2012 年臺灣國際科學展覽會

優勝作品專輯

國家：**United States**

編號：**110007**

作品名稱

Computer Vision for Alternative Input Systems

得獎獎項

一等獎

作者姓名

Matthew Thomas Vernacchia

Abstract

In the fast-paced environment of a hospital intensive care unit (ICU), good doctor-patient communication is essential. However, medical conditions and devices often inhibit a patient's ability to speak, write or type. Current assistive communication devices are either prohibitively expensive or cumbersome and time-consuming, creating a gap in communication during a patient's first days in the ICU. This project applies computer vision to develop a low-cost software solution that bridges this gap by enabling patients to generate words with eye movements. In the system, a webcam acquires an image of the patient, and an image processing algorithm classifies patient's gaze as pointing in one of eight directions. Each direction corresponds to an option on a graphical menu presented to the patient on the computer's display. The patient can use the menu to select a preformed phrase from a list of common phrases. Patients desiring to express more complex ideas can type custom words using the menu as an ambiguous keyboard (similar to a phone keypad). In either case, the patient-generated text will be displayed on screen and read aloud through the computer's audio system. The only hardware requirements are an existing computer and a $6 webcam. The program can process and respond to an image in 148ms. A new user can be trained in approximately 10 minutes, and after training can type a simple phrase such as "hello world" in 40 seconds. While further testing and improvement is required before the system will be ready for implementation, the project shows promise as a low-cost solution to ICU communication.

Project Summary

Problem Statement

In the fast-paced environment of a hospital Intensive Care Unit (ICU), good doctor-patient communication is essential. Medical conditions (paralysis, throat injury) or devices (intubation) often inhibit speech, and IVs or weakness can prevent writing. Eye movement is often these patients' only remaining form of communication. Computer-based systems that interpret a patient's eye movements are available, but are highly expensive, with some systems costing over $8000 ("Components"). Therefore, most communication in the ICU is currently done via pointing cards, a tedious and time-consuming process. An open-source eye movement- based communication system that uses existing hardware would be a significant aid in bridging this communication gap.

Design Goals

The goal of this project is to develop a program that grabs images of a user's eyes from a webcam and processes those images into commands. Through those commands, the user will control a graphical user interface (GUI) in order to type words or sentences. The image processing software must be able to reliably and accurately interpret user commands from eye movements, and the GUI must be straightforward and easy to operate. Additionally, the system must use only low-cost or pre-existing hardware.

Details of the Image Analysis Module

The first task of the image analysis software is to determine the points which represent the centers of the user's eyes. This could be accomplished with face recognition software; however face recognition software could be confounded by breathing tubes, bandages or bruises which may obscure a patient's face. Therefore, an alternative approach is required.

One of the visual characteristics which differentiates the eyes from the rest of the face is their ability to reflect light. Shining a light on the eyes produces a bright spot at the center of the eye (figure 1).

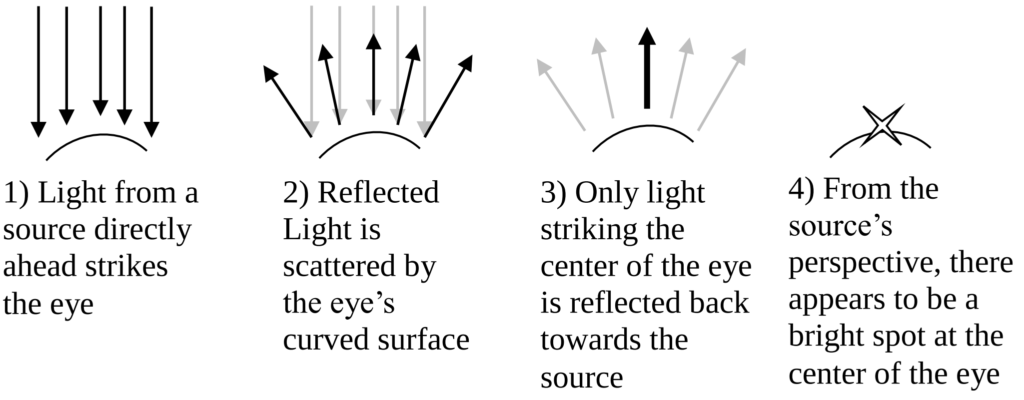

Figure 1 – the Optics of the Eye Detection System

Using this setup, image analysis software can easily find the center of the eyes by simply looking for the brightest spot in the picture. There are some obvious drawbacks, though, as this procedure would require constantly shinning a bright light into the user's eyes while operating in a relatively dark room. This discomfort can be avoided, however, by using near-infrared (IR) light. Since IR light is outside of the visual spectrum, the user would be unable to see it and experience no blinding or discomfort. IR webcams equipped with IR light-emitting diodes (IR-LEDs) can be purchased for around 6 USD.

