digital economy. They are fast-moving environments with high demanding reliability and availability requirements. Especially in the data center, there are applications like connecting cables, network connectivity, power distribution, storage units, etc. which are running all at once. Also, the system produces a huge amount of data from security tools and network monitoring tools. Nowadays all applications and activities are based on providing service in the least response time. As AR applications are now gaining popularity in the industries for the ease of development we would like to incorporate this technology to manage the network designs and its maintenance.

The goal of this research is to bridge the fields of Augmented Reality and Network Maintenance by developing a framework along with a prototype and demonstrate the benefits of using an Augmented Reality interface in this particular field.

II. RELATED WORK

In recent and past years, Several studies have been conducted on network device identification and management system to resolve the issues related to the development and managing the network devices with a key focus on Augmented Reality(AR). Using the AR is the key concept to bring a newer shape in the maintenance system.

In [2], Nishino et al. proposed an AR assistance system for Campus Area Network management. In this proposed system, which used a marker-based AR system for detecting network devices. In case a large network system marker-based AR system sometimes posed some kind of problem for installing distinct markers for every device on the network. But naturally, the campus Area Network was relatively small. In this case, marker-based AR proved useful. We had to store the information beforehand into a Database or server for displaying the information.

After this pioneering work from [2] further emphasis was given in this domain to improve the efficiency of the system. Haramaki et al. [3] proposed a system that will assist network administrators using an AR platform in their HMD device. They use a marker for identifying the network devices for bridging the gap between the network devices and their internal information. The visualization contents like IP addresses and VLAN configuration information were stored in a server named information management server. The system is configured beforehand by registering all network topology information in this server. They later introduce a marker-less system for reducing the load. This system provides a hands-free mobile operation environment and helps the user to concentrate on the administration tasks in a real machine room.

In [4], the authors modified the current system by using the Wi-Fi signal. The user will simultaneously sense the Wi-Fi signal from his current location and thus the system will identify the user's current location at any time. Both marker-based and marker-less AR have been introduced and implemented for the detection of devices. Though implementing marker-less AR requires much more complex programming, it provides better flexibility for the client for performing their tasks efficiently. So there is a trade-off between marker-based and marker-less implementation of the system.

So far we have seen that the client or network administrator has to store the topological information of a network device on the information management server beforehand and after the device have been recognized either by marker-less or marker-based recognition method, then it has to look up for the required information in the database. Maintaining and updating an information server includes an added burden as the topological information of a server changes quite frequently.

In [5], Flinton et al. proposed an AR system where we could acquire the information from the detected device. After detecting a device by a specific marker on that device it uses SSH or Telnet command to retrieve the network information of that device. That information is then displayed over the real-time image of that device.

We have observed that various researchers tried to implement the network management system differently. Each framework had its limitations and strengths. But no one provided a layered framework to fit in any type of AR technologies for network management which would help the developers in implementing the system easily. The spectrum of users will also increase by this method. Another challenge of this method is maintaining an information server and connecting the server with the AR application. A dynamic update and maintenance would reduce the burden of the network administrators by a huge margin.

III. Proposed Framework

After an extensive study on the design and development of network management system architecture and framework with the help of AR, we analyzed some limitations of the prevailing frameworks. We have proposed a layered framework to mitigate the limitations of the previous frameworks and which will help in the smooth operation of AR technology. Firstly we designed a framework that can fit into any problem raised or in any research area and provide a solution with AR technology.

A. Framework for Platform Independent Model

AR technology has the potential to help find a visual and easily understandable solution to many functioning systems. Keeping that in mind we designed a framework for Platform Independent Model (PIM). A platform-independent model (PIM) in software engineering is a model of a software system or business system that is independent of the specific technological platform used to implement it [6].

