CERN SUMMER STUDENT PROGRAMME 2017

CERN: EP-AID-DA

ALICE Experiment

Final Report:

More than ALICE: Development of an augmented reality mobile application for the ALICE detector

Supervisors
Barthélémy von Haller, Jeremi Niedziela

Student
Anastasia Stamatouli

24 June - 22 September 2017

CONTENTS

INTRODUCTION

More Than ALICE is a mobile application for iOS and Android devices. This project concerns the development of the v2.1 of the application which is meant to enhance the capacity of tracking quickly and reliably parts of the detector and its paper model. It recognises different parts of it and displays labels explaining its structure. Additionally, visualisation of the collisions can also be shown on the top of the camera image. *More Than ALICE* aims to increase the public awareness of the research goals of the ALICE collaboration. The application provides an Augmented Reality (AR) interface to track the detector during underground visits or its paper model which can be purchased at the ALICE secretariat. For those without access to either the detector or the paper model, the app provides the virtual model of the detector where the users can explore and understand the different parts of the detector and see real-time collisions.

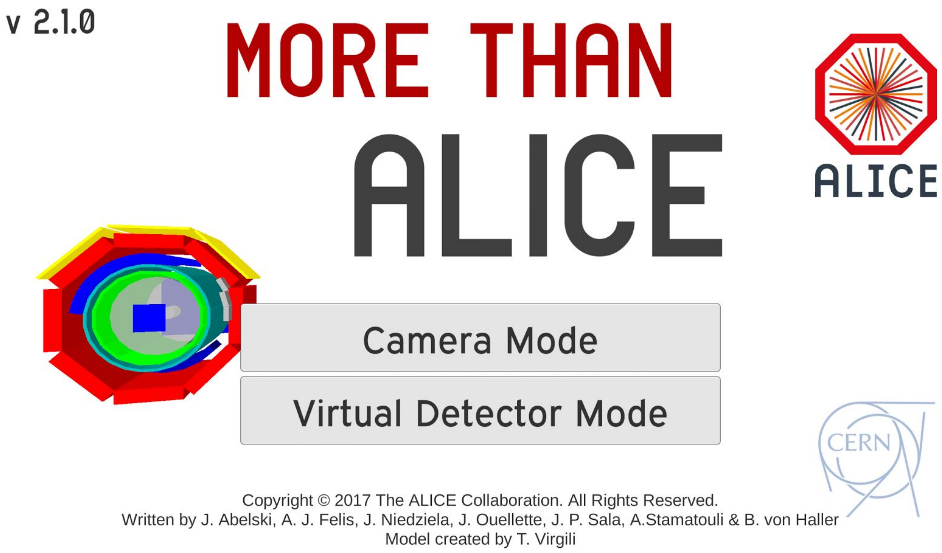

Figure 1: The Main Menu shown to users upon application launch.

MOTIVATION

The application was originally conceived by three CERN summer students (Jakub Abelski, Adam Janusz Felis and Jakub Pawel Sala) in 2014. In the first version (v1) of the application the Augmented Reality mode was designed for the real detector. The second version (v2) written by Jeff Ouellette[3] extended the AR mode to include the paper model of ALICE and a significant code refactoring took place.

However, in v2 the object recognition in the AR mode regarding the paper model was less efficient than the one for the real detector which means more research on this topic was needed in order to track quickly and reliably parts of the detector. It was crucial to understand how Vuforia SDK[1] performs in order to use the object recognition at the highest of its performance. Additionally, the algorithm of the collisions visualisation was enhanced in order to place the collisions on the correct part of the component. Furthermore, newer editions of the software tools that we use have been released, that are not backwards compatible, requiring significant code restructuring.

DESCRIPTION OF THE SOFTWARE TOOLS

- **The Unity Engine**

Unity[2] is a very popular game development engine. It is compatible with multiple operating systems and supports the development both of Android and iOS applications which was the main reason that led us to choose Unity for the development of our application. For more information, on how to use the Unity Engine, it is recommended to consult the main webpage.

