

Customizable Virtual Keyboard

Laser tracking based keyboard

B. Priyalakshmi¹, Achari Ayyappan S², Shubho Jyoti Chatterjee³, Aniruddha Maiti⁴

Department of Telecommunication Engineering, SRM University, Kattankulathur Chennai-603203, India

Abstract— It is becoming increasingly difficult for users to interact with the slew of portable gadgets they carry, especially in the area of text entry. To solve this problem, we proposed a Virtual Keyboard. This device will replace a physical keypad with a customizable keyboard printed on a standard A3 size paper whose “keystrokes” are read and translated to real input. In addition, buttons on this device can be reconfigured on-the-fly to give a new keyboard layout using a GUI we built in Mat lab and then transferring that data to the device using a computer’s serial port.

during implementing the real time vision based human – computer interaction system.

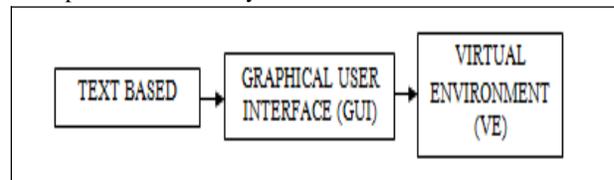


Fig. 1 Evolution of Human interface.

Index Terms—Keystroke, A3, GUI, Serial port.

I. INTRODUCTION

With rapid growth and latest advances in computer vision field in terms of processing capability and cost effectiveness, vision based human-computer interface (HCI) which offers contact interaction with a device is becoming a promising field. In virtual environment, instead of using conventional keyboard, virtual keyboard is the intuitive, immersive and cost-efficient HCI device. This paper presents an implementation of customizable virtual keyboard which appoints fingertips of user as the intuitive input device by image processing made as on a one of its kind MATLAB program.

In this era of technological advances, readily available and cost effective devices such as webcam and increased processing capability with the latest advances in computer vision field, the real time vision and virtual based human-computer interaction is becoming the most promising approach which can be well integrated and embedded into human world. This approach enables us to give inputs, instructions and commands to the computing devices by using the most natural, intuitive and convenient ways which is to employ human body such as human hand as an input device .

Uncontrolled environment, differences of human skin color intensity, rapid human hand movement, lighting source and conditions are the challenges and the important considerations

II. INTRODUCTION TO TOOLS

1. **Image Processing with MATLAB:** Images are everywhere, from everyday devices like cameras and smart phones to specialized devices for medical imaging, automotive safety, industrial automation, and more. Each of these uses for image processing has unique challenges.

MATLAB and Image Processing Toolbox provide a flexible environment to explore design ideas and create unique solutions for Matlab toolbox used in our project are as follows:

- Image acquisition Toolbox.
- Image processing toolbox
- GUI builder.

a) *Image Acquisition Toolbox:* The Image Acquisition

Toolbox is a collection of functions that extend the capability of the MATLAB® numeric computing environment. The toolbox supports a wide range of image acquisition operations, including. Acquiring images through many types of image acquisition devices, from professional grade frame grabbers to USB- based Webcam .Viewing a preview of the live video stream. Triggering acquisitions (includes external hardware triggers). Configuring call back functions that excuse when certain events occur. Bringing the image data into the MATLAB workspace.

b) *Image Processing Toolbox:* Image Processing Toolbox provides a comprehensive set of reference-standard algorithms, functions and apps for image processing, analysis, visualization and algorithm development. You can perform image enhancement, image deblurring, feature detection, noise reduction, image segmentation, geometric transformations, and image registration. Many toolbox functions are multithreaded to take advantage of multi core and multiprocessor computers.

Image Processing Toolbox supports a diverse set of image types, including high dynamic range, gigapixel resolution, embedded ICC profile, and tomographic. Visualization functions let you explore an image, examine a region of pixels, adjust the contrast, create contours or histograms, and manipulate regions of interest (ROIs). With toolbox algorithms you can restore degraded images, detect and measure features, analyze shapes and textures, and adjust color balance.

c) *GUI Builder:* A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB® tools, can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

- $L^*a^*b^*$ Colour Space is used for colour segmentation and to identify the colour of the rows (data) and to process each of them.
- A Lab colour space is a colour opponent space with dimension L for lightness and a and b for the colour-opponent dimensions, based on nonlinearly compressed CIE XYZ colour space coordinates.

