



ORBOID: Robotic VR based Application for Remote Regions

Heena M. Agrawal¹, Shyamal G. Mundada²

¹Department of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, 440013, (M.S.), India
agrawalhm2@rknc.edu

heena.agrawal12@yahoo.com

²Department of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, 440013, (M.S.), India
mundadasg@rknc.edu

ABSTRACT

In today's world protection of human and resources is very important in accordance with the rapidly growing population. Security is one of the major concerns of the globe. With the rapidly growing population it has become a global concern for protection of resources to serve the population. Industries all over the world spend a lot of capital in maintenance of resources and building of security system to protect international borders as well as domestic regions. The traditional approaches for security and maintenance are static in nature and those of current portable are highly expensive. Maintenance of gases is important yet difficult. Pipes may undergo corrosion and may lead to leakage of gases. The proposed paper aims to eliminate these hurdles by proposing a cost effective application based and portable solution making use of smart phones and a robot machine. This system enables a user to set up a smart phone on the robot machine and move the robot with the help of controller. The user can then see all the data captured by the device on robot via live streaming or images stored in the firebase. The user can view the remote region in surrounding of robot by moving its own head wearing the VR which compel the movement of robot carrying the device in the same direction as that of user. Thus, this system proves to be of benefit for the maintenance team of pipes and well as for the security team to survey remote regions with human intervention on site.

Key words: Remote, System, Orboid

1. INTRODUCTION

With the rise of technical advancements, various industries are making use of ever developing technologies in order to continue their products/services swiftly. With the development of VR systems, new interaction methods have been developed that allow the user to step into computer generated, or Virtual Environments (VEs) which is achieved by immersing the user in the synthetic environment. In today's time where the humans have high reach from sky to underground levels, these synthetic environments of virtual reality can make a person feel established in unreachable environments for real. Such a concept proves beneficial for humans in various situations.

The main gas pipelines are one of the major sources of fuel for human habitat. These pipes are often of diameter more than 1 meter. The oil and gas companies have to dedicate a good capital and human resource towards the maintenance of these pipes. The proposed approach aims to wave the finance required by making use of a robot and smartphones along with the VR set to observe the insides of the pipes and know about the condition for future workings.

With the growing threats in the 21st century, the concern for security has risen in individuals and also at community levels. The proposed system can serve as a moving security inspection without the intervention of human at the actual site. The human can control the movements from remote place and can view the surrounding to detect potential threats.

Since the user at the remote may not always be able to identify the objects seen on the screen via VR, the proposed system provides the option to analyse the image and return the probable list of objects that could match the objects on the image. The video recording done by traditional approaches can be viewed on the screen to which camera and storage are connected and cannot be viewed at a remote location in real-time. The proposed system presents an approach to view the recording from a remote location via live streaming.

2. REVIEW OF LITERATURE

VR research and development projects are currently underway world-wide in large organizations such as NASA, IBM, Intel, Boeing and Rolls Royce. Concepts came up with a concept which differs from conventional ones. This system use data transmitted from sensors embedded in devices without processing. It made use of embedded sensors that analyses collected data and provides it to customers in an easy-to-process form [14]. Despite the development in virtual reality (VR) technology, haptic interaction within large virtual environments (VEs) remains limited. One of the recent projects developed a VR system incorporating haptic interface using a cable-driven parallel robot with accessible motion capture with multiple Kinect sensors for rehabilitation. [15]

Walking machines which form the robot machine of proposed system have been simulated in various Dc motor

applications providing varied purpose to the developed technology [13].