For the smooth integration of the AR technology, there are some requirements that the system and clients need to ensure. Firstly, the client needs to understand the problem statement and need to focus on the subject domain with which need to be integrated into the AR technology. Alongside this, we need to specify the work environment in which the system will perform. Another important requirement for the framework is to specify the

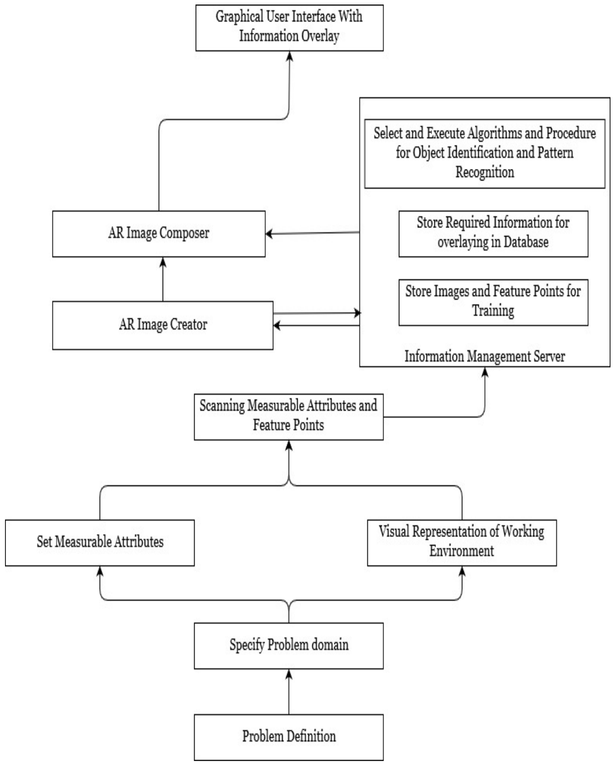

Fig. 1. Framework for platform independent model

measurable attributes in the environment by the system administrator or client.

For creating a visual representation of the working environment we need a camera to capture the video stream of the working environment. This operation can be easily done by the smartphone cameras which is readily available to most users.

Users then need to set various features which will be detected by the system after we have access to the visual representation of the environment. These attributes can act as the trigger or marker which will initiate the AR operation of the system. Instead of a marker, it can be a specific object in the environment which will be detected by the system.

Then the objects which are required to be scanned by the system are scanned by the camera. By scanning, we imply extracting the feature points of the object. The scanning process is based on the Image registration process. The next step in this process is using the data obtained in the previous step while detecting feature points to create a real-world coordinate system. If there is partial information about the geometrical coordination in the image space then Simultaneous Localization Mapping (SLAM) algorithm is used to map relative positions. If we do not have any information about the scene geometry then the SLAM algorithm will not work. In that case, we have to undertake the Structure from Motion (SFM) method like Bundle Adjustment. These complicated processes are done by various AR SDK which performs the back-end calculation and execution of these algorithms to present us an object with detected feature points.

After scanning the object we need to store the feature point information in a database or server. There are many methods for executing this process. The most common method is using the database of SDKs which have the service of saving the targets or objects. But this internal database saves only the object information and sometimes that is not sufficient. Because some applications may need to access information from a different database to access the information for displaying on the virtual display. So this integration is done using PHP and C# code in the engine where we are performing the application.

Next step in this framework is related to augmenting the real world image with virtual information. Once the application recognizes the required object accessing the information from the database it starts the execution of the AR application as per the requirement. It starts accessing the information required to display from the database and places them in the image space. The system administrator will define the position of virtual information to be displayed.

After creating the augmented image it is rendered in the engine and we then have a view of virtual information over live video streaming of the working environment. This rendering may take some time but excessive information will generate higher response time which will diminish the usefulness of the application.

The system administrator needs to develop a user-friendly GUI to display the information. There should not be an overload of information which will create confusion among the user. The information should be precise such that the user can easily understand the information the application trying to convey.

B. Framework for platform Specific Model

For developing an AR app for Network Device Management we proposing a platform-specific layered framework. We divided our framework into four layers: **User Layer**, **Analytical Layer**, **Cloud Layer**, **Augmentation Layer**. Each of the layers here has different functions. We have differentiated each of the tasks in different layers for the ease of understanding and visualization. Figure 2 represents our proposed platform specific model.

