2012 IEEE/SICE International Symposium on System Integration (SII)
Kyushu University, Fukuoka, Japan
December 16-18, 2012

Laser Spotlight Detection and Interpretation of Its Movement Behavior in Laser Pointer Interface

Romy Budhi Widodo, Weijen Chen, and Takafumi Matsumaru, *Member, IEEE*

***Abstract*—A laser pointer can be used as an input interface in human-machine interaction. Such utilization, however, can be problematic and one of the main issues is the lack of good reliability in the laser spotlight detection. Another problem is how to interpret the user's movement of the spotlight into commands for the application. This paper proposes a method for a laser spotlight detection. The aim is to improve the practicality and reliability of the previous approaches. We use the maximum pixel value as a multiplier in determining the threshold. Maximum pixel value is obtained from environment brightness at a specified time. For the second problem we propose a simple interpretation of incidents that allows the user to use the application, with three main events: laser-move, hover, and single-click. There is no need for users and program to wait a specified time span to be able to interact with each other, and the user can directly give commands to the application after the single-click event. These approaches result in better reliability, easier operation of the application by the user, and allow opportunity for development of a system for rehabilitative, recreation, and input interface devices in the future.**

I. Introduction

THIS paper aims to present the proposed method used to identify the spotlight from a laser pointer, as well as to find methods of spotlight behavior interpretation. The program converts the moving spotlight and related events into commands for the specific application uses spotlight behavior interpretation.

An application was developed to test the proposed method to comprehend the reliability of the system. A laser pointer is used by the user as an input for running the application as well as a PC, LCD projector, and web-camera.

Section 2 discusses background and related work and section 3 discusses the system specifications and previous approaches. Section 4 discusses the operation, which includes the spotlight detection, direction screen area recognition, and interpretation of spotlight behavior. Section 5 discusses the analysis and testing of the proposed system. Section 6 presents some conclusions.

II. Background and Related Work

A. Background

Advances in technology and knowledge can be utilized to create a variety of human-machine interaction devices. The advantages of using a laser pointer instead of a PC mouse as an input interface are laser pointer interface requires no additional apparatus such as a mouse pad and desk; in addition it ensure the movement flexibility for the user. However, the disadvantages of this interface are that it needs hand movement stability, although the user can be trained in a few moments; the other disadvantage is it has higher operational hassle if we have to imitate the PC mouse operation precisely such as: left-click, right-click, double-click, scroll-up and scroll-down. There is a great optimism regarding continuing increases in effectiveness and reliability of using a laser pointer as an input interface along with its use as a human-machine interaction device. This is a motivation and a challenge for our study.

B. Related Work

Laser spotlight detection as well as the interpretation of its movement has been studied by several researchers and in our previous study. But the reliability and effectiveness of the system should be improved. Several studies about spotlight detection and behavior interpretation of its movement will be discussed in more detail below.

1) Laser Spotlight Detection: According to their respective field of application, researchers have performed various methods of handling spotlight detection. For the spotlight detection in the previous study [10] we have tested Mixture of Gaussian method (GM-method), which is especially well-suited for a static instruction display. Another method is the color-threshold method based on detecting an HSV color space of the frame and implements thresholds for hue, saturation, and value channel [10, 15]. In [10], the threshold value adapts according to the lighting environment, but the interference of the LCD projector is still affecting the spotlight detection. Other researchers in [11, 13] use a static threshold value in order to detect the spotlight, which means that the spotlight is identified when the pixel value exceeds the threshold limit. Threshold used in this case is static so it can not adapt to the lighting environment. While [12] uses background removal process,

Manuscript received September 23, 2012. This research project has been supported partially by the following organizations and here we express gratitude; MEXT (Ministry of Education, Culture, Sports, Science and Technology) through its Grants-in-Aid for Scientific Research <KAKENHI> (Number: A11522500) in 2011 and 2012 fiscal year.

Romy B. Widodo, Weijen Chen, and Takafumi Matsumaru are with Graduate School of Information Production and System, Waseda University, Kitakyushu, Japan (phone: +81-93-692-5241; fax: +81-93-692-5021; e-mail: [romy.budhi@fuji, weijen0312@ruri, matsumaru@aoni].waseda.jp).

