Walchand College of Engineering, Sangli
(An Autonomous Institute)

Department of Computer Science and Engineering

A Mini- Project Report on

VirtualCampus: Prototyping 3D Mobile Augmented Reality Systems for Exploring the College campus using Android app

under the guidance of

Mrs. T. T. Kulkarni

Submitted by

Vaibhav Ananda Kumbhar	**2012BCS057**
Akshay Shirish Habbu	**2012BCS095**
Machchindra Sanjay Pol	**2012BCS091**

2014-2015

Walchand College of Engineering, Sangli
(An Autonomous Institute)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Third Year B.Tech. project entitled VirtualCampus: Prototyping 3D Mobile Augmented Reality Systems for Exploring the College campus using Android app is a bonafied work carried out by the student team- **Mr. Vaibhav Anand Kumbhar** , **Mr. Machchindra Sanjay Pol** and **Mr .Akshay Shirish Habbu** - in partial fulfillment of the completion of 5th semester B. Tech. course during the year 2012 2013. The project report has been approved as it satisfies the academic requirement with respect to the project work.

Guide **H.O.D**

Mrs. T. T. Kulkarni **Dr. B. F. Momin**

Contents

List of Figures

ACKNOWLEGEMENT

We wish to take this opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during my project work. It is our pleasure to acknowledge the help of all those individuals.

We would like to thank our project guide Mrs. T. T. Kulkarni mam, Computer Science and engineering Department for her guidance and help throughout the development of this project work by providing us with required information. With her guidance, cooperation and encouragement we had learnt many new things during our project tenure.

We specially thank Dr. B. F. Momin Head, Computer Science and Engineering Department for his continuous encouragement and valuable guidance in bringing shape to this dissertation.

We specially thank Dr. G.V. Parishwad Director of Walchad College of Engineering Sangli for his encouragement and support.

In completing this project successfully all our faculty members have given an excellent cooperation by guiding us in every aspect. We also thank our lab faculty and librarians.

Kumbhar Vaibhav A. [2012BCS057]
Pol Machchindra S. [2012BCS091]
Habbu Akshay S. [2012BCS095]

ABSTRACT

Our College premises are rapidly developing with so many departments and so many laboratories with multi-disciplinary environment. For any student, teachers and staff members it will be great if all the related information regarding any infrastructure in the college is being shown on their smartphones. So our project is the same Android application which will display every single information related to the scene that being recorded by the users smartphone i.e. Names of every buildings, labs, departments, offices, open spaces, etc.

We describe a project work that combines together the overlaid 3D graphics of augmented reality with the untethered freedom of mobile computing. The goal is to explore how these two technologies might together make possible portable android application that can support users in their everyday interactions with the world. We introduce an application that presents information about our college campus, using a mobile phone installed with our android application. We provide an illustrated explanation of how our project is used, and describe our rationale behind designing its software infrastructure.

1 Introduction

The field of Augmented Reality (AR) has existed for just over one decade, but the growth and progress in the past few years has been remarkable.

1.1 Augmented Reality?

The basic goal of an AR system is to enhance the users perception of and interaction with the real world through supplementing the real world with 3D virtual objects that appear to coexist in the same space as the real world. Many recent papers broaden the definition of AR beyond this vision, but in the spirit of the original survey we define AR systems to share the following properties:

1. Blends real and virtual, in a real environment
2. Real-time interactive
3. Registered in 3D

We describe a project work that combines together the overlaid 3D graphics of augmented reality with the untethered freedom of mobile computing. The goal is to explore how these two technologies might together make possible portable android application that can support users in their everyday interactions with the world. We introduce an application that presents information about our colleges campus, using a mobile phone installed with our android application. We provide an illustrated explanation of how our project is used, and describe our rationale behind designing its software infrastructure.

1.2 Problem definition

Think of a situation where we are moving in any new premises of any college and you need to know every single information about that place. So in the conventional method of current scenario we will google the information about the place not so sure we will either can also go to google maps but the problem is the real time scene and infrastructure you are experiencing is way much different than

the conventional maps will show you. So this is how you come to a situation where you can use this kind of app which will show you the real time buildings with names and details on your mobile screen.

