

# Intelligent System Design for Security using Augmented Reality(AR)

Hameed Ullah Jan, Salman Ahmed, Aftab Wahab

**Abstract**— Augmented Reality(AR) revolutionized Military training, education, games and e-commerce industries by generating 3D virtual objects in the real world. The main problem we tackled in this research is the security and privacy issues of military bases using the technology of Augmented Reality(AR). This problem is solved using 3D object scanning via Vuforia and face detection, face recognition, facial expression and age estimation using Visage SDK. Augmented Reality(AR) is an emerging technology, and this technology can be used to give a complete solution in the security sector. Using this research and development, more precise and accurate solutions can be developed for more complex security systems. The visage SDK gives best results for face recognition and detection and with the change of lighting condition and pose variation, it gives 99% accuracy in facial detection and approximately 98% accuracy in face recognition. 3D object scanning gives better result if the object is placed in the same orientation as the training dataset, with a change in the orientation the detection is acceptable but not very accurate.

**Index Terms**— Augmented Reality(AR), Virtual Reality(VR), Software Development Kit (SDK)

## 1 INTRODUCTION

In recent year's security and privacy issues is the main concern of every organization especially sensitive regions like military bases, technology is playing a vital role in providing the solutions for this problem in which Augmented Reality, Object Recognition, and behavior Recognition is of prior importance because of its performance and accurate results. Augmented Reality (AR) is considered a new research area, which involves the overlay of computer graphics in the real world [1]. Augmented Reality (AR) is based on techniques developed in Virtual Reality [2] and interacts not only with a virtual world but has the ability to Augment virtual objects in a real-world environment. Augmented Reality along with pattern recognition and advanced digital image processing can easily solve the security problems and can give better results than conventional methods of securing sensitive areas. Object and Behavior detection in augmented reality ensures detection of any 3D object and its behavior in real time, and also with greater accuracy than other technologies used. AR Technology is used to optimize the production of such products; remote assistance and design of security based systems, or object and pattern recognition in real time. Using these technologies combine we can scan the scene under observation in real time and can even manage it remotely, and instructions from multiple stations can be sent by experts and can communicate with on-site available resources who receive real-time audio and video communication as well as AR instructions. Using Unity 3D as a tool for the implementation of all these technologies can add to this research because of its wide range of features and support for these technologies especially AR with Vuforia, ARCore, and ARKit.

As from the title of the research it is clear that we will be designing a real-time security system for sensitive areas by using Augmented Reality(AR). One of the main problems in this research is Object recognition in AR, as it is important for the detection of danger and suspicious behavior of external entity entering into the area. Using the technologies developed for Augmented Reality (AR), like Visage SDK and Vuforia, the problem of object detection and recognition can be solved if we

use a combination of these to SDK's with a little modification.

For the simulation of this system we have used Unity3D. Unity 3D is used for wide range of applications of computer vision, Games and 3D animations but after the release of AR technology for unity 3D, like Vuforia, ARKit by Apple and ARCore by Google, Visage SDK, Wikitude SDK for Unity 3D, it is now used for Image processing and object recognition also, and giving better results. Unity3D is the most widely used game Engine due to its cross-platform nature; Application developed in Unity3D can work on different platforms like Android, IOS, Windows, etc. Unity3D is considered among the best platforms which provide a lot of plugins and algorithms for Augmented Reality(AR) and Computer Vision, and as it is cross-platform so a complete system for a lot of operating systems can be developed with one code base, and can maximize its reach to multiple platforms.

Object recognition in augmented reality is the key element, which involves registration of a 3d object in Real-time environment [3]. Object recognition in Augmented Reality(AR) must be done accurately to combine the real camera scene and rendered graphics simultaneously. In non-real scenario the objects in Augmented reality are mostly pre-defined and static, but for systems in real life the object is neither pre-defined nor a single set of objects in our data set that can be targeted, so an intelligent system is required to map both the rendered scene and camera, with initialized object and also pattern of initialization and behavior of object initialized in the scene [4]. Object recognition or object detection in AR often refers to identify both the form and shape of an object and its position in real time in the device camera. Object recognition is very severe across all AR application which needs enhancement of the real world scenes and also an overlay of computer-generated graphics or text and video. Most of the tools available for implementing these features have one or more flaws and so needs to be addressed accurately else will be losing too many features.

