Implementation of Image Segmentation Using RGB Color Detection on Virtual Reality Robot as Spy Tool

Rossi Passarella
Lab. Industrial Automation
Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya
Inderalaya, Palembang, Indonesia
passarella.rossi@gmail.com

Sutarno and Ahmad Rezqy FF
Department of Computer Engineering
Faculty of Computer Science, Universitas Sriwijaya
Inderalaya, Palembang, Indonesia
tarnoabufaruq@yahoo.com
rexisco@yahoo.com

***Abstract*— Humans cannot spy in small places that difficult to pass. We need small size robot which remotely controlled so it can be used for spying. The virtual reality technology in this research was created using image segmentation with RGB color detection and Pythagoras algorithm. A wireless camera is used for replacement of human eye to be able to look at the circumstances around a robot. The camera can turn to the left or right in accordance with the user's head movement. The webcam is used as a sensor to detect the movement of a human head. Systems were created to be able to detect color and get the angle value and work well to move the camera in accordance with the user's head movement. In this way, the user can act as though they are on the robot and perform the surveillance**

Keywords*— Image processing; Pythagoras algorithm; Spying tool; Virtual reality robot

I. Introduction

Robot can assist human, in many security sectors, one of them is surveillance. Robots can replace humans to carry out a dangerous mission. When humans are doing surveillance, they must know the circumstances. Humans can see with their eyes and turn his head to left or right. The robots that replacing humans should also be able to perform it. Therefore, virtual reality technology is chosen. In the robotics field, virtual reality can be a solution to the problem of movement of feedback [1].

In order to look at the circumstances around the robot, a wireless camera is used replace of the human eye for carrying out surveillance [2, 3]. The camera is mounted on the robot and captured image is displayed in a monitor in within real-time. Virtual reality system is used to capture the movement of the human [4]. In this study the movement of human head was captured. Camera on the robot can turned to the left or to the right in accordance with the movement of human head.

The webcam can be used as a sensor [5]. Using cheap low cost webcam, the image result can be as good as the input for image processing [6]. A webcam camera is used as a sensor to detect the movement of a human head.

There are many technologies which are important to create virtual reality system, among them are the tracking system and interaction techniques [7]. In this research we used a helmet as a tracking system. The helmet has two colored signs which are red and blue dots which can be are read by the camera sensor. Then it will be processed using the image segmentation with RGB color detection techniques and Pythagoras algorithm as an interaction techniques to make the user's head movement and the movement of the camera on the robot aligned.

II. Methods

The virtual reality robotic system has 3 sub-system consisting of the movement of the camera on the robot, capturing images around the robot, and controlling the movement of the robot. In this paper, we will only discuss the movement system of the camera on the robot. Figure 1 shows the sub-system of the virtual reality robot.

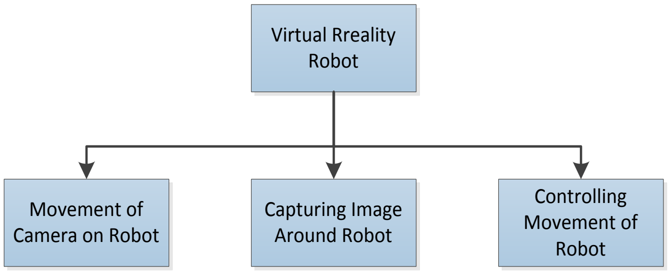

Virtual Rreality Robot
Movement of Camera on Robot
Capturing Image Around Robot
Controlling Movement of Robot

Fig 1. Sub-system of Virtual Reality Robot

This system consists of the input digital image, taken from a webcam, which then processed on a computer. The process consists of image segmentation process to obtain red and blue dots, then continued by a process that uses a Pythagoras algorithm to obtain angle values to be sent from the computer to the microprocessor in robot using Bluetooth.

A. Image Segmentation

The first step to extract the information contained in an image is the image segmentation. Image segmentation technique is basically to divide an image into regions that did not overlap. Image segmentation with RGB color detection techniques are used to get the two dots from two colors on the helmet. The colors used are red and blue. Two dots of these colors will be a parameter for measuring angles.

