

Automated Guided Vehicle (AGV) Driving System Using Vision Sensor and Color Code

Jun-Yeong Jang , Su-Jeong Yoon  and Chi-Ho Lin * 

School of Computer Science, Semyung University, Jecheon 27136, Republic of Korea; win4723@nate.com (J.-Y.J.)

* Correspondence: ich410@semyung.ac.kr

Abstract: Recently, the use of Automated Guided Vehicles (AGVs) at production sites has been increasing due to industrial development, such as the introduction of smart factories. AGVs utilizing the wired induction method, which is cheaper and faster than the wireless induction method, are mainly used at production sites. However, the wired guidance AGV operation system has the disadvantage of being limited to small-batch or collaboration-based production sites, since it is difficult to change the driving route. In this paper, we propose an AGV line-scan algorithm that can perform route recognition, driving commands, and operation through color-code recognition using an Arduino controller and a low-cost vision sensor, instead of the optical sensor conventionally used for these functions. When the proposed algorithm is applied to the AGV car, the CMUcam 5 Pixy2 camera identifies the driving path to follow by tracking a black line using the Otsu method. In addition, it can be confirmed that the driving command is executed using the proposed color code by applying the color recognition function of the CMUcam 5 Pixy2.

Keywords: AGV; color code; HSI; RGB; vision sensor



Citation: Jang, J.-Y.; Yoon, S.-J.; Lin, C.-H. Automated Guided Vehicle (AGV) Driving System Using Vision Sensor and Color Code. *Electronics* **2023**, *12*, 1415. <https://doi.org/10.3390/electronics12061415>

Academic Editor: Chiman Kwan

Received: 31 January 2023

Revised: 28 February 2023

Accepted: 11 March 2023

Published: 16 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, the Fourth Industrial Revolution has been in the spotlight due to the remarkable evolution of information and telecommunication technology. Consequently, various research studies are being conducted on the core technologies of the Fourth Industrial Revolution, such as big data, artificial intelligence, and the Internet of Things. In addition to the core technology of the Fourth Industrial Revolution, the industries affecting change through the revolution are also attracting a great deal of attention. For example, smart factories are leading a wave of major change, with many additional cases in other fields that began to develop during the Fourth Industrial Revolution.

Among the various technologies being utilized by smart factories, one technology that leaders in the industry are paying close attention to is AGV technology [1,2]. The AGV was first used in 1953 as a means of pulling a trailer. By the early 1960s, it was being used in various factories and warehouses. Subsequently, in the 1970s, the use of AGVs expanded throughout the industry, with AGVs replacing the conveyor previously used in the assembly line of Swedish automaker Volvo. The modern AGV is powered by a battery. The AGV travels to the desired position along given travel lines [3–6]. The methods of guiding the traveling route of the AGV include laser navigation and wireless guidance using RFID. In addition, wired guidance methods such as magnetic, optical, electromagnetic, and magnetic-gyro guidance may be utilized [7–17]. As for the wireless guidance method, this is a method of guiding the AGV to the target point by first creating a virtual route in advance, and then using the current position of the AGV obtained through the positioning sensor to navigate. Since there is no guidance line, the AGV can travel autonomously, making it easy to change and maintain the travel line. However, the wireless induction may cause a risk of error in measurement due to obstacles and signal distortion triggered by the disturbance of radio waves or the refraction of light. Furthermore, it

has the drawback that the difference in accuracy becomes significant, depending on the workplace environment and the characteristics of the transceiver installed in the AGV. At present, at industrial sites, the wired guidance method is mainly used due to the advantage of inexpensively implementing the tracking method and driving the AGV. However, the wired guidance method has the drawback of posing difficulty in changing and maintaining guidance lines, as it is a method of embedding or attaching guidance lines to the floor of the workplace. Therefore, various studies are underway to overcome such problems [18–21]. Various attempts have been made, such as a pattern-recognition method that uses simple barcode-like location-specific symbols, which are then identified using a line-scan camera or other method of recognizing the surrounding environment, such as a deep learning-based vision system [22–25].