978-1-4673-1497-8/12/$31.00 ©2012 IEEE

there is a comparison between a reference image and a real captured image. Other studies done in [16] and [17] use FRBS (fuzzy logic rule-based system) and combine with genetic algorithm for tuning optimization, but [12], [16], and [17] require expensive computing.

2) Spotlight Movement Behavior Interpretation: The other issue is interpretation of a laser spotlight behavior. A few researchers have attempted to find ways to provide instruction via the command behavior of the laser spotlight. Ghosh et al. in [1, 2] and Kim et al. in [15] use four types of laser spotlight behavior to substitute a PC mouse click operation, namely: "mouse move," "single-click," "double-click," and "right click." Other researchers, Kirstein and Heinrich [3], use three types of laser spotlight behavior, namely "button press," "button release," and "mouse move." "Button press" operation was recognized when the spotlight was stable in the specific area for a specified time, on the other hand "release button" operation is generated if the spotlight is turned off for a specified time, while the "mouse move" operation is obtained from the motion of the spotlight it-self. All of those studies tried to replace the mouse action for desktop-based applications and the reliability gained about 50% in [3]. Whereas in [2] the user should take the specified time span t for delay process in order to give time for the program to recognize a certain action, which will not be easy for common users to fulfill and may cause errors in behavior interpretation.

Fukuda et al. [8] and Takahashi et al. [9] use a laser pointer only on the ON-OFF state to control the direction of a wheelchair to avoid obstacles [8] and to order the robot arm to retrieve certain items for bedridden users [9], respectively. While [4], [5], and [14] use additional hardware other than a laser pointer. In [4] and [14] there are additional buttons to simulate left-click and right-click action, while in [5] such information is transmitted as a mouse click by changing the modulation of the spotlight, then frequency-demodulation CMOS image sensor is used to translate it.

Zhang and Chen in [6] divide the instruction screen into nine segments (space segmentation), spotlight is maintained ON continuously, and the interpretation of command instruction is obtained from a displacement of the spotlight of a particular segment to another segment. However, it is specifically designed for presentation application that needs "next-page" and "previous-page" command action.

3) Proposed study: The proposed system offers a practical method for detecting the spotlight; the method used is an adaptive threshold based on the environment lighting. The adaptive threshold is implemented on all the RGB elements of colors captured by the camera. On the spotlight behavior interpretation issue, the proposed system does not require a specified time span in recognizing the movement behavior and also no additional hardware required. The proposed method to interpret a laser spotlight behavior is using "laser move," "hover," and "single-click" event. Hover is a state when a spotlight enters a certain item's area like a cursor of a PC mouse when entering an item on Windows application. Based on the testing of employed applications, the proposed method has high reliability and efficiency.

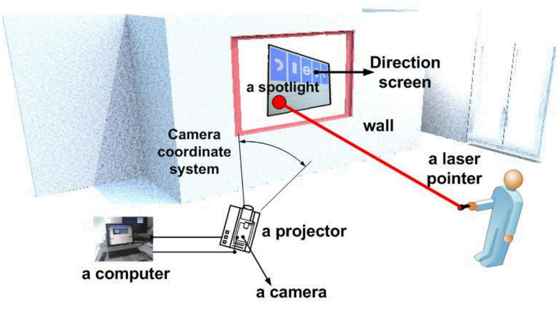


Fig. 1. Illustration of the proposed system

Fig. 2. Equipment in this system

Improvements have been made from the previous system [10], such as the proposed system have a high reliability on spotlight detection, the proposed system has a new object handling function as well as able to interpret the spotlight movement, and the system can work on a single PC.

III. System Configuration and Previous Approaches

A. System Configuration

The projector used to display the screen direction was pointed onto the floor, ceiling or wall. Users only need to point the laser pointer to the desired place within the direction screen. A web-camera is ready to capture the spotlight from a laser pointer and a computer processes it in advance. Figure 1 describes the system illustration. This system consists of a projector, web-camera, laser pointer, and computer; as shown in Fig. 2.

