© 2018 JETIRAugust2018, Volume 5, Issue 8www.jetir.org (ISSN-2349-5162)

Design and implementation of Air-Writing Recognition

[1]Dr.D.VISHNU VARDHAN,[2]A.HEMANTH KUMAR
[1]Associate Professor,[2]PG Student
[1]Department of ECE, JNTUA College Of Engineering, Ananthapuramu.
[2]Department of ECE,JNTUA College Of Engineering,Ananthapuramu

***Abstract-** Air-writing refers to writing of linguistic characters or words in a free space by hand or finger movements. Air-writing differs from conventional handwriting; the latter contains the pen-up-pen-down motion, while the former lacks such a delimited sequence of writing events. We address air-writing recognition problems in a pair of companion papers. In Part1 recognition of characters or words is accomplished based on six-degree-of-freedom hand motion data. Part 2 addresses detecting and recognizing air-writing activities that are embedded in a continuous motion trajectory without delimitation. Our main contribution is to build an air-writing system encompassing both detection and recognition stages and to give insights into how the detected writing segments affect the recognition result. With leave-one-out cross validation, the proposed system achieves an overall segment error rate of 1.15% for word-based recognition and 9.84% for letter-based recognition.*

***Index Term-** Air-Writing, Linux, Raspberry pi , Character recognition, Object tracking, Human-machine interaction, Open CV*

I. INTRODUCTION

The project is aimed at evaluating the performance of an operating system on an embedded system. Before delving into its implementation, an introduction is needed to the parts involved in the project. The whole report is centered around the field of embedded systems and the use of Linux to run applications on them. Hence an introduction to Embedded Systems and using Linux as an OS in them is provided.

II. EXISTING METHOD

Digital pen with trajectory recognition can be done by using accelerometer. The digital pen consists of a triaxial accelerometer, a microcontroller, and an RF wireless transmission module for sensing and collecting accelerations of handwriting and gesture trajectories. Our embedded project first extracts the time- and frequency-domain features from the acceleration signals and, then transmits the signals by using RF transmitter. In receiver section RF signals can be received by RF receiver and given to micro controller. The controller processes the information and finally the results can be displayed on Graphical LCD.

III. PROPOSED METHOD

In proposed system we are going to use webcamARM microcontroller and display unit. Here, we are using pen or hand for drawing in front of the camera then whatever we are going to draw in front of it will be displayed on the display unit. Our embedded system is capable of translating time-series acceleration signals into important feature vectors. Users can use the pen to write digits or make hand gestures etc can be displayed on the display unit.not only this we are going to perform on and off operation of dc motor.

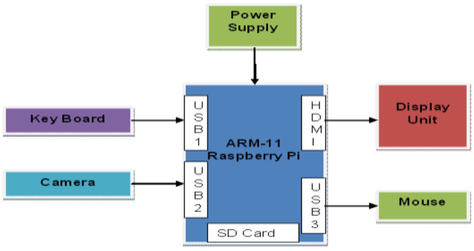

Fig 1: Block Diagram

IV.HARDWARE IMPLEMENTATION

A.RASPBERRY PI BOARD:

The Raspberry Pi is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. The Raspberry Pi is manufactured in two board configurations through licensed manufacturing deals with Newark element14 (Premier Farnell), RS Components and Egoman.

Fig 2: Raspberry pi Board

These companies sell the Raspberry Pi online. Egoman produces a version for distribution solely in China and Taiwan, which can be distinguished from other Pis by their red coloring and lack of FCC/CE marks. The hardware is the same across all manufacturers. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, Video Core IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded to 512 MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and persistent storage.

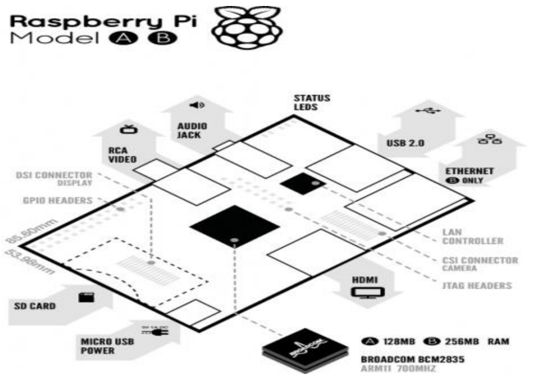

Fig 3: Board features

The Foundation provides Debian and Arch Linux ARM distributions for download. Tools are available for Python as the main programming language, with support for BBC BASIC (via the RISC OS image or the Brandy Basic clone for Linux), C, Java and Perl.

