Design and Implement of a Kind of Virtual Keyboard Based on Microcomputer and CMOS Camera

Zhiyuan Zheng, Kun Yang, Ji Pei
School of Electronic and Information Engineering
Beihang University
Beijing, China
Jesse.zhengzhiyuan@gmail.com kevinyang7684@gmail.com iampgzp@yahoo.cn

***Abstract*— the standardized keyboard is difficult for people with disabilities to use, and the bacteria which lives in the keyboard are not suitable for medical environments. To solve this problem, we give a new insight into the characteristics of CMOS camera and microcomputer, and use them to improve a kind of customizable keyboard based on some prior accomplishments. In this paper, dominate features and expected applications of CMOS camera will be presented. This kind of customizable virtual keyboard will be widespread in medical environment, transportation, and future computers.**

Keywords- CMOS Camera; AVR Microcomputer; Virtual Keyboard

I. INTRODUCTION

Nowadays, an increasing number of new devices have come out which had dramatically changed people's life, but the quagmire in text entry for disabled people have not been completely solved. The standardized keyboard is hard for handicapped people to use, and it is not necessary to design a standard keyboard if only several keystrokes will be used. Therefore, a new kind of keyboard is undoubtedly helpful to address these problems.

We have found many virtual keyboards available now, among those the one designed by Naweed Paya and Venkat Ganesh from Cornell University stands out. It is a customizable keyboard printed on a flat black board to take place the traditional physical keyboard. Based on their accomplishment, we try to improve its function and accuracy better to meet the needs of disabled people and make it more personally. First of all, we adopt a high speed microcomputer to replace the slower one. Second, we design a variety of keyboards to make it more personally. Moreover, we also add Braille to keyboards so that blind people could operate it with ease. Finally, we conduct many operations to test the accuracy under different lighting environments and draw conclusions from that. This paper will elaborate the design and improvement of customizable virtual keyboard in detail.

II. FEATURES OF CMOS IMAGE SENSOR

Along with the CCD image sensor, CMOS have excellent imaging performance. CMOS image sensor has advantages such as low power consumption, low fabrication cost, compatibility with VLSI integration, the possibility of smaller systemsize and radiation hardness. [1]

The disadvantages of CMOS (Complementary metal-oxide-semiconductor) image sensors, such as low-sensitivity, high noise, dark current and low filling degree have been resolved by technologies like DRSCAN noise[2] cancellation technology and it is now widely used in surveillance, aviation and detection equipment, medical equipment, eye mask recognition and visual communications.

III. HARDWARE IMPLEMENTATION

Based on the characteristics of the CMOS camera, we improved and implemented a new kind of virtual keyboard by referring to some relative kinds of virtual keyboards. [3] This part we will discuss the hardware implementation from the following five aspects.

A. Overall Descriptions

The Virtual Keyboard mainly contains four units, which are AVR Microcomputer, CMOS Camera, Laser Module, and the printed keyboard. The four units connect to each other and make up the whole project as follows.