3. SYSTEM ARCHITECTURE

The project has been divided into 2 sides for operations from two end. One is the user who shall operate the application and another is the robot which will move and capture image or stream live. The overall working will consist of two devices having the application installed. One application will operate in the “user” mode which requires human intervention, another will operate in the “robot” mode which will be placed onto the case on the robot machine as shown in Fig.1. The working can be summarized as:

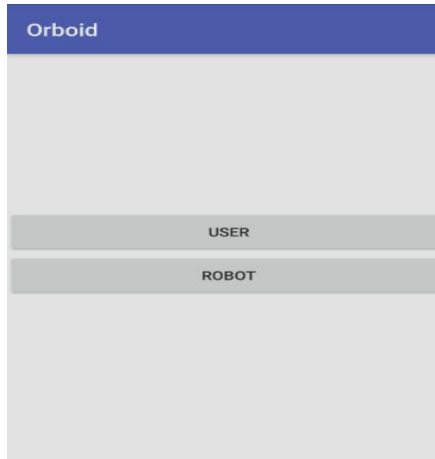


Figure1: Operating Options in Orboid

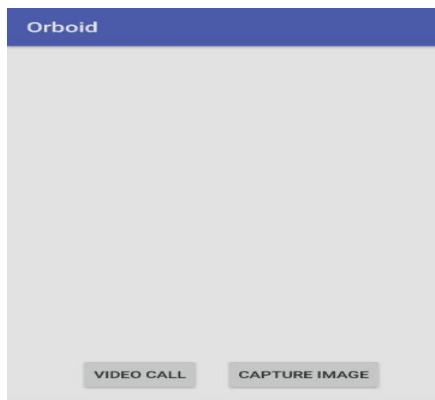


Figure 2: Operations of User Mode.

3.1 User End: The user operating in this mode has two choices of operation as shown in Fig.2

a) **Video Call:** This will connect a video call with another device which is operating in the “robot” mode. The UI has a call button which starts the connection process when pressed. During the preview mode when the call is not connected one can view the host details, token generated for the communication, display name and resource id as shown in Fig 3. Once the call is connected the screen splits into two, to facilitate view from the VR

box. Now the user can see whatever is captured and streamed by the camera on the robot machine via video call. As the user wearing the VR box moves the position of device with “user” mode, the movement of device on the robot machine change. This enables the maintenance team of pipes to view in 180 degree movement from the remote locations and scrutinize to gather information about the condition of pipes[16].

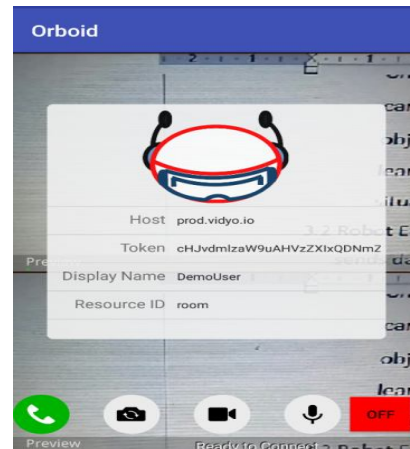


Figure 3: Video Call for Live Stream

b) **Capture Image:** As soon as the user gets into this option, the UI has two options shown in Fig 4. Either you can capture the image or view the previously captured ones by the robot machine which are stored in the database. When the user hits the capture buttons the device on the robot machine will automatically capture image in background and upload it onto the storage which can later be viewed from the gallery view shown in Fig 7. These images later can be selected to give input to the analysis API which returns the list of probable objects in the image called as “tags” shown in Fig 8. This information helps the maintenance team learn in detail about the health of pipes and act accordingly to avoid any catastrophic situation.

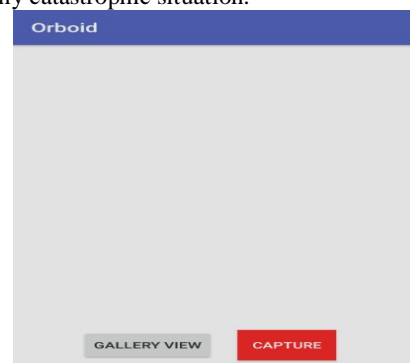


Figure 4: Operations in Capture Mode

3.2 Robot End: The device with this mode is placed onto the robot machine which then either sends data via live stream or uploads image to the database.

3.3 Robot Machine: This machine is placed over wheels which are connected with the servo motors carries the device with “robot” mode to the remote locations shown in Fig 11. The case on which device is placed is connected to

motor and programmed which we shall see in the next chapter. When the user moves the position of device with “user” mode, this case also changes its angles which the motor receives. This feature enables user has a 180 degree view. The combined use of robot machine and application is of robust nature which can perform seamlessly.

