"Text Recognition and Dynamically Loading Related Images in Augmented Reality"

SEVENTH SEMESTER MINI PROJECT REPORT
FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

BY

Abhishek Jaiswal (IIT2013129)
Shubham Chaudhary (IIT2013149)
Tanushree Anand (IIT2013192)

UNDER THE SUPERVISION OF

Dr. Shirshu Varma
IIIT-ALLAHABAD

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD

(A UNIVERSITY ESTABLISHED UNDER SEC.3 OF UGC ACT, 1956 VIDE NOTIFICATION NO. F.9-4/99-U.3 DATED 04.08.2000 OF THE GOVT. OF INDIA)
A CENTRE OF EXCELLENCE IN INFORMATION TECHNOLOGY ESTABLISHED BY GOVT. OF INDIA

2nd December, 2016

CANDIDATES' DELARATION

We hereby declare that the work presented in this project report entitled "**Text Recognition and Dynamically Loading Related Images in Augmented Reality**", submitted towards fulfillment of 7th Semester report of B.Tech. (IT) at Indian Institute of Information Technology, Allahabad, is an authenticated record of our original work carried out from August 2016 to December 2016 under the guidance of **Dr.Shirshu Varma**. Due acknowledgements have been made in the text to all other material used. The project was done in full compliance with the requirements and constraints of the prescribed curriculum.

Place: Allahabad

Date:

Abhishek Jaiswal (IIT2013129)

ShubhamChaudhary (IIT2013149)

Tanushree Anand (IIT2013192)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Place: Allahabad

Dr.Shirshu Varma

IIIT Allahabad

ACKNOWLEDGEMENT

We would like to acknowledge and extend our heartfelt gratitude to **Dr. Shirshu Varma**, Indian Institute of Information Technology Allahabad, who guided us through this project. His keen vital encouragement, superb guidance, and constant support are the motive force behind this project work. We would like to thank all our friends for their continuous motivation and encouragement during this project. We are very thankful to all the technical and non-technical staff of the college for their assistance and co-operation.

Abhishek Jaiswal
IIT2013129

Shubham Chaudhary
IIT2013149

Tanushree Anand
IIT2013192

ABSTRACT

Augmented reality (AR) is a live direct or indirect view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data. The fact is to create an artificial world that provides sense of a real world where real life is enhanced by virtual elements in real times.

This project deals with the development and implementation of Augmented Reality software which recognized English words in real time, crawls the web to extract useful related images and meaning of the detected and augments it to the word detected in real time. Most of the applications in AR uses a database to prestore the contents to be augmented as a map to the given marker elements. We, on the other hand, are recognising the markers (as a word) by text recognition and are downloading the required contents from the web in the real time through web crawling and regular expression pattern searching techniques. This overcomes the shortcoming of manually downloading and mapping the contents to the million potential word markers present in our word corpus. Also storing the contents for the million of the words would take a considerable amount of space in the database which will not be suitable to be used on the mobile devices which have limited storage capabilities.

The position of the virtual contents augmented to tracked words are updated in each frame respective to updated coordinates values of the tracked words thus providing the real world feel to the virtual objects.

Table of Contents

1. Introduction

Augmented Reality (AR) is the concept of superimposing digital content (such as online information, sound, video, graphics or GPS data) on top of a view of the real world as seen through the viewfinder of a camera. Augmented Reality is similar to Virtual Reality except it seeks to enhance your perception of the real world and is not a fantasy place. 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 originally AR systems share the following basic properties:

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

There are various versions of textual and visual dictionaries, encyclopedias and other such resources. A visual dictionary is a dictionary that primarily uses pictures to illustrate the meaning of words. Visual dictionaries are often organized by themes, instead of being an alphabetical list of words. For each theme, an image is labeled with the correct word/term to identify each component of the item in question. Search the themes to quickly locate words, and the meaning of a word by viewing the image it represents. We present a more realistic form of data dictionary that lets you visualize and feel the meaning of a word with the help of meaning and model or illustration of the word without the effort of searching or typing, just by hovering over your camera over the printed word i.e. all-in-one source of reference. Such a dictionary helps you learn English in a visual and accessible way. It is ideal for teachers, parents, translators and students of all skill levels. It is handy and perfect to discover a visual world of information! The best part is you can use this tool/app in your smart phone. The user is not even required to type a word for searching. Instead the user just places his phone on top of any text/word and he has the option of viewing the attributes related to the word which will be popping up in the real world when seen through his camera.