The Flow diagram of the stages of image segmentation techniques to detect red dot and blue dot is shown in figure 2.

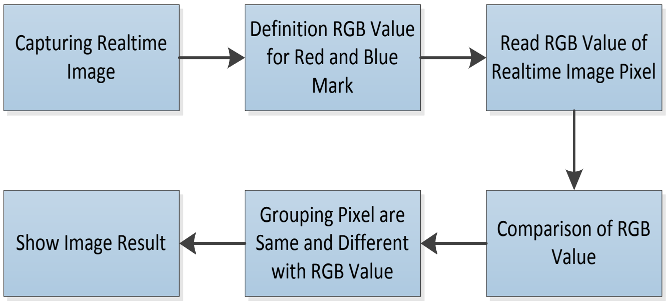

Capturing Realtime Image
Definition RGB Value for Red and Blue Mark
Read RGB Value of Realtime Image Pixel
Comparison of RGB Value
Grouping Pixel are Same and Different with RGB Value
Show Image Result

Fig 2. Flow line diagram of image segmentation with color detection

Capturing Digital Image. The first step is to obtain a digital image taken from a webcam in real-time. At this stage, the program will be made in compiler of Open Source Computer Vision (Open CV) [8]. Figure 3 is a simplified block diagram of a real-time image capture system.

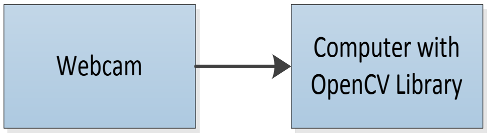


Fig 3. Block diagram to get digital image

Real-time image captured by default will have a resolution of 640x480 pixels. This resolution will be scaled down to 240 x 180 pixels in order to speed up the process. Here's an example of a real-time image taken from the top of the user is shown in figure 4.

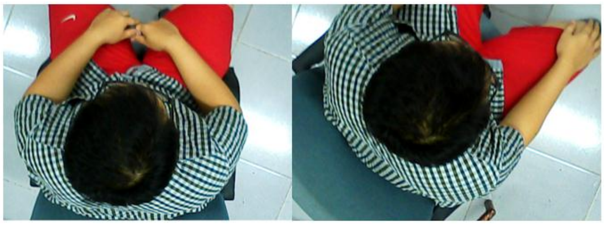


Fig 4. Real-time Digital Image

Definition RGB Value. The next stage is the determination of the RGB values for the red mark and blue mark. The initial step of this stage is capturing a sample image of a mark on the helmet. An image with a resolution 240x180 pixels will be cut at contained color marks. Figure 5 is shown an image after cropping.

An image will be reading each pixel to obtain a value [9]. In this study, the value that want to read and found is RGB values. The next step is iteration from pixel (x0, y0) to the last pixel (xn, yn) of an image that has been cut to get the RGB value of each pixel. RGB values of the pixels are then summed to obtain the average RGB value. This value will be the value of the RGB color mark.

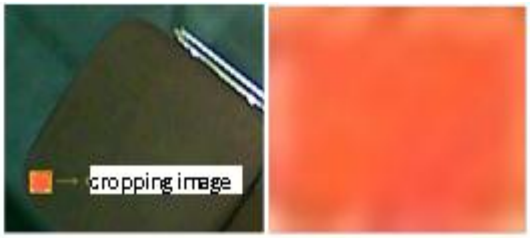


Fig 5. Cropping Image of Color Mark

Reading, Comparison and Grouping of RGB Value. The next stage is reading RGB values of the real-time image captured by the webcam. RGB color space is sensitive to light changes [10]. In this study light changes must be avoided. Then performed the iteration from the beginning pixel of an image to the end pixel of the image. In each pixel iteration, comparison will be made between the RGB pixel values of the real-time image with RGB value of mark that have been obtained previously. A new image of the same size, which is 240 x 180 was prepared as a result of this comparison. If the RGB pixel value of real-time image included in the criteria of RGB value of the mark, then it will be an RGB value in the new image pixel. Conversely, if the RGB pixel value of real-time image is not included in the criteria of the RGB value of the mark, then pixel in the new image will be given a value of white (R = 255, G = 255, B = 255). Image results of image segmentation is shown in figure 6.