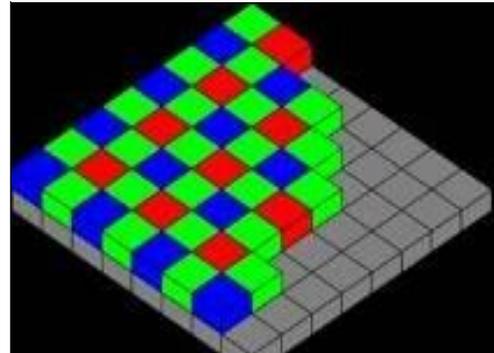


Fig 2: Bayer color Pattern

2. Webcam:

This Webcam is used to capture and recognizes an object in view and tracks the user's hand gestures using computer-vision based techniques. It sends the data to the computer. The camera, in a sense, acts as a digital eye, seeing what the user sees. It also tracks the movements of the thumbs and index fingers of both of the user's hands. The camera recognizes objects around you instantly.

The live stream images are transmitted to the computer system and is processed by the code in MATLAB.

- High quality digital CMOS image sensor
- With a USB 2.0 cable attached to it.

Fig.3: Webcam



Camera Specification:

Hardware	Specification
Sensor	High quality digital CMOS image sensor
Interface	USB 2.0
Drive	Free, plug and play
Automation	Colour compensation, White balance, Exposure
Frame Rate	VGA: 30 f/s, CIF:15 f/s
Focus Range	5cm - infinity
Image Format	BMP/JPG
Video Format	AVI
Max Resolution	1280*1024 pixels
Colour Depth	24 bit RGB
Interpolated Resolution	3 Mega Pixel
Dynamic Range	>72 Db(30F/s,220LUX)
Power Dissipation	<0.7W
Working Temperature	-10-75°C
Hi-resolving power	640*480/ 320*240

3. Laser

The laser is used for indicating the position of the finger stroke by shining a bright red light on it, and this red light is captured by the webcam by the color recognition codes in our MATLAB program.

Laser Type: LED

Voltage specification: 3.3 V

We are using a power adapter to step down the voltage from a USB port which has an output of 5 volts.

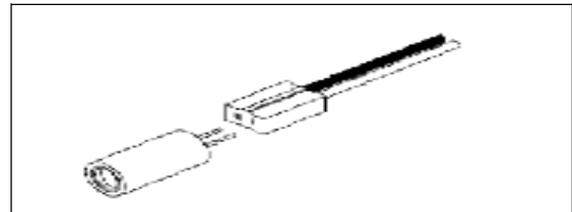


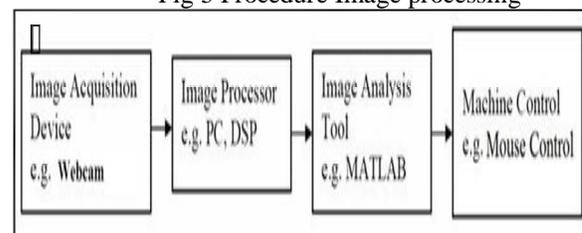
Fig 4: Laser

III. WORKING

Pictures are continuously received from the webcam. All the grids are pre specified for their own functions so whichever grid contains the changes picture the respective function is called. As the related work shows, the choice of the algorithms chiefly depends on the environment in which the system is required to operate. A larger setup of the system in an open environment, such as a shopping mall or outdoors requires robust hardware that can cope with many users and difficult lighting conditions such as sunset. The choice of hardware to capture the users is a defining element of the reliability of the final system.

- Image acquisition setup: It consists of a web camera with suitable interface for connecting it to PC.
- Processor: It consists of personal computer or a dedicated image processing unit.
- Image analysis: Certain tools are used to analyze the content in the image captured and derive conclusions e.g. MATLAB 7.0.
- Machine control: After processing, some conclusions have to be made in order to initiate control actions. In our case control actions are desktop control via mouse control.

Fig 5 Procedure Image processing



Implementation of image processing:

As we all know, an image is defined as 2D function of $f(x,y)$ where x & y are spatial coordinates and amplitude at any pair of coordinates is called intensity of image at that point

1. Initialize the webcam and capture the image.
2. Then convert the captured colored image to gray scale image using image processing toolbox. The gray scale contains only brightness information about the image.
3. Later use image segmentation to locate objects and their boundaries in an image.

In image segmentation, thresholding is their gray scale value. If pixel intensity is less than threshold, corresponding pixel is black in resulting image. If pixel value has intensity greater than threshold, the resultant pixel is white which represents object.

4. Thus binary Images contain only black and white pixels which indicate that it keeps only the significant part of the image and removes the noise.

5. Thus the required target is white and the surrounding background is black. This is how image processing is carried out.

changed coordinate within all the grids. All the grids are pre specified for their own functions so whichever grid contains the changes picture the respective function is called.

The rectangle containing all the pixels that have changed is calculated. Since the laser only generates a small spot, we check the size of the rectangle:

This allows the webcam to adjust itself to the new laser stroke and execute event as per it. If the rectangle size makes sense, the list of hotspots is searched for a hotspot in which the rectangle fits.

