ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 8(34), DOI: 10.17485/ijst/2015/v8i34/71902, December 2015 ISSN (Online) : 0974-5645

An Intelligent Attempt to Export Files into Cloud in Handheld Devices through Gesture Recognition

Shriram K. Vasudevan[1] and C. Vivek [2]

[1]Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham (University), Coimbatore - 641112, Tamil Nadu, India; Kv_shriram@cb.amrita.edu
[2]Department of Electronics and Communication Engineering, M Kumarasamy College of Engineering, Karur - 639113, Tamil Nadu, India; Vivekc.phd@gmail.com

Abstract

Objective: The idea is to upload files into cloud using gestures. A "Gesture" is a form of movement of part of a body, especially the hand or the head, to express an action. Waving of the hand over the phone is used as a gesture to export files into the cloud. **Methods/Analysis:** The type of the file to be uploaded into the cloud can be either a picture or a video. This is accomplished using an application similar to gallery. To perform the task, the application is invoked to view a picture or video, when the hand is waved over the phone. The gesture is recognized and the currently viewed file is uploaded into the cloud. The cloud service used here for uploading is Dropbox. The size of the media files is generally large and uploading it will take time, so the size is reduced through a compression algorithm. There is a buffer which tracks the uploading part and will see to that, that the entire media file is uploaded into the cloud. **Findings:** Through this application the file can be uploaded into the cloud through a gesture easily and effectively. Also, it is found that the application developed is found to be stable in tough testing conditions. **Application/Improvements:** This app reduces the human effort to a large scale and most importantly it does not occupy more space in the handheld device.The same app can be further ported to IOS for apple phones and IPads.

Keywords: Cloud, Data Compression, Dropbox, Efficient and Easier Interaction,Wave Gesture

1. Introduction

Cloud storage offers a centralized location that can be accessed from anywhere, any time, and ideally from any device. This is already a huge improvement from local storage. Cloud storage differs from traditional storage infrastructures in regard to three key aspects: accessing files remotely over the network, accessing files on object-based storage, and the unique cost structure.

As an inheritance and emergence of cloud computing and mobile computing, mobile cloud computing has been devised as a new phrase since 2009. Cloud storage provides geographically dispersed users with storage capacity managed from a central location.

With Android Surpassing One Billion Users Across all Devices in 2014 (statistics from http://www.gartner.com/newsroom/id/2645115[1]), working with the majority is ideal. Data management on Android is easy but one can still find oneself low on empty space very quickly.

Today's Smartphone not only serves as the key computing and communication mobile device of choice, but it also comes with a rich set of embedded sensors, such as an accelerometer, digital compass, gyroscope, GPS, microphone and camera. So with the use of these advanced sensors we are accomplishing our target.

Initially as an improvisation to the traditional upload button, uploading files into cloud using tilt gesture was done.Thus we are uploading the file through the available

* Author for correspondence

sensors and sending it to the cloud. Usually the size of the media file will be large and uploading it might take time. Hence to be more efficient, compression techniques will be incorporated in. Therefore the file will be loaded efficiently and easily.

With an overall efficiency and ease brought about by this application, users will be able to upload media files into cloud without hesitation of time consumption.

2. State of the Art

In Janna Anderson and Lee Raine's "The future of Cloud Computing" it is known that many people today are primarily using smartphones, laptops, and desktop computers to network with remote servers and carry out their various tasks. There is a huge transition from static devices to dynamic device usage. This is so because, people prefer to get their work done on the go. With busy schedules, people prefer to save their time to fnish off their work when possible. This led to the research analysis of the most commonly used OS of all the the dynamic hand-held devices. In accordance to Gartner, Android users surpass One Billion Users Across all Devices in 2014 (statistics from[1]). Hence working with the majority seemed more ideal.

