

Distance Estimation based on Color-Block: A Simple Big-O Analysis

Budi Rahmani¹, Hugo Aprilianto², Heru Ismanto³, Hamdani Hamdani⁴

^{1,2}Program Studi Teknik Informatika, STMIK Banjarbaru, Kalimantan Selatan, Indonesia

³Department of Informatic Engineering, Musamus University, Merauke, Indonesia

⁴Department of Computer Science, Faculty of Computer Science and Information Technology, Mulawarman University, Samarinda, Indonesia

Article Info

Article history:

Received Okt 15, 2016

Revised Jan 22, 2017

Accepted Feb 6, 2017

Keyword:

Arduino nano

Big O

Complexity

Object detection

Pixy cmucam5

ABSTRACT

This paper explains how the process of reading the data object detection results with a certain color. In this case, the object is an orange tennis ball. We use a Pixy CMUcam5 connecting to the Arduino Nano with microcontroller ATmega328-based. Then through the USB port, data from Arduino nano re-read and displayed. It's to ensure whether an orange object is detected or not. By this process, it will be exactly known how many blocks object detected, including the X and Y coordinates of the object. Finally, it will be explained the complexity of the algorithms used in the process of reading the results of the detection orange object.

Copyright © 2017 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Budi Rahmani,
Program Studi Teknik Informatika,
STMIK Banjarbaru,
Kalimantan Selatan, Indonesia.
Email: hugo.aprilianto@gmail.com

1. INTRODUCTION

Robot navigation is the way of the robot to change its position or to go to its goal position from its stationary position without hitting any obstacle. Navigation may define as the ability to move in any particular environment [1]. Other researchers define it as a science in guiding a robot to move to its environment [2]. As for the problems that accompany this process is the navigation can be defined in three questions, namely: "Where am I", "Where am I going", and "How do I get there". For the first and second questions can be answered by completing the appropriate robot with sensors, while the third question can be done with an effective planning system navigation [3]. The navigation system itself is directly related to the presence of the sensors used in robots and the structure of the environment, and that means there will always match the purpose-built robot and the environment in which the robot will be operated [4-6]. Vision-based navigation tremendous progress with the implementation of various autonomous vehicles, like autonomous ground vehicles (AGV), autonomous underwater vehicles (AUV), and unmanned aerial vehicles (UAV) [7-9].

Regardless of the type of vehicle or robot is built, then the system utilizes a vision sensor for navigation purposes, can roughly be divided into two general categories: systems that require prior knowledge of the environment in which it will be operated and systems who see environmental conditions to which he will navigate. Systems that require a map can be subdivided into folders using systems, map-building systems, and topological map-based systems [10]. As the name suggests, it is the map using navigation systems need to include a complete map of the environment before starting navigation. While the

metric map-building systems all over the map itself built and used in the next phase of navigation. Furthermore, other systems that are in this category is a system that can perform self-localize on the environment simultaneously performed during map construction purposes [5]. And other types of map-building navigation systems are encountered, e.g. visual sonar-based systems or local folder-based systems. Both of these systems collect environmental data when navigating, and build a local folder that is used to support in order to be an appropriate navigational purpose. As for the local map includes mapping the barrier and space, and this is usually a function of the angle of view of the camera [10]. The last system, namely: a topological map-based, which builds a topology map that consists of nodes are connected by a line, where the vertices represent the place/specific positions on the environment, and links represent the distance or travel time between the two nodes [10], [11].

The next navigation system is mapless navigation systems which mostly include the reactive technique that uses visual cues are built from image segmentation, optical flow, or the search process features between image frames obtained. There is no representation of the environment on these systems, and environments seen/perceived to navigate the system, recognize objects, or browse landmark [5]. One of the most important parts of the process of vision-based robot navigation is how to process the data obtained from the camera sensor used into useful information for a particular robot to navigate toward a specific point on the environment or certain specified arena. On this occasion, the navigation is done by a humanoid robot soccer players is how to object orange balls [12], using data obtained readings from sensors used CMUcam5 pixy camera. Then it will be analyzed more specifically the complexity of the algorithms used to read the results of the detection of the object [13].