1.3 Overview

VirtualCampus is an android application that is been developed using Wikitude SDK. Wikitude is a mobile augmented reality software which is developed by the Austrian company Wikitude GmbH (formerly Mobilizy GmbH) and was first published in October 2008 as freeware. It displays information about the users' surroundings in a mobile camera view, including image recognition and 3d modeling. Wikitude was the first publicly available application that used a location-based approach to augmented reality.

2 TECHICAL SPECIFICATIONS

2.1 Introduction to Augmented Reality

2.1.1 History of Augmented Reality

1966 Invented Head-Mounted Display which was the first step in making AR a possibility.

1992

- Coined the term Augmented Reality.
- Developed Complex Software at Boeing to help technicians assemble cables into aircraft.

1999

- In 1999, Hirokazu Kato of the Nara Institute of Science and Technology released the ARToolKit to the open source community.
- Although the smartphone was yet to be invented, it was what allowed a simple, handheld device with a camera and an internet connection to bring AR to the masses.

2.1.2 What is Augmented reality?

Augmented reality is the display of computer graphics and media, overlaid and registered on real-world environments, interactively in real time.In other words,

Augmented reality (AR) is the act of enhancing reality with computer-generated content that, in this case, is activated by holding a marker in front of a webcam.

2.1.3 Uses of Augmented Reality

- Archaeology
- Architecture
- Art
- Commerce
- Education

- Gaming
- Military
- Navigation
- Translation

2.2 Features and Concepts of Wikitude:

2.2.1 Image Recognition

Image recognition helps to recognize images in the viewfinder and overlay it with images. Furthermore it shows how to recognize multiple different images and how to react on user clicks on the overlaid elements.

The image recognition engine developed by Wikitude is based on the Natural Feature Tracking (NFT) principle. As the name implies, this particular field of computer vision (CV) uses natural features to track images. The alternate approach is the use of marker tracking, which uses artificial images, barcodes for example, to track a scene using artificial shapes.

To implement image recognition engine several steps need to execute:

1. **Preprocessing:** The target image that should be tracked is analyzed. Significant areas in the image, so called Feature Points, are extracted and stored. How the Feature Points are detected and stored is the essential essence of the algorithm used. The preprocessing step is executed only once per tracked image, and can run offline and asynchronously.

2. **Feature Point Detection:** On the device, similar to the preprocessing step, the current camera image is analyzed for keypoints.

3. **Tracking:** These recognized keypoints are then compared with the keypoints generated from the target images in step 1. If a pre-defined similarity is determined, the image is considered a positive match and then tracked. Several algorithms exist to determine a threshold of similarity, which one to chose is up to the implementation.

These are few terms which defines the Wikitude concepts about AR.

- **Target:** A target image and its associated extracted data that is used by the tracker to recognize an image.
- **Target collection:** An archive storing a collection of targets that can be recognized by the tracker. A target collection can hold up to 1000 targets.
- **Tracker:** The tracker analyzes the live camera image and detects the targets stored in its associated target collection. Multiple trackers can be created, however only one tracker can be active for recognition at any given time.

2.2.2 3D Models

Wikitude also helps to augment a target image with 3D content. It starts by displaying a 3D model on a target and advances by adding displayed animations and interactivity.

3D content within Wikitude can only be loaded from Wikitude 3D Format files (.wt3). This is a compressed binary format for describing 3D content which is optimized for fast loading and handling of 3D content on a mobile device. You still can use 3D models from your favourite 3D modelling tools (Autodesk Maya or Blender) but you'll need to convert them into the wt3 file format. The Wikitude 3D Encoder desktop application (Windows and Mac) encodes your 3D source file. The Encoder can handle Autodesk FBX files (.fbx) and the open standard Collada (.dae) file formats for encoding to .wt3.

3D content is rendered on top of 2D objects. This limitation exists because of some SDK internal restrictions.

Four flavours of 3D model AR:

1. 3D Model on Target Image
2. Appearing Animation
3. Interactivity
4. Snap to Screen

2.2.3 POINT OF INTEREST (POI)

The Point Of Interest (POI) will show how you can create a marker that is placed at a specific geolocation. The example is split into four different parts that depend on each other. You will have a complete and reusable marker at the end of the series which has a title, description, a selected and an idle state which animates smoothly from one to another.

