

Review on Laser Beam Operated Windows Operation

Prof. Swati Shinde¹, Mayur Dhaigude², Prashant Dorage³, Sachin Ghadge⁴, Soumya Dhingra⁵
Department of Electronics and Telecommunication
Mumbai University
Mumbai, India

Abstract — We describe how to use a laser pointer and a camera to emulate a mouse, allowing a presenter to control a computer remotely. The utility of this system would be in a large classroom environment or a presentation where an instructor or presenter can simply use the laser pointer and move the mouse on the projected screen without using the computer mouse. To move the mouse to a particular position on the screen, the user just needs to point the laser at that particular position and that does the job. The laser pointer can also be used to emulate mouse clicks by turning the laser off and on at the same position on the screen. The main advantage of this system would be in a large classroom environment where an instructor is moving around in the class and presenting something on a computer screen (say a power point presentation) projected by means of a projector. The instructor does not at every instant need to go to the computer mouse and change the slides; rather he can just use this software and can emulate a mouse (motion and clicks) by the use of just a simple laser pointer.

Keywords – Laser; Mouse Operations; Webcam; MATLAB.

I. INTRODUCTION

Consider a large room where a PowerPoint presentation is being made using a projector. The presenter normally uses a laser pointer to point on to the screen to some details in the current slide[1]. But imagine if the presenter could also be able to change the slide using the same laser pointer and also the presenter could also be able to do other activities like switch windows, open a new application etc, which he wouldn't be able to do otherwise. This could be realized if the laser pointer is somehow used as a device which would be able to emulate the mouse on the screen. This idea gives rise to this project where we design an implement a simple tool which could generate mouse events by detecting a laser pointer on the screen.

The basic idea of the operation of this tool is to capture the image of the projected screen using a camera and then in this captured image, we detect the laser pointer and identify its corresponding position on the projected screen and move the mouse accordingly [2]. The tool is designed to work on Microsoft's windows operating system.

In the present paper we have developed a full suite of interactive techniques that can work as practical information

manipulation tools. Kirstein and Muller [3] have reported a similar approach to interactive input. Their approach was to map the laser appearance, movement and disappearance to mouse down, move, and up events in X- Windows. But this type of simple mapping is not sufficient for general information manipulation. More over Kirstein and Muller [3] presented a system that uses a laser pointer as a pointing device but the system suffers has got a severe drawbacks. Their system may cause false triggering in case of dynamic background of the screen. Moreover they reported that the reliability of their system is only 50% i.e. their system is able to detect laser spot in only 50% of the frames. It means the chances of failure are more than 50% and so the scope is limited. A single-pointer system for executing more complicated tasks is presented by Olsen and Nielsen [4]. This paper has an interaction technique that uses widgets. The selection of widget works well with user interface but requires a feedback procedure. It also projects an alien object on the user interface, which creates distraction.

Our tool runs in two main stages namely, Initialization and Detection. In an ideal situation, the camera would be able to see only the projected screen and thus, the corners of the image created by the camera would co inside with the corners of the projected screen. Thus, just detecting the laser pointer and scaling the image to the actual resolution of the screen would be enough to calculate the corresponding mouse position on the screen. But this does not happen in actuality. The camera does not see only the projected screen; it also sees the surrounding background with it. Thus, it becomes necessary to input into the system the position of four corners of the projected screen on the captured image so that the program can calculate the position of the mouse. Calculations of the position of the mouse pointer are made based on these four corner points. This is done in the initialization stage. This calibration is done by showing the user an image of the screen captured by the camera. The user clicks on the four corners of the screen in the image and these coordinates are saved by our program as the four corners of the screen and all the further calculations of the mouse pointer are made using these coordinates as reference points.

After the initialization the camera starts capturing the image of the projected screen and passes this image to the image processing module. The laser detection module checks the

image received from the camera for the presence of a laser point. Depending upon the presence or the absence of the laser, various decisions of moving the mouse or performing clicks are taken.

II. SYSTEM ARCHITECTURE

Our system has two distinct parts:-

A. Hardware Part

The hardware system consists of LCD beam projector, laser pointer, camera and the computer. The beam projector functions for the projection of image. The laser pointer functions for interaction with the projected image. The webcam captures the image and obtains the position of laser. The computer functions as the processing unit. Software part for the image processing consists of color setting module and calibration module. Fig 1 presents the basic diagram of the interacting module.