Augmented Reality (AR) is the new technology which is still in development and enhancement stages, for this project in spe-

cific and AR projects in general, there are three main characteristics of augmented reality: (1) Combination of the 3D objects with the real world objects; (2) Interactivity among virtual objects and real environment; (3) Registration in 3D [5].

This paper is organized in the following manner, in Section II, we presented the implementation and methodology of the proposed algorithm; a detailed flow chart of the whole algorithm is presented in the first part of the same section. The experimental results of face recognition, face detection, expression recognition, age estimation, and 3D object scanning are presented in Section III. In Section IV, the future work and opportunity for further improvements are being discussed, and a summary of the whole algorithm is being concluded in the same section.

## 2 METHODOLOGY AND IMPLEMENTATION

The main purpose of this research is the detection of danger and suspicious behavior of external non-related entities in the surrounding of a sensitive area and its detection using augmented reality powered by Vuforia and Visage SDK. The complete implementation of this system includes face detection, face recognition, pattern recognition. Face Detection, Face Recognition and expression recognition is done by using Visage Software Development Kit (SDK), inside Unity3D. The pattern of the wearing Uniform is being detected and recognized inside Unity3D by using Vuforia. Expression detection and age estimation is done through visage SDK, the purpose of expression detection is to make it sure that a person entering the base is wearing a face-mask or not, this process is also carried out through Visage SDK.

The complete process behind this research is explained in the flow chart shown in Fig. 1. The first stage involves the detection of a new entity entering to the sensitive area, when the system detects a new entry it stores the information of the face and uniform of the incoming entity. The face detection phase involves a check for the detection of the face-mask, if the new entity is wearing a mask the system will generate an Alert, specified as Alert\_1. Alert\_1 represents that the new entity is wearing a face-mask and should be reported as suspicious behavior, this alert is being sent to the control unit through firebase notification, which is done inside Unity3D. After getting this notification the control unit will take the corresponding action. If the new entity does not wear a face-mask then the system will go to the face recognition phase, which is done by using visage face recognition algorithm, the system will check if the new entity exists in the database or not, if the new entity exists in the database, the entity is given access to enter the base, if the face doesn't exist in the database, check for the uniform of the new entity if the uniform is being detected as the uniform of other employees, send Alert\_2 to the control unit. Alert\_2 means that the new entity is wearing an employ's uniform but the face doesn't exist in the database, which is also notified as suspicious behavior, and the control unit will take action accordingly. The third case is if the new entity's face doesn't exist in the database and also the uniform is different, the system will generate Alert\_3. Alert\_3 means that the new entity is not a known person, the control unit will take action accordingly. After looking into all these parameters, the system takes intelligent decision based on all the features considered in this process.