The user layer is responsible for executing the local and user side operations. Multiple AR devices such as mobile, tablets or Head Mounted Display (HMD), simultaneously start with sensing the real environment, producing raw videos, and capturing user's gestures via their cameras and sensors. The video streams contain the raw video data which is further duplicated into two copies with one stored in the up-link cache for data transmission and the other stored in the local cache for subsequent processing. This raw video data will be sent later on the analytical layer.

The analytical layer plays a critical role in AR applications. As it receives the uploaded data from AR devices, the raw videos are then converted into several frames through an image processing module. The clipping module slices one representative frame (or image) from each raw video for subsequent processing. Features of various interests are being extracted from these frames for the next step. Once an image is an input into the feature extraction and matching module of the analytical layer, the feature extraction algorithm will immediately search its inherent salient interest points, which are used to estimate the similarity between this image and the standard images pre-stored in the databases. This procedure can be done via some well-known machine learning algorithms, such as convolutional Neural Network (CNN) and Support Vector Machines (SVM). After obtaining the best-matched standard image, the result will be sent on the augmentation layer for further processing.

The cloud layer contains a large cloud database for storing the additional data that are not cached in the analytical layer due to its limited memory size. It contains all the information about the network devices along with the information of the target/marker. Based on the information given by the analytical layer the cloud layer search for necessary and relevant data of the image and pass that information in the augmentation layer

The augmentation layer provides the user with the final augmented model. It collects necessary information from the cloud database i.e. relevant information, 3D model files, etc. Then this layer superimposes the virtual object in the real-time raw video with the required information and sends it to the down-link cache in the user layer. The down-link cache renders the video and shows the output in the display to the end-user.

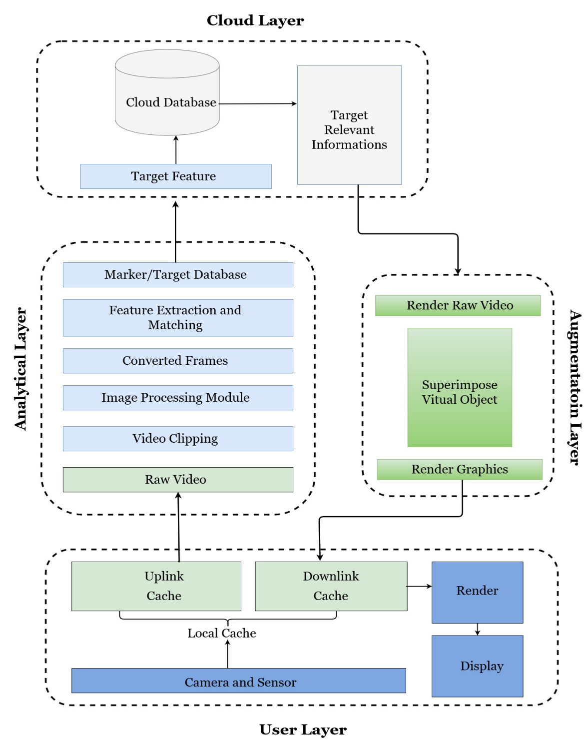

Fig. 2. Layered representation of platform specific model

C. Proposed Application Model

For application development, we will focus on the basic modules based on the layered framework discussed earlier. In this section, we have shown the basic workflow/system architecture at run-time of our proposed application model in figure 3.

In figure 3, a camera-equipped device (left top) reads a video stream that is rendered as a video background to generate a see-through effect on the display. The camera device will be placed in front of the network device which the user wants to identify. Different tracking modules are available for tracking devices precisely. In the proposed model we selected Vuforia as a tracking module because it incorporates computer vision technology to recognize and track planar images and 3D objects in real-time which is fast and accurate.Tracker can load and activate multiple data-set at the same time which contains the computer vision algorithms that detect and track real-world objects in-camera video frames. Device Database to store marker targets in the device itself and lastly cloud database which stores the target in the cloud. After tracking and detecting the required object is completed, data are sent into a classifier that is used for the learning process to classify new records (data) by giving them the best target attribute (prediction). These data are further forwarded to a network topology database.