Mobile Cloud Computing (MCC) combines mobile computing and cloud computing. It has become one of the industry buzz words and a major discussion thread in the IT world since 2009. MCC is still at the early stage of development. Cloud computing through mobile computing is one of themost up to date technologies used world-wide[2]. This triggered us to make use of this technological advancement in our application. Further in the paper, a review on the background and principle of MCC, characteristics, recent research work, and future research trends are clearly illustrated, followed with a discussion on characteristics and recent research work.

Incorporating gesture techniques using proximity sensors, into our application was inspired by earlier report[3]. The usage of proximity sensors eradicates the need to use or wear specific sensors. The creation of their system allows a user to interact with mobile devices using intuitive gestures, without touching the screen or wearing/holding any additional device. Their system also shows the efficiency of using proximity sensors. Gestures are recognised with a precision of over 98% in real time. They further on go on to evaluate the various results of system. Hence we were able to conclude that the usage of proximity sensor is not only highly efficient, but also hassle free and also a further step into the many milestones of technology. Incorporating the proximity sensing technology into our application would enable our users a hassle free and efficient method of uploading media files to cloud.

In a recently approved and accepted application "TransmitMe", media files are uploaded to cloud using Tilt gestures. The tilt gestures are recognized if two tilts are given within a span of two seconds. When this gesture is recognized, the media files are uploaded into cloud such as DropBox. Though a very efficient technique, there are cases at which the tilt gesture may fail. Cases includes dropping the phone accidently, placed on the car dock when the car is moving etc. Hence to minimize the faults occurred through these such cases "TransmitMe" application is improvised by replacing the tilt gesture with a wave gesture. This eradicates the shortfalls that comes from the tilt gesture given as the user gesture to trigger the function of uplaoding media files into cloud.

3. Proposed Architecture

We have divided this application into three modules; [Module 1-Gesture Recognition, Module 2- Cloud Connectivity, Module 3-Uploading files into cloud]. The modules are built individually. Then they are integrated together as "TransmitMe" application. The various components involved in each module are described by the following description of the modules.

3.1 Module Gesture Recognition

Today touch less technology is at the peak. Everywhere people are expecting simplicity. So as to bring in simplicity the gesture is implied in Figure 1. Gestures are recognised through sensors. There are various sensors available in smart phones. The sensor used here is proximity sensor. The input gesture given by the user is a wave over of the hand held device. Once the wave gesture is given, the function assigned to it is invoked[4]. The function in this mobile application is cloud connection establishment and file uploading into the cloud.

The **gesture command generator** component of the architecture from Figure 1 listens to events that are emitted from the accelerometer sensors, so it steadily monitors the acceleration of the device. A gesture is detected if the absolute value of the acceleration exceeds a special threshold for a sufficient period of time. For instance in the "TransmitMe" android application, the tilt gesture is recognised the earth value recorded is greater

than 9.8m/s. Thus, it is assured that noises produced by unintended movements of the hand do not emit gesture.

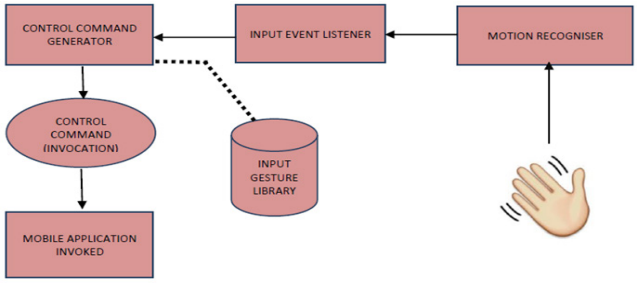

Figure 1. Gesture Recognition.

The **input gesture library** provides classes to create, recognise,load and save gestures. A touch screen smart mobile has the capability of understanding and recognising touch and air gestures on teh screen and around the mobile. These gestures can be recognised and stored. A collection of such gestures are that, tahta make up the input gesture library. For an application, if a usergesture is given, it can be recognized by matching it with the various gesture functions stored in the library.