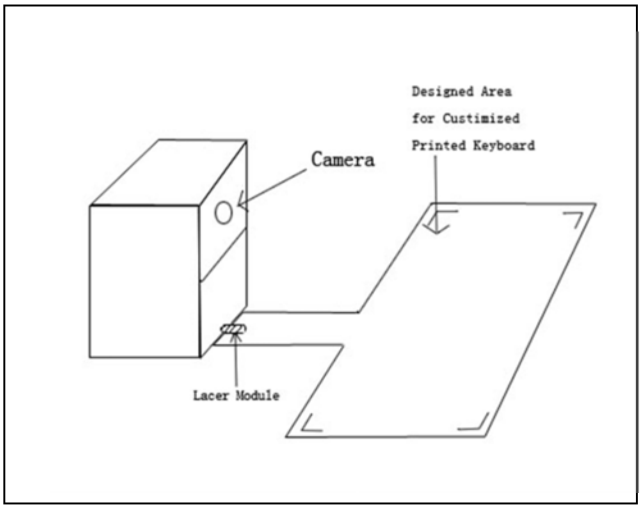

Figure 1. Example of a Customizable Virtual Keyboard

978-1-61284-307-0/11/$26.00 ©2011 IEEE

B. *CMOS Camera*

In the project, we use Omni Vision's CMOS image sensor OV6630 as our capturing images device, because it is relatively cheaper and it can output image color data in progressive scan mode. The camera, which is similar to OV6130, lies in the front with two roles of interfaces on the back. It turns out that we can get the expected results with OV6630 finally. Progressive scanning function is a crucial factor as we do not have enough power available to process the entire frames at once. As we use red laser module to obviate the finger we are pressing, the 8-bit red channel output is the only value we get to PORTA on the Mega32.

C. *AVR Microcomputer*

ATmega32 is a high-performance, low-power AVR 8-bit Microcontroller. It works above 5V power resources with 0-16MHz speed levels. We uses COM PORTA on Atmega32 to communicate with computers and the computer offers power to the microcomputer through USB. Of course, AVR Microcomputers with higher performance is welcomed; besides, we can also choose double microcomputers to enhance its processing speed.

D. *Laser Module*

It is a red laser module with a line-generating DOE attached. We decided to use a Class Ⅱ 635 nm Red laser because when fingers are covered by the light, they can be easily distinguished from black background and moreover, the laser is harmless to people. We have considered to use infrared laser. If we use an infrared laser to detect button presses, the user would have no idea when he or she is staring at the laser directly, plus there would be no available ways to prevent eye damage. The laser module would emit a plane of red laser which covers the keyboard plane several millimeters above the board, and Figure2 depicts it. To be specific, the laser module doesn't cast directly on the board and the black background can absorb all color elements. Therefore, the camera cannot distinguish the red button unless the finger is pressed on the keyboard. The finger blocks out the light and turns red. The camera then detects the red signal and process the value and position to microcomputer.

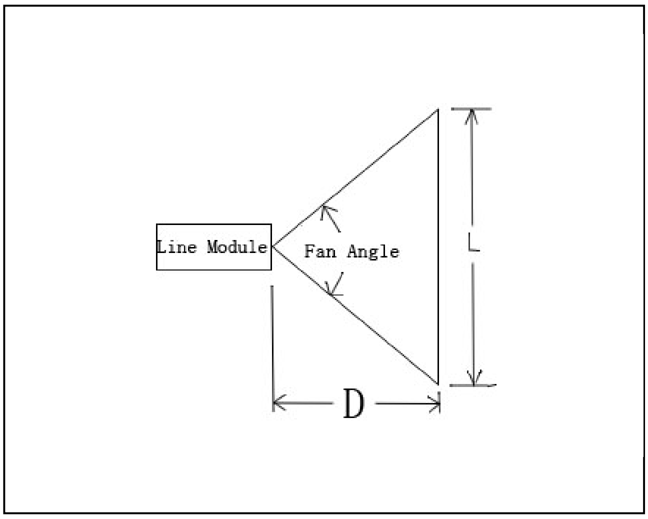

Figure 2. Example of Line Module

E. *Customized Keyboard*

The design of the keyboard is the uniqueness of the device, because user can design it according to our minds. No matter you prefer rectangular or circular, it's up to you, and the size or the content of the button are also highly personalized. It will change ways of inputting texts dramatically. We designed three types of customized keyboard as paragons, and also, the traditional printed one which is better for us to do accuracy test. The color of the keyboard should be dark color, such as black, and the buttons can be blue, which distinguishes it from the red finger. In addition, it is highly suggested that the material of the keyboard surface should not be reflective.

We take the needs of blind people into account as well, so we add Braille to keyboards for their convenience, plus this feature is very unique.