Visual-based navigation method divided into three categories, i.e. map-based, map building, and mapless navigation. One of map-based navigation method was described in [14] that uses a stereo camera on a wheeled robot as a golf ball collector. It's associated with a wide-angle camera mounted on top of the golf course with a viewing area covers area 20m x 20m, with a viewing angle of 80° vertical and horizontal viewing angle of 55° with the camera installation height is 7m above the ground. Resolution capable captured by this camera is equal to 1280x780 pixels. The camera catches the image is then processed by a computer server which then builds navigation map grid-shaped or there is a call later with questions of occupancy (occupancy folder) that can indicate where to position the robot makers ball on the golf course, and where the ball position to be taken off the map, is determined by the server and the robot can move to a certain position based on the information sent wirelessly [14], [15]. The next category of navigation method is map building navigation described in [11], [12]. A new algorithm that was developed using a stereo camera [16]. The aim is to estimate the free space of the vehicle (car) which is in front of where the system is run, particularly for navigation on the highway. By utilizing the difference map (disparity map) [17], it will obtain information on whether the condition of the vehicle in front of him was 'solid' or 'rarely'.

Further to the development of scores folder, because the sensor used is a stereo camera, which is carried out next is a stereo matching process based on image data that is passed longitudinal road surface. Detection of free space is done by extracting the road surface without any obstacle. The last category is mapless navigation method. Optical flow is one of the best methods that mimics how the visual behavior of animals, bees, which are determined by the basic robot movement speed difference between the image seen by the eye (camera) and the right image seen by the eye left, and the robot will move to the side of the speed change is the smallest image [5]. This method is widely used to detect an obstacle, and following this method developed further in many algorithms, including the Horn-Schunck algorithm (HSA) and Lucas-Kanade algorithm (LKA). In the HSA, proposed an Equation of the optical flow method that meets the constraints and smoothness constraints globally. While LKA proposed quadrat Equation of optical flow with the smallest weight [18-20].

Furthermore, the optical flow is the instantaneous velocity of each pixel in the image at a certain point. Optical flow field is described as a field of gray of all pixels in the image of the observed field. This optical flow field reflects the relationship between the time domain and the position of adjacent frames of the same pixel position [18]. Inconsistencies between the directions of optical flow field and optical flow and major movements can be used to detect an obstacle. Uses optical flow field computed multi-image obtained from the same camera at different times, and the main movement of the camera in the estimation based on OFF. Optical flow can be generated from two camera movements that flower and flowerpot, if the distance of the object in the field of view and the camera is d and the angle between the object and the direction of translational movement is θ , then the camera will move with translational velocity (TV) v and angular velocity (AV) ω . Next optic flow generated by the object can be calculated by Equation (1) [18]:

$$F = -\omega + \left(\frac{v}{d}\right) \sin \theta \quad (1)$$

Where:

F = The amount of optical flow

v = velocity or displacement translation (translational velocity)

d = object in the field of view of the camera

ω = angular velocity

θ = the direction of translational movement

2. RESEARCH METHOD

Here is the algorithm which will detect the presence of objects with specific color by using the camera pixy CMUcam5 whose output is fed to the controller (Arduino Nano with ATmega328) via serial communication lines. The program code is executed by the controller continuously read serial data, the existence of an object with a specific color (in signature orange 1), if the object in question then it is detected as the 'blocks' of those colors. Every block is detected by the camera will be sent the data to the controller with a speed of 50 frames per second [21]. Hence the controller will also create delays the process of reading that much, this is done for asynchronous data sent by the camera and the data received by the controller. The controller will then communicate back the received data to the user via the universal serial bus (USB) port to the user that the information it contains 'blocks' color of the object detected by the camera, then its position in the field of x and y , as well as the size of the width and height of the block. The. Diagram of a system built presented in Figure 1 fit the above description.

