JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-04, Issue-04 E-ISSN: 2347-2693

Laser Gesture Recognition for Human Machine Interaction

Umang Keniya[1*], Sarthak Kothari[2], Akash Mehta[3], Bhakti Palkar[4]

[1,2,3,4] *Department of Computer Engineering*

K.J. Somaiya College of Engineering, Vidyavihar, Mumbai-77, Maharashtra India

www.ijcseonline.org

Received: Mar/19/2016 Revised: Mar/26/2016 Accepted: Apr/12/2016 Published: Apr/30/2016

Abstract— Proposed is an application that gives visual commands to the computer without touching the keyboard or mouse for performing certain actions. This puts together a simple laser gesture recognition application and used it to control Windows Media Player and Microsoft PowerPoint. This is far more comfortable than using a remote control because you don't have to look for the correct buttons in the dark. All you have to do is to make a few simple gestures anywhere in the camera's field of view with a laser pointer.

Keyword— Laser spot detection, Gesture Recognition.

I. Introduction

An alternative computer control method based upon a standard laser pointer and LCD projector, used to control applications and recognize specific contextual gestures at a significant stand-off distance. The laser pointer is being used as a traditional mouse for controlling the media player remotely. The webcam detects the gestures of laser pointer such as left, right, top and bottom. Its working is simple. The program searches for the brightest pixel in the camera's field of view. After it detects the brightest pixel, it will track the further motion of pointer and perform the action. [1]

II. Computer Applications

The application detects the laser spot pointed at the projector's projection and then to recognize the movement of the laser pointer so as to analyze the gesture performed using the laser pointer through a camera (laptop's embedded webcam) and as per the gesture analyzed, the system will perform the assigned task associated with the gesture.

A. *Power point presentations*

Group meetings and other non-desk situations require that people be able to interact at a distance from a display surface. This project describes a technique using a laser pointer and a camera to accomplish just such interactions. Calibration techniques are given to synchronize the display and camera coordinates.

B. *Windows Media Player*

Listening to music, but changing tracks is a real pain since you have to reach for the keyboard every time you feel likechanging a song. So, we thought that it would be great if we could somehow give visual "gesture" commands to our machines and control Windows Media Player without having to touch the keyboard.

III. Working Principle

The application accesses the laptop's integrated camera or a separate camera to continuously capture images of the projector screen. This is a continuous process i.e. the application keeps on capturing the images after a certain fixed interval of time from the time it is started up to the time it is stopped. These captured images would be processed in the next function. There are two major parts of this system (1) Laser Spot Detection. The laser spot which is projected on the screen needs to be detected. (2) Gesture Recognition The gesture performed by moving the laser needs to identified and appropriate action needs to be performed. [1]

A. *Laser detection*

The laser spot detection is about finding out the HSV (Hue-Saturation Value) levels of the RGB image frame captured by the camera, converting the image into binary format (0 black, 1 white) and identifying the laser spot. (White pixel).The color of laser always shows a characteristic value in Hue and Saturation range which is from 0 to 255. It uses a threshold values to compare each pixel and if the pixels are in range of laser characteristic the pixel value is set to 1 or else 0 for binary conversion. (Later this binary image is processed to get the exact pixel point). For a typical red laser, the threshold value for hue ranges between 0 & 5 and saturation value ranges between 235 & 255. [4]

The R, G, B values are divided by 255 to change the range from 0...255 to 0.1:

$$R' = R/255;\ G' = G/255;\ B' = B/255$$

$$\text{Cmax} = \max(R', G', B');\ \text{Cmin} = \min(R', G', B')$$

$$\Delta = \text{Cmax} - \text{Cmin}$$

Saturation = 0 if Cmax=0 OR 1- [Cmin/Cmax] if Cmax <> 0

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$

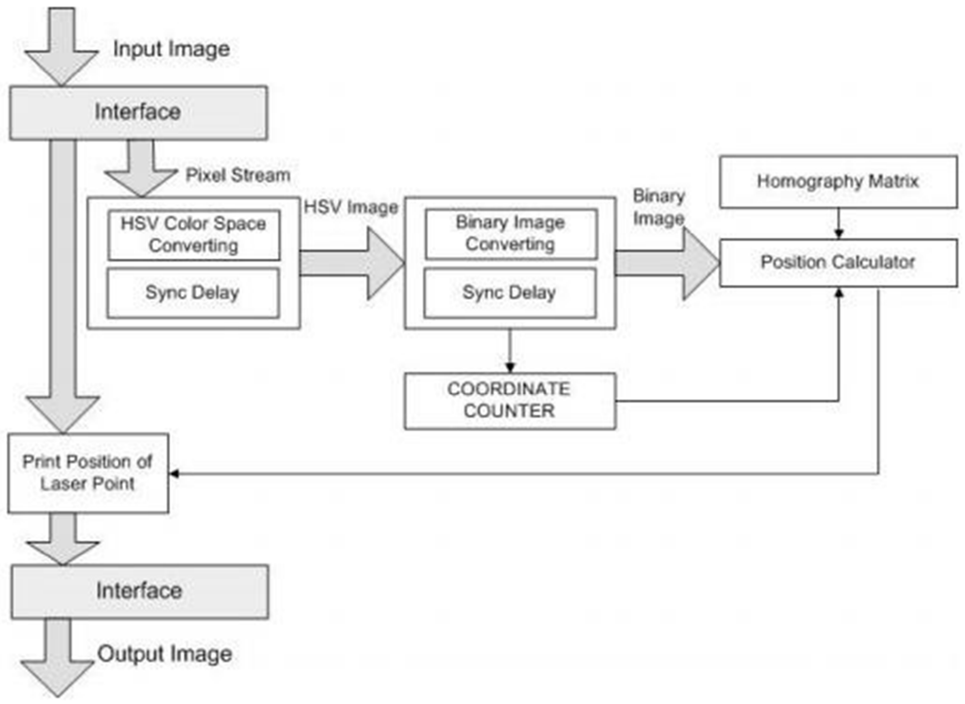

Fig 1: Laser Detection [2]

One of the most common approaches is to compare the current frame with the previous one. It's useful in video compression when you need to estimate changes and to write only the changes, not the whole frame. But it is not the best one for motion detection applications. Assume that we have an original 24 bpp RGB image called current frame (image), a gray scale copy of it (currentFrame) and previous video frame also gray scaled (backgroundFrame). First of all, let's find the regions where these two frames are differing a bit. For the purpose we can use Difference and Threshold filters. [2]

1. Create filter
2. Set background frame as an overlay for difference filter
3. Apply the filters

On this step we'll get an image with white pixels on the place where the current frame is different from the previous frame on the specified threshold value. It's already possible to count the pixels, and if the amount of it will be greater than a predefined alarm level we can signal about a motion event. But most cameras produce a noisy image, so we'll get motion in such places, where there is no motion at all. To remove random noisy pixels, we can use an Erosion filter, for example. So, mostly only the regions where the actual motion was are resulted.

1. Create erosion filter
2. Apply the filter on the image where the current frame is different from previous frame.

The simplest motion detector is ready! We can highlight the motion regions if needed.

1. Extract the red channel from the original image.
2. Merge the red channel with motion region.
3. Replace the red channel in the original image.

The disadvantage of the approach is that if the object is moving smoothly we'll receive small changes from frame to frame. So, it's impossible to get the whole moving object. Things become worse, when the object is moving so slowly, when the algorithms will not give any result at all.

There is another approach. It's possible to compare the current frame not with the previous one but with the first frame in the video sequence. So, if there were no objects in the initial object independently of its motion speed. [4]

But, the approach has a big disadvantage - what will happen, if there was, for example, a car on the first frame, but then it is gone? Yes, we'll always have motion detected on the place, where the car was. Of course, we can renew the initial frame sometimes, but still it will not give us good results in the cases where we cannot guarantee that the first frame will contain only static background. But, there can be an inverse situation. If you'll put a picture on the wall in the room, it'll get motion detected until the initial frame will be renewed.

The most efficient algorithms are based on building the so called background of the scene and comparing each current frame with the background. There are many approaches to build the scene, but most of them are too complex. We'll describe here the approach for building the background. It's rather simple and can be realized very quickly. [5]

As in the previous case, let's assume that we have an original 24 bpp RGB image called current frame (image), a gray scale copy of it (currentFrame) and a background frame also gray scaled (backgroundFrame). At the beginning, we get the first frame of the video sequence as the background frame. And then we'll always compare the current frame with the background one. But it will give the result is described above, which we obviously don't want very much.

