

AUGMENTED REALITY SYSTEM FOR ROUTE HIGHLIGHTING

Valeriu Manuel IONESCU¹, Ene George ALEXANDRU²
University of Pitesti, Department of Computer Science, Pitesti, Romania
¹valeriu@ieee.org, ²alex_ene_97@yahoo.com

Keywords: Augmented Reality, Raspberry Pi, Unity, mobile device, Android

Abstract: This paper presents the implementation of an Augmented Reality system that enhances a mobile robot control system. The mobile system uses the Raspberry Pi3 platform and an Android mounted device for sending a video stream to a PC in order to be remotely controlled by a user. The application running on the mobile system will recognize markers in the video stream and replace them with virtual 3D objects, created using the Unity platform. The purpose is to help the user to control the mobile system by having additional information offered by Augmented Reality.

1. INTRODUCTION

The term of Augmented Reality (AR) has long been used, but it did not become widespread until Android and iOS, smartphones were equipped with GPS and high-performance cameras that have AR capabilities. It is considered an extension of the Virtual Reality[2], a space where the user can surpass the limits of physical reality. There is a variety of development kits for AR applications dedicated to Android and iOS platforms. If you want to see what a merchant's furniture will look like in your room before you buy it, it is now possible with an AR app. For example Nissan in the 2008 LA Auto Show [1] and Best Buy (in 2009) [3] have used AR to showcase their products.



Fig. 1. Best Buy AR application

The number of AR applications for mobile devices has increased and automobile vendors are particularly interested in these applications.

The automobile industry navigation systems benefit greatly from AR as it can enhance visibility in low light conditions but also help drivers with on screen HUDs that display useful information regarding speed, traffic and road conditions.



Fig. 2. BMW Head-Up Display

In entertainment industry, AR is used to create museums and expositions because it offers to possibility to change the attractions displayed with minimal const. This is the case of the Universal Studios Orlando resort [4] as seen in Fig. 3. This shows that Augmented Reality has applications in many areas, such as marketing, automotive, advertising, medicine, military and continues to expand to other new areas.



Fig. 3. AR in Universal Studios, USA

In this paper an Augmented Reality system is presented that can be used in automobile industry in order to highlight characteristics for recognized markers (that can be in a real world application road signs) for novice drivers in order to learn more about them.

The 3D models displayed when the markers are detected were created in Unity and the mobile device where the camera will be mounted is a Raspberry Pi remote controlled robot. Unity is a platform launched in 2005 for creating 2D/3D models for use in simulations and Virtual Reality. For this paper the API used is C# that allows user interaction and network communication.

The reminder of the paper will present the general system architecture, then the implementation details and finally the problems that were encountered during the system development.

2. SYSTEM ARCHITECTURE

The system's objective is to make an Augmented Reality application that will run on a mobile device that is controlled from the laptop.

The system has the following components:

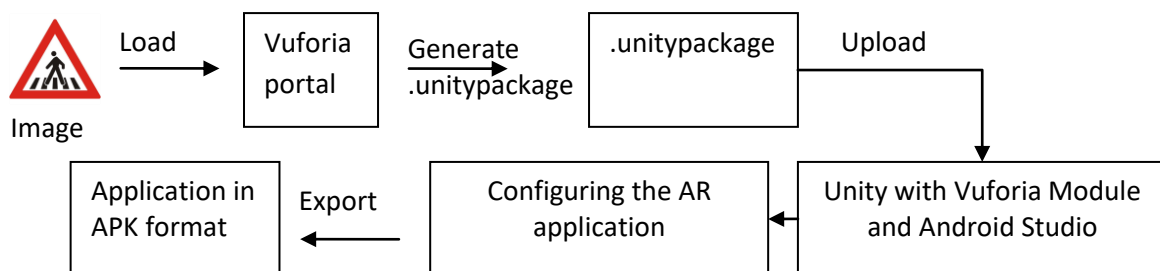


Fig. 4. Creating the AR APK using Unity with the Vuforia module and Android Studio

-the mobile device (a robot with 2 motor wheels) that is controlled remotely by the user, based on the Raspberry Pi 3 Model B+ platform;

-the Android system, mounted on the robot, with the camera that sends to the user the video stream using the Wi-Fi connection;

-the PC where the user sees the video stream and controls the robot using the keyboard.