In this paper, we propose an AGV line-scan algorithm that can perform route recognition driving commands, and operation through color-code recognition using an Arduino controller and a low-cost vision sensor, instead of the existing method of recognizing and driving along a line using an optical sensor. The proposed method is a path-recognition technology that reduces computing and is easy to maintain and change at a low cost compared to methods using images or RFID tags, by utilizing pattern recognition with an existing line-scan camera and a simple barcode-like identification symbol. In addition, the proposed method has the advantage of being able to solve the problem of driving via path recognition, which used to be performed using various sensors, using only a vision sensor.

When the proposed algorithm is applied to the AGV car, the CMUcam 5 Pixy2 camera identifies the driving path to follow by tracking a black line using the Otsu method. In addition, it can be confirmed that the driving command is executed using the proposed color code by applying the color recognition function of the CMUcam 5 Pixy2.

2. AGVA-Robot Driving System

2.1. Materials

Introducing the composition of the AGVA-Robot used for the study discussed in this paper: The main CPU of the AGVA-Robot was manufactured by ARDUINO, which is open-source based hardware with good accessibility. An L298N, which is highly versatile, was used as a driver for operating the motors of the AGVA-Robot. A CMUcam5 Pixy2 was used as a vision sensor.

2.1.1. CMUcam5 Pixy2

The vision sensor selected for the study discussed in this paper is CMUcam5 Pixy2. The CMUcam5 Pixy2 vision sensor is a fast image sensor that tracks objects and can directly connect to Arduino Uno R3 through the ICSP port on the Arduino Uno R3 circuit board. CMUcam5 Pixy2 has its own powerful processor with which to process the image. Since CMUcam5 Pixy2 has its own processor, it processes the images captured from the sensor and extracts the useful information. Additionally, CMUcam5 Pixy2 comes with a color-recognition algorithm that enables it to detect an object's color. Normally, RGB (red, green, and blue) is used to represent colors. Instead, CMUcam5 Pixy2 calculates the hue, color, and saturation of each RGB pixel from the image sensor and uses these as the primary filtering parameters.

2.1.2. Arduino Uno R3

Arduino Uno R3 is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to begin.

2.1.3. L298N Motor Driver

This L298N-based motor driver module is a high-voltage, high-current DUAL FULL-BRIDGE driver suitable for driving DC motors and stepper motors. It can control up to four DC motors, or two DC motors with directional and speed controls.

2.2. Circuit Design

Figure 1 shows the overall block diagram of the AGVA-Robot we used in the paper. Arduino Uno was used as the main board and an ISP port was used to interface with the vision sensor [26]. The CMUcam5 Pixy2 camera was used for color recognition and line tracking [27]. Two servo motors were used to allow the vision sensor to move up, down, left, and right. As the engine of the AGVA-Robot, four servo-type DC motors were used (two on the left and two on the right), and the L298N was used as a driver for operating the four motors [28]. An independent power supply was also equipped.

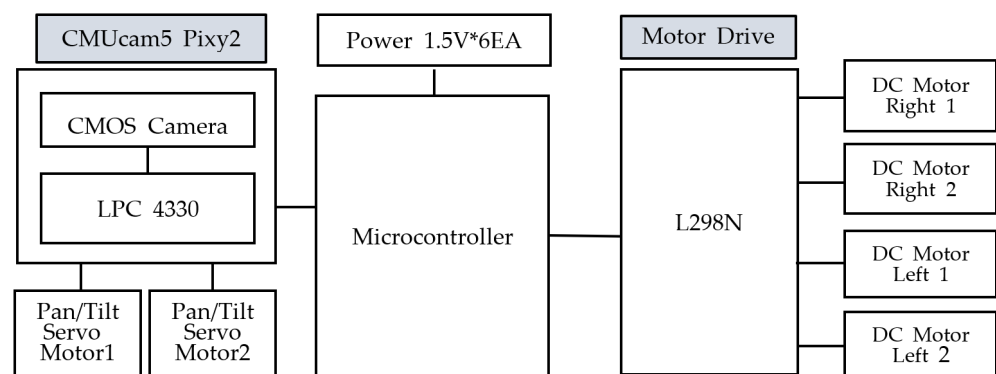


Figure 1. Block diagram of the configuration of the AGVA-Robot. Schematics follow the same formatting.