F. *Casing the Hardware*

The camera of the device is designed at a fixed position such that it overlooks the printed keyboard from a particular angle and it can cast all the appropriate region of the keyboard. Besides, the laser module is to be attached at a particular angle so that the plane of red light completely covers the area above the printed keyboard. The distance of the laser module to the keyboard should be around 40cms or so, to make sure all the regions are covered by the plane of red laser. Connect the laser module to a 3V power source, and you will see a line of red light comes out. Press the keystroke with your finger, which will turn red immediately. At the same time, the camera above detects the signal and sends keystroke's position to the microcomputer to determine the input content.

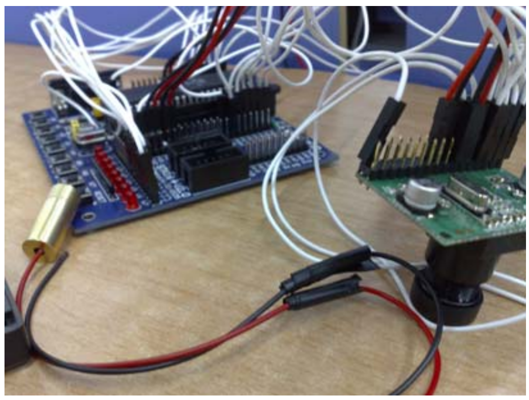

Figure 3. OV6630 Camera Connecting to ATmega32 Microcomputer

IV. SOFTWARE IMPLEMENTATION

The main operation of the software implementation is: First, read red values of the keyboards by camera through

I2C protocol; Second, process these image values in the microcomputer to obtain the pressed key; finally, convert the pressed key into computers and obtain the key.

A. Camrea Operation and Image Processing

First of all, we set the frame rate to 6 frames per second, the enough time to distinguish two different pressed keys. And then operate the getRedIndex function to get values of the boundaries of the camera, with which we can determine the areas of keys. As we only obtain the red value of the key, and threshold is set to 19200, UBRRL=51, if the camera detect the red value above the threshold, it will send the position value of the key to the microcomputer, if else, no data will be sent. Third, if the queue is empty, the getRedIndex function will be operated to read from camera; finally, the microcomputer will send the data if the line is idle.

The new frame data will arrive on the UV and Y lines. In determining the pressed keys, we use an array of 88 values in which we obtained the data on the UV line every 2 PCLKS, which means the pressed keys within 2 PCLKS are considered as the same key. The camera will run the same set of pixels for consecutive rows and two lines processed to the same pixels. We do the processing after each vertical line as we do not have enough space to store the whole values. HREF will stay negative for about 0.8ms and the camera data will be invalid just after each vertical line of valid data, which gives us enough time to process the data. If the pixel exceeds the red threshold, we will check if the pixel was part of a contiguous line of red pixels, which would help determine the pressed key. We bounce the key by examine the contiguous four presses. If it is valid, we then added the key to the queue of keys and send them to PC.

B. I2C Communication

The I2C communication is served as the bridge between the camera and the microcomputer. There are about 92 registers available for us to set up the camera. We use CodeVision communicate with the camera. The protocol uses a 2-wire communication scheme, activated by a 10kOhm pull-up resistor. The SCL line gives clock signal to the camera, and the frequency is set by: (16 + 2 x (TWBR) (4TWPS)). We defined constants for TWCR Register, codes for TWI Master Status, and some other functions. The bit rate register (TWBR) is set to 72 and the status register (TWSR) is set to 0. The rest values are set according to the standard protocol, and with the above setting, we can get the expected results.

C. Determing The Edge

One of the difficulties is to determine the edges of the camera vision. The OV6630 has a broad vision and after several experiments, we are sue it could capture about 248x 160 pixels, but the accuracy near the edge is low. As we hope to capture the image of the whole A3 size keyboard, we set the camera to 176x 144. With this resolution, we set the output format to capture 16 bit UV/Y data [2], where UV had BRBR data and Y had GGGG data. The Y data was completely ignored.