1.1. Motivation

Augmented reality has many applications in a variety of fields – IoT, military, industrial, and medical, commercial and entertainment areas, sign boards, video games, industrial design, sports, education.

While reading whenever we come across words whose meaning is unclear to us, each time we have to refer to a paper dictionary or open a dictionary app, type the word and what we get is only the meaning of the word in written form. We felt the need of having software that lets you visualize and feel the meaning and model or illustration of the word without the effort of searching or typing, just by hovering over your camera over the

printed word. Also the illustrations are displayed in the real world alongside the queried word using augmented reality so that you can also feel and visualize them.
Our idea can be developed and used for educational purposes as well as by language translators. Text, graphics, video and audio can be made into a database of words relating to specific subjects or teaching newbies and primary school children with the help of illustrations. We can extend our dictionary to any other language and a larger database of words which would then involve providing the translation of a particular word in different languages along with other attributes related to its meaning.

2. Problem Definition

To build an interactive and dynamic software which:

- Recognizes printed English words,
- Dynamically positions meaning/definition and images/model for the recognized word in augmented reality.

3. Objectives

We intend to make an efficient software keeping the following in mind:

- To be accurate enough to recognise as many words of English text.
- Should be lightweight so it can be operated on mobile devices.
- To be dynamic so as to crawl the web and download the meaning and image specific to any given English word.
- To use power of augmented reality to make understanding of words more useful by attaching virtual objects(meaning, models/ images) with the recognized text in the region of interest

4. Literature Survey

Sl.No	Title/Year	Journal/ Conference	Authors	Objective	Methodology
1	Word Recognition Incorporating Augmented Reality For Linguistic E-Conversion	International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)	Faustina Jeya Rose.R, Bhuvaneswari.G	Development and implementation of Augmented Reality software for linguistic conversion on smart phones which can detect and translate English text into Tamil Language in realtime.	The camera image is preprocessed by mono chrome image recognition, digital overlay on mono chrome image, CMYK image recognition and digital overlay on recognized CMYK image, followed by character and word recognition using the Vuforia word list.A Hash Table is created to map the Tamil word equivalent for each English word which is then superimposed on the English word in augmented reality.
2	Design of an optical Character Recognition System for Camera based Handheld devices	International Journal of Computer Science Issues-2011	A F Mollah, Nabamita Majumder, Subhadip Basu, Mita Nasipuri	To develop a complete Optical Character Recognition system for camera captured image/graphics embedded textual documents for ndheld devices	Text regions are extracted from images, skew corrected, binarized and segmented into lines and characters. Each character is sent into the recognition module where it is resized by its bounding box, normalized to a standard dimension and compared to a class template; it belongs to the class for which the maximum correlation is found.
3	Automatic text detection for mobile augmented reality translation	ICCV Workshop 2011	Marc Petter, Victor Fragoso, Matthew Turk, Charles Baur	fast automatic text detection algorithm devised for a mobile augmented reality (AR) translation system on a mobile phone	Initial search to find a single letter. Detecting one letter provides useful information that is processed with efficient rules to quickly find the remainder of a word. Also present a method that exploits the redundancy of

					the information contained in the video stream to remove false alarms.
4	Object Detection and Pose Tracking for Augmented Reality: Recent Approaches	FCV 2012	Hideaki Uchiyama, Eric Marchand	This paper classifies and summarizes the recent trends on how to compute relative camera pose with respect to target object so as to visually merge a virtual object onto a real scene with geometrical consistency	Explained projection models and poses estimation depending on the shape of objects. Then, classification and summarization of the recent progress of detection and tracking techniques. Also, some evaluation datasets and evaluation procedures are introduced.
5	Augmented Reality: Applications, Challenges and Future Trends	Applied Computation -al Science, WSEAS 2014	Mehdi Mekni, Andre Lemieux	This paper surveys the current state-of-the-art in augmented reality	Describes work performed in different application domains and explains the exiting issues encountered when building augmented reality applications considering the ergonomic and technical limitations of mobile devices. Future directions and areas requiring further research are introduced and discussed
6	Tideland Animal AR: Superimposing 3D Animal Models to user defined targets for Augmented Reality Game	International Journal of Multimedia and Ubiquitos Engineering-2014	Youngo Lee, Jongmyong Choyee	To develop a mobile app that superimposes 3D models of tideland animals sequentially when users make image targets in real time	With Vuforia SDK they augment 3d models on 2d plane,cylinder and cubical shape;use an array data structure to manage several 3d models and image targets;whenever user takes a new picture,save it into the array and assign a 3d model to it removing the last one(least recent one)