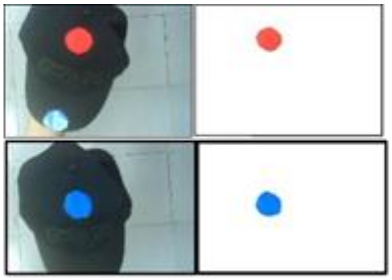


Fig 6. Image Result of Image Segmentation

B. *Pythagoras Algorithmts*

At this stage the red dot and blue dot that have been obtained from the previous process will be a parameter to get the value of the angle of the user's head movement. From the two dots will be made 3 pieces of line that will form a triangle. Figure 7 will be explained the steps to get the value of the angle.

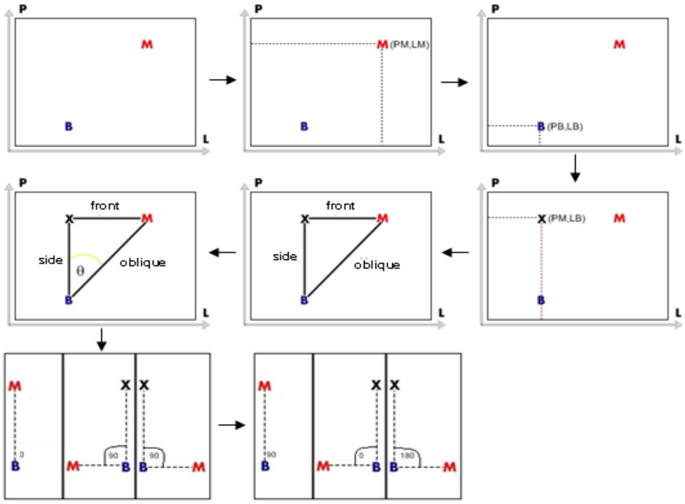


Fig 7. Steps to Get the Value of Angle

Description of Figure 7:

1. Detecting red dot and blue dot by using image segmentation.
2. Knowing the location coordinates of the red dot with coordinates locate the length and width of the red dot (PM, LM).
3. Knowing the location of the coordinates of blue dot with coordinates locate the length and width of the blue dot (PB, LB).
4. Define Point X from the coordinates of the length of red dot and width of the blue dot (PM, LB) to form a triangle.
5. Find the value of each side of the triangle.

$$\text{Side of triangle} = PB - PM \quad (1)$$

$$\text{Front of triangle} = LB - LM \quad (2)$$

$$\text{Oblique of triangle} = \sqrt{\text{side}^2 + \text{front}^2}\ [11] \quad (3)$$

6. After getting the value of each side of the triangle, the next step is to find the value of the angle by using the formula of Pythagoras

$$\text{Cos } \theta = \text{Side of Triangle / Oblique of Triangle} \quad (4)$$

$$\text{Cos } \theta = \cos^{-1}(\cos\theta)\ [11] \quad (5)$$

7. 0 degrees obtained when the red dot and blue dot is vertically aligned. 90 degrees obtained when the red dot is horizontally aligned on the right or on the left of blue dot. With this condition, the wide angle is obtained only from 0 to 90 degrees only, while the wide angle that wants to come is from 0 to 180 degrees. To get it, use the value of the front side (front = width of blue - width of red) which has been obtained from the previous stage. The value of the front side can be positive and negative values. If the value obtained is negative, final angle to be gained is between 0-89 degrees. The following formula for final angle:

$$\text{Final Angle} = 90 - \text{angel} \quad (6)$$

If the value obtained is positive, final angle to be gained is between 91-180 degrees. The following formula for final angle

$$\text{Final Angle} = 90 + \text{angel} \quad (7)$$

C. *Control of Servo Motor*

Virtual reality system can be used to control robot [12, 13]. In this study, virtual reality is used to control the movement of the camera mounted servo motor on the robot. Once the angle is obtained from the program in a computer, the next step is the angle obtained is sent via Bluetooth from the computer to the microcontroller on the robot. In the microcontroller, the angle received from computer is value of the angle in degrees (0-180), the servo motor does not recognize the value of the angle in degrees, but recognize the value of angle in width of pulse that given to the servo motor [14].