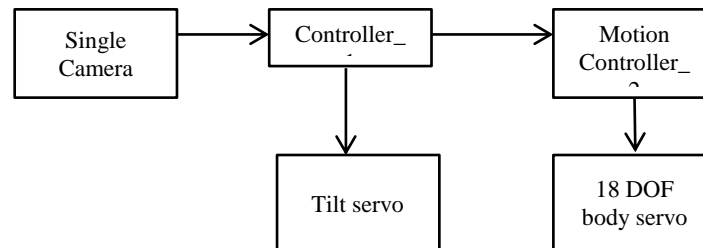


Figure 1. Block diagram system

Below is shown an algorithm that reads the data of the detection object orange balls [21], [22].

```

void loop()
{
  static int i = 0; // initializing the variable y is zero with static data type integer
  int j;           // j variable declaration
  uint16_t blocks; // blocks is the number of pixels of color of the object to be detected
  char buf[32];    // buffer variable declaration with 32 array characters type

  // 'blocks' variable declaration to accommodate the results of the function pixy.getBlocks()
  blocks = pixy.getBlocks();

  // If the blocks or object are found
  // Print the results of blocks every 50 frames
  // And subsequently fed to arduino if i modulo 50 = zero
  if (blocks)
  {
    i++;
    if (i%50==0)
    {
      sprintf(buf, "Detected %d:\n", blocks);
      Serial.print(buf);
      for (j=0; j<blocks; j++)
      {
        sprintf(buf, " block %d: ", j);
        Serial.print(buf);
      }
    }
  }
}
  
```

```

    pixy.blocks[j].print();
  }
}
}
}

```

3. RESULTS AND ANALYSIS (10 PT)

Here are the results display on the personal computer (PC) over the data that is read by the controller shown in Figure 2.

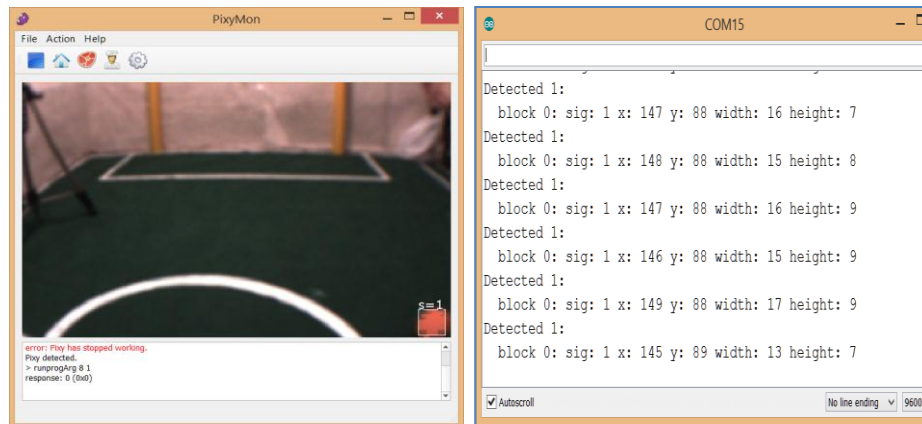


Figure 2. Results of colored object detection (via USB)

The above algorithm will be analyzed one by one in a way that is [23]:

a. Number of instruction

The number of instruction from the algorithm is:

```

static int i = 0;           → 1 instruction
int j;                     → 1 instruction
uint16_t blocks;          → 1 instruction
char buf[32];              → 1 instruction
blocks = pixy.getBlocks(); → 2 instruction

if (blocks)                → 1 instruction
{
  i++;                      → 1 instruction
  if (i%50==0)              → 1 instruction
  {
    sprintf(buf, "Detected %d:\n", blocks);
    Serial.print(buf);

    for (j=0; j<blocks; j++) → 2 instruction in loop
    {
      sprintf(buf, " block %d: ", j);
      Serial.print(buf);
      pixy.blocks[j].print();
    }
  }
}
}