5. Methodology

Our project consists of three main phases:

- Text Recognition (Identifying words in ROI by a Bounding Box)
- Real time Web Crawling for meaning and images related to the word
- Dynamic Positioning of crawled contents relative to the word in Augmented Reality

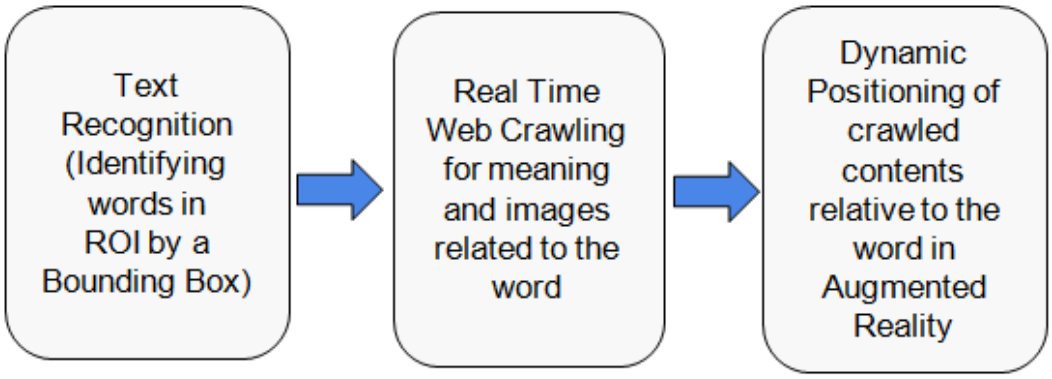

Figure 1: Three Main Phases of Project

5.1. Text Recognition

In our project, text recognition includes detecting and tracking of printed English words within a defined detection window referred to as a region of interest (ROI).

The words are then matched against a defined list of words. We are using the base set of words in the Vuforia word -list as a reference for matching text elements that appear in the ROI in camera field-of-view.

Word lists are loaded from a binary file encoded in the Vuforia Word List (VWL) format. A default VWL file contains over 100,000 high-frequency English words and is distributed in a bundle with the Vuforia SDK. The default word list can be extended with additional word lists and the word list can be filtered.

The Vuforia text recognition engine words and wordlists utilize the UTF-8 character encoding standard and recognise the following characters - space, ‘ , -, A to Z, a to z.

5.1.1. Canvas GUI Design

The canvas of our app is segmented into five different sized rectangles.

- Region Of Interest (ROI) or Text Recognition Region Rectangle
- Background Rectangles A, B, C, D.

Basically we want to define a region of text recognition so that when we point our camera to a page containing lot of words, we don't end up recognizing all the words on that page. We only need to recognise one word at a time.

The text area rectangle serves as our region of interest and enables user to effectively position the camera over the desired word. To effectively cover up the background region we have to divide the leftover area into five rectangles namely A, B, C, D.