In ATmega32 microcontroller there are PWM (Pulse Modulation width) that can be utilized [15]. PWM technology can produce accurate servo motion to adjust the resolution that be desired. Pulses can be generated from the OCR pin on the microcontroller. Need to set a register timer in microcontroller that can be generated a width pulse that we want. Two parameters are needed to find the constants timer, microcontroller clock value and the timer clock value (determined from the prescaler). From two parameters, we can design a width of high pulse and low pulse with a specific frequency to control the servo motor movement.

An external clock that used is 11.059200 MHz and timer1 is used to generate the PWM. The PWM mode used is Phase Correct PWM top = ICR1 with prescaler 8 so that the value of the timer clock is 11.0592 / 8 = 1382.4 KHz. Thereby the increase in the timer counter can be calculated from the period: 1/1382.4 KHz = 7.2338x10^{-7} s or approximately 72.34 microseconds.

A PWM period that we want to produce is 20 ms which is the period of the servo motor in one iteration, because we use the Phase Correct PWM mode is the two time periods of the timer counter increment, so that the counter value of the timer is 10 ms. Then the timer period is 10 ms / 7.2338x10-7 = 13823.99 or the integer 13824.

To adjust the angle of rotation servo, we need to calculate the range of values that need to be given to OCR1A in order to move servo motor from 0^o to 180^o. For 0^o, given pulse ON is 1 ms. because using Phase Correct PWM mode, then it becomes half of the pulse ON. So 0.5 ms / 7.2338x10-7 s = 691.19 or 691 in integer. As for 180^o, given pulse ON is 2 ms. So 1 ms / 7.2338x10-7 s = 1382.39 or 1382. Next step is the calibration of the angle 0^o to 180^o with the increase in angle of 1 degree and generate the following formula:

$$x = (y+180)/0.2605 \qquad (8)$$

Description:

x = PWM value for OCR1A pin in microcontroller.

y = Angle value received from the computer (0 – 180)

The next is testing the percentage error of the angle obtained from the program with the movement of the servo angle is directly calculated manually using a bow. Tests done looping of angle 0 degrees to 180 degrees with a range of 5 degrees and the servo was given a pen as a marker that can be counted manually using a bow degrees. In Table 4 it can be seen from the results of tests measuring the angle of the servo motor as follows.

$$\text{Error percentage} = \frac{\text{Total of Error}}{\text{Total of Testing}} \qquad (9)$$

III. Result and Discussion

A. Design Physical Virtual Reality Robot Model

A hardware design is the first thing done. The hardware will be designed in Virtual Reality Robot includes the shape design and system design. Figure 8 is shows the shape design of virtual reality robot system

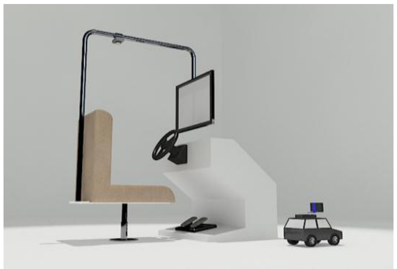


Fig 8. Design Physical of Virtual Reality Robot System

Design of virtual reality robotic system is divided into two, design of robot and design of navigation robot tool. The design of the robot will be assembling a mobile robot with components such as microcontrollers, driver motor, motor dc, servo motor, gearbox, wheel robot, Bluetooth, and wireless cameras. Then on designing robotic navigation tool will be assembling a navigation tool and monitoring robot consisting of a gas pedal, brake pedal, steering wheel, monitor to display image of the camera on robot, user chair seat, and webcam are used as sensor.

B. Definition RGB Value for Red Mark and Blue Mark

By using the image of the red mark with a size of 10x8 pixels will take the RGB value of each pixel and then obtained the average RGB values for red marks.