Figure 2 shows the overall circuit diagram of the AGVA-Robot we used for the study discussed in this paper.

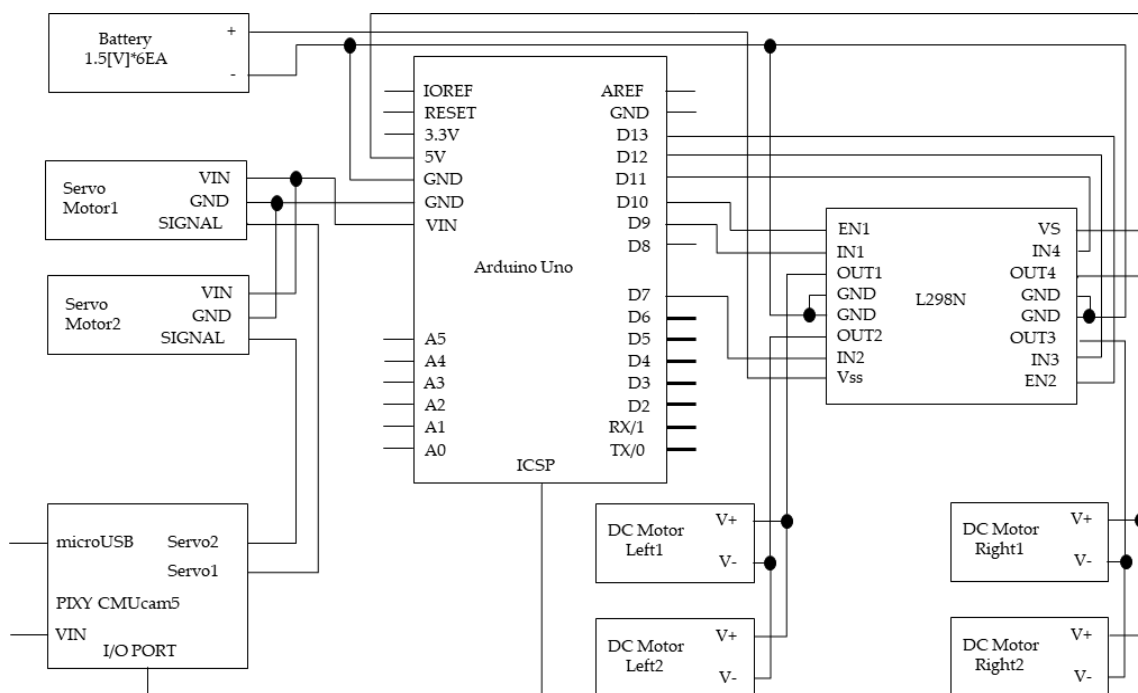


Figure 2. Schematic circuit diagram of the AGVA-Robot.

Figure 3 shows the final result of the AGVA-Robot's hardware design. It consists of: a CAMcam5 pixy2, two pan/tilt motors, an Arduino Uno R3, a power pack, a motor drive L298N, and four DC motors.