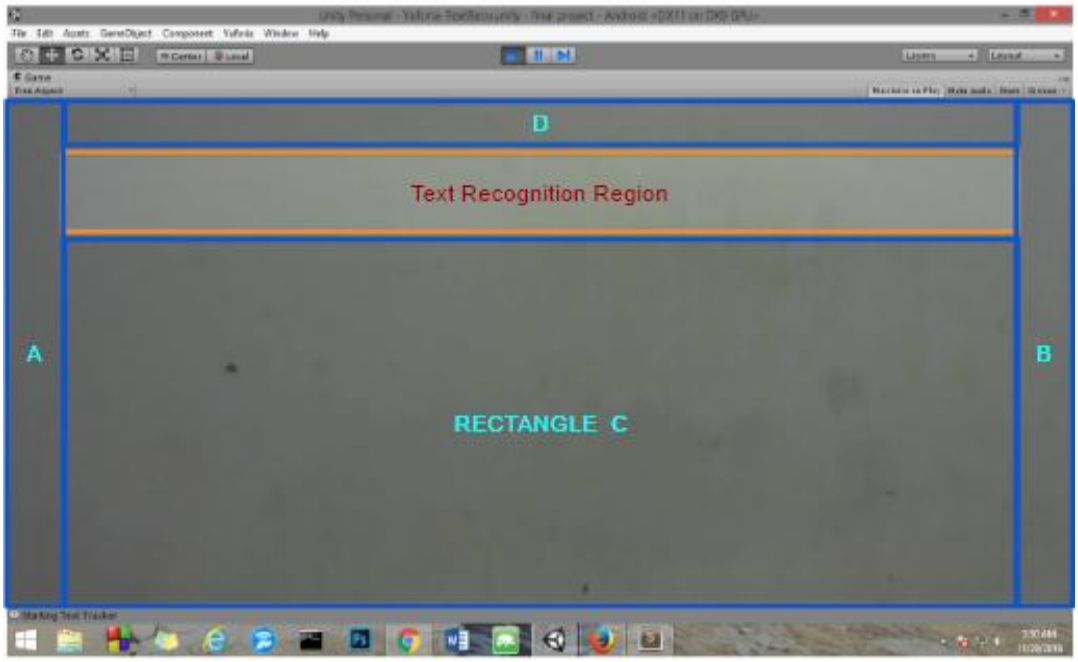

Figure 2 : Canvas with the ROI and background rectangles

5.1.1.1. Creating Region of Interest Rectangle:

We want width of our ROI rectangle to be 90% of the screen width and height to be 15% of the screen height.

- We initialise our mLoupeWidth with value 0.9 and multiply it with screen.width to get 90% of the screen.width in pixel value.
- mLoupeHeight with 0.15 to get 15% height of the screen in pixel value.
- Left offset is half of the region left after substracting ROI rectangle width from screen.width.
- Top offset is take equal to be left offset as it gives suitable position to the ROI rectangle in the run time.
- Finally rectangle have been initalised with X offset equal to leftOffset and X width equal to textAreaWidth, Y offset equal to topOffset, Y height equal to textAreaHeight which is 15% of the screen height.

```
722  private void CalculateLoupeRegion()
723      {
724          var textAreaWidth = mLoupeWidth * Screen.width; // 90%
725          var textAreaHeight = mLoupeHeight * Screen.height; // 15%
726          var leftOffset = (Screen.width - textAreaWidth) * 0.5f;
727          var topOffset = leftOffset;
728          mDetectionAndTrackingRect = new Rect(leftOffset, topOffset,
729                                               textAreaWidth, textAreaHeight);
730      }
731
```

Figure 3 : Snippet for creating ROI Rectangle

5.1.1.2. Creating background Rectangles:

To design the background region we had to break it into five rectangles and design those rectangles individually.

- the rectangle A has 0, 0 X and Y offset, it's width is equal to the min X coordinate value of ROI rectangle and height equal to screen height.
- The rectangle D has X offset equal to X min coordinate of ROI, Y offset set to 0, X width equal to ROI width and height equal to top offset of ROI rect.
- The rectangle B has X offset equal to X_{min} coordinate of ROI, Y offset set to ROI bottom side distance from top, X width equal to ROI width and height equal to left over of screen height after subtracting Y_{max} of ROI.
- The rectangle C has X offset equal to X max coordinate of ROI, Y offset set to 0, X width equal to screen width - ROI right side X coordinate value and height equal to top offset of ROI rect.

```
604  private void DrawOuterRectangle(Rect rectangle)
605      {
606          GUI.DrawTexture(new Rect(0f, 0f, rectangle.xMin, Screen.height), mBackgroundTexture);
607          GUI.DrawTexture(new Rect(rectangle.xMin, 0f, rectangle.width, rectangle.yMin), mBackgroundTexture);
608          GUI.DrawTexture(new Rect(rectangle.xMin, rectangle.yMax, rectangle.width, Screen.height - rectangle.yMax), mBackgroundTexture);
609          GUI.DrawTexture(new Rect(rectangle.xMax, 0f, Screen.width - rectangle.xMax, Screen.height), mBackgroundTexture);
610      }
611
```