R average = (amount of R)/(amount of pixel)
R average = 20219/80
R average = 253

G average = (amount of G)/(amount of pixel)
G average = 10327/80
G average = 129

B average = (amount of B)/(amount of pixel)
B average = 10348/80
B average = 129

The same thing performed to get the average RGB values for the blue mark. By using the image of the red mark with a size of 10x8 pixels will take the RGB value of each pixel and then obtained the average RGB values for red marks.

R average = (amount of R)/(amount of pixel)
R average = 3231/80
R average = 40

G average = (amount of G)/(amount of pixel)
G average = 14855/80
G average = 185

B average = (amount of B)/(amount of pixel)
B average = 20348/80
B average = 254

After doing the calculations, the value obtained for the red color mark is RGB (253,129,129), and for the blue color mark is RGB (40,185,254). The next step is reading a pixel of real-time image and comparing the RGB value of the pixel real-time image with the RGB value of the color mark. The condition of comparison is R>253, G<=129, B<=129 for the red color and R<=40, G<=185, B>254 for the blue color. Figure 9 shows the results of the detection of color.

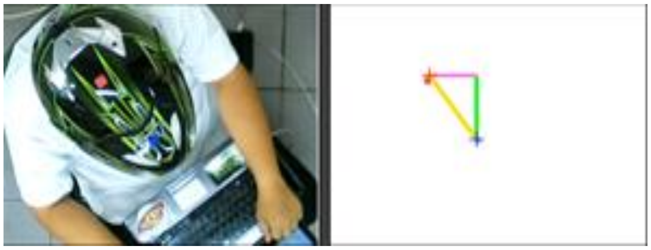
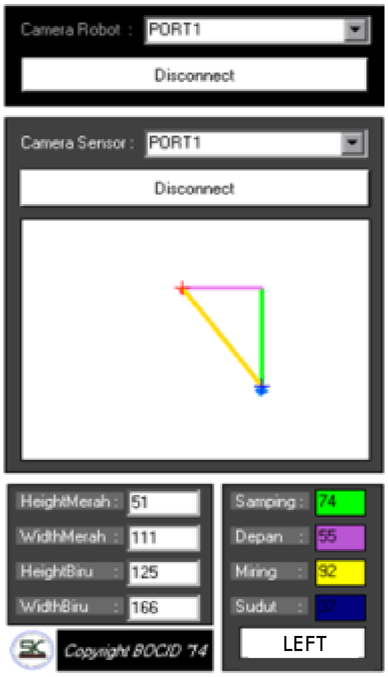


Fig 9. The triangle formed by the two dots

C. *Get the Value of Angle*

Figure 10 shows, a program created to get the value of angle from the user's head movement. In the program there is one frame to display the results of image segmentation using RGB color detection. On the program are shown the values of the parameters to get the angle, the value of the angle is obtained, and the direction of the angle.



Camera Robot : PORT1

Camera Sensor : PORT1

HeightMerah : 51

WidthMerah : 111

HeightBiru : 125

WidthBiru : 166

Samping : 74

Depan : 55

Miring : 92

Sudut : 37

LEFT

Fig 10. Program to Get Value of Angle

D. *Testing of the Value of Angle*

In the microcontroller, the final angle is 127 degrees received, because the value of front side obtained is a positive 55, then the final angle is:

Final angle = 37 + 90

Final angle = 127 degrees

Furthermore, the angle is processed by using a formula. Calculations can be seen below as follows.

PWM value = (Angle+180)/0,2605

PWM value = (127+180)/0,2605

PWM value = 307/0,2605

PWM value = 1178

The maximum PWM value is 13824. PWM values obtained for the angle of 127 degrees is 1178. Servo motor pulse signal with the input angle of 127 degrees is shown in Figure 11.