Four flavours of POI:

1. POI at Location
2. POI with Label
3. Multiple POIs
4. Selecting POIs

2.2.4 RETRIEVING POI DATA

There are several ways to request and work with POI detail information in an ARchitect World. Depending on your application and use case, one might fit better than the other.

Retrieving data consists of three parts

1. From Application Model
2. From a Local Resource
3. From a Webservice

Displaying numerous POIs in the camera is a challenge. How many POIs should be offered? How to deal with POIs in same direction? What is the maximum range to show POIs and how to display a long description? The following example cover frequently asked questions related to the POI browser use case and consists of five parts plus a bonus section :

1. Presenting Detail
2. POI and AR Radar
3. Limiting Visible POIs
4. Reloading POI Data

5. Native Detail Screen
6. Capture Screen Bonus

2.2.5 VIDEO DRAWABLES

Besides images, text and HTML content you are able to display videos in augmented reality using the Wikitude SDK. With the help of AR.VideoDrawables you can add a video on top of any target image (AR.Trackable2DObject) or have it displayed at any geo location (AR.GeoObject). Like any other drawable you can position, scale, rotate and change the opacity of the video drawable.

SUPPORTED VIDEO CODECS AND HOSTING SERVICES

To support all platforms make sure to use a H.264 encoded video with a maximum resolution of 720p (1280x720 pixel).

H.264 defines different profiles. Make sure that you are using either one of the following :

- Baseline
- Extended
- Main

If the profile differs Android devices will most certainly misbehave (fail to play or crash entirely) when playing back those videos.

If you like to add a YouTube video, which is then played fullscreen in the native player, rather use an AR.ImageDrawable showing a poster or play-button and add the URL to the YouTube video to the onClick trigger. Similar to what we do in the image recognition sample.

VIDEO SAMPLE

This is the way to augment a target image. Furthermore it shows how to react on playback states and concludes with how to use transparent videos.

So it consists of four parts :

1. Select a Video and add it to a Target Image

2. Control Video Playback
3. Snapping Video
4. Transparent Video

2.2.6 IMAGE RECOGNITION AND GEO

The Wikitude SDK allows you to combine location based augmented reality scenes with vision baed scenes to create a seamless experience for users. This tutorial will show you how to accomplish this and will provide you with additional advices.

2.3 Project Architecture

2.3.1 DEVELOPMENT WORKFLOW

The following section describes a default development workflow for writing AR content using the ARchitect JavaScript API. It demonstrates the code test and debug cycle and provides useful tips for each step.

CODE, TEST, DEBUG :

1. Write your HTML, JavaScript and CSS using the text editor of your choice
2. Test in your desktop browser
3. Debug in your desktop browser using e.g. WebInspector
4. Test on a physical device
5. Debug on a physical device
6. Rinse and repeat

You can use any editor for writing the ARchitect World. We particularly like Sublime, which has a great selection of plugins for web developers.

The next step is to test it out in a desktop browser. To have the ARchitect JavaScript API available in the desktop browser you'll need to include the ARchitect Desktop Engine (ADE). See the chapter ARchitect Desktop Engine for instructions on how to use it. While this is limited in visualizing the experience, it greatly helps

in finding errors in the JavaScript code and reduces the time it takes to see effects of changes you have made to the HTML and CSS parts. Desktop browsers come with great debugging tools that allow you to easily debug your JavaScript code and that you should make full use of when debugging ARchitect Worlds.

Once you have verified the JavaScript is working properly you should test it on the device of your choice. Either start your application that loads the ARchitect World or run it inside the Wikitude World Browser app.

AR experiences can be tested on Android using the Wikitude World Browser for Android. Download the Wikitude app from Google Play. Launch the application and press the menu button in the upper left corner of the screen. Then tap the Developer button to open the developer login page. Enter your username and password of your Wikitude developer account and tap the login button to get to your user account. If you don't have a developer account yet register at the Wikitude developer page.

In this view, tap on the text box below the Launch via URL label and enter the URL of your ARchitect World. After tapping the Launch button your AR experience will be opened in Wikitude.