User has to provide all the information needed to be displayed into the topology network database on the server before the operation in the Network Topology Database. The recorded data fall into two types of records: physical topology and logical topology information. The physical information includes the individual information of each network device such as MAC address, machine name, manufacturer, model number, and device type (Router, L3 switch, or L2 switch). The Device Attribute Information includes IP address, VLAN configuration, location, Feature ID assigned to the registered devices.

For augmenting the virtual object with the real environment all these data are sent to Unity 3D. It is a powerful cross- platform 3D engine and a user-friendly development environment. Unity SDK supports C# interface and acquiring all the information it superimposes the virtual object with the raw input and after rendering graphics it shows the output in the display.

IV. Developed Prototype

After proposing and discussing the frameworks for both platform-independent and platform- specific model we worked on a prototype for establishing the foundation of our conceptual framework. A Proof-of-Principle Prototype (PoP) also known as Proof of Concept (PoC) is a type of prototype which is used to verify some key functional aspects of the intended design, but usually does not contain all the functionalities of the final product.

A. Marker-less Prototype

We scanned a router as our network device. For scanning the router at first we used an android application developed by Vuforia named Vuforia Object Scanner.For scanning, the router few conditions were fulfilled for the successful completion of the scanning. The object was scanned under moderately bright and diffuse lighting avoiding any direct lighting. Because scanning objects with reflective surfaces under direct lighting can introduce areas with no tracking points. During the scanning session, the Object Scanning target defines our Object Target's position and orientation relative to the origin of its local coordinate space. A printable Object Scanning target is included in the Vuforia Object Scanner download package. Thus we have scanned our required object and upload the required *.od file in the Vuforia database. This is the process of detecting the object for the implementation of Markerless AR using the Vuforia platform. Figure 4 and Figure 5 demonstrate the outcome of our developed prototype.

Fig. 5. Port identification (Marker-less Approach)

B. Marker-based Prototype

In Marker Based AR we used the image in Figure 6 as a marker for identifying our device, which is attached to our object. Vuforia has a target database where we can upload our marker which can later be detected while capturing the live stream of video of the working environment. So, first, we uploaded our image to the target database.After the integration of Unity 3D and Vuforia, we downloaded the marker image which was uploaded in the target database in the Unity editor. In the Unity editor, we designed the text,