B.TFT display unit:

TFT stands for Thin Film Transistor, and is a type of technology used to improve the image quality of an LCD. Each pixel on a TFT-LCD has its own transistor on the glass itself, which offers more control over the images and colors that it renders. While TFT-LCDs can deliver sharp images, they also tend to offer relatively poor viewing angles, meaning they look best when viewed head-on. If you view a TFT-LCD from the side, it can be difficult to see. TFT-LCDs also consume more power than other types of cell phone displays.

C. UVC Driver Camera:

Fig 4: UVC Driver camera

A UVC (or Universal Video Class) driver is a USB-category driver. A driver enables a device, such as your webcam, to communicate with your computer’s operating system. And USB (or Universal Serial Bus) is a common type of connection that allows for high-speed data transfer. Most current operating systems support UVC. Although UVC is a relatively new format, it is quickly becoming common.

There are two kinds of webcam drivers:

1)The one included with the installation disc that came with your product. For your webcam to work properly, this driver requires some time to install. It is specifically tuned for your webcam, designed by your webcam manufacturer and optimized for webcam performance.

2)A UVC driver:-You can only use one driver at a time, but either one will allow you to use your webcam with various applications.

The following Logitech webcams support UVC: Logitech® Quick Cam® Pro 9000 for Business, Logitech® Quick Cam® Pro for Notebooks Business, Logitech® Quick Cam® Communicate MP for Business, Logitech® Quick Cam® Deluxe for Notebooks Business, Logitech® Quick Cam® 3000 for Business.

V.SOFTWARE REQUIREMENTS

A. *Linux Operating System:*

Linux or GNU/Linux is a free and open source software operating system for computers. The operating system is a collection of the basic instructions that tell the electronic parts of the computer what to do and how to work. Free and open source software (FOSS) means that everyone has the freedom to use it, see how it works, and changes it. There is a lot of software for Linux, and since Linux is free software it means that none of the software will put any license restrictions on users. This is one of the reasons why many people like to use Linux.

A Linux-based system is a modular Unix-like operating system. It derives much of its basic design from principles established in UNIX during the 1970s and 1980s. Such a system uses a monolithic kernel, the Linux kernel, which handles process control, networking, and peripheral and file system access. Device drivers are either integrated directly with the kernel or added as modules loaded while the system is running.

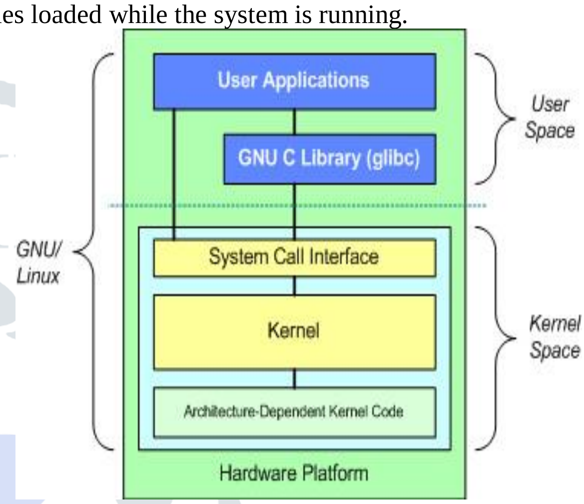

Fig 5: Architecture of Linux Operating System

B. ***Qt for Embedded Linux:***

Qt is a cross-platform application framework that is widely used for developing application software with a graphical user interface (GUI) (in which cases Qt is classified as a widget toolkit), and also used for developing non-GUI programs such as command-line tools and consoles for servers. Qt uses standard C++ but makes extensive use of a special code generator (called the Meta Object Compiler, or moc) together with several macros to enrich the language. Qt can also be used in several other programming languages via language bindings. It runs on the major desktop platforms and some of the mobile platforms. Non-GUI features include SQL database access, XML parsing, thread management, network support, and a unified cross-platform application programming interface for file handling. It has extensive internationalization support.