This system consists of four steps, namely:

Step 1: Spotlight detection, extracting the foreground object

Step 2: Coordinate of spotlight calculation

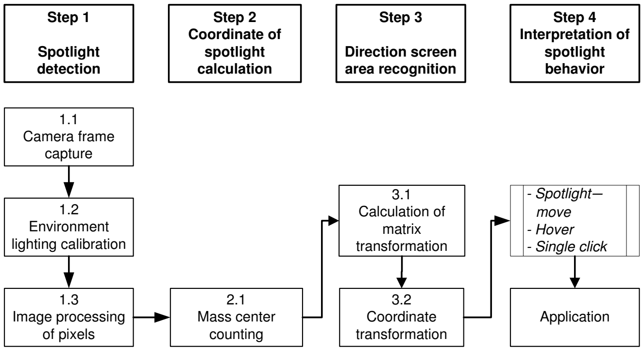
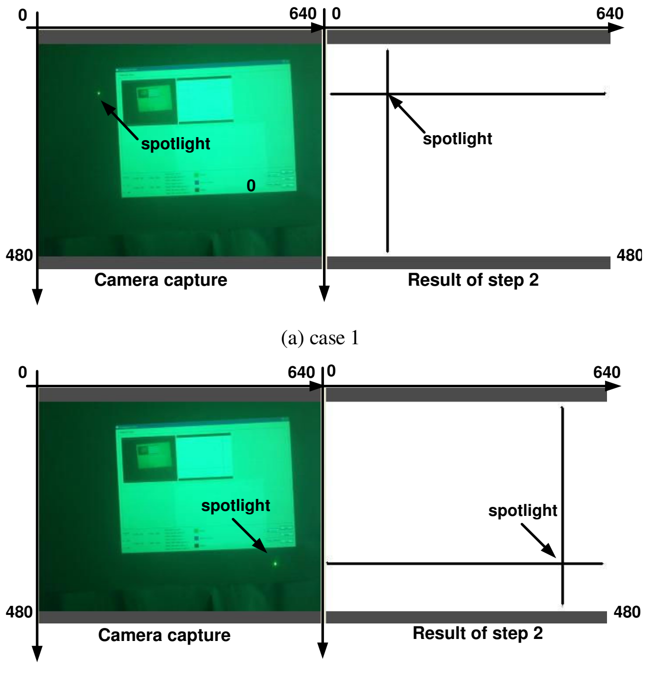

Step 1
Spotlight detection
Step 2
Coordinate of spotlight calculation
Step 3
Direction screen area recognition
Step 4
Interpretation of spotlight behavior
1.1
Camera frame capture
1.2
Environment lighting calibration
1.3
Image processing of pixels
2.1
Mass center counting
3.1
Calculation of matrix transformation
3.2
Coordinate transformation
- Spotlight—move
- Hover
- Single click
Application

Fig. 3. Schema of the design procedure



(a) case 1

(b) case 2

Fig. 4. Two results example of step 2.

Step 3: Direction screen area recognition process
Step 4: Interpretation of spotlight behavior

Step 4 is a new step while steps 1 to 3 have been done in previous studies, but in this proposed system only step 2 uses a method similar to previous studies. We have modified step 1 and step 3 from the previous study so that the system has better reliability. Section 4 will discuss steps 1, 3, and 4 in more detail. Figure 3 shows the schema of all steps.

B. *Previous Approaches*

1) Step 1: A brief overview of the previous studies [10] of step 1 follows. Two methods of spotlight detection steps were studied, namely: Mixture of Gaussian (GM-method) and color-threshold method. In the previous study the GM method is based on such principles as extracting the foreground object and building a model of the background. Our experiment shows the GM method is only appropriate for static backgrounds and items. If the background is dynamic, the GM method takes 3-6 seconds to achieve stability, depending on what learning rate is used. Some advantages of this method are that the user can select any color of laser spotlight, item, and background of application display. On the other hand, the color-threshold method is based on detecting an HSV color space of the frame and implements thresholds for hue, saturation, and value channel. This method is an appropriate choice in a situation where the direction screen always changes, but it requires a spotlight color consistency. Based on our experiment the H, S, and V channel from the spotlight are not detected well because of the interference from the brightness of LCD projector. It needs adaptive threshold for H, S, and V channel to overcome the unstable environment brightness.

2) Step 2: Step 2 relies on the mass center coordinates of the camera point of view as spotlight coordinates (1). The result of this process is the Cartesian coordinate system with ranges from (0,0) to (640,480).

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}} \qquad (1)$$

where,

$$m_{pq} = \sum_{i=1}^{n} I(x, y).x^{p} y^{q} \qquad (2)$$

Note: pq = 00, 10, 01 for m_{00}, m_{10}, and m_{01}, respectively. I = image of spotlight obtained from step 1, while the m_{00} is the total number of pixels of the frame area.