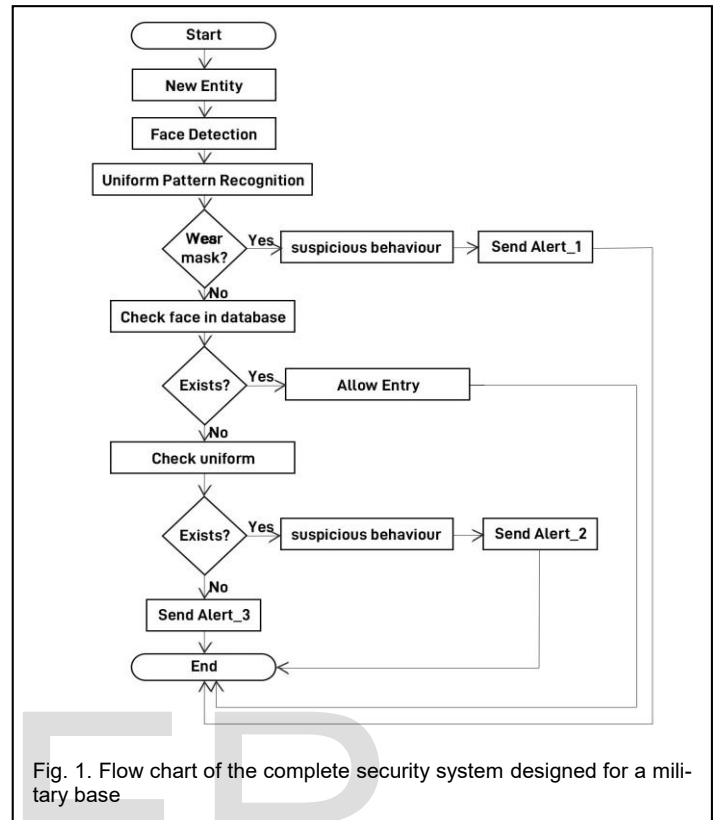


Fig. 1. Flow chart of the complete security system designed for a military base

### 2.1 Integration and Configuration of Visage SDK

The implementation of Face Recognition, Face Detection, Pose Estimation and Age Estimation through Visage SDK involves the following steps:

1. Create a new Unity Project
2. Import VisageTrackerUnityDemo.unitypackage (provided in Samples/Android/VisageTrackerUnityDemo folder) by selecting Assets->Import Package->Custom Package... item from the menu.
3. Select and open the "MainScene"
4. Following files need to be copied to the folder Assets/Plugins/Android:
  - 1) Libraries libVisageVision.so and libVisageTrackerUnityPlugin.so (located in lib/armeabi-v7a)
  - 2) AndroidCameraPlugin.jar and AndroidManifest.xml (located in Samples/Android/AndroidCameraPlugin/app/release)
5. Following files and folders (located in Samples/data) need to be copied to the folder Assets/StreamingAssets/Visage Tracker:
  - 1) btdsdata folder
  - 2) jk\_300.wfm
  - 3) jk\_300.fdp
  - 4) configuration file (e.g. Head Tracker.cfg)

After the implementation of these steps you can build your project of any android device. After running, the application displays three buttons: Detect, Match Faces and Analyze Face. De-

tect button loads an image from the device gallery. If any features were detected, they are drawn on the loaded image and shown on the screen. Match Faces button loads an image bundled with the sample application and performs face recognition on it. As a result, the name of the recognized person (or the name of the person with the most similar face found in the VisageFaceRecognition gallery) will be drawn on the loaded image and shown on the screen. Analyze Face will perform Face Analysis on faces detected in the image loaded from the device gallery and display gender (if estimated gender is female, pink circle will be drawn around the face and if it is male, the circle will be blue) and age estimation for all faces and emotion estimation on the biggest face found in the image.

Visage SDK comes with several standard configuration files aimed at common usage scenarios. Table.1 provides an overview of all available configurations.

TABLE 1  
STANDARD CONFIGURATION FILES FOR VISAGE SDK

Configuration file name	OverView
Head Tracker.cfg	Optimized for high-performance head pose tracking.
Facial Features Tracker - High.cfg	Facial features tracker optimized for real-time operation from camera or video files on high-performance mobile devices such as iPhone6 as well as all desktop/laptop computers. Tracks head pose, mouth, eyebrows and eye motion.
Facial Features Tracker - Low.cfg	Facial features tracker optimized for real-time operation from camera or video files on low-performance mobile devices such as iPhone4S. Tracks head pose, mouth, eyebrows and eye motion
Face Detector.cfg	Used in face detection.

Some of the Genral configuration and setup guidelines are give bellow:

1. Determine camera\_focus parameter
2. The room and the face should be well lit. User can experiment with different types of lighting (indirect daylight is usually the best, neon lights the worst).
3. User should disable automatic adjustment of the camera settings by the driver like gain, exposure, white balance and similar and set them manually, if possible, depending on the camera used and lighting conditions

## 2.2 Integration and Implementation of Vuforia inside Unity3D

Object Recognition using 3D scan technique through Vuforia enables you to make applications that can perceive 3D objects, such as vehicles, human body, toys and so forth. In this section, we will show you how to add Object Recognition and Object Targets to a Unity3D project.