To find the center of the eye, the software then scans the image for the brightest pixels. Since the image from the camera is grayscale, this simply consists of checking whether one of the RGB values for the pixel is over a certain threshold. If it is, the pixel is added to an ArrayList of bright Points.

Sometimes there are extraneous bright spots in an image due to reflective objects or lights in the background. In order to prevent these from interfering with locating the eyes, the image must be "sanitized" to remove extraneous bright spots. Two methods do this: first, the software removes any large "clumps" of bright Points, as the spot on the eye is only 4-8 pixels in size. Second, each bright spot undergoes a counterpart check. Since the face is symmetrical, the "true" bright points will have a counterpart on the other side of the image at about the same height. Any points that do not have a counterpart are removed from the list. The remaining points on each side of the image are averaged to find the center of the eye.

Once the center of the eye is thus determined, the image analysis software next must find the location of the iris. Iris location is performed by looking for a circular figure near the center of the eye. First, edge detection (Sobel filter) is performed on the region around the center of the eye (Figs 2 &3).

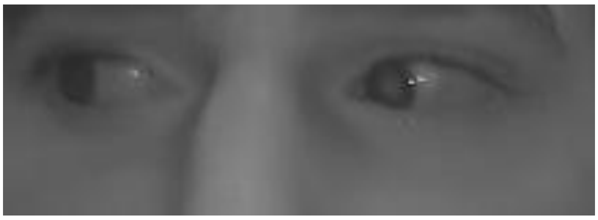

Figure 2 – Original Image

Figure 3 – Results of edge detection on Fig 3 (edges in white)

The edges of the iris will create a circle of edge points. A fitting algorithm such as the Hough transform can then be used to locate the center of a circle which best matches the edge points (Lazebnik np) (Fig 4 & 5), thereby locating the iris (Fig 6).

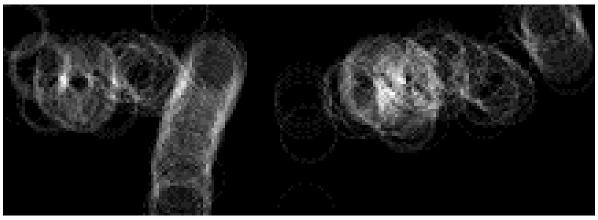

Figure 4 – Hough Space generated by the Hough Circles algorithm (lighter regions are better candidates for circle centers)

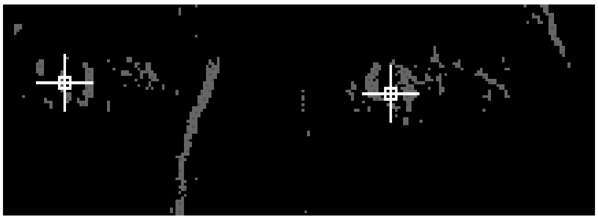

Figure 5 – Circles identified by the Hough Circles algorithm (centerpoint denoted by crosshairs)

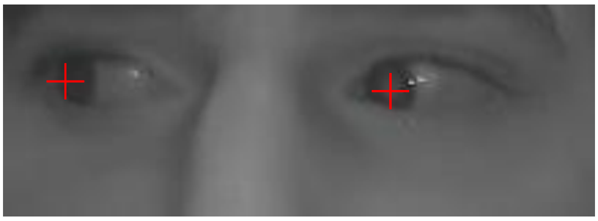

Figure 8 – Location of circle centerpoints on original image

Next, the program subtracts the coordinates of the iris Point from the coordinates of the eye center Point. The results become the x and y components of the gaze vector. The vector for the other eye is then found, and the two vectors are averaged. The resulting vector is classified as one of eight directional commands (up, down, left, right, up-left, up-right ...).

<u>Details of the Typing Interface Module</u>

The typing interface has two modes: Quick Phrase and Custom Phrase. In Quick Phrase mode, the GUI displays several icons to the user. Each icon consists of a common phrase and a representative image. The user then selects the desired icon via directional commands. The user then advances the Review Menu, and has the option to read the phrase aloud, start a new phrase, or add a word to the phrase.

In Custom Phrase mode, the user uses an ambiguous keyboard to type out words using the directional commands. In the system, the letters of the alphabet are divided up amongst six of the eight commands. One of the two non-letter commands (looking left) will be a backspace key to delete a mistaken letter command. To type a word, the secondary user will select the commands containing each letter in the desired word. For example, to type "cat', the user would select the [abcd] command, the [abcd] command again, then the [rstu] command. As the user enters letter commands, the program will present the user with a list of possible word options. A sub-routine will search a list of several thousand common words and return those that match the commands given by the user. In the above example, the user will be presented with the most common words whose first and second letters are (a, b, c, or d) and whose third is (r, s, t or u), i.e. "cat," "bat," "act," "car" etc. Once the desired word is seen near the top of the list, the user moves to the next menu to scroll through the list and select the desired word. A method in the Typing Interface module advances a word's position in the list each time it is used, so that more frequently selected words appear first in the menu. After typing and selecting the desired word, the user is brought to the Review menu, and again may choose to read the phrase aloud, start a new phrase, or add a word to the phrase.