As it can be seen there are two independent paths in the current implementation of the system. One is the video camera live feed from the Android device to the PC and the other is the control system for the mobile robot. The connection between the two is:

- the fact that the Android device is mounted on the robot;

- the fact that the user reacts and controls the robot based in the live video feed.

3. SYSTEM IMPLEMENTATION

An AR application had to be implemented first for the mobile phone that captures the videos stream in order to recognize certain images called markers that will be superimposed with 3D models created in Unity. The application runs on the Android device and is be able to track the marker if it is moving.

Vuforia [5] is the software development kit responsible for the augmented reality part of the application. The process of creating a APK package for Android is presented in Fig. 4. A module had to be installed for integration with Unity.

The images were be uploaded to the Vuforia web portal, to be recognized by the AR application. For the proper functioning of the application, the photos chosen must contain lines and colors that are easy to see even by the human eye. These were uploaded to the web application made available free of charge by Vuforia, processed and then they will receive a rating in the application.

As the rating approaches five (5 is the maximum rating), the application will work properly and the images will be recognized faster. If the rating is small (below 2) the markers will be harder to recognize.

Unity [6] is the platform that integrates Vuforia and allows the easy and fast configuration of the AR application. Through Unity and the libraries contained by Android Studio we can export the application that can later be installed on a compatible mobile phone. Confirmation for the access to the video camera

was necessary for the application on the Android device.

The robot (Fig. 6) was running an application programmed using Python that is able to accept remote commands sent from the PC. In the initial testing (without the network communication) the commands were sent by connecting in a remote session to the Raspberry Pi device. The robot carrying the Android device on its chassis, will be able to navigate in a controlled way along a path where different images will be placed and recognized by the AR application running on the Android device.

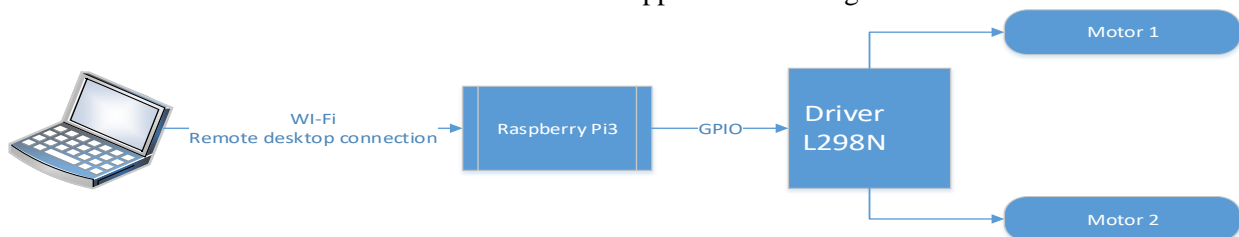


Fig. 5 Diagram of the communication between the PC and the robot's motors

The AR application communicates with the Raspberry Pi and through the UDP protocol (Fig. 5), the AR application is a client and the Raspberry Pi represents the server that will receive the information. The application interface contains four buttons that transmit information to the server via the UDP protocol. The server is a program written in Python code that interprets the information received from the user and gives the correct command for the robot's movement.

In order to use the UDP connection we must import the socket library that allows us to create a socket that allows the connection to be created. The connection is created using bind method with the two arguments: the first argument represent the host address, in this case the server address, and the second argument represents the port on which the information will be received.

The advantage of using the UDP protocol is the fact that the overhead of the communication is smaller when compared to TCP and that the Wi-Fi communication is prone to disconnects; therefore there is no need for keeping the TCP connection up. Finally the implementation had to take into account the possible disconnects and implemented a timer where a command is only executed for a limited time. If the key is pressed longer, multiple commands will be generated, each commanding

the robot for a few milliseconds. This implementation allows the robot to stop almost immediately when commands are no longer received.

The robot uses the L298N driver for the two wheel motors. The batteries for Raspberry and motor were separated in order to easily change them individually. For the motor control in Python, the GPIO library was used. On any application restart (when no commands were received), the pins were set to command the robot to stop.