4. IMPLEMENTATION

This section lays down the path to execute the system architecture discussed in the previous section. Orboid App system is backed by strong backend: Google’s Firebase which is real-time syncing cloud database. Data is stored and synced across connected devices in milliseconds. Due to its robust and scalable architecture of the robot machine it can travel any distance allowing surveillance and remote and highly compact regions where only a device of this size can move. Orboid is divided into functional modules for different operations to be performed[17]. The system has two modes of operation one is the “user” another is “robot” as shown in fig 5. The user has two options either to choose live stream or gather images of location. There is an initialized variable of two bits to switch operation. When live stream is chosen, bit1 of variable is set to 1 and function of live video stream is activated (explained further). When option of image capture is chosen bit 1 is set to 2. From here there are two operations, to capture image bit 2 if variable is set to 1 and image capture function is activated. To view the gallery and perform analysis of images, bit2 is set to 2 and analysis of image function is called. The movement of robots are controlled by a remote controller. Data transmission takes place through raspberry pi 3 set on the robot machine which also enables the movements of machine[18].

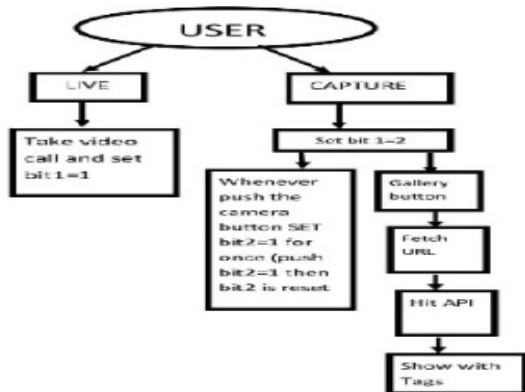


Figure 5: Architecture flow of system

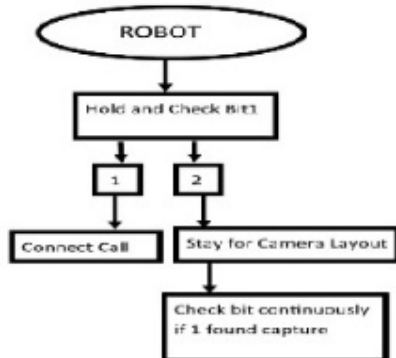


Figure 6: Architecture flow of system

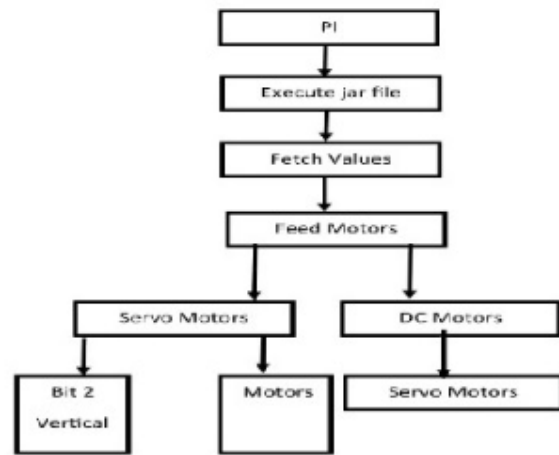


Figure 7: Architecture flow of system

4.1 Live Video Streaming

Live Video Streaming module deals with passing of video streams from mobile device placed at the robot to the mobile device inside the VR Headset. External SDK, VidyoClient SDK [9], for android is used to make an application that can efficient video streaming. Steps for this module are as follows:

1. Register an account in www.vidyo.io which provides SDK for video streaming and generate application ID and Developer Key.
2. Download SDK and place it in main/src/libs folder of the android application.
3. Open a new application in Android Studio and import `:com.vidyo.VidyoClient.Connector.VidyoConnector`
 - `com.vidyo.VidyoClient.Connector.Connector`
 - `com.vidyo.VidyoClient.Endpoint.VidyoLogRecord`
 - `com.vidyo.VidyoClient.VidyoNetworkInterface`
4. In MainActivity, implement interfaces :
 - `VidyoConnector.IConnect`
 - `VidyoConnector.IRegisterLogEventListener`
 - `VidyoConnector.IRegisterNetworkInterfaceEventListener`
5. Make UI such that there are two FrameLayouts to be better viewed in VR Box, buttons for Connect and Disconnect in the view.
6. Initialize Connector inside Activity :
 - `Connector.SetApplicationUIContext(this)`
 - `Connector.Initialize()`
7. Implement Start() method and register for events and camera :
 - `VidyoConnectorvc = new VidyoConnector(null, VidyoConnector.VidyoConnectorViewStyle.VIDYO_CONNECTORVIEWSTYLE_Tiles, 16, "", "", 0)`
 - `RegisterForVidyoEvents()`
 - `vc.SelectDefaultCamera()`
8. Generate Token from the DeveloperKey and ApplicationID by command :


```
java -jar generateToken.jar --key=rUlaMASgt1Byi4Kp3sKYDeQzo --appID=ApplicationID --userName=user1 --expiresInSecs=10000
```

9. Pass it to the Connect() method in application when Connect button is clicked :
 Boolean s =
 vc.Connect("prod.vidyo.io",token,"DemoUser","DemoRoom",this)
10. Pass the video streams to FrameLayout by :
 vc.AssignViewToLocalCamera(videoFrame, vidyoLocalCamera, true, true)
 vc.ShowViewAt(videoFrame, 0, 0, videoFrame.getWidth(), videoFrame.getHeight())
11. Implement Disconnect() method when Disconnect button is clicked: vc.Disconnect()

4.2 Image Capture

This function required implementation of camera functions and thread to capture images in the background onto the device with user mode. The on tick function of the code shown in Fig 6 is an in built thread which takes time in seconds as parameters to perform the activity assigned [11]. Here the value is 2 sec therefor images are captured after 2sec of capture button being hit. Life of thread is 3 sec. Object of camera is used to click pictures. Here, mcamera functions takes the picture when called [11]. A file is written and image is stored. Another path is created in local database and image is stored in the local storage of that device. This local path is then taken in a URI variable and is converted to the type URI [11]. The image is then uploaded to firebase from local storage along with this converted value which forms URL of the image [8]. This uploaded image then serves as input to analysis step.

```
//inbuilt thread for clicking pic in background
//parameters are active till 3 sec and click on every 2 sec
public void onTick(long millisUntilFinished) {
    mCamera.startPreview();
    mCamera.takePicture(null, null, mPicture);
    i++;
    Toast.makeText(getApplicationContext(), "Captured " +k,
        Toast.LENGTH_SHORT).show();
}
```

Figure 6: onTick Function of Image Capture

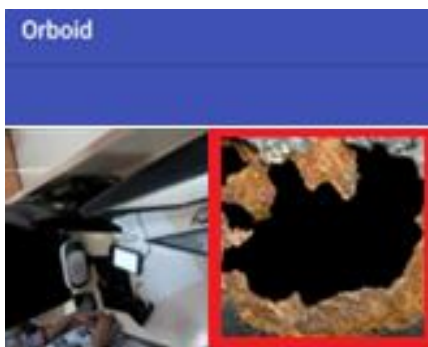


Figure 7: Gallery to view Captured Images

4.3 Analysis of Image

When the robot moves on a horizontal surface, it may encounter obstacles on its path. We need to identify those obstacles and analyse them as well. For that purpose we are using an API which takes as input images and returns the tags associated with those images. The Clarifai API is used for image recognition. This API automatically tags the images and videos by content. It searches and analyses the visual data without relying on additional text descriptions.

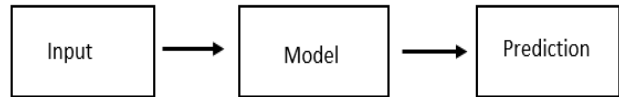


Figure 9: Analysis of images Clarifai API

The Clarifai API offers image and video recognition as a service [12]. Whether we have one image or billions, we are only one step away from using artificial intelligence to recognize our visual content. The API is built around a simple idea. We send inputs (an image or video) to the service and it returns predictions.