To add target object into your project, navigate to GameObjects>Vuforia>3D Scan, this will create an object "ObjectTarget" GameObject in the hierarchy. You will be using the "ARCamera" and "ObjectTarget" GameObjects. The following steps should be followed to create a dataset which contain the "ObjectTarget":

1. Navigate to the Target Manager
2. Click the Device Databases tab. You will find a list of existing databases
3. In this list click on the one where you want to add the Object Target
4. Click Add Target, the adding target menu will appear
5. Select one of the presets, that you want to use as a target
6. Enter the name of the target, the name must be unique, descriptive, and can only contain numbers, letters and underscore.
7. In the right bottom, you will see Add button, click this button and wait for a while until the processing is done when the processing finishes another page will open which will show that the target is uploaded.
8. Now you can download that database as a \*.unitypackage
9. Open the \*.unitypackage, and import it into unity, you will find the database into the "StreamingAssets" folder, inside Unity3D.

To configure the target object, the following steps should be followed:

1. Create an "ARCamera" by navigating to [GameObject>Vuforia> AR Camera], There must be only one camera in the scene so delete all the other cameras from the scene.
2. Create an "ObjectTarget", Navigate to [GameObject> Vuforia> 3D Scan], do not make the "ObjectTarget" or "ARCamera" as a child object of any other GameObject
3. Click the "ObjectTarget", and see in the inspector you will find the "Object Target Behaviour" component. Choose the Dataset name and Object Target name that you want to associate with the "ObjectTarget".
4. Set the position of the "ObjectTarget" to (0,0,0)
5. Add any content that you want to augment the object to the scene
6. Use the bounding box to position and scale your content in relation to the target
7. Make the content a child of the "ObjectTarget"
8. For custom event handling you can either Extend the Default "TrackableEventHandler" attached to the "ObjectTarget" GameObject, or you can create your own event handler script, which must implement ITrackableEventHandler interface.
9. Now the basic scene is ready, you can test by pressing the play button in unity editor.

You can use a boundary box in vuforia to register a 3D object. There are some basic steps that should be followed while positioning the content.

1. Use 2 by 3 layout system in the unity for better positioning, align the GameView and the Scene view side by side, so that both are visible
2. Select the "ARCamera" in the hierarchy and choose the webcam that you will be using for the augmentation
3. Place the object in front of your webcam, so that it is visible to you
4. For better positioning, play the scene from unity and bring your object in front of the camera, align the object properly in the bounding box, when you are completely satisfied by the position, copy that position in the play mode, exit the play mode and give that position to your content object.

### 3 EXPERIMENTS AND RESULTS

The Experimental results can be divided into two parts, in the first part we will discuss the experimental results obtained from Visage SDK and in the second part we will see the 3D pattern recognition done through vuforia.

#### 3.1 Experimental results achieved through Visage SDK

This section includes three major parts: Face Detection, Face Recognition and Expression Detection.

Visage SDK Tracking algorithm tracks 3D head pose, which includes Eye tracking, facial features tracking. All these features are detected in a video coming from the camera. The experimental results from an Android device are shown in Fig.2. This Figure shows single face detection under different background condition. It is clearly shown how Visage SDK tracks eyes, lips, and other facial features.

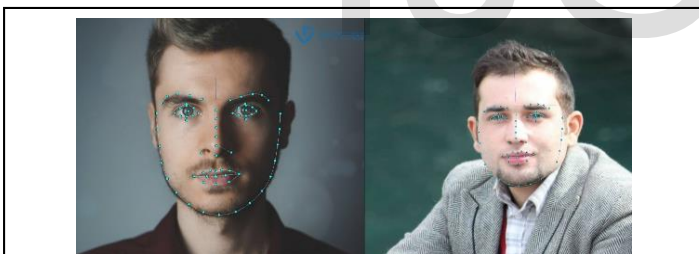


Fig. 2. Single face detection with a variation in background conditions

Fig. 3 shows face detection under different lighting condition and pose variations.



