Real Time Gesture Controlled Electrical Automation

Deepa B[1], Mohammed Jubail[2], Mohammed Saif[3], Sohana N[4]
[1]Asst. Professor, Department of Electrical And Electronics Engineering, New Horizon College of Engineering
[2,3,4]Student, Department of Electrical And Electronics Engineering, New Horizon College of Engineering
Outer Ring Road, Marathalli, Bengaluru- 560 103

Abstract- *Automation is the essential need for the present world. There are various types of automation like building automation, industrial automation, home automation, artificial intelligence etc. However the methods used in automating are complex and not helpful at times, therefore automating the same processes using gesture can be one solution to this problem.*

The objective is to implement Image processing 1 algorithms for automation purposes effectively for smarter control of machines. Our project focuses on real time image processing algorithms that can be used to detect, track objects and persons and can also recognize their activities and gestures in indoor scenarios. This project demonstrates home lighting and fan control where these algorithms can be used to achieve effective control.

Hand gestures are powerful means of communication among humans and sign language is the most natural and expressive way of communication for dump and deaf people. In this real time hand gestures system is proposed. Experimental setup of the system uses fixed position low cost web camera with 10 mega pixel resolution mounted on the top of the monitor which captures snapshot using red green blue (RGBP) colors 2 colors from fixed distance. This work is divided into four stages such as image processing, Region extraction 3 , Feature extraction 4 and Feature matching 5.

Keywords-

1. Image processing: An image is an array, or a matrix of square pixels arranged in columns and rows. In a grey scale image each square pixel has an assigned intensity that ranges from 0-255
2. RGB colors: It is the basic color model used in computers and for web graphics, but it cannot be used for print production. The combination of RGB colors in full intensity makes white.
3. Region extraction: The binary image is calculated using bwlebel and regionprops is applied to find bounding box for white pixels of hand image edge detection of extracted image using 'Sobel' method.
4. Feature extraction: Feature vectors of training images are stored in mat file of MATLAB and feature vector of input hand gesture image are calculated at run time.
5. Feature matching: Feature matching is done using Euclidian distance. Euclidian distance between feature vector of input real-time image and feature vector of each training image is used for recognition of perfect matching hand gesture.

I. INTRODUCTION

Robots are used successfully in many areas today, particularly in industrial productions, military operations, and deep sea drilling and space exploration. This success drives the interest in the feasibility of using robots in human social environments, particularly in the care of the aged and the handicapped. In Social Environments, humans communicate easily and naturally by both speech and gesture without the use of external devices requiring special training. Robots have to adapt to human modes of communication to promote a more natural interaction with humans.

Given a choice between speech and gesture, some researchers have opined that gesture recognition will be more reliable than speech recognition because the later would require a large number of training data sets to deal with the great variability in human speech and voice. Therefore this project is about implementing the control of a robot through simple hand gestures. The main motivation is the desirability of developing robots that can interact smoothly with humans without the need of any special devices.

II. METHEDOLOGY

During the project, four gestures are chosen to indicate different inputs to the equipment's in use varying the voltage levels at each gesture. A simple computer vision application is written for the detection and recognition of the gestures and their translation into the corresponding commands for the applicationsThe appropriate OpenCV functions and image processing algorithms for the detection and interpretation of the gestures were used. Thereafter, the program was tested on a webcam with actual hand gestures in real-time and the results were observed.The results of the project demonstrated that a simple computer vision application can be designed to detect and recognize simple hand gestures for robot navigational control based on simple heuristic rules.

The program was able to correctly interpret the gestures and translate it into the corresponding commands most of the time.

III. LITERATURE SURVEY

The sense of sight is arguably the most important of man's five senses. It provides a huge amount of information about the world that is rich in detail and delivered at the speed of light. However, human vision is not without its limitations, both physical and psychological. Through digital imaging technology and computers, man has transcending many visual limitations. He can see into far galaxies, the microscopic world, the sub-atomic world, and even “observe” infra-red, x-ray, ultraviolet and other spectra for medical diagnosis, meteorology, surveillance, and military uses,all with great success.