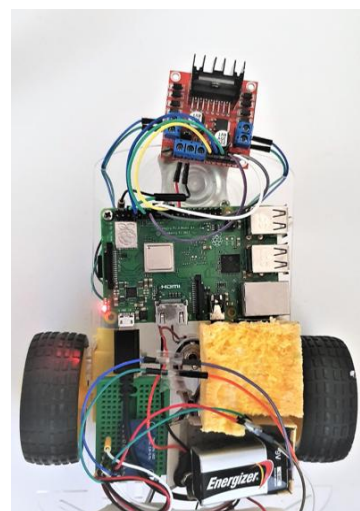


Fig. 6. Remotely controlled robot used to carry the Android device that transmits the video stream

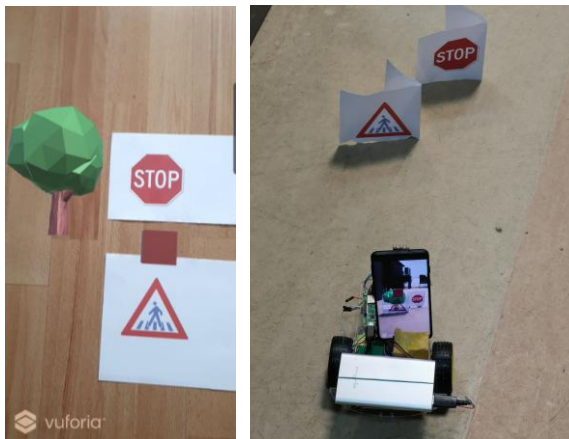


Fig. 7. Markers recognized at the same time and 3D model displayed over live video stream

The implementation results are visible in Fig. 7 where two road signs are detected in the same image in the Android application.

4. CONCLUSIONS

Augmented reality is a new technology. The major leap for this technology is when it has been able to run on smartphones. What is impressive is that AR applications have managed to run on various other platforms before the smartphone comes out, but with today's smartphones and tablets, AR has become much easier to use, being an increasingly popular aspect of our lives.

This paper presents the implementation of an Augmented Reality type application that runs on an Android device fixed to a mobile robot (using the Raspberry Pi3 model B+ platform) which will recognize the images received on the route through the camera, and overlay them with virtual objects and 3D models created using the Unity platform.

Over the last 30 years, important researches lead to the development of Augmented Reality. The rapid evolution of mobile phone cameras and 3D rendering possibilities have made it possible to migrate AR technology and lead the research towards its best use on mobile platforms.

The problems encountered were related to the wireless nature of the communication, namely the frequent disconnects and the communication latency. Due to the different usage of power between the Raspberry Pi and the motors, it was chosen to have different batteries powering these subsystems. Raspberry Pi was using more power and it was used more often when compared to the robot motors.

The images taken for the Vuforia application had to be correctly chosen in order to have a rating above 4 as this would give good marker recognition.

Finally another system development difficulty was that the system development used C# (in Unity), Java (in Android) and Python (for robot network communication and motor control) to complete the communication.

5. REFERENCES

- [1]. Tony Sperry, "Beyond 3D TV", Sperry Publishing.
- [2]. Elizabeth Zak, "Do You Believe in Magic? Exploring the Conceptualization of Augmented Reality and its Implications for the User in the Field of Library and Information Science", *Information Technology and Libraries* 33(4):23-50, 2014, DOI: 10.6017/ital.v33i4.5638
- [2]. Abbey Klaassen, Abbey Klaassen, "Best Buy's augmented-reality ad dazzles, but does it work?" 5.09.2009, web, accessed 20.06.2019, <https://adage.com/article/digital/advertising-inside-buy-s-augmented-reality-ad/138313>
- [3]. Zach Perilstein, "Universal Orlando Resort Patents Showcase Augmented Reality", 7.09.2019, web, accessed: 20.06.2019, <https://boardwalktimes.net/universal-orlando-resort-patents-showcase-augmented-reality-2b2f3e386fc7>
- [4]. Vuforia, "Vuforia Engine 8.3 is Available!" web, accessed: 20.06.2019, <https://developer.vuforia.com>
- [5]. Unity Technologies, "Asset packages", *Publication: 2019.2-002Xi*, web, accessed: 20.06.2019, <https://docs.unity3d.com/Manual/AssetPackages.html>