An **event listener** is an interface that exists in the view class. It consists of several call back methods. These methods will be called by the Android framework when view to which is been assigned is triggered by the user interation.For instance, when a View (such as a Button) is touched, the onTouchEvent() method is called on that object. In our case its the wave of the hand over the proximity sensor is the action to be recognised as the user gesture.Hence the event listeners are commonly used to listen for user interaction. In order to build a custom component, the deafault event behaviours are defined using the event handlers class.

The **control command generator** then calls the functionality class that is triggered by the user gesture. Here when the wave of the hand is recognised, modules Cloud Connectivity and Uploading files are executed.

3.2 Module Cloud Connectivity

There are various types of cloud services available for cloud storage purposes. These services have two use cases. The first is personal storage, or extending one's own file system to the cloud. The second is sharing, especially of moderately size to large documents. Sharing was looked at in both a one-to-one and group sharing, with people who had accounts on the service and those who did not. Figure 3.2 shows how the mobile is connected to the cloud. Cloud connectivity is achieved through Application Programming Interface (API). The different APIs are available for specific cloud services[4].

The **Mobile Backend client library** includes a server that stores the user's data with Application, a client library and sample app for Android that make it easy to access that data. Google Cloud Messaging (GCM) and continuous queries can also be added. These features help to notify the user application the status of the events that are needed to be triggered by the user gestures. To keep users data secure, Mobile Backend Starter also includes built-in support for Google Authentication.

Hence in this module, cloud connectivity is established[5]. Figure 2 shows how the mobile is connected to the cloud. The cloud storage service used is Dropbox. Dropbox is a file hosting service operated by Dropbox Inc[2] which offers cloud storage, and is perhaps the most popular cloud storage and sync app around.

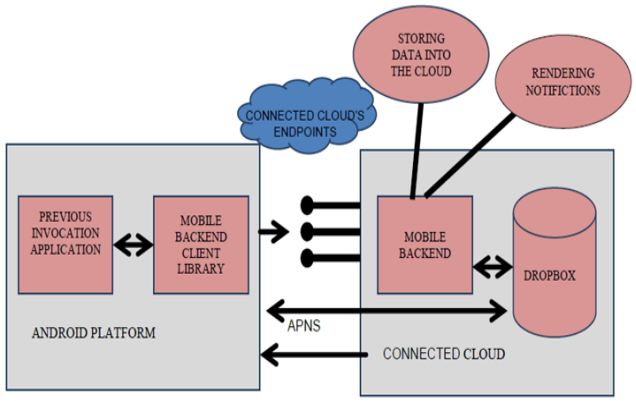

Figure 2. Cloud Conectivity.

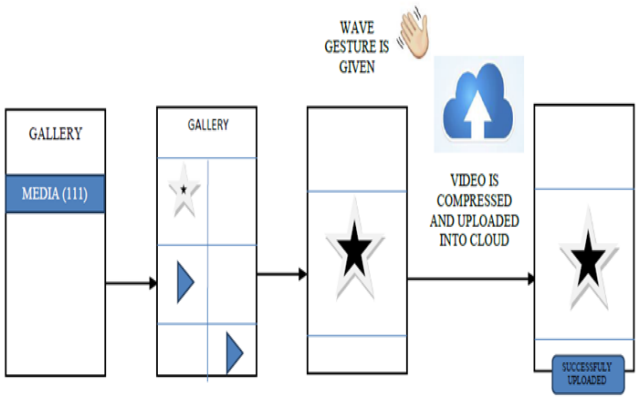

Figure 3. Further Application Innvocation.

Dropbox is a free service that lets a user bring all their photos, documents, and videos anywhere. With DropBox, any file saved to Dropbox will automatically. With the Dropbox application included, a user can take everything that matters with them on the go.DropBox is one of the safest cloud storages modes.

3.3 Module Uploading Files

When a wavegesture is recognised a connectivity is established and the currently viewing photo or video is uploaded into the cloud service.

To enable more efficiency and time consumption, compression technique is added in. FFMPEG[3] video compression is applied to compress the videos that is uploaded into the cloud. The compression ratiois 10:3 on an average. This ratio varies in accordance to the quality of the camera in which the video is recorded from.