Evaluation and Testing

Test 1 – Time required to type "hello world" (for a user who has been trained in the operation of the program).

Average result: 38.3 seconds

Test 2 - Time to Detect and Respond to a Directional Command

148 ms - average of 50 commands, recorded and calculated by the program during a test run.

Test 3 – Error Rate

18% - During a test run, the user made 50 directional commands. The command was correctly identified 41 times, and incorrectly identified 9 times.

Discussion and Conclusion

The design approach taken generated a functional solution that addresses the problem of doctor-patient communication in the Intensive Care Unit (ICU). The product requires low hardware costs, as a 6 USD webcam is the only additional equipment that must be purchased. It is simple to use, and most new users can be trained in approximately 10 minutes. The image processing algorithm can handle a variety of image inputs, even if there are extraneous bright objects in the image. It does not require any indicators or markers to be placed on the patient's face. The system is also relatively efficient, being able to detect and respond to an eye movement in 148 ms. Additionally, the system enables the user to communicate more rapidly than the solution currently in use. However, the observed error rate of 18% is higher than desired.

There are two sources of this error. First, inaccuracies in the Image Analysis algorithm may cause the gaze direction vector to indicate one direction when the user in fact looked in a different direction. This problem must be remedied by rewriting and improving the Image Analysis algorithm. Incorporation of Intel's OpenCV Computer Vision Library in future versions will make it possible to quickly implement and test more advanced algorithms, enabling an improvement of the project.

Second, the user may intend to look in one direction, but in fact look in another direction. Looking up and down with accuracy is easy. However, humans usually rotate their heads to look to the side, and may have some trouble making sideways eye motions that are distinguishable from diagonal motions. However, this difficulty can be partially overcome with training. An alternative approach would be to develop an eye-typing system based on four commands. A comparative study could then determine whether the higher error rate of a eight-command system or the limited menu options of a four-command system is a more significant hindrance.

Menu layout optimization would be another approach to improving the system. The ambiguous keyboard could be improved by arranging the letters so that the most frequently used letters appear on the easiest to access buttons (up and down). However, a non-alphabetical rearrangement may confuse users and making finding a letter difficult.

A better candidate for optimization is the icon menu. A survey of hospital staff and patients should be conducted to determine which phrases should be included in the icon menu, and in which order.

Even with these areas of potential improvement, the system shows great promise for enhancing doctor-patient communication in the ICU. A future version encompassing the aforementioned enhancements could be truly revolutionary.

References

Bennet, Nathan. Personal Interview. 25 December 2010.

"Components and Prices of the Eyegaze Edge." *Assistive Technology.* LC Technologies, Inc., 2009. Web. 19 December 2010. <http://www.eyegaze.com/content/components-and-prices-eyegaze-edge>.

"Development of Gaze Tracking Technology." *Research*. ITU GazeGroup, n.d. Web. 8 August 2010. <http://www.gazegroup.org/research>.

"Facts You Should Know About ALS." *About ALS*. ALS Association, 2007. Web. 19 December 2010. <http://www.alsa.org/als/facts.cfm>.

Lazebnik, Svetlana. **"**Edge detection**."** *COMP 776: Computer Vision*. University of North Carolina, 29 Jan. 2008. 26 Feb. 2011. <http://www.cs.unc.edu/~lazebnik/research/spring08/lec06_edge.pdf>.

Nesbat, Saied B. "A system for fast, full-text entry for small electronic devices." *Proceedings of the 5th international conference on Multimodal interfaces* (2003): 4-11. Web. <http://doi.acm.org/10.1145/958432.958437>.

"OpenCV." *OpenCV.* Willow Garage, 2010. Web. 1 May 2011. <http://opencv.willowgarage.com/wiki/>

Rus, Daniela. "Cameras, Images, and Low-Level Robot Vision." *Robotics: Science and Systems I*. MIT Computer Science and Artificial Intelligence Laboratory, 16 Feb. 2011. Web. 25 Feb 2011.

"T9®: The Global Standard for Mobile Text Input." *Nuance*. Nuance, 2010. Web. 3 Sept. 2010. < http://www.nuance.com/for-business/by-product/t9/index.htm>.

評語

Good application.

Good presentation.