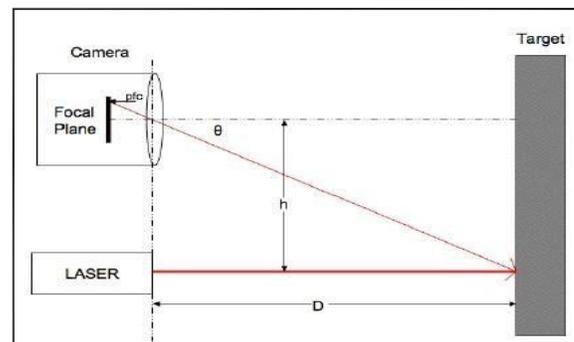


Fig 7: Basic layout

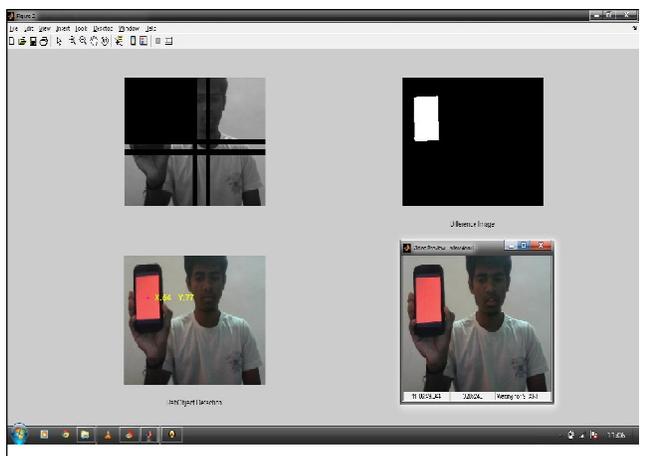


Fig 6: Color recognition

Pictures are continuously received from the webcam. Here basically we put an infinite loop for capturing the frame and put image in buffer so that we can compare it with the next frame image. A very simple method to track the laser point is to recognize the part where the frame difference is found. So we divide the entire surface into 6 by 5 Grids and look for the

IV. IMPLEMENTATION

There are some factors we should consider before implementing the whole system. While typing due to lag the system might repeat the same alphabet together and also might misalign a particular box.

A. Anti-repeat

To prevent a hotspot from being activated more than once because you left the pointer inside it a little too long, the following strategies are used: Once a hotspot is activated, the following pictures containing changes (valid or not) are ignored until a picture without changes (not laser pointer) is received. (Note: This does not apply to repeatable zones) When a hotspot is activated and the hotspot is disabled during one second. Take the camera, point it at something. Shine the laser inside the area where the camera can see. That's it in a nutshell. However, there are some additional considerations.

First, the more direct the camera, the more accurate it will be. If the camera is off to the side, it will see a skewed wall, and because one side is closer than the other, it's impossible to focus perfectly, and one side of the wall will be more precise



International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE '14)

Organized by

Department of ECE, Aarupadai Veedu Institute of Technology, Vinayaka Missions University,
Paiyanoor-603 104, Tamil Nadu, India

than the far side of the wall. Having the camera point as directly as possible and at the hooked surface is the best option.

Second, the distance to the surface matters. A camera that is too far from the surface may not be able to see a really small laser point. A camera that is too close will see a very large laser point and will not have great resolution. Its tradeoffs and you just have to experiment to find the best distance.

Third, if using wall as the projected area, the camera should be able to see slightly more than the projected area. A border of a few inches up to a foot is preferable, as this space can actually be used for input even if it's not in the projected space.

Fourth, it's important to control light levels. If there are sources of light in the view of the camera, such as a lamp or window, then it is very likely the algorithm will see those points as above the threshold, and will try to consider them part of the laser pointer (remember white light is made up of red, green, and blue, so it will still be above the red threshold).

Also, if using a projector, the laser pointer has to be brighter than the brightest the projector can get, and the threshold has to be set so that the projector itself isn't bright enough to go over the threshold. And the ambient light in the room can't be so bright that the threshold has to be really high and thus the laser pointer isn't recognized.

Again, there are a lot of tradeoffs with the light levels in a room.

B. Basic Theory

The basic theory behind the project is the following; a laser pointer shines on a surface. The web camera is looking at that surface. Software running on a computer is analyzing each frame of that camera image and looking for the laser pointer. Once it finds that pointer, it converts the camera coordinates of that point into screen coordinates and fires an event to any piece of software that is listening. That listening software can then do something with that event. The simplest example is a mouse emulator, which merely moves the mouse to the correct coordinates based on the location of the laser.

C. Implementation Theory

To implement this, I have the JMF library looking at each frame. I used this frame access.java example code as a starting point. When looking at each frame, I only look at the 320x240 frame, and specifically only at the red value. Each pixel has a value for red, green, and blue, but since this is a red laser I'm looking at, I don't really care about anything but red. I traverse the entire frame and create a list of

any pixels above a certain threshold value. These are the brightest pixels in the frame and very likely a laser pointer. Then I average the locations of these points and come up with a single number. This is very important, and I'll describe some of the effects that this has later. I take this point and perform the affine transform to convert it to screen coordinates. Then I have certain events that get fired depending on some specific things:

Laser On: the laser is now visible but wasn't before.