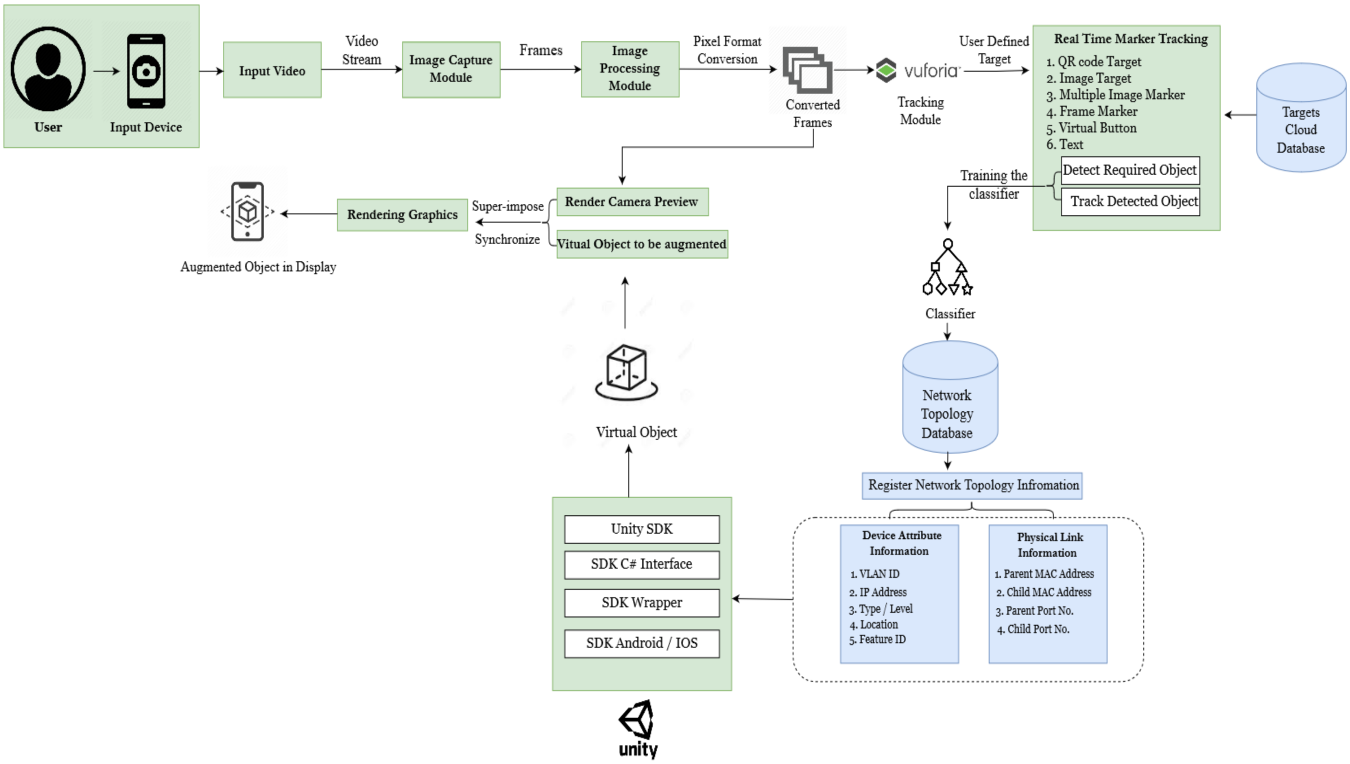

Fig. 3. Proposed Application Model

Fig. 4. Physical information of a network device (Marker-less Approach)

cubes, and lines which will be required to show the different ports of the router. Then, we augmented the router's different ports concerning that marker and placed the designed virtual information in desired places of the screen. After that, we built an application using Unity 3D and C# language in Visual Studio code that runs on the smartphone and displays our required virtual information. Using this app, we could recognize the router's different ports. Figure 7 shows the prototype of marker-based approach.

Fig. 6. Target image

Fig. 7. Port identification of a network device (Marker-based Approach)

V. Conclusion

Our objective was to build a framework that can act as a guide for developing a Network Maintenance system based on AR. We proposed frameworks for both platform independent model and platform specific model and showed the implementation concept of developing systems following that framework. The developed prototype was a Proof of Concept (PoC) for our proposed framework. The main purpose of our prototype was to show that following the steps of our framework we could build a system for network management with AR technologies. This technology eliminates the needs of a maintenance operator for assisting others in network issues. This technology can provide the detail information of the network devices to any user so that they can alone handle situations may arise in data centers. Our research has some limitations the framework which was not was not implemented in a real distributed network. The system was not tested by the technical persons. Therefore the friendliness of the system to users was not measured. This also warrants a real-time simulation. We will further improve our system by integrating a database with our developed prototype. We can also use the system for detecting any error in network connection and indicate the point of error which will reduce the time required for error detection by the network administrator. We can measure the operating performance of our application by involving the user in a real environment. In the future, we will emphasize on user acceptance of our application. The frameworks will act as the basis of further improvement in incorporating more AR functionalities for network management.

References

[1] Wikipedia, "Augmented reality," Available at https://en.wikipedia.org/wiki/Augmented_reality.

[2] H. Nishino, Y. Nagatomo, T. Kagawa, and T. Haramaki, "A mobile ar assistant for campus area network management," in *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*. IEEE, 2014, pp. 643–648.

[3] T. Haramaki and H. Nishino, "A network topology visualization system based on mobile ar technology," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*. IEEE, 2015, pp. 442–447.

[4] ——, "A device identification method for ar-based network topology visualization," 11 2015, pp. 255–262.

[5] C. Flinton, P. Anderson, H. P. Shum, and E. S. Ho, "Netivar: Network information visualization based on augmented reality," in *2018 12th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*. IEEE, 2018, pp. 1–8.

[6] Wikipedia, "Platform-independent model," Available at https://en.wikipedia.org/wiki/Platform-independent_model.