Figure 4 shows some results of step 2. In the two cases the spotlight was detected and the coordinate was calculated in the program.

IV. Operation of the Proposed System

In the proposed system we improved the method of step 1 to get a better reliability, changed the method of step 3 to be a manual method, and in the step 4 we added a new object handling to interpret the spotlight movement. The next sub section will discuss the proposed method in more detail.

A. *Spotlight Detection (step 1)*

The method used in the proposed system is more suitable for interfaces using a laser pointer, although it requires an additional device as a filter that is placed in front of a web-camera, as done in [7]. The color of the filter is the same as the color of the laser pointer spotlight so that the spotlight can be passed by the filter, while other colors should not be passed by the filter. In this study green is the color of both the filter and laser spotlight. Fig. 5 shows the schema of the software design of step 1 in more detail.

The image from a web-camera is captured and converted to gray scale while the program got the maximum pixel value. The value of maximum pixel will be used for the image

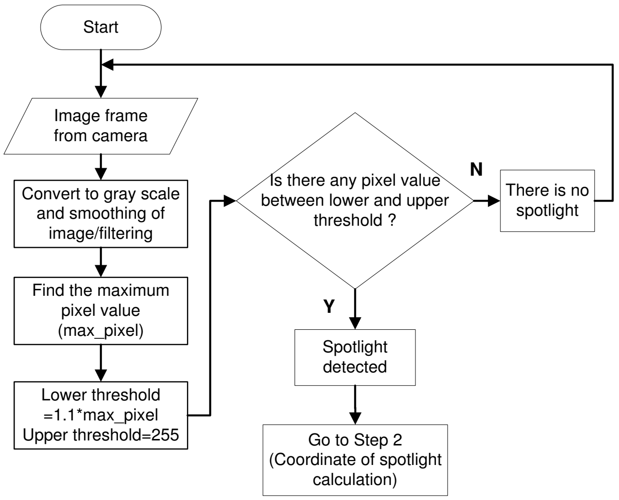

Start
Image frame from camera
Convert to gray scale and smoothing of image/filtering
Find the maximum pixel value (max_pixel)
Lower threshold =1.1*max_pixel Upper threshold=255
Is there any pixel value between lower and upper threshold ?
N
There is no spotlight
Y
Spotlight detected
Go to Step 2 (Coordinate of spotlight calculation)

Fig. 5. Schema of software design of step 1: Spotlight detection

processing step (see step 1.3 in Fig. 3).

The image processing method is as follows: maximum pixel value is used as a lower-threshold for all elements of the RGB. Based on the experiment the best lower-threshold we obtained is 1.1 x maximum pixel value while the upper-threshold value is 255 applied to all elements of the RGB color. If the pixel value of laser spotlight is between lower and upper-threshold then a spotlight is deemed to have been found, the process has completed one cycle and should go to the next step. Thus the threshold used in this study is adaptive according to the ambient environment lighting condition at the time. In the experiment, the user should perform an environment lighting calibration before the application starts.

B. *Coordinate of Spotlight Calculation (step 2)*

Step 2 uses a mass center calculation, which is the same method as the previous study. This step already discussed on the section III B.

C. *Direction Screen Area Recognition (step 3)*

Figure 6 illustrates the camera view that contains a direction or application screen. The purpose of this step is to perform perspective transformation of coordinates on the application screen to the camera coordinate system (640 x 480). In the calibration process, four pairs of corner coordinates of direction screen/application screen, namely: top-left, top-right, bottom-right, and bottom-left; are taken by the user. Users direct the laser spotlight on every corner alternately and the program will capture the spotlight coordinate as illustrated in Fig. 7.