Figure 4 : Snippet for creating the four background rectangles

5.1.2. Dynamically creating Bounding Box around Recognised words

Our text tracker algorithm can recognise multiple words at a time. But since the user can not know which word it has recognised and is showing results based on the word it detected. There could be situations where it can recognise words partially or incorrectly if the camera is not properly focused onto the word.
So we are creating a bounding box around the words it has detected so that user can know if it has correctly recognised the word or not. User can then change the orientation and position of web cam to properly focus the right word before the cam.

Figure 5 : Bounding Boxes (in red) around words "Corporate", "Governance"

Following are the steps to create bounding boxes are given below:

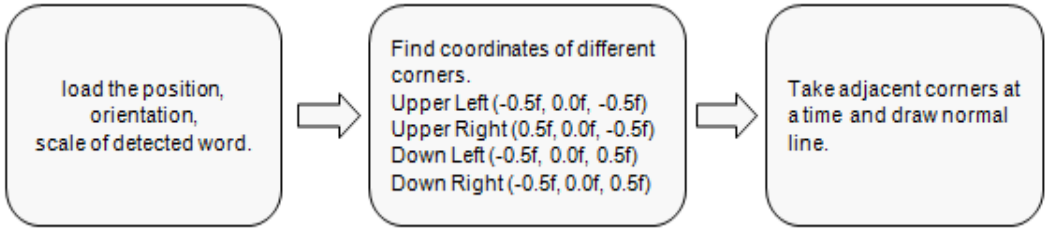

Figure 6 : Steps to create bounding boxes

- **Load the position, orientation, scale of detected word**

First we are loading the run time position, orientation, size of the detected word into the temporary variables. These values are updated in each frame based on the movement of detected word in front of the camera.

```
foreach (var word in mSortedWords)
{
    var pos = word.Position;
    var orientation = word.Orientation;
    var size = word.Word.Size;
    var pose = Matrix4x4.TRS(pos, orientation, new Vector3(size.x, 1, size.y));
```

Figure 7 : Snippet to load the position, orientation, scale of each detected word

- **Find coordinates of different corners.**

We then get coordinates of four corners of the word by using the width, height and pos of the word.

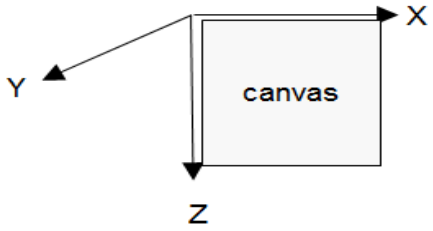

Figure 8 : Alignment of X, Y, Z axes in the co-ordinate system being followed

The coordinates of detected word lie in X-Z plane where X is along the width of word, Z is along height of word.

- Upper Left corner is calculated by subtracting half of width from center X coordinate of word.
- Y coordinate is calculated by subtracting half of height of word from center Y coordinate of word.

Similarly other coordinates are calculated.

```
var cornersObject = new[]
    {
        new Vector3(-0.5f, 0.0f, -0.5f), new Vector3(0.5f, 0.0f, -0.5f),
        new Vector3(0.5f, 0.0f, 0.5f), new Vector3(-0.5f, 0.0f, 0.5f)
    };
var corners = new Vector2[cornersObject.Length];
for (int i = 0; i < cornersObject.Length; i++)
    corners[i] = Camera.current.WorldToScreenPoint(pose.MultiplyPoint(cornersObject[i]));
```

Figure 9 : Snippet to calculate co-ordinates of corners of detected word

- **Take adjacent corners at a time and draw normal line**

Now we are taking adjacent corners in pair and drawing a line between them as normals.

```
var normals = new Vector2[4];
for (var i = 0; i < 4; i++)
{
    var p0 = corners[i];
    var p1 = corners[(i + 1)%4];
    normals[i] = (p1 - p0).normalized;
    normals[i] = new Vector2(normals[i].y, -normals[i].x);
}
```

Figure 10 : Snippet drawing a normal between every two corners to form a rectangle

5.2. Real Time Web Crawling

Web crawling makes it easier for search engines to return the most relevant results to users after they enter a search query. The "web crawlers" systematically crawl pages and look at the keywords contained on the page, the kind of content, all the links on the page, and then returns that information to the search engine's server for indexing. When a search engine user enters a query, the search engine will go to its index and return the most relevant search results based on the keywords in the search term.
Here we crawl the links for Google images and thefreedictionary.com to get the most relevant content and closest possible match for our target word.