Laser Off: the laser is no longer visible.

Laser Stable: the laser is on but hasn't moved.

Laser Moved: the laser has changed location.

Laser Entered Space: the laser has entered the coordinate space.

Laser Exited Space: the laser is still visible, but it no longer maps inside the coordinate space. For most of these events, the raw camera coordinates and the transformed coordinates are passed to the listeners. The listeners then do whatever they want with this information.

Limitations:

For most of these events, the raw camera coordinates and the transformed coordinates are passed to the listeners. The listeners then do whatever they want with this information.

V. CONCLUSION

This project presented a laser-based light tracking system that does not require any special device or instrument and can operate in real-time on a commodity PC with low-cost cameras and laser. Specifically, the system can track the tip positions of the laser light that fall on the index finger for each hand, assuming that a calibrated camera is viewing the hands placed in front of it. The motivation for this laser tracker was a desktop-based two-handed interaction system in which a user can select and manipulate 3D geometry in real-time using natural hand motions. The algorithmic details for the virtual keyboard were presented in this document. Virtual keyboard integrated with cloud computing will be capable of recognizing an object and will display the information about it automatically. It will also allow us to access information regarding the specific object in any way we need. The Virtual Keyboard prototype implements several applications that demonstrate the usefulness, viability and flexibility of the system. Laser Powered Virtual Keyboard uses webcam to monitor the segmentation area, detects if any laser light comes to any segment and then does the equivalent of pressing a specified keyboard key. Here to interact with media device i.e. camera we use MatLab. Laser Control with a MatLab program that using a webcam detects when a laser pointer is in a



specified hotspot and then does the equivalent of pressing a specified keyboard key. Thus we can control Media Portal or applications through the hot grids. The hotspots can be set by the user as well as the actions taken. We can make our wall as a remote control.

ACKNOWLEDGMENT

We would like to take this opportunity to express our deep gratitude to Mrs. B. Priya Lakshmi (Assistant professor) for her constant encouragement, help and valuable suggestions.

We would also like to sincerely express our thanks to Dr. T Rama Rao, (Head of Department, Information and Telecommunication) for her timely cooperation and support.

REFERENCES

- [1] Kazim Sekeroglu, "Virtual Mouse Using a Webcam" July 9, 2012
- [2] Y. Sato, Y. Kobayashi, H. Koike "Fast tracking of hands and fingertip in infrared images for augmented desk interface", 2000
- [3] Z. Zhang, Y. Wu, Y. Shan, S. Shafer "Visual panel: Virtual mouse keyboard detection" July, 2003
- [4] Shahzad malik "Real time hand tracking and fingertip tracking for interaction CSC2503E Project Report December 18, 2003 [5]
- [5] Georg Hackenberg, Rod McCall, and Wolfgang Broll "Lightweight palm and finger tracking for real time 3D gesture control", March
- [6] Wayne Westerman "Finger identification and choric manipulation of multi-touch surface", Spring 1999
- [7] <http://www.slideshare.net/renjithravi/sixth-sense-technology>.
- [8] Arjun, K. R "Sixth Sense Technology" Seminar Report 2012.
- [9] Mistry Pranav "Sixth Sense." Weblog post Sixth Sense. Pranav Mistry, 2010. Web..
- [10] Yanhua Sun, Noriyaki Kuwahara, Kazunari Morimoto "Analysis of Recognition System of Japanese Sign Language using 3D Image Sensor, 2013:
- [11] Jiayang Liu, Zhen Wang, and Lin Zhong, Department Of Electrical Computer Engineering Rice University, Houston TX 77005 and Jehan Wickramasuriya and Venu Vasudevan, Pervasive Platforms & Architectures Lab Applications & Software Research Center, Motorola Labs, 2009; <http://www.ruf.rice.edu/~mobile/publications/liu09percom.pdf>
- [12] UW team works on gesture recognition technology <http://allsee.cs.washington.edu/files/allsee.pdf>
- [13] M. Bartlett, G. Littlewort, B. Braathen, T. Sejnowski, and J. Movellan. A prototype for automatic recognition of Spontaneous facial actions. In S. Becker, S. Thrun, and K. Obermayer, editors, Advances in Neural Information Processing Systems, volume 15, pages 1271– 1278, 2003.
- [14] T. N. Bhaskar, F. T. Keat, S. Ranganath, and Y.V. Venkatesh. Blink detection and eye tracking for eye localization. In Proc. TENCON 2003, volume 2, pages 821– 824, 2003.