The main difficulty for computer vision as a relatively young discipline is the current lack of a final scientific paradigm or model for human intelligence and human vision itself on which to build a infrastructure for computer or machine learning.

The use of images has an obvious drawback. Humans perceive the world in 3D, but current visual sensors like cameras capture the world in 2D images. The result is the natural loss of a good deal of information in the captured images. Without a proper paradigm to explain the mystery of human vision and perception, the recovery of lost information from 2D images represents a difficult hurdle for machine vision. However, despite this limitation, computer vision has progressed, riding mainly on the remarkable advancement of decades-old digital image processing techniques, using the science and methods contributed by other disciplines such as optics, neurobiology, psychology, physics, mathematics, electronics, computer science, artificial intelligence and others.

Computer vision techniques and digital image processing methods both draw the proverbial water from the same pool, which is the digital image, and therefore necessarily overlap.

Image processing takes a digital image and subjects it to processes, such as noise reduction, detail enhancement, or filtering, for the purpose of producing another desired image as the end result. For example, the blurred image of a car registration plate might be enhanced by imaging techniques to produce a clear photo of the same so the police might identify the owner of the car.

On the other hand, computer vision takes a digital image and subjects it to the same digital imaging techniques but for the purpose of analysing and understanding what the image depicts. For example, the image of a building can be fed to a computer and thereafter be identified by the computer as a residential house, a stadium, high-rise office tower, shopping mall, or a farm barn.

Open CV

OpenCv is a widely used tool in computer vision. It is a computer vision library for real-time applications, written in C and C++, which works with the Windows, Linux and Mac platforms. It is freely available as open source software from http://sourceforge.net/projects/opencvlibrary/. OpenCv was started by Gary Bradsky at Intel in 1999 to encourage computer vision research and commercial applications and, side-by-side with these, promote the use of ever faster processors from Intel.

OpenCV contains optimised code for a basic computer vision infrastructure so developers do not have to re-invent the proverbial wheel. The reference documentation for OpenCV is at http://opencv.willowgarage.com/documentation/index.html. The basic tutorial documentationis provided by Bradsky and Kaehler. According to its website, OpenCV has been downloaded more than two million times and has a user group of more than 40,000 members. This attests to its popularity.

A digital image is generally understood as a discrete number of light intensities captured by adevice such as a camera and organized into a two-dimensional matrix of picture elements or pixels, each of which may be represented by number and all of which may be stored in a particular file format (such as jpg or gif) [8]. OpenCV goes beyond representing an image as an array of pixels. It represents an image as a data structure called an IplImage that makes immediately accessible useful image data or fields, suchas:

- width– an integer showing the width of the image in pixels
- height– an integer showing the height of the image in pixels
- imageData– a pointer to an array of pixel values
- nchannels– an integer showing the number of colors per pixel
- depth– an integer showing the number of bits per pixel
- widthStep– an integer showing the number of bytes per image row

DESIGN

Software is developed in C++ which is an object oriented language like Java therefore will need to apply software engineering in an object oriented environment. For the development of high quality software, it must meet the user requirements set at the starting of the project. The software should be reliable with few bugs and should be well maintainable.

Functional Requirements

1. The camera used will be able to capture user images from the video sequences.
2. The software will be able to produce multiple frames and display the image in the RGB colorspace.
3. The software will be able to display the converted RGB image in a new window.
4. The software will be able to detect the skin regions of the user in the image captured.
5. The software will be able to detect the contours of the detected skin regions.

UML Diagrams

UML (Unified Modelling Language) is a most useful method of visualization and documenting software systems design.UML includes a set of graphic notation techniques to create visual models of software systems. It is used to specify, visualize, modify, construct and document the artefacts of an object-oriented software system under development.

WORKING

Gathering the requirements and the proposed design concepts, the algorithm can be implemented to capture the image, detect the skin in the image captured and detect the contour that will make tracking of the hands easier for gesture recognition.

To do this, the algorithm must be implemented using a flexible programming language, which has already been used by other programmers for developing such software.