FFMPEG is one of the best multi-encoding media tools available. It made portable to android by Faywong. This tool enables us to reduce the size of the video by encoding into acceptable quality. Comparing it with android's default 'MediaEncoder', FFMPEG is faster with encoding the same video to the same encoding options. It also supports various options such as compressing any format of video. Hence if the developer wants to use it in the future or enable applications to choose quality options based on network speed or user preferences, it is possible.

4. Hardware Requirements

Smartphones come with various inbuilt sensors these days. Sensors are devices which measure the physical energy and converts it into a signal (typically analog signals). Later this signal can be read or observed from an instrument or an electronic device (egmicrocontroller*)* connectedto the sensor. There are more than hundreds of sensors are available. Most Android powered phones, Windows Phones, iPhone, Blackberry Os and other Os based smartphones have built-in sensors that measure the orientation, motion, light and other environmental conditions. These sensors are built to measure and provide a high precision and accurate data to the reader. Proximity sensor is the major sensor used in this application[5].

Proximity sensing technology is becoming popular across a wide range of industries.Proximity sensor is a hardware-based sensor.It measures the proximity/ position of an object in cm relative to the screen of a device. This sensor determines the position of the phone w.r.t the object. For instance, it measures the distance between the phone and face when the phone is brought near to face during a call. Today, mobile phones use IR-based proximity sensors to detect the presence of a human ear. This sensing is done for two main purposes: Reduce display power consumption by turning off the LCD backlight and to disable the touch screen to avoid inadvertent touches by the cheek.Proximity sensor provides continuous or binary values. It is safest to assume binary values for compatibility. Hence most of the devices assume binary values. These sensors usually only emit data within 5cm proximity. Thus these sensors can be used to detect wave or swipe gestures. But the wave gestures should be within 5 cm from the sensor. The space between fingers and thumb can also be detected by the proximity sensor, which can effect wave gesture detection. Proximity sensor is good for gesture validation. The ambient light sensor is susceptible to light changes in environment. Light sensor can detect type of gesture but the proximity sensor can be used to determine that a gesture actually did take place. The number and positions of the proximity sensors in various devices vary accordingly. High-end smart phones have multiple proximity sensors present on the top and the bottom of the devices. Whereas mid-range smart devices have either single or multiple sensors on the top of the device. These proximity sensors works by measuring the intensity of light exposed to it. It has Boolean values of 1, for maximum intensity of light falling on it and 0, for minimum intensity of light. Hence by making use of this, proximity sensors can be used in accordance. A mid-range or a higher level smartphone is the major hardware requirement of our application.

5. Illustrations

The illustration in further application innvocationis the diagrammatic form for easier understanding.

6. Testing Results

Any product or software becomes stable and the confidence to the developer gets gained through a rigorous testing. This research is no different. There is a lot of emphasis given towards testing the developed product

From all the perspectives. The testing includes hardware, software and operating system dependent test scenarios. It is observed that the system we developed is

consistent and efficient under tough testing conditions. The following are the summary of the test scenarios and the results obtained.

6.1 Hardware Oriented Test Cases

The application was tested on various hardware components to check whether it works. The various test cases and the results obtained is given in Table 1.

Table 1. Hardware oriented test cases

Sl.No	Test Cases	Results Obtained
1.	With different equipment. (MotoG, MotoE, Samsung Galaxy Grand2, Asus Zen phone 5)	PASS
2.	With different memory ranges. (16GB SD Card, 32GB Card)	PASS
3.	Launch of application with equipment being charged	PASS
4.	Launch of application with battery having completely charged	PASS
5.	App launch in tablets (IPad and Samsung Tab)	PASS

It was very much visible that irrespective of the hardware changes and platform, the application we developed has shown a tremendous consistency. There were some problems with respect to resolution in the tablets as the screen size is different. It has been taken as a challenge and the team is working towards curbing this minor issue.