These four pairs of coordinates of the corner are treated as the source coordinate, each pair of coordinates will be the x and y value in (5), while the destination coordinate (x' and y') is (0,0), (640,0), (640,480), (0,480) for top-left, top-right, bottom-right, and bottom-left, respectively. Refer to (5), if x, y, x’, and y’ are known then perspective transformation matrix could be calculated. Finally, this matrix can be used to calculate the newest spotlight coordinate. The following transformation equations are:

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \qquad (3)$$

$$M = \begin{bmatrix} A & B \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & b_0 \\ a_{10} & a_{11} & a_{12} & b_1 \\ a_{20} & a_{21} & a_{22} & b_2 \end{bmatrix} \qquad (4)$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = M \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & b_0 \\ a_{10} & a_{11} & a_{12} & b_1 \\ a_{20} & a_{21} & a_{22} & b_2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (5)$$

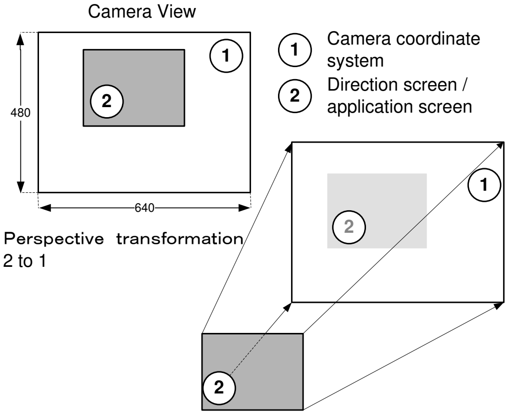
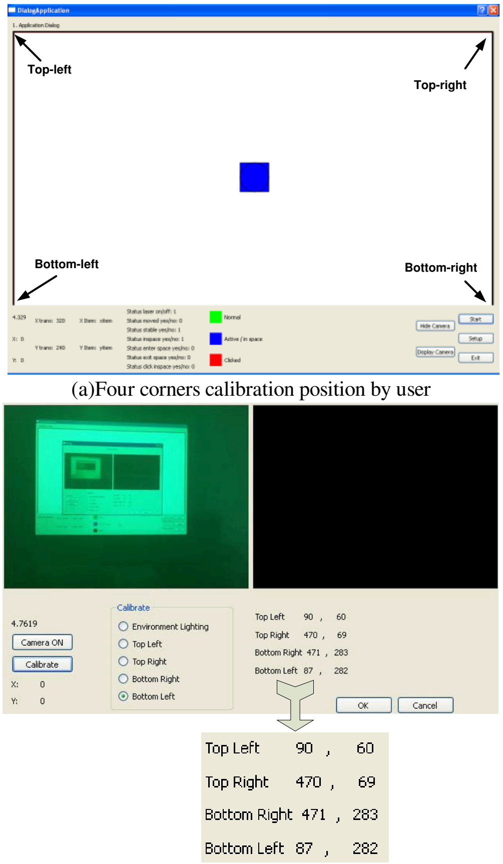

Camera View
480
640
1
2
1 Camera coordinate system
2 Direction screen / application screen
Perspective transformation 2 to 1
1
2
2

Fig. 6. Illustration of perspective transformation



4.329 X trans: 320 X Item: xItem
X: 0 Y trans: 240 Y Item: yItem
Y: 0

Status laser on/off: 1
Status moved yes/no: 0
Status stable yes/no: 1
Status inspace yes/no: 1
Status enter space yes/no: 0
Status exit space yes/no: 0
Status click inspace yes/no: 0

🟩 Normal
🟦 Active / in space
🟥 Clicked

(a)Four corners calibration position by user

4.7619

X: 0
Y: 0

Calibrate
○ Environment Lighting
○ Top Left
○ Top Right
○ Bottom Right
● Bottom Left

Top Left	90 ,	60
Top Right	470 ,	69
Bottom Right	471 ,	283
Bottom Left	87 ,	282

Top Left	90 ,	60
Top Right	470 ,	69
Bottom Right	471 ,	283
Bottom Left	87 ,	282

(b)Four pairs of corner coordinate

Fig. 7. Spotlight coordinate on each corner.

D. *Interpretation of the Spotlight Behavior (step 4)*

Interpretation of the spotlight behavior consists of three main events, namely: “spotlight move,” “hover”, and “single-click.” Fig. 8 illustrates a flowchart of this step.