D. EEPROM Programming

Because we hope to be able to change the key assignments accordingly when the program is operating, we stored an array of scan codes corresponding to each key in EEPROM and turn on the RS-232 receive interrupt. We use a java applet to help user to changes the keys they desire and transmit it to the AtMega32 through a standard COM port on the PC.

E. Environment Requirements

Because we use the red pixel as the threshold to distinguish the pressed key to the rest, so the surrounding conditions would be expected to be dark and to make sure the red pixel of the background bellows the threshold. If strong light is offered in the background, it may mistakenly output some keys. The expectation lighting values of background wound be about 10-70 Lm. We also conducted the experiments (testing 100 times and reporting the keys accuracy) under several lighting circumstances and come up with the results listed as below.

Lighting Environment (Lumen)	Average Accuracy of keys (%)	Central Accuracy of Keys (%)	Peripheral Accuracy of Keys (%)
10	79.249	83.765	74.733
30	77.158	82.917	71.399
50	71.877	77.389	66.356
70	66.921	72.941	60.901

V. TESTING

To increase the credibility of the testing results [4], we firmly tested the accuracy of the keystrokes for 100 times of the standardized keyboard. Final design for the customized keyboard and testing results for this layout are given in Figure 5. We set the acceptance threshold at 80% for central areas and 70% for side areas. We tested in four different lighting conditions and the overall testing results meet our expectation. Finally, we suggest the lightening condition to be 30, with relative practical and expected results. From the testing results, we can say confidently that the customized keyboard by using CMOS camera and AVR Microcomputer is applicable.

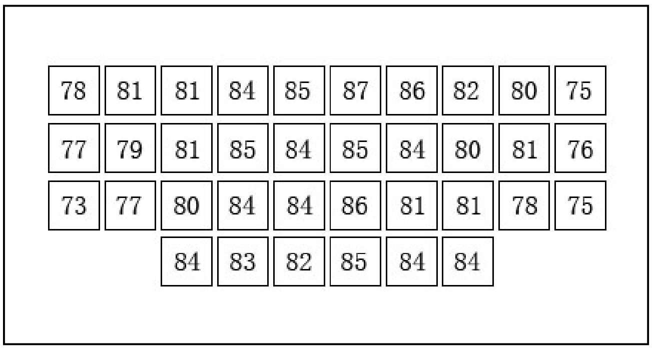

Figure 4. Accuracy of Key Strokes (under 30 Lumen)

VI. CONCLUSION

Compared with traditional keyboard, Virtual Keyboard obviously has much more advantages in the personalization, convenience, and applicable to special environments. It could be widely used for people with disabilities, [5] and for special uses when only several key strokes are used. Although there are many kinds of virtual keyboards on markets, this is one of the few who uses microcomputer to implement the function.

In this paper, we demonstrate the hardware and software implementation in detail and the testing results shows the accuracy of keys. From this we can confidently draw conclusion that this implementation is absolutely practical and attainable. This paper is served as the instigator to illustrate the application of CMOS camera and AVR Microcomputer in virtual keyboard. We hope this paper could make some contributions to the fast developing computer industry.