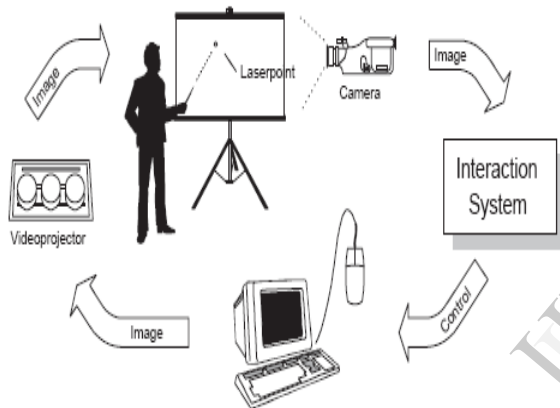


Figure 1- Scenario of the interactive system

B. S/W Part

Software part for the image processing consists of initialization and detection module. The initialization module further consists mainly of the calibration module. The calibration module finds out the position of laser pointer. The detection module further mainly comprises of color setting module and transformation module[5]. The color setting module can find the laser point by input image. To obtain the laser point, the color setting module is subdivided into RGB, the red layer being removed and then the input image is converted into gray scale image. We propose the hardware system which is inexpensive technique whereas every person in the room using a laser pointer can interact with the information on a large projected display. Interaction is performed by using the laser to point at displayed widgets to manipulate their functions [6].

III. WORKING

A .Initialization

The program enters the initialisation stage when first run.

Here, user is asked to input the four corners of the screen in the projected image. Once the four corners of the rectangular screen are input by the user, they are used as the reference points in all the further calculations of the mouse pointer locations.

After the user clicks on the four corners, the escape button on the keyboard is pressed, which completes this stage and the program now goes into the detection stage.

B .Detection

The actual working of the program takes place in the detection phase which is followed by the initialization stage, where the four corners of the projected screen are input by the user. Now the program knows the four corners of the screen. The program now captures the images from the camera at regular interval and processes these images to detect the position of the laser in the image. This function is performed by the laser detection module. After the laser pointer is detected in the image, the coordinate of the point is given to the transformation module which transforms the position of the laser dot on the screen to its corresponding coordinates on the computer screen.

B.1 Image Capturing

Images from the camera are captured. If no camera is connected or, if the program isn't able to detect the camera, an error message is displayed on the screen. If the camera is detected, it is used to capture images of the screen.

B.2 Colour Setting

After the Image has been captured, the image is passed to the laser detecting module. This module detects the presence of the laser dot on the image if any and on successful detection , it returns the position of the laser pointer in the image.

We work on the basis that the laser is supposed to be the brightest red point in the whole image. We are treating each frame as an image and there we remove the Red layer of the RGB mode and are working on the Green and Blue layers. Furthermore we turn that image onto a greyscale image and from that the brightest spot is detected. We use a constant value of B_{thres} G_{thres} any value exceeding the threshold value is considered as the laser point [7] and the co-ordinates of the corresponding pixels are found out, assuming that to be the cursor position.

B.3 Transformation

If the laser dot is detected, the program gives the co-ordinates of this detected laser dot to the transformation module. The function of this module is to calculate the position of the mouse pointer on the screen corresponding to the current position of the laser dot in the image being processed. The position is

calculated in terms of the fractional distance of the position of the pointer from the left edge of the screen (fraction of the width of the screen) which when multiplied with the width of the screen gives the x co-ordinate of the point and the fractional distance of the position of the pointer from the top edge of the screen (fraction of the height of the screen) which when multiplied with the height of the screen gives the y co-ordinate of the point [8].

The following paragraph explains the whole process.

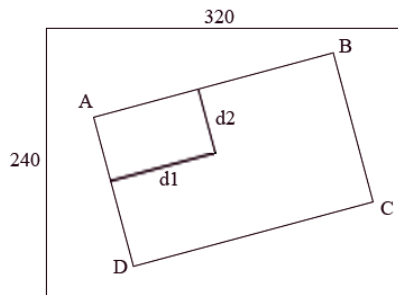


Figure 2 - Sample transformation from a captured image.

Consider the diagram shown above. We consider points A, B, C, D as the four corners of the screen. The user needs to input only three corners into the program during the initialization stage namely the top left, the top right and the bottom left (A, B, D). Our assumption that the screen is viewable as a parallelogram can be used to calculate the 4th point, the bottom right one easily. Now assume X is the position of the laser pointer detected by the laser detecting module. We then calculate the distances of this particular point X from the left edge of the screen and call this distance as d1 and from the top edge of the screen and call this distance as d2. We then use these distances d1 and d2 respectively to find out the actual mouse position on the screen in the following way. We first calculate the length of the left edge of the screen. In this diagram, this can be depicted by the distance AD. We also calculate the length of the top edge of the screen, which can be seen from the diagram as the distance AB. We also have the distances d1 and d2 as mentioned earlier[9]. Using these distances we calculate the fractional position of the point on the screen.

$$\text{Fractional horizontal distance} = AB/d1$$

$$\text{Fractional vertical distance} = AD/d2.$$

IV. SIMULATING THE MOUSE EVENTS

Since in contrast to the mouse the laser pointer has no buttons, their properties have to be simulated from the possibilities of the laser pointer to be switched on and off. The simulation of clicking a mouse button is as follows:

A. Cursor Hovering

The fact that cameras, projectors and rooms are all different in their optics and their positioning poses a problem. What is needed is a function that will map a detected laser spot (x, y) position in the camera image to the corresponding position in the coordinates of mouse move interactive display. The laser spot is lighted on the display surface if user wants a mouse move interaction. We return it as the laser position (x, y).

