

# Building Mobile and Web Apps

*Ramanpreet Kaur(Mat. No. 746930)*  
[ramanpreet.kaur@informatik.hs-fulda.de](mailto:ramanpreet.kaur@informatik.hs-fulda.de)  
*Mythri Manjunath(Mat. No.344560)*  
[mythri.manjunath@informatik.hs-fulda.de](mailto:mythri.manjunath@informatik.hs-fulda.de)  
*Global Software Development(W.S 2016/2017)*  
*Fulda University of applied Sciences, Fulda*

## 1 Introduction

This project is based on the paper “A Markerless Augmented Reality System for Mobile Devices” [1]. In this paper an end to end markerless AR pipeline is developed.

“**Markerless AR**” is a term used to denote an Augmented Reality application that does not need any pre-knowledge of a user’s environment to overlay 3D content into a scene and hold it to a fixed point in space. Until recently, most AR fell under the category of “marker-based AR,” which required the user to place a “tracker” — an image encoded with information that’s translated by complex software to produce a 3D object that maintains special orientation within a scene in order to achieve the desired effect [2].

Marker-based implementation utilizes some type of image such as a QR/2D code to produce a result when it is sensed by a reader, typically a camera on a cell phone. Fiducial marker which is traditional marker in which the camera recognizes the image and augments respect to that image. Markerless AR is often more reliant on the capabilities of the device being used such as the GPS location, velocity meter, etc. It may also be referred to as Location-based or Position-based AR [3].

In this project, feature/edge detection is implemented using OpenCV’s Canny Edge Detection Algorithm, and two client applications are developed using Markerless AR.

## 2 System Architecture

Natural Feature Tracking is an image-based tracking technique that detects and tracks the features that are naturally found in the image itself. These could be corners, edges, blobs, etc., without using specifically designed ID markers. The tracking pipeline can be sketched as shown in Figure 1. [4].

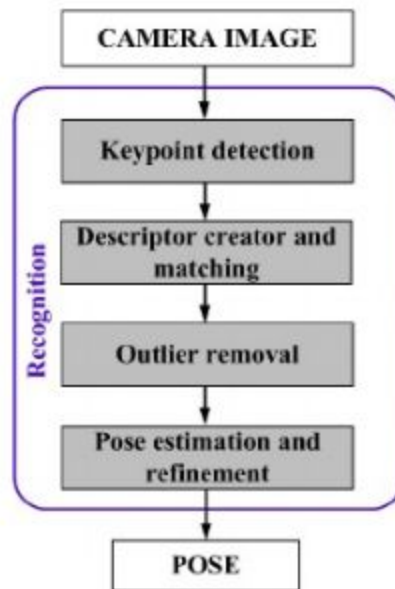


Figure 1 : Natural Feature Detection AR Pipeline

### 2.1 Feature/Edge Detection

The system uses the OpenCV's Canny edge Detector implementation to determine the edges of the image and display them on the camera. The **Canny edge detector** is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. The Process of Canny edge detection algorithm can be broken down to 5 different steps:

- Apply Gaussian filter to smooth the image in order to remove the noise
- Find the intensity gradients of the image
- Apply non-maximum suppression to get rid of spurious response to edge detection
- Apply double threshold to determine potential edges
- Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