The various services provided by Clarifai are as follows [12]:

- Identify unwanted content including gore, rust, etc.
- Recognize clothing, accessories, and other fashion related items.
- Predict the age, gender and cultural appearance of detected faces.
- Detect the presence and location of human faces with a bounding box.
- Computes various embedding vectors using the Face Detection model.
- Identifies textures and patterns within an image.

Clarifai is a REST API of version 2.0 [12]. It offers numerous pre-built and custom models which we can access for free up to 5000 operations. It is built and developed in a secure environment and also supports cloud infrastructure. The API also offers custom model training. We can choose between a model tuned and trained according to the specifications, or we can opt for a brand new neural architecture for even better accuracy [12].

| |
|-------------------|
| Tag: cave |
| Tag: texture |
| Tag: old |
| Tag: dirty |
| Tag: rust |
| Tag: closeup |
| Tag: rusty |
| Tag: invertebrate |

Figure 8: Tags identified by CLARIFY API

The predict API analyses the images or videos and returns a list of concepts with corresponding probabilities of how likely it is these concepts are contained with the image or video.

4.4 Database Configuration

Created nodes in database[8] as follows:

- Bit : First Bit is 1 for live video streaming and 2 for image capturing. Second bit initially set to 0 indicating no images to be captured, when set to 1 indicates capture image
- RobotData: Two subnodes to hold at a time two movement data. First bit indicates direction to move to for dc motor of robot(F-forward, B-backward, L-left, R-right). Second bit indicates angle for servo motor which moves camera on robot vertically. Third bit indicates angle for servo motor which moves camera on robot horizontally.
- Pic_path: This node contains urls of image generated when image is uploaded to firebase server



Figure 10: Configurations of Database

4.5 Real Time Data Transmission

We used client server model for real time data transmission

Client module(user) : Orboid (Android Application)

- Established database connectivity(used firebase as database)
- Establish mechanism for collecting sensor data and processing the data obtained.
- Created a thread which sends all the sensors data with a delay of 0.05 seconds.
- Values of two sub-nodes in RobotData node as mentioned in Fig 5 is set to null one by one as soon as both the nodes are filled.

4.6 Gathering data from sensor:

Collection of Gravity Sensor Data module deals with gathering the readings of gravity sensors of the device placed inside the VR Headset[18]. This is the data needs to be passed in real time to device on the robot through Module 5.1.2 . Steps to get gravity sensor data are as follows :

- Register Sensor Manager[10]

```
SensorManagerSensorManager =
SensorManager.getSystemService(SENSOR_SERVI
CE);
```

- Register Accelerometer and Magnetic Field Sensors
`Sensor mAccelerometer =mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);`
`mSensorManager.registerListener(this,mAccelerometer,SensorManager.SENSOR_DELAY_GAME);`
`Sensor mMagneticField =mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);`
`mSensorManager.registerListener(this, mMagneticField,SensorManager.SENSOR_DELAY_GAME);`
- Getting data from onSensorChanged event
 - Collect data of two sensors in two separate float arrays.
 - Use these two arrays to get RotationMatrix
 - Use Rotation Matrix to get Orientation Float Array which contains 3-dimensional rotational axis data in Radians Measurements.

4.7 Processing Gravity Sensor Data

Processing Gravity Sensor Data module deals with further processing of gravity sensors data. Data collected in Module 5.1.3 from gravity sensors is not in useable form(raw)[17]. We use following algorithm to convert it into required form as follows:

- First convert the required two axis (horizontal,vertical) data in degrees.