The approach is to "move" the background frame to the current frame on the specified amount (used 1 level per frame). We move the background frame slightly in the direction of the current frame - we are changing colors of pixels in the background frame by one level per frame. [4]

1. Create filter.
2. Move background towards current frame.
3. Dispose old background.

And now, we can use the same approach we've used above.

1. Create processing filters sequence.
2. Apply the filter.
3. Extract the red channel from the original image.
4. Merge the red channel with moving object borders.
5. Replace the red channel in the original image.

There is another approach based on the idea. As in the previous cases, we have an original frame and a gray scaled version of it and of the background frame. But let's apply Pixelate filter to the current frame and to the background before further processing. So, we have pixelated versions of the current and background frames. Now, we need to move the background frame towards the current frame as we were doing before. The next change is only the main processing step:

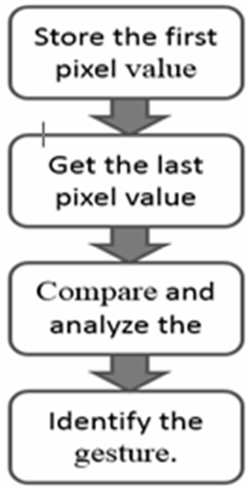

Fig 2: Gesture recognition

A. *Gesture Recognition*

After it finds the pixel, it analyzes how much that point moved along the x and y axes. Based on these parameters, the application tries to recognize the movement. For example, if the laser dot's movement along the x-axis is much more than its movement along the y-axis, the application will determine that it was more or less a horizontal movement. Then, based on the initial and final position of the laser dot, it will determine if the movement was towards the left or towards the right. It uses a similar technique to detect upward, downward, and diagonal movements. [6]

IV. ADVANTAGES

It will be easy for the user to use the gesture. User will just have to use his/her laser pointer to give commands to the computer (User Friendly). The technology also has the potential to change the way users interact with computers by eliminating input devices such as mice and keyboards and allowing the unencumbered body to give signals to the computer through gestures such as pointing laser. Unlike haptic interfaces, here the only thing you need is a laser pointer to give your commands. The gestures of the laser are read by a web camera which is readily available nowadays with the computer instead of some sensors to be attached to the computer. [3]

V. ACKNOWLEDGMENTS

The authors would like to thank Prof. Bhakti Palkar of K. J. Somaiya College of Engineering for providing specifications about the application scenario.

VI. CONCLUSION

The proposed laser gesture recognition system will control the above mentioned applications i.e. Windows Media Player and Microsoft PowerPoint for a few gestures which are left to right, right to left, top to bottom, bottom to top, for performing certain operations. This will be a better alternative to other controlling devices such as a mouse or a keyboard or a remote control which cannot be feasibly used during dark. Also, many more applications with many other gestures can be introduced after the successful implementation of the project.

REFERENCES

[1] Aryuanto Soetedjo, Ali Mahmudi, L. L., "Detecting Laser spot in shooting simulation using an embedded camera", Int. Journal of Smart Sensing and Intelligent Systems, Volume-07, Issue-01, March 2014.
[2] ByungMoo Jeon, "Hardware architecture for detecting laser point using FPGA", Int. Conference on Control, Automation and Systems, Page No (199-203), October 2012.
[3] Xingyan Li, "Gesture Recognition Based on Fuzzy C-Means Clustering Algorithm", April 2002.
[4] Jean-Francois Lapointe, Guy Godin, "On screen Laser spot detection for large display interaction", IEEE International Workshop on Haptic Audio Visual Environments and their applications, Page No (72-76), October 2005.
[5] Darren Phi Bang Dang, "Template Based Gesture Recognition", June 1996.
[6] Khushboo Arora, Shrutika Suri, Divya Arora and Vaishali Pandey, "Gesture Recognition Using Artificial Neural Network", Int. Journal of Computer Sciences and Engineering, Volume-02, Issue-04, Page No (185-189), April 2014.