Fig. 3. Face detection under different lighting conditions and pose variations

The face recognition experiment was done by adding the Olivetti Research Lab (ORL) dataset as a gallery to the visage SDK. ORL database consists of 400 images of 40 subjects these images were taken under different lighting conditions and pose

variations. Each subject has 10 images so there are 40 subjects, the total dataset contains 400 images. The face recognition experiment through visage SDK was done by following these steps:

1. Store these 400 images as Gallery for the visage SDK
2. Provide one image for testing
3. Visage SDK will show the result as to which category this image belongs

As the ORL dataset consist of 400 images, out of 400 images 399 images are stored in gallery for the visage SDK and provided one image for testing, the visage SDK checks if the current test image or face is present in the gallery. Since there are 40 subjects and each subject has 10 images, an id is being assigned to each subject. For testing purpose one image is provided as test to the visage SDK, the visage SDK returns the id of the image. This id specifies that whether the required test image's id is same as the output or not. If the id is same, it means that it is recognized correctly otherwise it is wrong. We checked all the 400 images out of which 396 were recognized correctly by the visage SDK. Now calculating the recognition rate:

Recognition rate (%) = (correctly recognized images/total images) \* 100

$$\text{Recognition rate (\%)} = (396/400) * 100$$

Recognition rate = 90%

For Expression detection and age estimation the visage SDK is being integrated into both IOS and Android. Fig.4 shows results from the real simulation in android and IOS devices



Fig. 4. Expression and Age estimation using visage SDK

#### 3.2 Experimental results achieved through Vuforia

The Vuforia 3D scanning requires a Target file, we did this test on a toy car which can be replaced by any other character like a military person, etc. This Toy car was first scanned using the Vuforia 3D scanning technique, after scanning, this was stored as a target file for the 3D scanning, Fig. 5 shows the experimental result, the green highlighted area shows the detected area.



Fig. 5. Object Scanned by Vuforia

The complete experiment through Vuforia involves the following steps:

1. There must be a suitable scanning environment, not too congested
2. Set proper lighting
3. Make a plane surface, and specify the area of the scanning
4. Position your target object over the plane surface
5. When all the above steps are done, the scanning process can be started by running the application on android or IOS device

## 4 CONCLUSION

Visage SDK gives approximately 99% accurate results for face recognition, facial features detection, expression recognition, and 85% accurate results for age estimation. The system has been checked for various scenarios such as different lighting conditions, pose variation, the distance between the subject and the camera and multiple subjects in one view but the accuracy remained almost the same. The 3D scanning through Vuforia is currently checked for one object but it can be modified for multiple objects scanning. Although the accuracy of 3D scanning through Vuforia highly varies with the orientation of the object, in all the cases it gives accuracy above 70%. The two main features that are 3D object scanning and facial recognition can be used for many security purposes in the future. For future work 3D object scanning can be improved to work better for different orientation and can be optimized for the detection of multiple objects. It can also be optimized for a huge dataset of objects which can contain multiple shapes i.e. humanoid, vehicles, etc.

ER

## REFERENCES

- [1] H. R. Babaei, P. L. Mohurutshe and A. H. Lashkari, "Image-Processing with Augmented Reality (AR)," Oct 2012, doi: 10.1117/12.2011323.
- [2] R. T. Azuma, et al, *A Survey of Augmented Reality*, Presence, vol. 6, no. 4, pp. 355-385, 1997.
- [3] M. Billinghurst, H. Kato and I. Poupyrev, "The MagicBook: a transitional AR interface," *Computers & Graphics*, vol. 25, pp. 745-753. Elsevier, Amsterdam. 2001
- [4] T. Hollerer, S. Feiner, T. Terauchi, G. Rashid and D. Hallaway, "Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system," *Computers & Graphics*, vol. 23, pp. 779-785. Elsevier, Amsterdam. 1999
- [5] F. Doil, W. Schreiber and T. Alt, "Augmented reality for manufacturing planning," *Proceedings of the workshop on Virtual environments*, pp. 71-76. ACM Press, New York. 2003