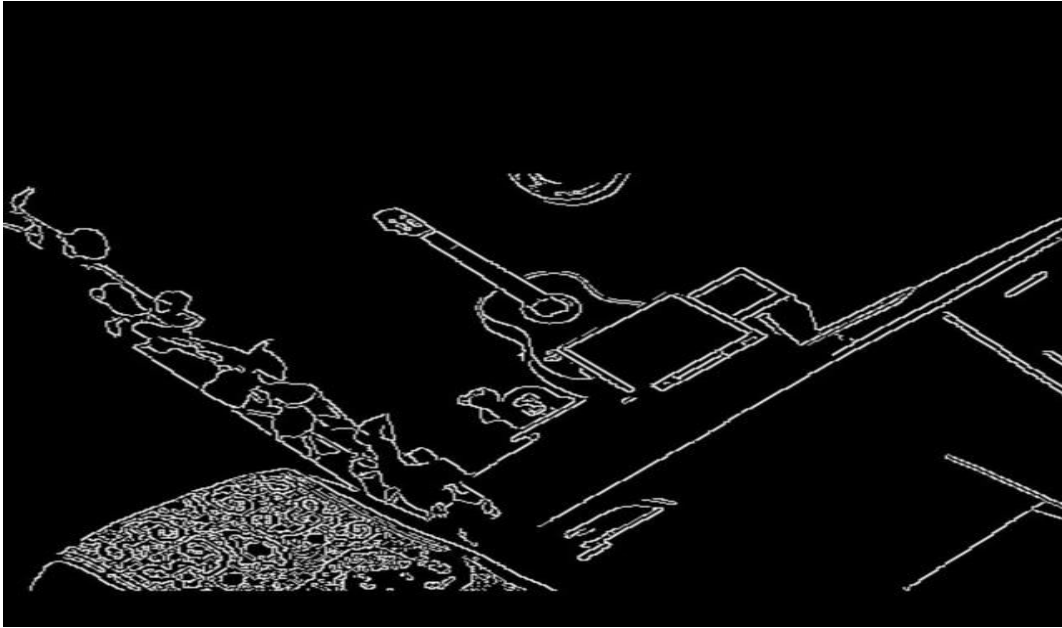


Figure 1: Output of Feature Detection

### **3 Client Applications**

#### **3.1 Jet Fighter**

This application is based on Markerless Augmented Reality(AR) and is built using Unity 3D game engine. It is a simple game where a user can fire bullets to kill the enemies which are visible on the top of the live video feed. The app will work for Android devices.



Figure 2: Output of Jet Fighter

### 3.2 3D Augmentation Overlays

This application for augmented reality uses the orientation of your mobile's gyroscope to orientate the camera in the virtual world to line up the virtual content over the top of a live video feed. This application displays a 3D model of a goblin over grid surface on the pop of application. This application is developed for android platforms and is tested on two android phones.

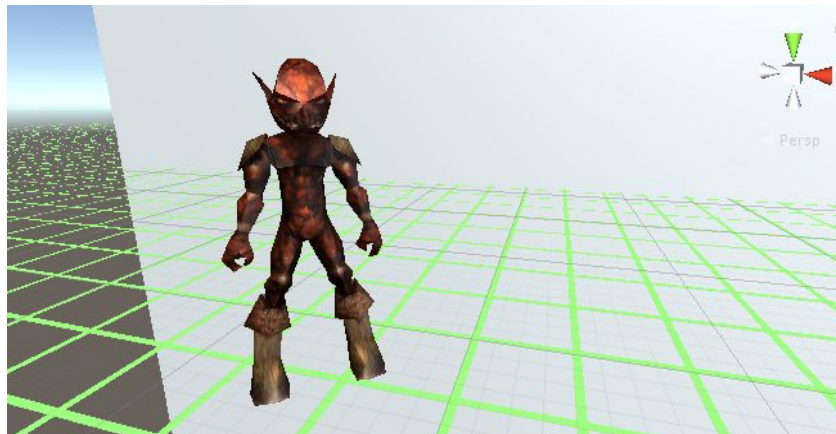


Figure 3: Output of 3D augmentation overlay

## 5 Summary

As a result of this project we will have two client applications: 1) JetFighter Game - A simple game where a user can fire the bullets and kill the enemies. 2) 3D Augmented Overlays - A 3D model of a goblin over a grid surface which appears on the pop of the application. A part of the Markerless AR pipeline i.e. Feature/Edge Detection - Displaying the edges of the images in the real world, is also implemented as part of the project. These applications are designed for android platforms and are tested on two android mobile phones. Due to shortage of personal apple devices, it was not possible to port these applications to the iOS platform.

## Bibliography

- [1] Alex Ufkes, Mark Fiala. “*A Markerless Augmented Reality System for Mobile Devices*” 2013 International Conference on Computer and Robot Vision DOI: 10.1109/CRV.2013.51
- [2] <http://www.marxentlabs.com/what-is-markerless-augmented-reality-dead-reckoning/>
- [3] <http://researchguides.dartmouth.edu/c.php?g=59732&p=382858>
- [4] [http://www.academia.edu/10351349/Marker\\_Based\\_vs.\\_Natural\\_Feature\\_Tracking\\_AR](http://www.academia.edu/10351349/Marker_Based_vs._Natural_Feature_Tracking_AR)
- [5] <http://opencv-python-tutroals.readthedocs.io>