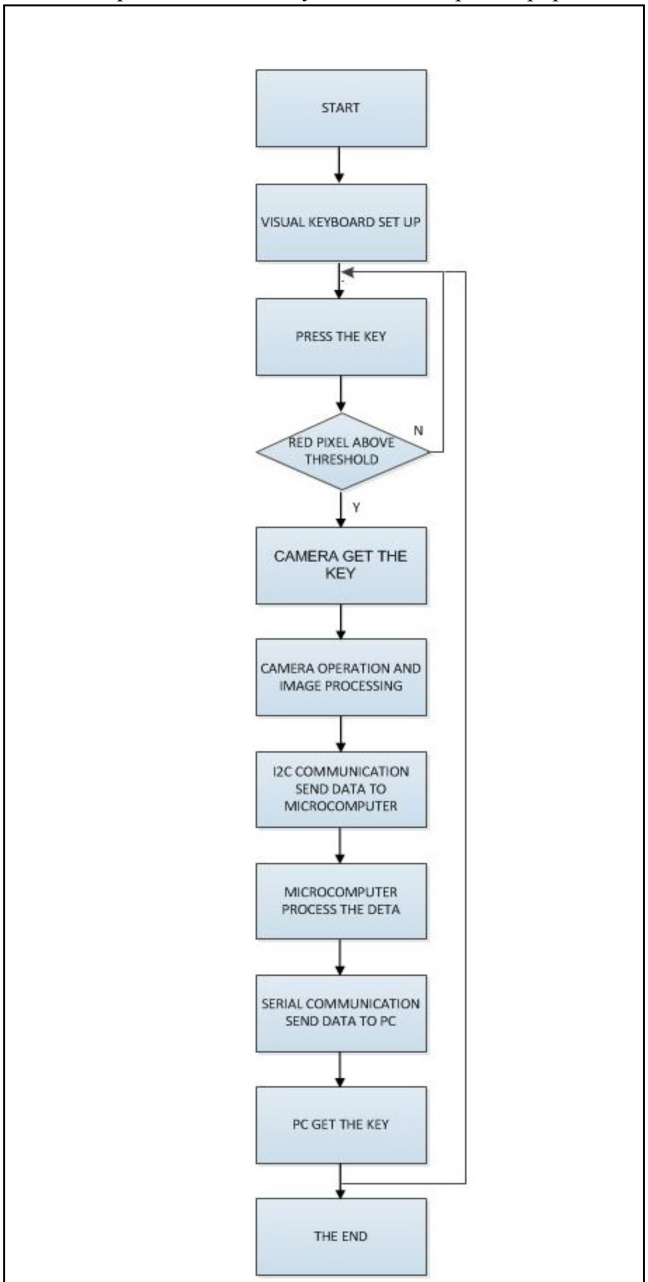

Figure 5. Flow Chart

ACKNOWLEDGMENT

We would like to acknowledge and extend our heartfelt gratitude to the following persons who have made the completion of this thesis possible: Professor Zhao Qi, Graduate students Jianfei Wang and Qingjun Cui.

Particular appreciation goes to Professor Qi Zhao not only for the countless hours of rough draft reading as it was in process, inspiring suggestions, encouragement, and great patience with my seemingly endless revisions and questions, but also for the immeasurable intellectual input and lively discussion throughout the duration of this project.

I am also grateful cordially to my friend Jianfei Wang and Qingjun Cui, who have contributed enormously to my understanding of microcomputer, camera and image processing. Without a doubt, they are the most gifted and natural friends I have ever known.

My sincere thanks also go to two friends, who have inspired me and give their generous ideas on the project, and helping with specific suggestions. They are Naweed Paya and Venkat Ganesh from Cornell University.

REFERENCES

[1] Zhongxiang Zhao, Zhihai Xu, Meijuan Bian, Analysis of Technological Features and Types of Digital Camera [J/OL]. CAMERA MAGAZINE. 2001,(7).

[2] Xudong Wang, Yutang Ye, Comparative research and future tendency between CMOS and CCD image sensor [J/OL].Electronic Design Engineering. 2010,18(11).

[3] Naweed Paya, Venkat Ganesh. Customizable Virtual Keyboard [EB/OL] . [2010-04-12]. http://courses.cit.cornell.edu/ee476/FinalProjects/s2008/nap33_vsg3/nap33_vsg3/index.html

[4] Li Pan, Qin Zheng. “Thumb Controlled Key Extraction Method of Virtual Keyboard for Touch-Screen Mobile”, Computer Systems & Applications, November 2010.

[5] Cheng-Huei Yang, Li-Yeh Chuang, Yu-Huei Cheng, Jun-Yang Chang, Cheng-Hong Yang. An Intelligent Chinese Phonetic On-Screen Virtual Keyboard System. [J/OL].2006,18(4)