3 SYSTEM ANALYSIS

3.1 Proposed System:

The proposed system is an android application which detects the places around the entire campus using augmented reality algorithms. The live stream of video data i.e. set of continuous images is fetched to the system, system then identifies the markers in the images and displays the appropriate response back on the live screens. So basically its a real time detection of places and it displays back the stored information about that particular place/ markers.

The VirtualCampus app is just a simple single screened application which directly jumps on to your camera view and starts detecting the live frames.

3.2 Harware Requirements

To work with project the following hardwares are required:

- Smart Phone with Android Operating System ,Camera, GPS.

3.3 Software Requirements

List of Softwares needed:

- Android Studio.
- ADB Device Support Driver.

4 SYSTEM DESIGN

4.1 Core concepts

As we will show later, many tracking algorithms can be split into functional blocks or even atomic operations which can be reused in other algorithms. The data types used to exchange data between these blocks also have certain similarities. To enable combination of different functional blocks we encapsulate these in abstract classes with a xed interface. The interface consists of only three functions for initialization, execution and de-initialization. A similar abstract class exists for data, which enables us to store it in a global data-set, where it can be accessed by pointer or name reference. We call these abstract classes Action and Data. The connections between the blocks are dened by the dataow between them. Each Action has a set of keys which is a reference to a data element, which can be retrieved from the data-set before execution. The indirection over a key element for data access allows the use of a graphical interface or a scripting language to setup the data-ow. Actions and Data also provide a reective type interface to access selected class variables either by their original type or even more general by string. This provides a generic way to access parameters and data at run-time in graphical interfaces.

4.2 Parallelization and Synchronization

In order to support state of the art multi-core and multi-processor systems, the system offers three different levels of parallelization. All threading and synchronization aspects are based on boost::thread library. The lowest level consists in the classical and straight forward multi-threading implementation achieved by the developer to improve performance of a specic algorithm. At this level it is the developers responsibility to implement correct parallel execution. The mid-level is done automatically and is based on dependency analysis of the data ow.

As a result we get a dependency graph which is used to select Actions with independent data inputs, able to run in parallel without any conicts. This method can be seen from the outside as a sequential processing and does not require precautions to protect the data from access by simultaneous running threads, due to the precondi-

tion of having independent data. The method does not require any knowledge about the algorithm, but relies on a ne granularity in the building blocks, the more atomic they perform their work the more can be parallelized. This method is the easiest to use for the developer, he can develop the tracking system in a clear hierarchical manner (top down) without caring about execution order. Although this method might not produce an optimal result, it is a convenient way to improve performance.

. As a third method, our system provides a Component which is a conguration of Actions, executed in its own thread. This concept is important for tracking with devices running at different framerates, like inertial sensors (e.g. 100Hz) and cameras (e.g. 25Hz). In such a case the sensor interface and a pose estimation algorithm, like a Kalman lter, would run in one component at sensor frame rate, while the slower camera and vision part runs in another component. In order to synchronize data exchange between these components, the data objects incorporate a locking mechanism. The same mechanism also handles signals to notify other components about changes in the data. Additionally our data objects can be time stamped to identify related data in an asynchronous setup, where multiple sensor inputs must be handled.

4.3 Extending the System

New Action and Data classes built in to shared libraries can be added to our framework by simply loading them during runtime. This makes the system extendable without rebuilding it and allows enhancements by adding more problem specic Actions while reusing the existing Actions for common processing steps. The system provides a type-factory where all available (linked) object are registered and can be instantiated by name. This comes in useful when reading congurations from a XML le or when composing new congurations with a graphical interface.

4.4 Interfacing

par To use the system as a library in once own application, only the instantiation of the run-time instance is necessary. It provides access to all Actions and data in the system. By accessing the Actions, all parameters can be varied during run-time. Results or

additional input data can be accessed through the global data-set where all data objects are stored. The run-time system also has a singleton which provides a static pointer to itself, which makes access from different classes easier. The run-time instance provides some general purpose functions useful in any application like loading and saving of xml conguration les, loading of extensions, and it denes abstract entry points for a graphical interface which allows separation of user interface and library. How this concept can be exploited to build certain types of applications, is outlined in the Application Framework section below.