```

b. Determine function of above algorithm

If the instruction before the loop for calculated, it would amount to approximately 9 instructions, therefore its function is $f(n)=9+2n$. From the function can be seen that it is a linear function.

c. Determine the complexity of the function obtained

Based on the function obtained from the above algorithm, we found the complexity of the function $f(n) = 9 + 2n$ with *big O* is n or $O(n)$.

To ensure that the algorithm analysis previously conducted in accordance with the results of experimental measurements of the distance the ball and robots in real-time, then the experiment measuring the distance the ball and robot re-done. The sketch prototype of robots used and the results obtained, shown in Figure 3 and Table 1, respectively.

In Table 1 are shown the results of experimental measurements of the distance the ball to the robot in real-time. Based on experiments conducted found a trend of rising pixel size color blocks the ball, at a distance of the ball on the robot is reduced. Based on experiments conducted the closest distance the ball and robots that can be detected is approximately 51 cm, and the greatest distance is about 210 cm. This condition is based on the construction of a camera mounted on a robot according to the prototype used and shown in Figure 4.

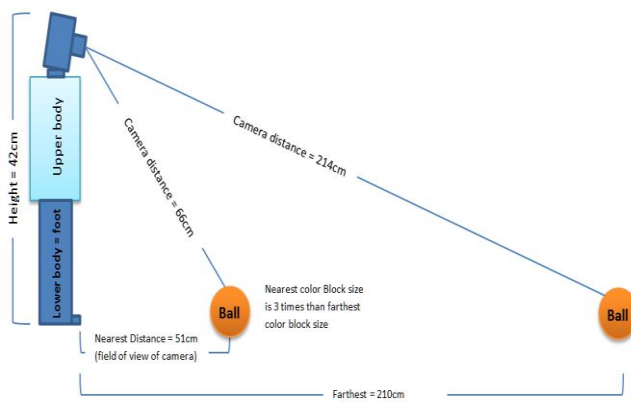


Figure 3. Sketch of prototype robot [2]

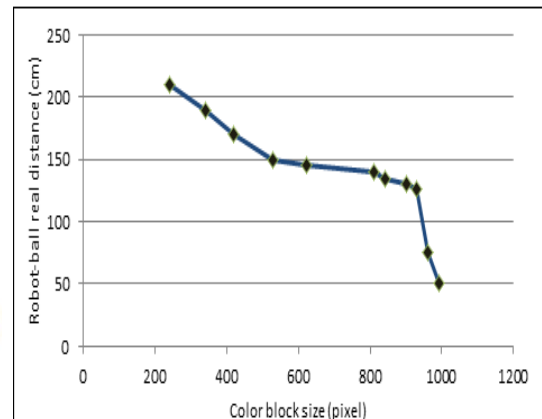


Figure 4. Linear graphic of color block size toward Robot-ball real distance

Table 1. Color block size toward Robot-ball real distance

Distance information	Height (pixel)	Width (pixel)	Color Ball block size (pixel)	Real-time robot-ball distance (centimeter)
Farthest distance	15	16	240	210
	18	19	342	190
	21	20	420	170
	23	23	529	150
	24	26	624	145
	28	29	812	140
	29	29	841	135
	30	30	900	130
	31	30	930	126
	31	31	961	76
Nearest distance	32	31	992	51

4. CONCLUSION

Based on the analysis of the algorithm used to detect orange tennis ball on the field or arena, colored dark green base, found a mathematical function $f(n) = 9 + 2n$ with the complexity of using the *big O* is n or $O(n)$. Its means that the complexity of the algorithm used is linear, and so, when compared with the results of experimental measurements of the distance the robot and the ball physically. This is shown on the graph of the results distances tend to fall with an increase in the size of the color blocks of balls that were

detected by the camera used. Hence the algorithms used analysis method is quite effective in analyzing an algorithm.