- **The Vuforia SDK**

Vuforia is an Augmented Reality Software Development Kit (SDK) for mobile devices that enables the creation of Augmented Reality applications. It seems to be the most suggested and well-performing tool for augmented reality applications. There is a big community which provides information on how to setup the development environment, install the appropriate SDK version and build a sample app. Additionally, it offers an implementation compatible with Unity. On the website, one can find links to download the SDK, as well as links to the documentation.

TASKS

Task 1: Evaluation of the AR frameworks

Among the variety of the available AR frameworks the most popular were chosen and compared in order to find the suitable one for our application.

- **Vuforia**

1. Standalone library
2. Object detection
3. Fast
4. Extremely well-integrated into Unity
5. Good Documentation and Examples
6. Android, iOS, Unity

- **Open CV[4]**

1. Open Source library
2. Complex Algorithms
3. A unique recognition algorithm is required for every single object
4. OpenCV loses performance with the increase of image size
5. Plug in for Unity

- **ARToolKit[5]**

1. Open Source
2. No 3D object tracking
3. Android, iOS, Unity

- **Metaio SDK[6]**

1. More features
2. Lack of updated tutorials and documentation
3. Flickering in tracking
4. Steals the rendering from Unity and uses it's internal renderer which makes the process of drawing custom objects to the augmented one more complicated.
5. Android, iOS, Unity

Weighing up the advantages and disadvantages of the available AR frameworks we found that Vuforia is the best tool to perform this task.

Task 2: Evaluation of Vuforia's object recognition technology

Vuforia's object recognition technology works by matching images caught by a camera with a pre-defined reference image. Thus, the most important part of this process is to catch correctly the feature points of the reference image. This was by far the largest difficulty especially for 3D objects, because it relies on high quality 3D scans and on the features that the object has. The more detailed the object, the better it is.
The recognition of a variety of objects was tested under different lighting settings by different devices and the following problems were faced:

1. Difficulty in recognizing the object in the dark
2. Slow recognition
3. Parts of the object that do not have details are not recognized
4. The camera should be really close to the object and the recognition does not work from longer distance

Problems regarding difficulty in recognition or slow recognition could be solved by having better object scans. However, it should be kept in mind that the framework does not always perform as it is expected and some tolerance is needed.

Task 3: New scans of the objects using QR codes

Following up the problems faced during the previous evaluation of Vuforia we ended up with the solution of using QR codes. QR codes are labels used to uniquely identify the item to which they are attached to. Usually, they contain specific details and information about the object that they're attached to. In this case they are content independent since we only need them in order to introduce more details in each object so it will be easier for the Vuforia Object Scanner to catch the object's features and recognize it afterwards. They are widely available on Internet and for the needs of *More Than ALICE* they've been downloaded from Google images. Finally, a unique QR code is placed on each component and a scan is made using Vuforia's object scanner. The difference in the recognition of the object comparing to the previous results is significant. However, more research needs to be done in order to test whether the components are recognized if the QR codes are removed and the database still includes the reference images of the components with the QR codes.

Figure 2: QR code.

Task 4: Evaluation of the Augmented Reality mode using new scans

The Augmented Reality (AR) mode works both for the real ALICE detector and its paper model. For the recognition of the real ALICE detector we have used Image Targets which are planar images and for the paper model Object Targets which are 3D scans made by the Vuforia Object Scanner application. Up to two Object Targets can be tracked simultaneously while up to five Image Targets.

Testing the performance of the Augmented Reality Mode on the paper model of ALICE we came up to the following observations:

1. Regarding the ALICE_EMCAL, ALICE_PHOS, ALICE_TRD_TOF and ALICE_SOLENOID component it is not a good practice to scan each of

them separately because when it's placed on the paper model most of its surface is covered due to other components. Thus, when it comes to the recognition, most of the times the components are not recognized because the reference image is completely different with the live one. That being said, each one of these components should be scanned as it appears on the model and not separately.

2. However, the previous technique introduces some fuzziness. For example, instead of scanning the ALICE_SOLENOID separately we scan the whole model including the ALICE_TPC, ALICE_TRD_TOF, ALICE_EMCAL, ALICE_PHOS and ALICE_PMD on it. But, most of the feature points of this scan which will be a type of anchor for the object recognition of the ALICE_SOLENOID come from the ALICE_TPC component. So, when it comes to the recognition of the whole paper model each time that ALICE_TPC is in front of the camera both the explaining labels of the ALICE_SOLENOID and ALICE_TPC appear because both of the reference images have the same features.