4.5 Tracking Framework

In order to simplify the implementation of tracking algorithms and reuse as much code as possible, we dene a tracking framework consisting of several base classes representing the major steps in tracking algorithms. We identied the following steps which are common to the most visual 3D tracking systems:

4.5.1 Common Processing steps-

- **Image processing-**In this, usually the rst step, the incoming video images are modied to t the requirements of the following processing steps. Examples are image and video capture, image read and write to le, gradient image, image pyramid, binarization, image conversion, undistortion, etc.
- **Feature detection-** at this step higher level information such as interest points of an image are generated. This step is specic to a given tracking procedure. Examples are Harris - corner detector, edge detection, KLT - Kanade Lucas tracker - the feature detection part, SURF
- **Tracking, matching or classication-** The process of correspondence determination is very specic to a tracking procedure and can consists in searching features seen in the past images (tracking), comparison with reference data (matching) or feature sorting (classication). Examples for tracking are KLT point tracking, line tracker, for matching features with references there is MSER and for classication we have randomized trees.

- **Pose estimation-** Computation of a camera pose with a given set of correspondences. Examples are generic pose calculation with Kalman Filter, RANSAC (linear or non linear estimation with inliers/outliers tests),POSIT, linear pose calculation: HEIV and nonlinear pose estimation (Levenberg-Marquardt).

4.6 Data Flow Diagram

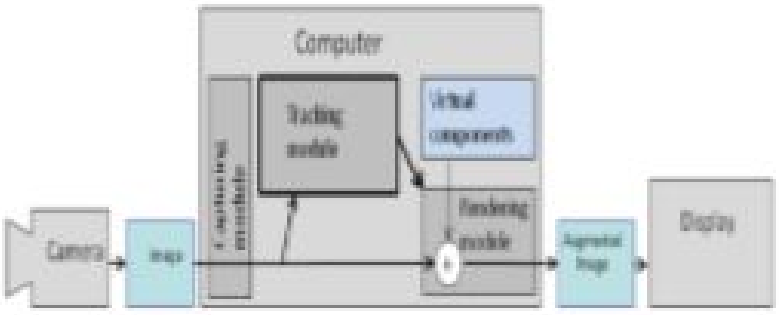

Figure 1: Flow Chart for a Simple AR System.

AR.jpg

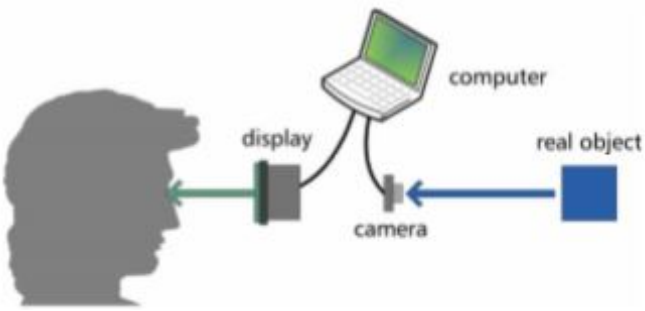

Figure 2: Simple AR System Flow.

4.7 Output Design

Designing computer output should proceed in an organized, well throughout manner; the right output element is designed so that people will find the system whether or executed. When we design

an output we must identify the specific output that is needed to meet the system. The usefulness of the new system is evaluated on the basis of their output.

Once the output requirements are determined, the system designer can decide what to include in the system and how to structure it so that the required output can be produced. For the proposed software, it is necessary that the output reports be compatible in format with the existing reports. The output must be concerned to the overall performance and the systems working, as it should. It consists of developing specifications and procedures for data preparation, those steps necessary to put the inputs and the desired output, i.e. maximum user friendly. Proper messages and appropriate directions can control errors committed by users.

The output design is the key to the success of any system. Output is the key between the user and the sensor. The output must be concerned to the systems working, as it should. Output design consists of displaying specifications and procedures as data presentation. User is never left with the confusion as to what is happening without appropriate error and acknowledges message being received. Even an unknown person can operate the system without knowing anything about the system.