REFERENCES

- [1] W. L. Fehلمان-II, M. K. Hinders, "Mobile Robot Navigation with Intelligent Infrared Image Interpretation", London: Springer-Verlag London, 2009.
- [2] B. Rahmani, A. Harjoko, T. K. Priyambodo, H. Aprilianto, "Research of Smart Real-time Robot Navigation System", in *The 7th SEAMS-UGM Conference 2015*, 2015, pp. 1–8.
- [3] B. Rahmani, A. E. Putra, A. Harjoko, T. K. Priyambodo, "Review of Vision-Based Robot Navigation Method", *IAES Int. J. Robot. Autom.*, vol. 4, no. 4, pp. 31–38, 2015.
- [4] P. Alves, H. Costelha, C. Neves, "Localization and navigation of a mobile robot in an office-like environment", in *2013 13th International Conference on Autonomous Robot Systems*, 2013, pp. 1–6.
- [5] A. Chatterjee, A. Rakshit, and N. N. Singh, *Vision Based Autonomous Robot Navigation*. Springer US, 2013.
- [6] I. Iswanto, O. Wahyunggoro, A. Imam Cahyadi, "Path Planning Based on Fuzzy Decision Trees and Potential Field," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 1, p. 212, 2016.
- [7] A. M. Pinto, A. P. Moreira, P. G. Costa, "A Localization Method Based on Map-Matching and Particle Swarm Optimization," *J. Intell. Robot. Syst.*, pp. 313–326, 2013.
- [8] R. Strydom, S. Thurrowgood, M. V Srinivasan, "Visual Odometry: Autonomous UAV Navigation using Optic Flow and Stereo," *Australas. Conf. Robot. Autom.*, pp. 2–4, 2014.
- [9] J. Cao, X. Liao, E. L. Hall, "Reactive Navigation for Autonomous Guided Vehicle Using the Neuro-fuzzy Techniques," in *Proc. SPIE 3837, Intelligent Robots and Computer Vision XVIII: Algorithms, Techniques, and Active Vision*, 1999, pp. 108–117.
- [10] F. Bonin-Font, A. Ortiz, G. Oliver, "Visual Navigation for Mobile Robots: A Survey," *J. Intell. Robot Syst.*, vol. 53, pp. 263–296, 2008.
- [11] M. S. Rahman and K. Kim, "Indoor Positioning by LED Visible Light Communication and Image Sensors," *Int. J. Electr. Comput. Eng.*, vol. 1, no. 2, pp. 161–170, 2011.
- [12] F. Maleki and Z. Farhoudi, "Making Humanoid Robots More Acceptable Based on the Study of Robot Characters in Animation," vol. 4, no. 1, pp. 63–72, 2015.
- [13] B. Earl, "Pixy Pet Robot - Color Vision Follower using Pixycam," *Adafruit Learning System*, 2015. [Online]. Available: <https://learn.adafruit.com/pixy-pet-robot-color-vision-follower-usingpixycam>.
- [14] C. H. Yun, Y. S. Moon, N. Y. Ko, "Vision Based Navigation for Golf Ball Collecting Mobile Robot," *Int. Conf. Control. Autom. Syst.*, no. Iccas, pp. 201–203, 2013.
- [15] R. L. Klaser, F. S. Osorio, D. Wolf, "Vision-Based Autonomous Navigation with a Probabilistic Occupancy Map on Unstructured Scenarios," *2014 Jt. Conf. Robot. SBR-LARS Robot. Symp. Rob.*, pp. 146–150, 2014.
- [16] K.-Y. Lee, J.-M. Park, J.-W. Lee, "Estimation of Longitudinal Profile of Road Surface from Stereo Disparity Using Dijkstra Algorithm," *Int. J. Control. Autom. Syst.*, vol. 12, no. 4, pp. 895–903, 2014.
- [17] S. A. R. Magrabi, "Simulation of Collision Avoidance by Navigation Assistance using Stereo Vision," *Spaces*, pp. 58–61, 2015.
- [18] Q. Wu, J. Wei, X. Li, "Research Progress of Obstacle Detection Based on Monocular Vision," in *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2014, pp. 195–198.
- [19] N. Ohnishi, A. Imiya, "Appearance-based navigation and homing for autonomous mobile robot," *Image Vis. Comput.*, vol. 31, no. 6–7, pp. 511–532, 2013.
- [20] M. L. Wang, J. R. Wu, L. W. Kao, H. Y. Lin, "Development of a vision system and a strategy simulator for middle size soccer robot," *2013 Int. Conf. Adv. Robot. Intell. Syst. ARIS 2013 - Conf. Proc.*, pp. 54–58, 2013.
- [21] A. Rowe, R. LeGrand, S. Robinson, "An Overview of CMUcam5 Pixy," 2015. [Online]. Available: <http://www.cmucam.org/>. [Accessed: 10-Jun-2015].
- [22] A. J. R. Neves, A. J. Pinho, D. a. Martins, B. Cunha, "An efficient omnidirectional vision system for soccer robots: From calibration to object detection," *Mechatronics*, vol. 21, no. 2, pp. 399–410, 2011.
- [23] D. Zindros, "A Gentle Introduction to Algorithm Complexity Analysis," 2012. [Online]. Available: <http://www.discrete.gr/complexity/>. [Accessed: 10-Apr-2015].