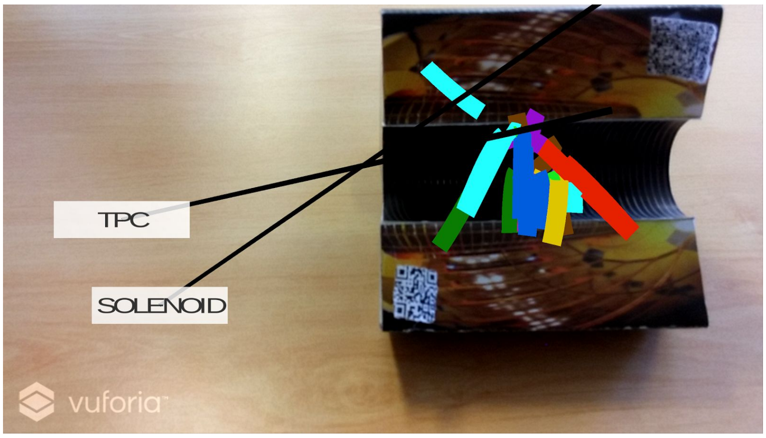

Figure 3: When ALICE_TPC is recognized both labels of ALICE_SOLENOID and ALICE_TPC appear.

3. Additionally, following the technique of scanning each component on the paper model and not separately, affects the individual recognition of the component outside of it. A possible solution to this problem is to include in the database both scans of the component itself and the one with the component on the model but this would increase the size of the database and the performance of the application.

4. The components that are being recognized effectively are ALICE_TPC, ALICE_TRD_TOF, ALICE_DIMUON and ALICE_SOLENOID. It takes more time to recognize the ALICE_PMD and ALICE_PHOS.
5. The ALICE_EMCAL and ALICE_VZERO_ITS are not recognized.

Task 5: Upgrade to the newer versions of Unity and Vuforia

Newer edition of Vuforia SDK had been released so the application had to be upgraded, requiring some updates of the API call. Additionally, the new version of Vuforia is not compatible with the latest version of Unity which means that a switch to a previous version of Unity was needed. We are using Unity 5.4.0f with Vuforia SDK 6.2.10.

Task 6: Obtaining the bounding box of the components

Obtaining the bounding box of the components was a challenging process since Vuforia does not provide a clear way and one has to deal with the pose matrix and complicated graphic transformations. Instead, the transform of each object has been used and the approximate distance from the middle point of the component until its edge was added.

Task 7: Collisions Mode

The collisions mode forms one of the core aspect of the application. They should be placed in the middle of ALICE_TPC component and for this reason code was written in order to place the CollisionContainer in the correct position by obtaining the bounding box of ALICE_TPC.

FUTURE PERSPECTIVE

While the object recognition in v2.1 was enhanced bringing the application in a more stable and reliable level, it is not complete yet and there are many elements that have to be implemented in v2.1. This includes multiple languages, options scene where the user can enable/disable the sounds or change the size of the exploded view.

A good direction of improvement regarding the collisions visualisation involves the use of a tool which already exists and not the custom one which was originally developed in v1 of the application.

In addition, regarding the quality of the object scans there is no need to focus on professional lighting and better camera quality because in the end what really matters is how detailed the object is. An average quality of camera and normal lighting conditions are able to give reliable results. In the current version of *More Than ALICE*, Vuforia has been used at the highest of its current capabilities but still there is work to be done as it is upgraded regularly.

Other future direction would give to the user the opportunity to observe in details the paper model and its components. The user will be able to have a close look around the component while Vuforia’s virtual buttons appear explaining specific details and making the experience more interactive.

Last but not least, the next version of the paper model could be more detailed with bright colors and contrast-based features which would enhance the recognition of Vuforia framework without using QR codes.

REFERENCES

[1] https://vuforia.com/

[2] https://unity3d.com/

[3] Jeff Ouellette. *More Than ALICE: Development of an augmented reality mobile application for the ALICE detector*. Project Report. CERN. August 12, 2016

[4] http://opencv.org/

[5] https://artoolkit.org/

[6] https://en.wikipedia.org/wiki/Metaio