The program monitors the coordinates of the spotlight continuously. There are some events namely: *laser ON, laser OFF, spotlight move, hover events,* and *single-click event*. *Laser ON* and *laser OFF* events are obtained from the presence or absence of spotlight in the area of application. *Spotlight move* event will be called if there is a difference

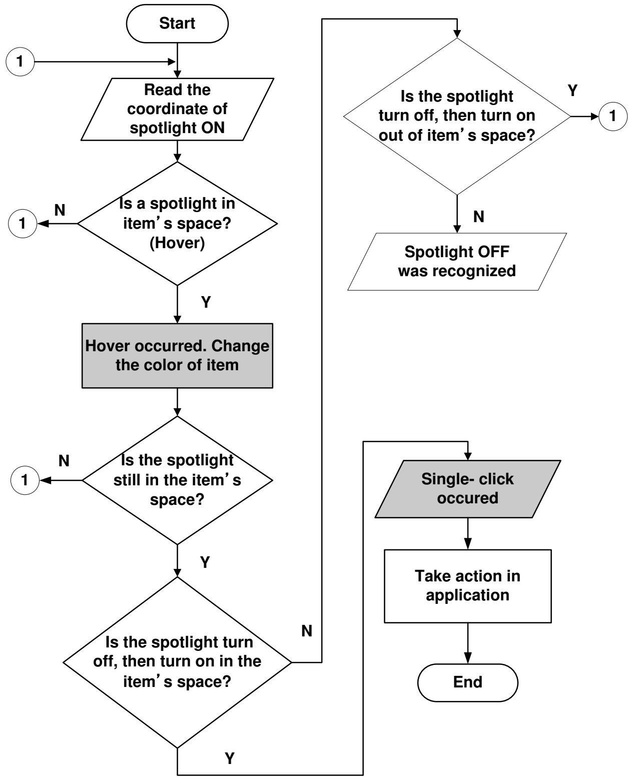

Start
1
Read the coordinate of spotlight ON
Is a spotlight in item' s space? (Hover)
N
1
Y
Hover occurred. Change the color of item
Is the spotlight still in the item' s space?
N
1
Y
Is the spotlight turn off, then turn on in the item' s space?
N
Y
Is the spotlight turn off, then turn on out of item' s space?
Y
1
N
Spotlight OFF was recognized
Single- click occured
Take action in application
End

Fig. 8. Flowchart of interpretation of spotlight behavior

between the pixel coordinates of the present spotlight and the coordinates of the spotlight 500 ms before that time. *Hover event* is obtained if the spotlight enters a specific widget item on the application and then the program changes the color of the item so the user knows that the item is ready to click. *Single-click event* occurs when the spotlight is in hover state and the user perform a spotlight "OFF" followed by "ON" on the same item. If the user wants to deselect the item the user simply turns on the spotlight in empty areas.

V. Testing and Analysis

To test the performance of the system, the experiment will be conducted on an application. The system was tested under various circumstances of surface illuminance that were affected by environment lighting. We used four values of surface illuminance as a variable of experiment: super bright (1600 lux), bright (950 lux), moderate (550 lux), and dark (80 lux). Some testing goals are:

1) Testing the system's reliability in recognizing a laser spotlight.
2) Testing the reliability of a program in identifying the laser spotlight event such as *hover* and *single-click.*

Table 1 describes the average result of ten times of data retrieval while Fig. 9 illustrates states of the spotlight interpretation behavior testing. The following is a brief explanation of Fig. 9:

TABLE 1
Reliability of the Proposed System

Surface illuminance	Laser spotlight detection (%)	Event	
		Hover (%)	Single-click (%)
Super bright (1600 lux)	100	100	100
Bright (950 lux)	100	100	100
Moderate (550 lux)	100	100	100
Dark (80 lux)	100	100	100