Development Tools

After looking at many past projects on similar software, the decision of using Microsoft Visual C++ as a programming environment was taken. Also, the OpenCV library, which is an open source library which contains functions that specializes in image processing and gesture recognition, used in this project is compatible with C++ programming language. Once the code has been implemented, the testing of the code is carried out. Testing ensures that the software developed functions as expected. There are two basic testing techniques available, Black box testing and white box testing.

The white-box testing focuses on the internal structure of the code. The white box tester (most often the developer of the code) looks into the coding of the system that’s has been implemented to look for any errors by writing test cases. Here the inputs are chosen specifically to determine an appropriate output the result can be evaluated more clearly as said in “Redstone Software Inc” (2008). White-box testing is often used for verification.

BLACK-BOX TESTING

Test	**Description**	**Desired Outcome**	**Actual Outcome**	**Result**
1	Initialize camera	Display user image in new window	Displayed user original image in new window	PASS
2	Image conversion	Display HSV image in new window	Displayed HSV image in new window	PASS
3	Filtering	Smooth out image	Smooth out image	PASS
4	Skin detection	Display skin areas in the image in a new window	Displayed skin regions as white with background removed	PASS
5	Contour detection	Draw contour around skin region	Green outline drawn along outline of the skin region detected	PASS
6	Exit	Exists when user clicks ‘ESC’ button	Exists when ‘ESC’ button is clicked by user	PASS

Table 1 shows the Black Box tests that were implemented for the code developed, tests labeled “pass” have passed the test at the first iteration of the code. Nearly all of the test cases passed in the software.

IV. LIMITATIONS AND FUTURE DEVELOPMENT

The popularity of gesture based systems have grown over the years and I believe that it will continue to expand and become an essential part of everyday like as the technology advances. This project has expanded my knowledge about the different techniques used in the develop of such systems and its useful application in different fields.

My system is now capable of performing basic image processing like conversion between two different colour spaces, differentiates between skin pixels and non-skin pixels in the image, performs background subtraction and cancels out major noises associated with the image.

However there are limitations to software functions and may not always produce expected outputs. One of the major limitations is that I have only tested my system in a few environments therefore low or varying lighting conditions may affect the outputs produced. Also a more complex image filtering should be applied to the image as it still contains some amount of noise associated with the final image.

In regards to further development, I feel that the project can be expanded further if the minor environmental and filter issues can be resolved. In order to improve the project further, research and further development on the filters used should be critically examined. I feel that once these issues have been further implemented, the algorithms developed can be used as a basis for future development. Functions for feature extraction, hand tracking and gesture recognition can be added to make the system usable. Right now the project just looks into image processing but after adding further functions to the current algorithm, the system can be used to assist disabled and elderly to interact with the computer with ease. The project therefore has immense scope in terms of the functionality that can be changed to gain better results.

Gesture based system can also be used in the gaming field to provide the gamers with an ultimate touch less gaming experience. Image processing can be used in robotics that can be used to further develop an aid to help elderly autonomously commute in a building/house, which will help immensely.

V. DISCUSSION AND CONCLUSION

The results also showed that the gesture recognition application was quite robust for static images. However, the video version was enormously affected by the amount of illumination, such that is was necessary to check and adjust the HSV values for skin colour when starting the program to get the proper output. Sometimes the adjustment was difficult to do because of the lighting conditions and the amount of objects in the background.

The application was very susceptible to noise on the video stream. Slight hand movements could affect gesture recognition. Nevertheless, if the hand is steady enough for long enough, the program outputs the correct command.

It was also observed that while the program was executing there were memory leaks. Attempts to remedy the problem were made by using the OpenCV functions to release memory. Despite this, the leaks continued. Perhaps the leaks were due to the implementation of OpenCV functions for the sequences behind the scenes.

For integrating the program with the robot in the future, it would be necessarily to consider other output such as speed or velocity as part of the navigational control commands.

Based on the results, a computer vision application could detect and recognize simple hand gestures for robot navigation using simple heuristic rules. While the use of moment invariants was not considered suitable because the same gestures could be used pointing in opposite directions, other learning algorithms like AdaBoost could be explored to make the program more robust and less affected by extraneous objects and noise.