5.2.1. Dynamically Loading Meaning and Images

Following is the flowchart of the per frame execution that is taking place inside our unity editor.

- We consider the first word from the list of the words that are detected.
- If meaning or image related to the word is not loaded we are calling suitable function to download meaning and images.
- We take care that at a time more than one download does not occur, that's why we have placed the download subroutine in the critical section and access to it is controlled by a mutex lock. At a time, only one of the download processes can occur so that speed of downloading does not further drop.

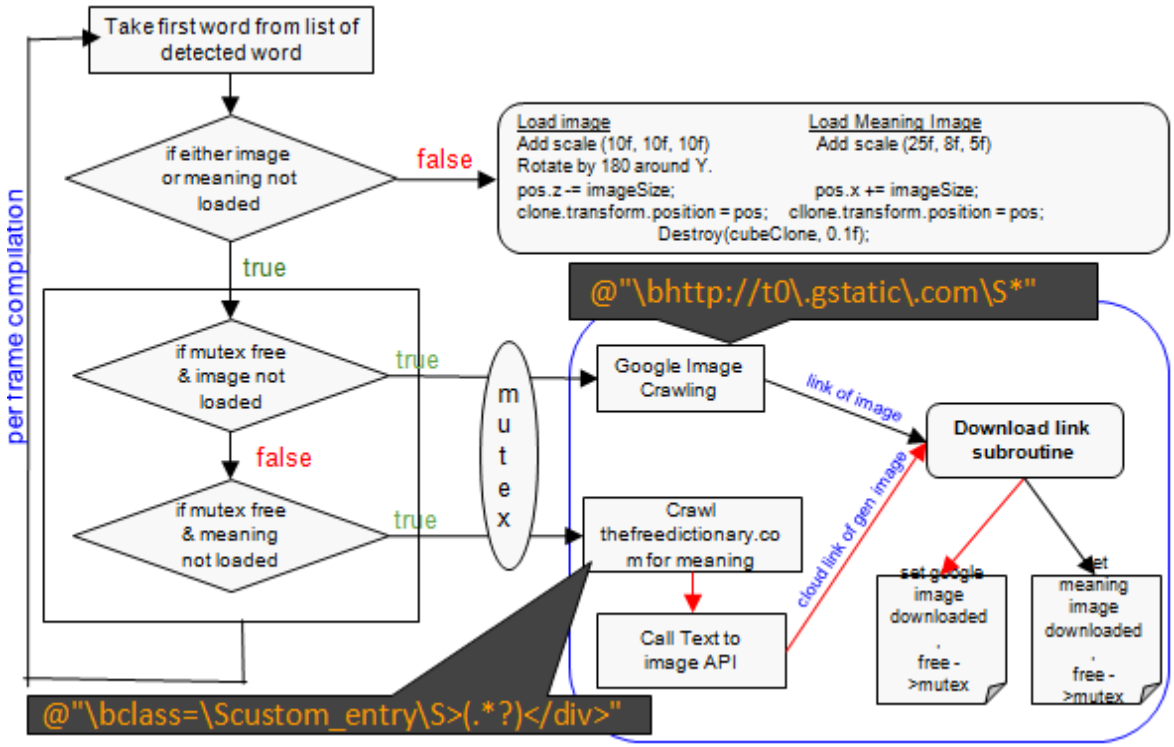

Figure 11 : Flow chart showing real-time crawling and loading of image and meaning

- Critical section is free from per frame execution paradigm as it is been executed in a subroutine so it is not constrained to be completed in each frame.
- When download of the contents is completed for the first word at the current moment in the list of detected words, the first 'if' goes false and we place the content relative to the word in augmented reality.