C. Open CV*:*

Open CV (Open Source Computer Vision) is a library of programming functions for real time computer vision. It is developed by Willow Garage, which is also the organization behind the famous Robot Operating System (ROS). Now you’d say MATLAB also can do Image Processing, then why open CV? Stated below are some differences between both. Once you go through them, you can decide for yourself. Advantages of OpenCV over

MATLAB (Collected from various blogs/forums):-

- Speed: Matlab is built on Java, and Java is built upon C. So when you run a Matlab program, your computer is busy trying to interpret all that Matlab code. Then it turns it into Java, and then finally executes the code. Open CV on the other hand, is basically a library of functions written in C/C++. You are closer to directly provide machine language code to the computer to get executed. So ultimately you get more image processing done for your computers processing cycles, and not more interpreting. As a result of this, programs written in Open CV run much faster than similar programs written in Matlab. So, conclusion? Open CV is damn fast when it comes to speed of execution. For example, we might write a small program to detect people's smiles in a sequence of video frames. In Matlab, we would typically get 3-4 frames analyzed per second. In Open CV, we would get at least 30 frames per second, resulting in real-time detection.

- Resources needed: Due to the high level nature of Matlab, it uses a lot of your systems resources. And I mean A LOT! Matlab code requires over a gig of RAM to run through video. In comparison, typical Open CV programs only require ~70mb of RAM to run in real-time. The difference as you can easily see is HUGE!
- Cost: List price for the base (no toolboxes) MATLAB (commercial, single user License) is around USD 2150. Open CV (BSD license) is free!
- Portability: MATLAB and Open CV run equally well on Windows, Linux and Mac OS. However, when it comes to Open CV, any device that can run C, can, in all probability, run Open CV.