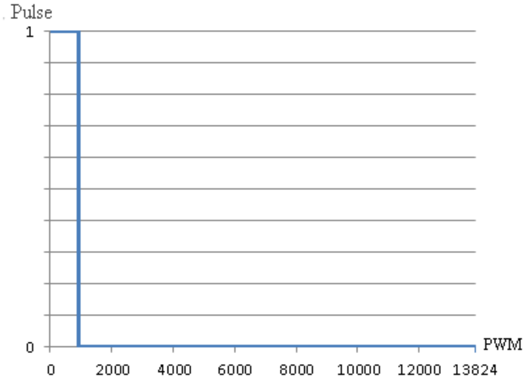


Fig 11. Servo motor pulse signal for angle of 127 degrees

Testing of the angle performed by measuring angle of servo manually using a bow. The test performed looping of angle 0 degrees to 180 degrees with a range of 5 degrees. Then find the average of angle difference.

$$\text{Error percentage} = \frac{\text{Total of Error}}{\text{Total of Testing}}$$

$$\text{Error percentage} = (41.82\ \%) / 37$$

$$\text{Error percentage} = 1.13\%$$

From the testing the angle of 0 degrees to 180 degrees with a range of 5 degrees, the result obtained is good accuracy with 1:13% error percentage.

IV. Conclusion

In conclusions, the image segmentation using RGB color detection techniques was able to detect red and blue dots colors within the range of RGB values. The system used Pythagoras algorithm to get the angle value which capable to work well with the camera harmonious movement as well as with the user's head movements. The result obtained was good accuracy within acceptable error percentage. A virtual reality system which was embedded in the robot, can align the movement of the camera on the robot and a head movement of the user.

ACKNOWLEDGMENT

We acknowledge for the support from Department of computer engineering and Faculty of Computer Sciences – Universitas Sriwijaya

REFERENCES

[1] Burdea, G. C. (1999). Invited review: the synergy between virtual reality and robotics. Robotics and Automation, IEEE Transactions on, 15 (3), 400-410.

[2] Patel, D. S, Dheeraj Mishra, Devendra Pandey, Ankit Sumele, & Ishwar Rathod. (2013, January). Mobile Operated Spy Robot (Spy Robot). IJETAE, 3 (2), 23-26.

[3] Qureshi, F., & Terzopoulos, D. (2008). Smart camera networks in virtual reality. Proceedings of the IEEE, 96 (10), 1640-1656.

[4] Patel, K., Bailenson, J. N., Hack-Jung, S., Diankov, R., & Bajcsy, R. (2006, August). The effects of fully immersive virtual reality on the learning of physical tasks. In Proceedings of the 9th Annual International Workshop on Presence, Ohio, USA (pp. 87-94).

[5] ShojaeipourShahed, Mohamed Haris S, Gholami E, Shojaeipour A, 2010. Webcam-based Mobile Robot Path Planning using Voronoi Diagrams and Image Processing. Bangi, Selangor, 43600, Malaysia.

[6] GiffordChristopher M, (2009). Low-cost Mobile Robot Localization Using Only a Downward-Facing Webcam. Department of Electrical Engineering and Computer Science.

[7] Brooks Jr, F. P. (1999). What's real about virtual reality?. Computer Graphics and Applications, IEEE, 19 (6), 16-27.

[8] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. O'reilly, 2008.

[9] Jianwei, W. A. N. G. (2014). A Noise Removal Algorithm of Color Image. TELKOMNIKA Indonesian Journal of Electrical Engineering, 12(1), 565-574.

[10] Bazmi, A. (2012). Increasing the Accuracy of Detection and Recognition in Visual Surveillance. International Journal of Electrical and Computer Engineering (IJECE), 2(3), 395-404.

[11] Farouki, R. T. (2008). Pythagorean—hodograph Curves (pp. 381-391). Springer Berlin Heidelberg.

[12] Miner, N. E., & Stansfield, S. A. (1994, May). An interactive virtual reality simulation system for robot control and operator training. In Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on (pp. 1428-1435). IEEE.

[13] Xu, Z., Zhao, Z. X., Wu, M. H., Liao, J. B., & Tian, G. Y. (1997). Virtual Reality Based Robot Graphic Simulation and Virtual Manufacturing System. The 1997 Chinese Automation Conference in the UK (CACUK'97).

[14] Alam, I. P., & Semarang, U. N. S. U. Pengendali Motor Servo Berbasis Mikrokontroler Basic Stamp 2sx Untuk Mengembangkan Sistem Robotika.

[15] AVR Microcontroller ATmega Datasheet. ATMEL.

[16] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.