5 SYSTEM TESTING

5.1 ON-DEVICE DEBUGGING

Remote debugging is available for devices running Android.

To enable remote debugging of a WebView add the following line to your CamActivity.

$if(Build.VERSION.SDK\_INT \geq 19)\{$
$WebView.setWebContentsDebuggingEnabled(true);$
$\}$

setWebContentsDebuggingEnabled is available for Android 4.4+, you may need to capsule that to avoid runtime errors on devices running Android lower version 4.4. Additionally set TargetSDK to 19 in your Android Manifest.

6 IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation of change over methods.

Apart from planning major task of preparing the implementation are education and training of users. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

6.1 Recongnition

There is one object of **Trackable2D**.This object is used for the recognition.

```
var CSEDeptT20 = new AR.Trackable2DObject(this.tracker, "CSEDept", {
    drawables: {
                cam: [  CSEDeptButton,CSEScholarButton,CSEDeptASLButton,CSEDeptDBLButton,CSEDeptRLButton]
    }
});
```

Figure 3: Code Snippet-Trackable2D.

6.2 Screen Overlay

These objects are used to add virtual world on camera view.

```
createOverlays: function createOverlaysFn() {
    this.tracker = new AR.Tracker("assets/VirtualCampus1.0.wtc", {
        onLoaded: this.worldLoaded
    });
    var CSEDeptButtonImg = new AR.ImageResource("assets/CSEDept/CSEDeptButton.jpg");
    var CSEDeptButton = new AR.ImageDrawable( CSEDeptButtonImg, 0.1, {
                offsetX: 0.1,
                offsetY: 0.1,
                zOrder: 1
                /*,
                onClick = function() {
                    AR.context.openInBrowser("");
                };*/
        });
    var CSEDeptASLButtonImg = new AR.ImageResource("assets/CSEDept/CSEDeptASLButton.jpg");
    var CSEDeptASLButton = new AR.ImageDrawable( CSEDeptASLButtonImg , 0.1, {
                offsetX: -0.9,
                offsetY: -0.1,
                zOrder: 1
                /*,
                onClick = function() {
                AR.context.openInBrowser("");
                };*/
        });
```

Figure 4: Code Snippet-Overlay Objects.

7 Screenshots and Results

7.1 Application Icon

Figure 5: Application Icon.

7.2 Application Entry View

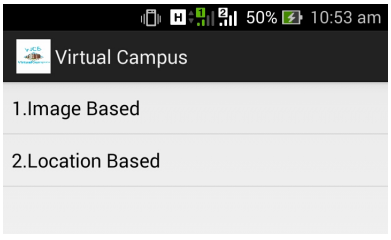

Figure 6: Application Entry View.

7.3 College's Front View Recognition

Figure 7: College's Front View Recognition .

8 Conclusion and Future Work

8.1 Conclusion

To conclude the description about the project: The project, developed by Augmented reality SDK- WikiTude using Android for development platform is based on the idea that fascinates our entire team, To make use of the latest technological advancements in day to day life. The idea of embedding the real life scenarios to the virtual world is always the motive behind our project work. This application helps navigating and sharing information about the college campus and the places around it.

The application is a navigation aid to people walking outdoors. These individuals could be students advancing upon their objective, faculties trying to gain information of the campus, or guests seeking directions to their intended destination. Today, these individuals must pull out a physical map and associate what they see in the real environment around them with the markings on the 2D map. If landmarks are not easily identifiable, this association can be difficult to perform. An AR system makes navigation easier by performing the association step automatically. If the user's position and orientation are known, and the AR system has access to a digital map of the area, then the AR system can draw the map in 3-D directly upon the user's view. The user looks at a nearby buildings and sees graphics directly overlaid on the real environment explaining the buildings name, what are the highlights of it, how far away it is, and List of faculties and students studying etc.

8.2 Future Work

Now the space ovehead is on mobile's memory. So in future, We can extend this project with cloud recognition facility. Real time 3D recognition also possible.

REFERENCES

1. Visual Tracking for Augmented Reality -Georg Klein.
2. Pro Android Augmented Reality -Raghav Sood.
3. Wikitude.com -Offcial Documentation and User Forum.