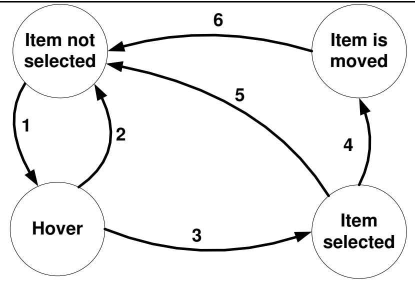


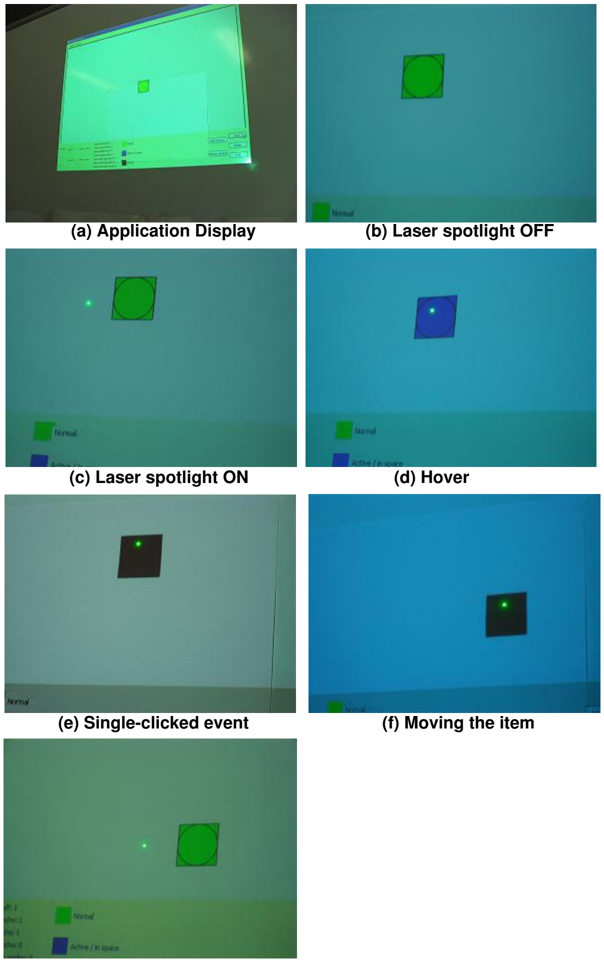
Fig. 9. States in the spotlight behavior interpretation testing

1) The default display is a green rectangle placed in the middle of the application screen (see top left of Fig. 10). If the user directs the spotlight on an item, hover event will occur and the color of item is changed to blue. We chose green as the color of item which is the same color as spotlight to test whether the spotlight can be recognized or not.
2) If the user directs the spotlight out of an item then the item will go back to green color as a sign that the item is not selected anymore.
3) If the user is still in the "hover" state and wants to move the item by moving the spotlight, the user should perform single-click event. "Single-click" is done by turning OFF followed by turning ON the spotlight, but the direction is still on the item. If the action succeeds the item color changes to red. Red color is a sign that the item was selected.
4) While the spotlight remains ON, item can be moved to follow the movement of the spotlight.
5) When an item is selected, if the user turns off the spotlight and then turns it back on but on the outside of the item's area then the state changes to "item not selected."
6) Similarly, when the state is in "item is moved" but suddenly the user turns off the spotlight and then turns the spotlight back on the area outside of the item's area then the state changes to "item not selected."

After designing the conditions/states as in Fig. 9, we tested the application. All worked well as desired as described in Table 1. Figure 10 illustrates some figures of the interpretation of spotlight behavior testing.

VI. Conclusion

This paper has discussed a method for laser spotlight detection and interpretation of its movement behavior to translate into commands. The proposed method is practical and highly reliable. We obtained the maximum value of the current pixel according to the brightness of the environment when the program started, and used it as one of the



(a) Application Display

(b) Laser spotlight OFF

(c) Laser spotlight ON

(d) Hover

(e) Single-clicked event

(f) Moving the item

(g) Deselect item

Fig. 10. Experiment of the interpretation of spotlight behavior

multipliers in the threshold calculation. This situation is helped by the installation of a filter in front of the camera. The color of this filter should be the same as the color of the laser pointer; therefore we can not use multiple colors or any color of laser pointer in this system. Of course, this becomes an opportunity for future research. Using this method the threshold is adaptive and we obtain consistently positive results of the testing system (see Table 1).

On the other hand, the interpretation of spotlight behavior has been proposed using three main events such as "spotlight move," "hover," and "single-click." The test results show that behavior can be recognized on a variety of surfaces' illuminance (see Table 1). The state changes can occur as desired in the design (see Fig. 9), and the user is free to move the spotlight similar to using a PC mouse without waiting for a certain time span. However, in this experiment we only imitate a left-click action of PC mouse. The application made in this study was limited to testing events in spotlight behavior interpretation. The advantages of using a laser pointer instead of a PC mouse as an input interface are it requires no additional apparatus such as a mouse pad and desk in addition it ensure the movement flexibility for the user. However, the disadvantages of this interface are it needs hand movement stability and it has higher complicated operational if the operation should imitate the PC mouse operation such as: left-click, right-click, double-click, scroll-up and scroll-down. Evaluation of laser pointer as an input interface will be the direction of our future work.