- Convert the data along one axis in 0 – 360° format by adding 360o in negative values.
- When we start the app, first values in both axis get fixed as initial values of axes.
- By using this initial value, we define the area of movements of robot that is area of 180°.
- The value received from sensors after the initial value should be convert according to this movement area. Consider value want to process is x.

```
if( initial > 269° ) {
    if ( x < 90° )
        x+=360°;
    }
    else if(initial<90°){
        if(x>270°)
            x-=360°;
        }
        if( initial -x > 0°&& initial - x < 91°)
            x=90° - ( initial - x );
        else if( initial-x < 0°&& initial - x > -91°)
```

```
x=90° - ( initial-x);
else
x=90°;
```

- Normalize the processed values and convert the Normalized values into their ASCII representation Consider processed value is x.
`x/=5; x+=35;`
`strX = Character.toString((char)x);`

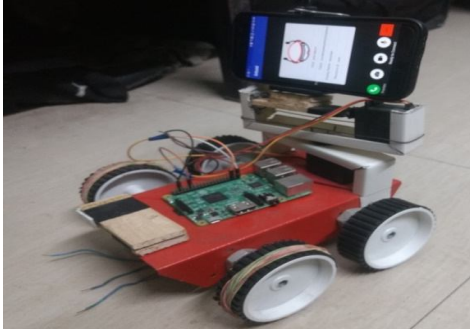


Figure 11 : Robot Carrying Device

4.8 Server (rasberry pi)

- . Execute java program which retrieves movement data from RobotData node of firebase database and controls two servo motors and a dc motor of robot as mentioned in Fig 10.
- Reset angles of motors to an initial position
- Establish connection with firebase[8] and start fetching the angles from subnodes of RobotData node.
- For every set of angles received at an interval of 100ms, start three threads each for three motors
- Threads of vertical and horizontal servo motors work in similar way as follows:
 - a) Take ascii character from data received and convert it into suitable angle(ver)
 - b) `ver = ver -35` : convert ascii value to real number for angle
 - c) `ver = ver *5` : lower degree of precision as we take angles in intervals of 5 degrees
 - o `ver = ver +62` : adding 62 because pi passes 62 as 0 degree angle to servo motor
 - d) Take two such angles for current and previous angles
 - e) If this current angle(ver) is greater than previous angle then run a loop to slowly which passes angles incrementally to servo with a delay of 10ms.
 - f) `s [2]: gpiopwm 1 + i; i -> angle`
 - g) These delays help in moving the robot smoothly.
- Thread for dc motor receives the angle and checks if indicates one of the following[3][4][5]:
 - a) N : No movement
 - b) L: Move left by running only one motor of right side so, robot moves towards left for 1 second

- c) R: Move right by running only one motor of left side so, robot moves towards right for 1 second
- d) F: Move forward by running both the motors in forward direction for 1 second
- e) B: Move forward by running both the motors in backward direction for 1 second

4.9 Raspberry Pi Configuration:

There is a 40-GPIO pin header[2]. Two 5V pins and two 3V3 pins are present on the board, as well as a number of ground pins (0V), which are not configurable[1]. The remaining pins are all general purpose 3V3 pins, meaning outputs are set to 3V3 and inputs are 3V3-tolerant. A GPIO pin designated as an output pin can be set to high (3V3) or low (0V).

For controlling motors, we can use PWM (pulse-width modulation) : Hardware PWM available on GPIO12(PIN 26), GPIO13(PIN23), GPIO18(PIN1), GPIO19(PIN24) as mentioned in Fig 12 [2].

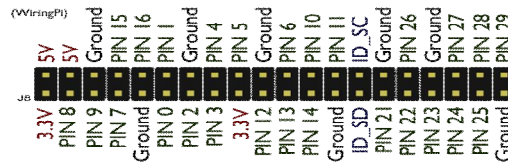


Figure 12: PIN configuration of Rasperry pi.