BIOGRAPHIES OF AUTHORS



Budi Rahmani received his bachelor's of Electrical Engineering, and Master of Informatics from Yogyakarta State University and Dian Nuswantoro University in 2003 and 2010 respectively. Currently he is a doctoral student at Department of Computer Science and Electronics, Universitas Gadjah Mada since 2014. He was interesting in Embedded System and Robotics. Currently his research focused is computer vision and control system for robot. His other research interests include decision support system using artificial neural network. He can be contacted by email: budirahmani@gmail.com <http://budirahmani.wordpress.com>



Hugo Aprilianto received his bachelor's of Informatics, and Master of Informatics from Universitas Dr Soetomo Surabaya and Sekolah Tinggi Teknik Surabaya in 1998 and 2007 respectively. Currently he is a lecturer at STMIK Banjarbaru. He was interesting in Computer Architecture, Embedded System, and Neural network. He can be contacted by email: hugo.aprilianto@gmail.com



Heru Ismanto received his bachelor's of Informatics, and Master of Computer Science from Universitas Padjadjaran and Universitas Gadjah Mada in 1996 and 2009 respectively. Currently he is a lecturer at Department of Informatics Engineering, Musamus University, Merauke, Papua and his a doctoral student at Department of Computer Science and Electronics, Universitas Gadjah Mada since 2014. He was interesting in Decision Support System, GIS, e-Government System. He can be contacted by email: heru.ismanto@mail.ugm.ac.id or heru.ismanto31@yahoo.com



Hamdani received bachelor's of Informatics from Universitas Ahmad Dahlan, Yogyakarta, Indonesia in 2002, received Master of Computer Science from Universitas Gadjah Mada, Yogyakarta, Indonesia in 2009. Currently, he is a lecturer at Department of Computer Science in Universitas Mulawarman, Samarinda, East Kalimantan, Indonesia and candidate his Ph.D. program in Computer Science at Department of Computer Sciences & Electronics in Universitas Gadjah Mada, Yogyakarta, Indonesia. His research areas of interest are group decision support system/decision model, social networks analysis, GIS and web engineering. Email: hamdani@fkti.unmul.ac.id and URL: <http://daniunmul.weebly.com>