References

[1] C. Garai, A. Ghosh, P. Hajra, and P. Paul, "Inclusion of scroll bars and shortcut icons: Improvement of the Laser Track Pad (LTP) for the physically challenged people," *International Journal of Research and Reviews in Computer Science*, vol. 3, no. 3, pp.1606-1610, 2012.

[2] A. Ghosh, C. Garai, P. Hajra, and P. Paul, "Designing a human computer interface using Laser Track Pad (LTP) for the physically challenged people," *IEEE International Conference on Communication and Industrial Application, 2011.*

[3] C. Kirstein and H. Mueller, "Interaction with a projection screen using a camera-tracked laser pointer," in *Proc. of the Conference on Multi Media Modeling,* IEEE ISBN: 0-8186-8911-0, 1998.

[4] K. Choi and G. Lee, "LaserPen: A new pointing device for a beam projector," in *International Symposium on Virtual Environment, Human-Computer Interfaces, and Measurement Systems*, Lugano, Switzerland, July 27-29, 2003, pp. 132-137.

[5] T. Wada, M. Takahashi, K. Kagawa, and J. Ohta, "Laser pointer as a mouse," in *Proc. SICE Annual Conference*, Sept 17-20, 2007, Kagawa University, Japan, pp. 369-372.

[6] L. Zhiang, Y. Shi, and B. Chen, "NALP: Navigating Assistant for Large Display Presentation using laser pointer," *IEEE 1st International Conference on Advances in Computer-Human Interaction*, 2008.

[7] S. Zhenying and W. Yigang, "Research on human-computer interaction with laser-pen in projection display," in *Proc. of 11th IEEE International Conference on Communication Technology*, 2008, pp. 620-622.

[8] Y. Fukuda, Y. Kurihara, K. Kobayashi, and K. Watanabe, "Development of electric wheelchair interface based on laser pointer," in *ICROS-SICE International Joint Conference,* Fukuoka, Japan, August 18-21, 2009, pp. 1148-1151.

[9] Y. Takahashi and M. Yashige, "Robotic manipulator operated by human interface with positioning control using laser pointer," in *Proc. 26th IEEE Annual Conf. of Industrial Electronics Society, IECON*, 2000, pp. 608-613.

[10] R.B. Widodo, W. Chen, and T. Matsumaru, "Interaction using the projector screen and spot-light from a laser pointer: Handling some fundamentals requirements," in *Proc. SICE Annual Conference*, Aug 20-23, 2012, Akita University, Japan, pp.1392-1397.

[11] J. Davis and X. Chen, "Lumipoint: Multi-user laser-based interaction on large tiled displays," *Displays*, 23(5), 205–211, 2002.

[12] B.A. Ahlborn, D. Thompson, O. Kreylos, B. Hamann B., and O. Staadt, "A practical system for laser pointer interaction on large displays," in *Proc. of the ACM symposium on Virtual Reality Software and Technology*, 2005, pp. 106–109.

[13] M.E. Latoschik and E. Bomberg, "Augmenting a laser pointer with a diffraction grating for monoscopic 6 dof detection," *Journal of Virtual Reality and Broadcasting*, vol. 4, no. 14, 2006.

[14] L. Zhang, Y. Shi, and J. Chen, "Drag and drop by laser pointer: Seamless interaction with multiple large displays," *UIC*, *LNCS 4159*, Springer, 2006, pp.12–20.

[15] N.W. Kim, S.J. Lee, B.G. Lee, and J.J. Lee, "Vision based laser pointer interaction for flexible screens," in *Proc. of the 12th International Conference on Human-Computer Interaction: Interaction Platforms and Techniques*, 2007, pp. 845–853.

[16] F. Chávez, F. Fernández, R. Alcalá, and J. Alcalá-Fdez, "Hybrid laser pointer detection algorithm based on template matching and fuzzy rule-based systems for domotic control in real home environments," *Applied Intelligence*, In Press, 2011. DOI: 10.1007/ s10489-010-0268-6.

[17] F. Chávez, F. Fernández, M.J. Gacto, and R. Alcalá, "Automatic laser pointer detection algorithm for environment control device systems based on template matching and genetic tuning of fuzzy rule-based systems," *International Journal of Computational Intelligence Systems*, vol. 5, no. 2, pp. 368-386, 2012.

[18] G. Bradsky and A. Kaehler, *Learning OpenCV: Computer Vision.* O'Really, 2008.