6.2 Software Oriented Test Cases (Application Level)

The application was tested for software issues and checked whether it was supported in all the devices. The application was also tested in all functional aspects. The various tests and the results are presented in Table 2.

Table 2. Software Oriented Test Cases

Sl.No	Test Cases	Results Obtained
1.	Launch of application along with multiple applications being launched in parallel	PASS
2.	Application kill/re-launch process	PASS
3.	Application stability cheque with having an incoming call as higher priority process	PASS
4.	Context switch check with other applications	PASS
5.	Battery consumption check with our app alone being launched for a long time	PASS

6.3 OS Level Test Cases

The OS level test cases showed how much the software was compatible, all the system related problems were identified and rectified in this testing. These tests make sure that application runs in all device. The various tests and results are given in Table 3.

Table 3. Operating system oriented test cases

Sl.No	Test Cases	Results Obtained
Case 1.	Launch of application immediately after system boot	PASS
Case 2.	Launch of application after 10 mins of system boot	PASS
Case 3.	Launch of application after a system restart	PASS
Case 4.	Launch of application after a sleep	PASS
Case 5.	Check for application's functioning in different versions of Android	PASS

Time taken for all the above test scenarios are summarized in the form of a graph (Figure 4)

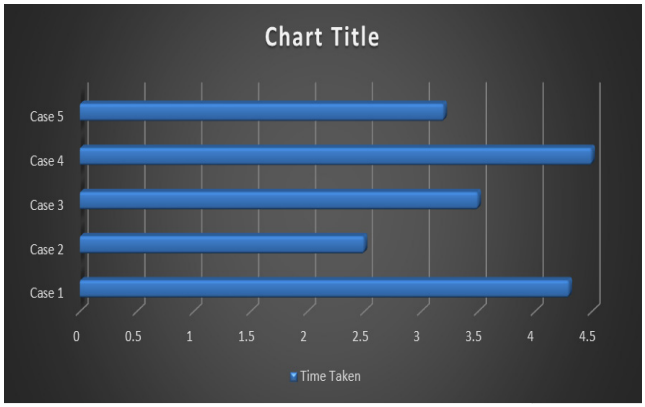

Figure 4. Analysis of time taken for the test scenarios.

It can be seen that the system is behaving consistently over all the operations.

6.4 Stress Testing Scenarios

All the functionality of the application were tested here. The different possible ways by which the application can run and its effectiveness was tested. The application launched and killed many times to observe its behaviour on all circumstances. The test cases and the results for the tests done here are summarized in the Table 4.

Table 4. Stress testing scenarios

Sl.No	Test Cases	Results Obtained
1.	kill/Launch of app so many times	PASS
2.	Kill/Launch of app so many times along with other apps	PASS
3.	Testing with same input image multiple times	PASS
4.	Testing with same input video multiple times	PASS

The system behaves quiet stable under all the above tough conditions.

7. Challenges Faced

One of our biggest problems faced during this project was compatibility. The question of the function of proximity sensors present and not present in various android devices. High level proximity sensors are required to be present in order for the direction of the hand to be recognized, having various functions for each direction.Another challenge faced was including more than one cloud service i.e., giving access to two or more cloud services. The combination of API's into a single unit brought lot of complications. Also the video compression technique which was to be had to be employed had to compress both the pictures and videos.

Figure 5. Functionlity of the developed application.

Figure 6. A Maximized View Of The Picture Or Video.

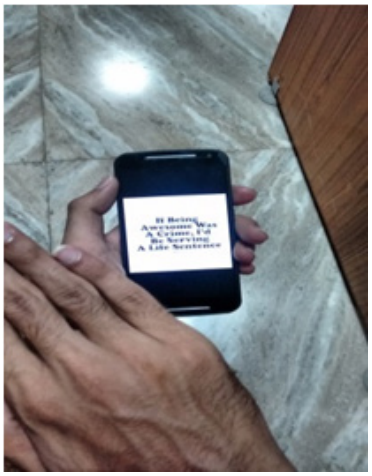

Figure 7. Gesture initiation.