5.3. Dynamic Positioning of contents in Augmented Reality

Once the position of the word is detected and downloading of the crawled content is finished, we move on to section of code to place the contents relative to the queried word at run time.

Positioning of the Google Image

We map the image to the game-object plane, locally scale it to 10 units of its original size, update its position to be below the word by value of its width (since image is square its height and width will be same).
We are destroying the image after completion of each frame so that previous copy of image from previous frame is not visible onto next frame execution. Here we have taken destroy rate to be 0.1 sec.

```
if(dict[str]){
    GameObject cubeClone = Instantiate( dict[str] ) as GameObject;

    cubeClone.transform.localScale = new Vector3(10f, 10f, 10f);
    cubeClone.transform.RotateAround(transform.position, transform.up, 180f);
    Debug.Log("size -> " + cubeClone.GetComponent<Renderer>().bounds.size.x);
    float imageSize = cubeClone.GetComponent<Renderer>().bounds.size.x;
    pos.z -= imageSize;
    cubeClone.transform.position = pos;
    Destroy(cubeClone, 0.1f);
}
```

Figure 12 : Snippet for the positioning of downloaded Google image

Positioning of the Meaning Image

We map the image to the game-object plane, locally scale it to 25 units in X:5 units in Y of its original size, update its position to be below the word by value of Google image's width and move in Z coordinate by value of its own width.
We destroy this image too after completion of each frame so that previous copy of image from previous frame is not visible onto next frame execution. Here we have taken destroy rate to be 0.1 sec as before.

```
if(dictMeaning[str]) {
    GameObject cubeClone = Instantiate( dictMeaning[str] ) as GameObject;

    cubeClone.transform.localScale = new Vector3(25f, 8f, 5f);
    cubeClone.transform.RotateAround(transform.position, transform.up, 180f);
    Debug.Log("size -> " + cubeClone.GetComponent<Renderer>().bounds.size.x);
    float imageSize = cubeClone.GetComponent<Renderer>().bounds.size.x;
  //  pos.z -= imageSize;
    pos.x += imageSize;
    cubeClone.transform.position = pos;
    Destroy(cubeClone, 0.1f);
}
```

Figure 13 : Snippet for the positioning of meaning image

6. Challenges

- 3D models are not available for free on the internet.
- API for Images and meaning of the word are not available.
- Downloading of the contents has to be done sequentially and through a mutex lock so that different words don't start the download consequently.
- Downloading of content has to be done in a separate subroutine independent of per frame sequential execution otherwise frames freeze.