VI.RESULTS

Fig-(a): In Existing system devices are connected to kit

Fig-(b): kit connected to display

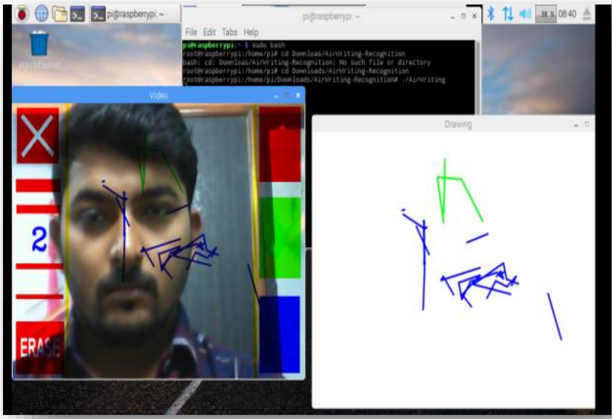

Fig-(c): Delay in camera with raspberry pi board

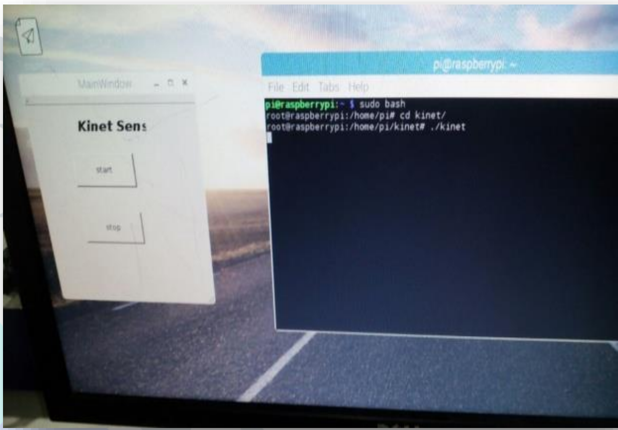

Fig-(d): Implemented system using linux operating system

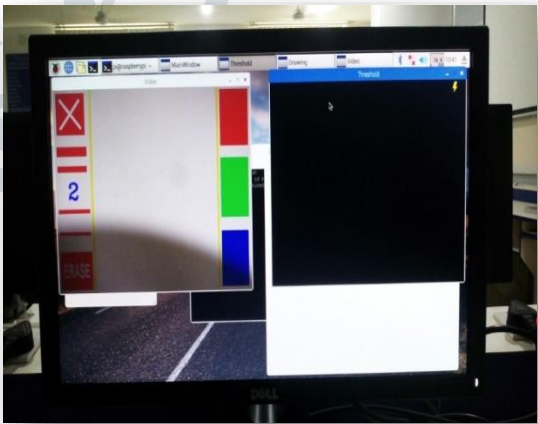

Fig-(e): Main window is opened

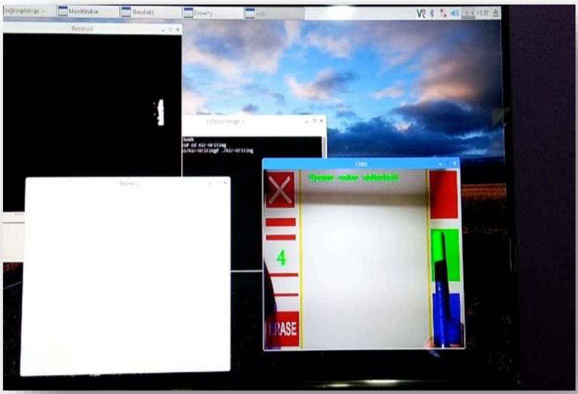

Fig-(f): selection of color

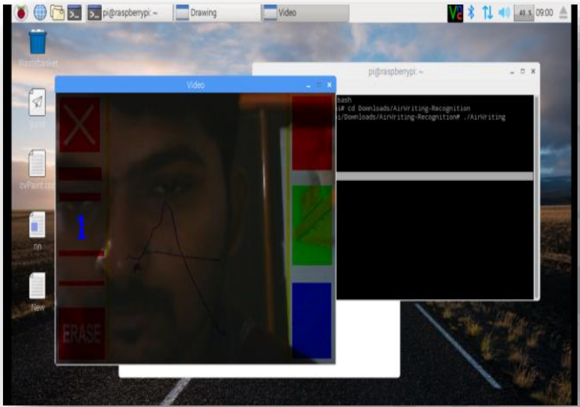

Fig-(g): Text recognized by the camera

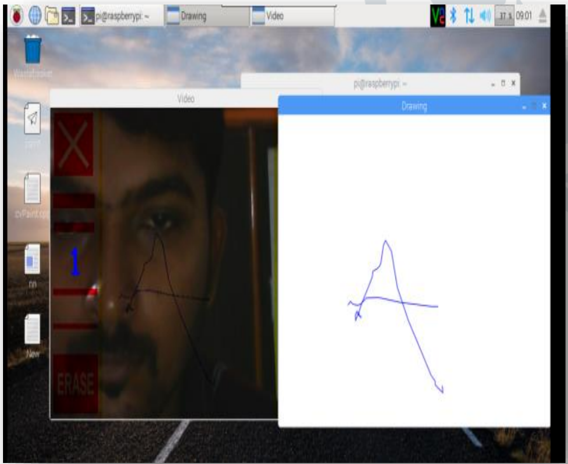

Fig-(h): Recognized output-1

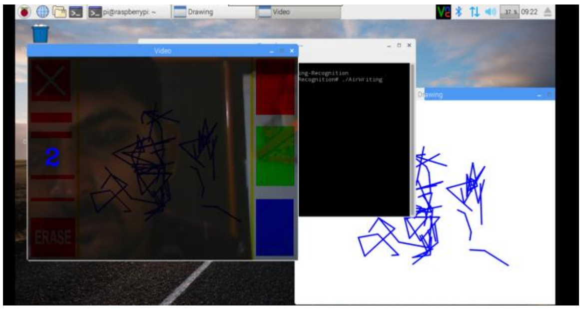

Fig (i): Recognized output-2

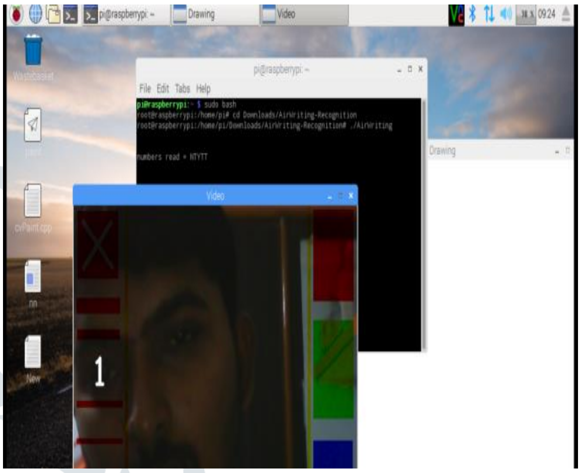

Fig-(j): Recognized output-3

VI.CONCLUSION

In this study, we attempt to recognize air-writing with a 6-DOF motion tracking system. The writing motion is tracked with the position and orientation in the global frame, and the acceleration and angular speed in the device-wise coordinates. The air-writing recording process is very time consuming. To make the recording process feasible, we place constraints on stroke orders and uppercase letters with limited vocabulary to refine the scope of air-writing data acquisition without losing too much generality. From these motion data, we derive five basic features for observations of HMMs and form the combinationof pure optical, pure inertial, and complete 6-DOF features. Although the handwriting is defined purely by the planar shape, we show that motion information beyond the spatial trajectory is informative for air-writing recognition. Air-writing is unistroke without pen-up/pen-down information. The writing style and motor control are different from ordinary pen-based writing due to lack of haptic and vision feedback. We separate air-writing in two levels: motion characters and motion words. Motion characters are handled similar to motion gestures, and each character is modeled with a HMM. A motion word can be modeled by concatenating character and ligature models. We present two approaches to model ligatures: hard clustering and decision tree. The former is provento be sufficient for word-based word recognition. The latter provides better capability of ligature modeling, which improves the performance of letter-based word recognition. The word-based word recognition achieves relatively low WER but is not able to recognize OOV words. The word-based recognizer is suitable for applications that have a limited vocabulary and stringent requirement on the accuracy. On the other hand, letter-based word recognition has around 10% WER but can handle arbitrary letter sequences and progressive decoding. To substantially improve the letter-based recognition accuracy, the system can provide suggestions with *n*-best decoding and lets the user choose the right one.