Actual configuration is:

- PIN 1: control pin for vertical servo motor is connected
- PIN 24: control pin for horizontal servo motor is connected
- PIN 23 & PIN 26: control pin for two dc motors are connected

4.10 Use of VR Box:

The use of this system requires that the user operating in “user” mode keeps the device steady. As and when this device moves, the angles are sent to the servo motor which then moves the head of case carrying the phone to that direction. Thus, it is highly feasible to view the stream via VR box[16]. VR box helps keep the device in a steady state and have stammer-less movements. In order to enable user experience virtual reality, display of stream is split into two. Major advantage of VR is that it helps avoid jitters and fluctuations s that those angles don’t cause random movements of head on robot machine. With the help of robot machine, it is highly feasible to travel to compact places. The movements of robot are controlled by a human. Two devices suffice the need of application. One with the user in the VR box ad another onto the robot machine. It is due to the highly scalable and advanced features of rasperry pi that there are no lags faced in the movement of user device and robot device. Either user can view the whole surrounding via live stream similar to a video call which sends data synchronously with movement of robot or capture image on user command. These images stored in database can be retrieved anytime because of strong backend of firebase which issues quick responses. Clarify API does the identification giving all possible

identifications. These components together make the system advanced and very useful.

5. CONCLUSION

Many a times humans need to get onto the compact places where they cannot reach but it is important to know the conditions of that region. At today's time, the industry is aiming to minimize human intervention by taking leverage of ongoing technical advancements. The project has been sub-divided as a versatile application which can be used by various domestic and commercial organisation. Due to the robust nature of system, it can be used for various purpose apart from video monitoring. The system is highly cost effective with utmost requirement of Smart Phone and connectivity. Orboid provides the ease surveil a remote locations or places such as inside of a pipe to detect corrosions and help the maintenance team to act accordingly saving the capital of industries. The robot machine is only used as a vehicle travelling to the places desired carrying the device operating in robot mode. The major working of system is application based which provides live steaming of surrounding of machine or captures images image and stores to database. The analysis of image plays a crucial role in inspection to detect any potential threats or damages. This system can be the best mechanism to safeguard in case of external threats with no requirement of human on site. The system is unlike traditional systems and can pave for more innovation.

REFERENCES

1. www.raspberrypi.org
2. www.pijava.com
3. <https://javatutorial.net/raspberry-pi-control-dc-motor-speed-and-direction-java>
4. <https://www.electronicshub.org/controlling-a-dc-motor-with-raspberry-pi/>
5. <http://www.instructables.com/id/DC-Motor-Control-With-Raspberry-Pi-and-L293D/>
6. Raspberry Pi Cookbook: Software and Hardware – by Simon Monk
7. Raspberry Pi Robotic Blueprints-Book by Richard Grimmett
8. www.firebaseio.com
9. <https://developer.vidyo.io/documentation/4-1-22-9/getting-started>
10. https://developer.android.com/guide/topics/sensors/sensors_position
11. <https://developer.android.com/guide/topics/media/camera>
12. www.clarifai.com
13. <https://www.micromo.com/applications/robotics-factory-automation/robotic-rover>
14. <http://www.fujitsu.com/global/about/resources/featurestories/2017060201.html>
15. Faure, Bouyer, Mercier, Robitaille, McFadyen, Fortin Cote, Cardou, Gosselin, Bonenfant, Laurendeau. **Development of a virtual-reality system with large-scale haptic interface and accessible motion capture for rehabilitation**, Faure, Bouyer, Mercier, Robitaille, McFadyen, Fortin Cote, ... Laurendeau. (2017). Development of a virtual-reality system with large-scale haptic interface and accessible motion capture for

rehabilitation, International Conference on Virtual Rehabilitation, 2017,

doi:10.1109/icvr.2017.8007519.

16. A. Rizzo and G. J. Kim, **A SWOT analysis of the field of virtual reality rehabilitation and therapy**, Presence Teleoperators Virtual Environ., vol. 14, no. 2, pp. 119–146, 2005.
17. M. Bonenfant, et al. **A Computer vision system for virtual rehabilitation**, CRVS, Conference on Computer and Robot Vision, Alberta, 2017. <https://doi.org/10.1109/CRV.2017.30>
18. A. Pfister, A. M. West, S. Bronner, and J. A. Noah, **Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis**, J. Med. Eng. Technol., vol. 38, no. 5, pp. 274–280, Jul. 2014. <https://doi.org/10.3109/03091902.2014.909540>