7. Results

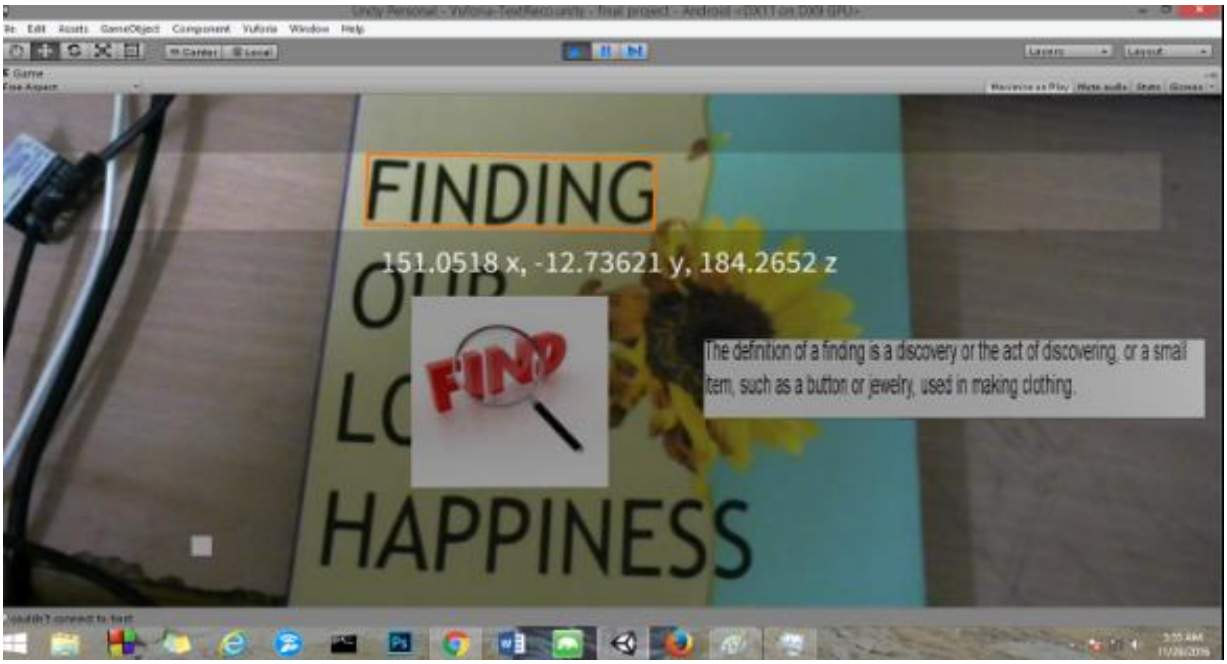

Figure 14 : Final output showing the image and meaning of the word "finding" in Augmented Reality

Figure 15 : Labeled output with the co-ordinates of each detected word, and the image, meaning of first word which is "Corporate" here

8. Hardware and Software Requirements

Software/Tools Used:

- C# Scripting
- Unity 3D
- Microsoft Visual Studio 2013 or later
- Vuforia SDK
- Sublime Text

Other Requirements:

- Web Camera
- **OS**: Windows 7 SP1+, 8, 10; Mac OS X 10.8+,Android
- **GPU**: Graphics card with DX9 or DX11 with feature level 9.3 capabilities
- **Android(if mobile app)**: Android SDK and Java Development Kit (JDK)

9. Activity Time Chart

Work done Pre-Mid Semester				Work done Post-Mid Semester		
Phase I : August 15 - August 20	Phase II : August 21 - August 27	Phase III : August 28 - September 4	Phase IV : September 5 - September 20	Phase V : October 15 - October 30	Phase VI : November 1 - November 15	Phase VII : November 16 - November 28
Literature Survey	Setting up Visual Studio and learn C# Scripting	Learning to work On Unity and Vuforia	Implement Text Recognition and its Mobile App	Learn mapping of text to illustrations in Augmented Reality	Dynamic Web Crawling for meaning and images	Positioning of crawled contents relative to the word in Augmented Reality

10. Conclusion and Future Scope

We successfully implemented real time text recognition for about one million English words in the Vuforia word list. We further extended our algorithm for dynamic real time web crawling of the meaning (from thefreedictionary.com) and image (from Google Images) of the detected words in the region of interest. We next dynamically positioned the crawled content relative to the recognized word at runtime in augmented reality. It is a noteworthy point that we have not restricted our software to work on a static and small database because it not only saves us from manually mapping images and meaning to every desired word but also would have restricted us to use only a subset of possible English words.

Though this software successfully gives a realistic insight into the meaning of English words still by extending our project to use 3D Models and/or Videos it will open another dimension in how we understand our day to day words we come across. We can also build a central database on cloud storing images, videos, 3d model, meaning related to word which will further make rendering of object in augmented reality much faster. We can also select more specific and relevant images or models for words according to our choice instead of simply relying on Google images. Support for different languages apart from English will enable people across world to use the app.

11. References

[1] Hideaki Uchiyama, Eric Marchand;" Object Detection and Pose Tracking for Augmented Reality: Recent Approaches"

[2] Youngo Lee, Jongmyong Choyee, "Tideland Animal AR : Superimposing 3D Animal Models to user defined targets for Augmented Reality Game"

[3] Marc Petter, Victor Fragoso, Matthew Turk, Charles Baur, "Automatic text detection for mobile augmented reality translation"

[4]https://unity3d.com/learn/tutorials/topics/scripting/coding-unity-absolute-beginner

[5] https://library.vuforia.com/tutorials

[6] http://www.thefreedictionary.com

[7] https://docs.unity3d.com/Manual/UnityManual.html

[8]https://library.vuforia.com/articles/Solution/How-To-Implement-Text-Recognition-in-Unity