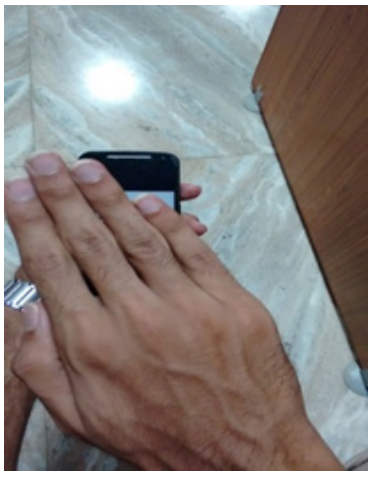

Figure 8. Gesture midway.

Figure 9. Gesture from left to right.

Figure 10. Status of upload.

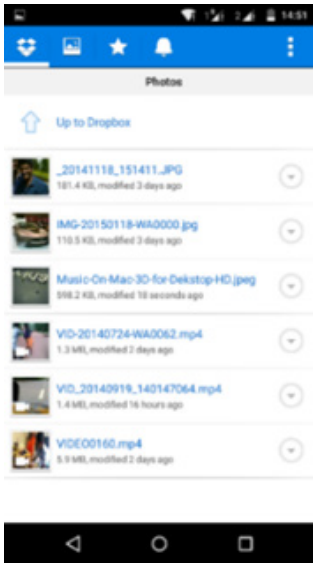

Figure 11. Successful upload.

8. Results

The functionalities of the developed application elaborates the functionality of the application. At the invocation of the application, it starts collecting all the available photos and videos in the device, and it displays a thumbnail view. The contents will be shown in a gallery format. The user canchoose his desired file. The maximized view of the photo or video is displayed.

If a file is considered or need to be viewed later, it can be stored in cloud. This can be easily done through this application. If that file is to be uploaded into the cloud, a wave gesture from left to right is displayed. The proximity sensor recoganizes the gesture and connectivity to the cloud is established.

The status shows the uploading of the file into the cloud. Once the gesture is recognised the uploading takes place, the file to be sent to cloud is compressed and then uploaded into the cloud space. And in case you want the process to be cancelled just click the 'CANCEL' button shown in the status of upload image. This ability of cancelling the process can be used whenever an accidental store is driven by the user. Successfully uploading the file into the cloud account can be viewed and deleted.

9. Conclusion and Future Scope

With almost all applications available from cloud these days, other cloud services apart from Dropbox can be implemented through their respective APIs into this project. This enables users to choose from their various accounts in the various cloud services provided. This also enable more flexibility and more options for the user. The videos, before getting uploaded are compressed to enhance the efficiency. To make this application even more efficient image compression can also be included.

Modern day mobile phones come with many sensors. The sensor used here is promixity sensor. It is found as a common sensor in all the smart phones in today's market, so it is possible to detect the tilt gesture, which can be recognized in any smart phone.Thereby ensuring that the application is feasible in all the smart phones. With gestures and cloud services playing a huge role in the world tomorrow, this application is the first step taken towards simplicity and efficiency.

10. References

1. Android to Surpass One Billion Users Across all Devices in 2014. Available from: http://www.gartner.com/newsroom/id/2645115 Date of Access: 2015 Jan.
2. Qi H, Gan A. Research On Mobile Cloud Computing: Review,Trend And Perspectives; Bangkok; 2012. p. 195–02.
3. Cheng HT, Chen AM, Ashu R, Elliot B. Contactless Gesture Recognition for Mobile Devices; USA. 2011 Jan 9-12. (Article from CMU repository).
4. Application Fundamentals | Android Developers. Available from: http://developer.android.com/guide/topics/fundamentals.html Date of Access: 2015 Jan.
5. Sensors used in smart phone. Available from: http://myphonefactor.in/2012/04/sensors-used-in-a-smartphoneq Date of Access: 2015 Jan.