A user study investigates input speed, motion footprint, physical strain, and subjective evaluation of two motion-based text input methods: air-writing and virtual keyboard. The results suggest that air-writing is suitable for short and infrequent text input on a motion-based user interface.

REFERENCES

[1] M. Chen, G. AlRegib, and B.-H. Juang, "Air-writing recognition—Part II: Detection and recognition of writing activity in continuous stream of motion data," *IEEE Trans. Human-Mach. Syst*, to be published.

[2] (2015). Leap motion [online]. Available: http://www.leapmotion.com

[3] M. Chen, G. AlRegib, and B.-H. Juang, "Feature processing and modeling for 6d motion gesture recognition," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 561–571, Apr. 2013.

[4] C. H. Blickenstorfer, "Graffiti: Wow!" *Pen Computing Mag.*, vol. 1, pp. 30–31, Jan. 1995.

[5] K. Perlin, "Quikwriting: Continuous stylus-based text entry," in *Proc. 11th Ann. ACM Symp. User Interface Softw. Technol.*, 1998, pp. 215–216.

[6] P. Isokoski and R. Raisamo, "Quikwriting as a multi-device text entry method," in *Proc. 3rd Nordic Conf. Human-Computer Interaction*, 2004, pp. 105–108.

[7] T. K¨oltringer, P. Isokoski, and T. Grechenig, "Twostick: Writing with a game controller," in *Proc. Graph. Interface*, 2007, pp. 103–110.

[8] (2013). Swype-Type fast, swype faster. [Online]. Available: http://www. swype.com/

[9] R. Plamondon and S. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 63 –84, Jan. 2000.

[10] J. Makhoul, T. Starner, R. Schwartz, and G. Chou, "On-line cursive handwriting recognition using hidden markov models and statistical grammars," in *Proc. Workshop Human Lang. Technol.*, 1994, pp. 432–436.

AUTHORS PROFILE

Dr.D.VishnuVardhan received his B.Tech Degree in ECE from RGM College of Engineering, Nandyal and M.Tech from JNTU Kakinada. Currently he is working as an Assistant professor in the department of ECE at JNTUA Ananthapuramu. His research interest includes Embedded systems, Image processing and VLSI system Design.

A. Hemanth Kumar received his B.Tech Degree in ECE from Anantha lakshmi institute of technology and science, Ananthapuramu. Currently he is pursuing M.Tech in embedded systems from JNTUA, Ananthapuramu. His research interest includes Embedded systems, Iot.