B. Left Click

Just one time the laser button is downed if user wants a mouse left click interaction. When the laser pointer is turned off, the target where the laser pointer was last detected is returned as the position on the display where the left click has to be simulated.

C. Left Double Click

The laser button is switched on off twice quickly at the same position if the user wants a mouse left double click interaction. We return it as the laser position when the laser pointer light is turned off at second time.

D. Right Click

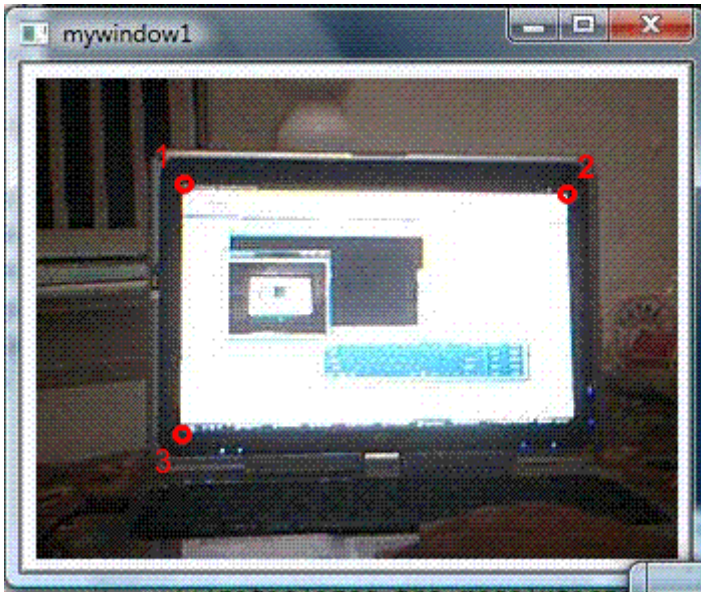
Mouse right button click action will be activated when the laser pointer is being kept fixed on the pointer target on for approximately 2 to 3 seconds. We return it as the laser spot (x, y) position.

E. Drag and Drop

Drag and Drop is performed when the left mouse button is down and the cursor is moved elsewhere and the target is placed where the left mouse button is up. Laser pointer which was on has to be turned off on the target and turned-on on a different position to initiate the drag the drop operation. Later the laser pointer is switched off at the desired position where the drop operation is to be performed.

V. PROGRAM DOCUMENTATION

1. Run the MATLAB code.
2. First, a window which will show the current vision of the camera is opened.
3. Adjust the camera, so that you see the entire projected screen.
4. Once that is done, fix the camera position and press <ESC> on your keyboard.
5. A still image of the projected screen will now appear on your screen. This is the initial calibration screen.
6. Now click on the 3 points on on the image namely, the top left, top right and the bottom left in order as shown by the following figure.



7. After this step, press <ESC> on your keyboard again and now the system will go in the detection state and will start detecting the laser pointer to perform mouse events.

REFERENCES

- [1] R. Sukthankar, R. Stockton, and M. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. In Proc. Of ICCV'01, pp.247–253, IEEE Computer Society Press, July 2001.
- [2] J.-Y. Oh and W. Stuerzlinger. Laser pointers as collaborative pointing devices. In *Graphics Interface 2002*, pp. 141–149, May 2000.
- [3] Kirstein, C. and Muller, H., Interaction with a projection screen using a Camera-tracked laser pointer, *Multimedia Modeling '98 Proceedings*, pp. 191-192.
- [4] Olsen, D. R. Jr. and Neilsen, T., Laser pointer interaction, *CHI'2001*, pp.17-22.
- [5] Laser Pointer Mouse, Xinpeng Huang and William Putnam, 2006
- [6] MacKenzie, I. S., Jusoh, S., An evaluation of two input devices for remote pointing. *EHCI 2001*, Heidelberg, Germany: Springer-Verlag.
- [7] Myers, B. A., Bhatnagar, R., Nichols, J., Peck, C. H., Kong, D., Miller, R., and Long, A. C., Interacting at a distance: measuring the performance of laser pointers and other devices, *Proceedings CHI'02*, to appear.
- [8] Mouse emulator using laser pointer and camera, Nerav A Vasa, University Of Columbia
- [9] MacKenzie, I. S. (1995). Movement time prediction in human-computer interfaces. In R. M. Baecker, W. A.S. Buxton, J. Grudin, & S. Greenberg (Eds.), *Readings in human-computer interaction* (2nd ed.) (pp. 483-493). Los Altos, CA: Kaufmann. [reprint of MacKenzie, 1992]