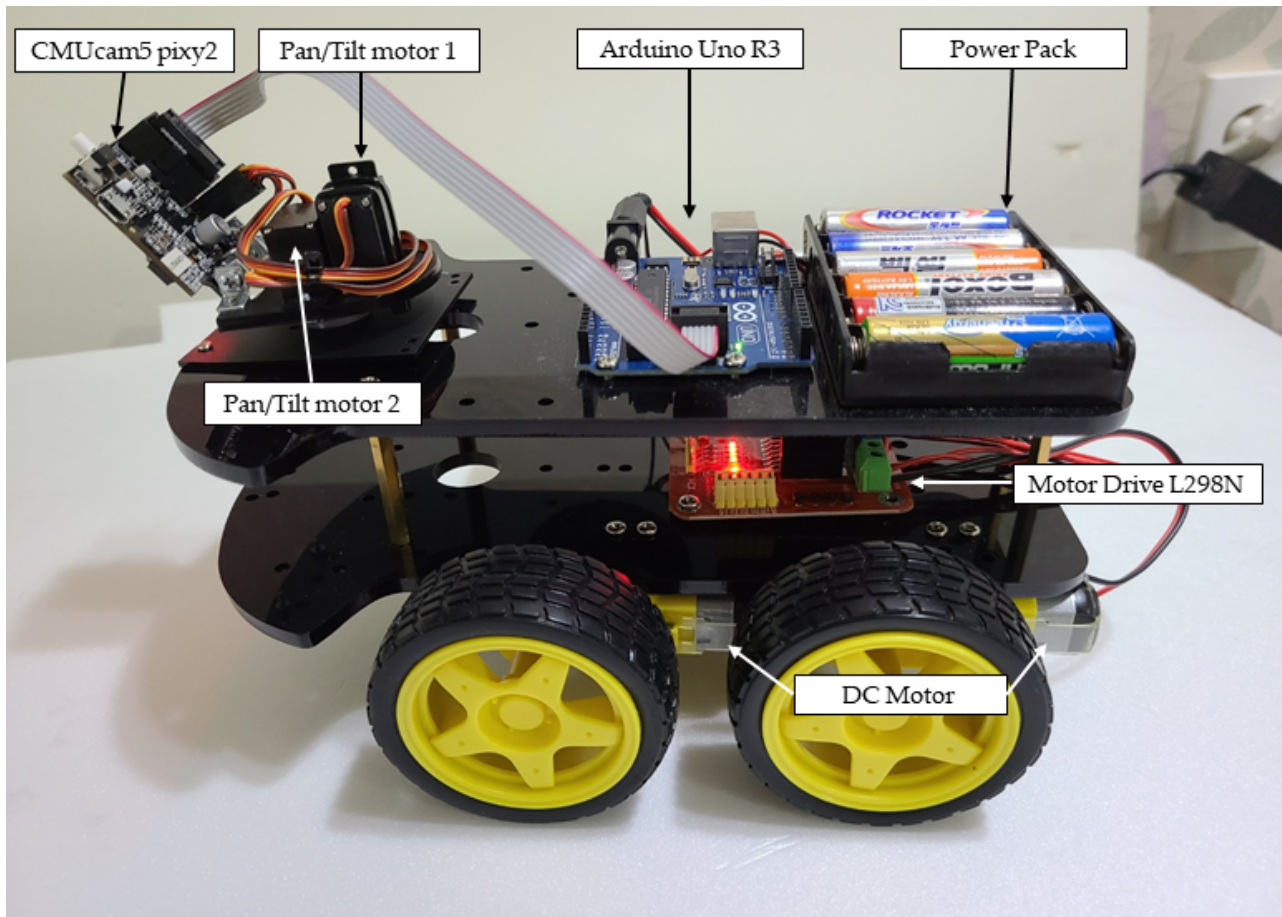


Figure 3. Final hardware design of the AGVA-Robot.

2.3. Operating Flowchart

The AGVA-Robot is designed to use a vision sensor to photograph the traveling route, indicated by a black line on a white background, recognize the route by means of the line-tracking algorithm, and follow the color code when it is detected during travel. Figure 4 shows the overall traveling algorithm. After extracting an image from a video, in order to obtain information from the image, it is necessary to separate the objects contained in the image. One of the representative methods for separating objects is binarization. For the study discussed in this paper, the Otsu method was used as the binarization algorithm for line recognition [22]. Additionally, the color coding function of CAMcam5 pixy2 was used to give the driving command from the color code. In CAMcam5 pixy2, the HSI color model was used for color detection. The order of the detected color models was coded and matched with driving commands to be used in the AGV's driving algorithm.

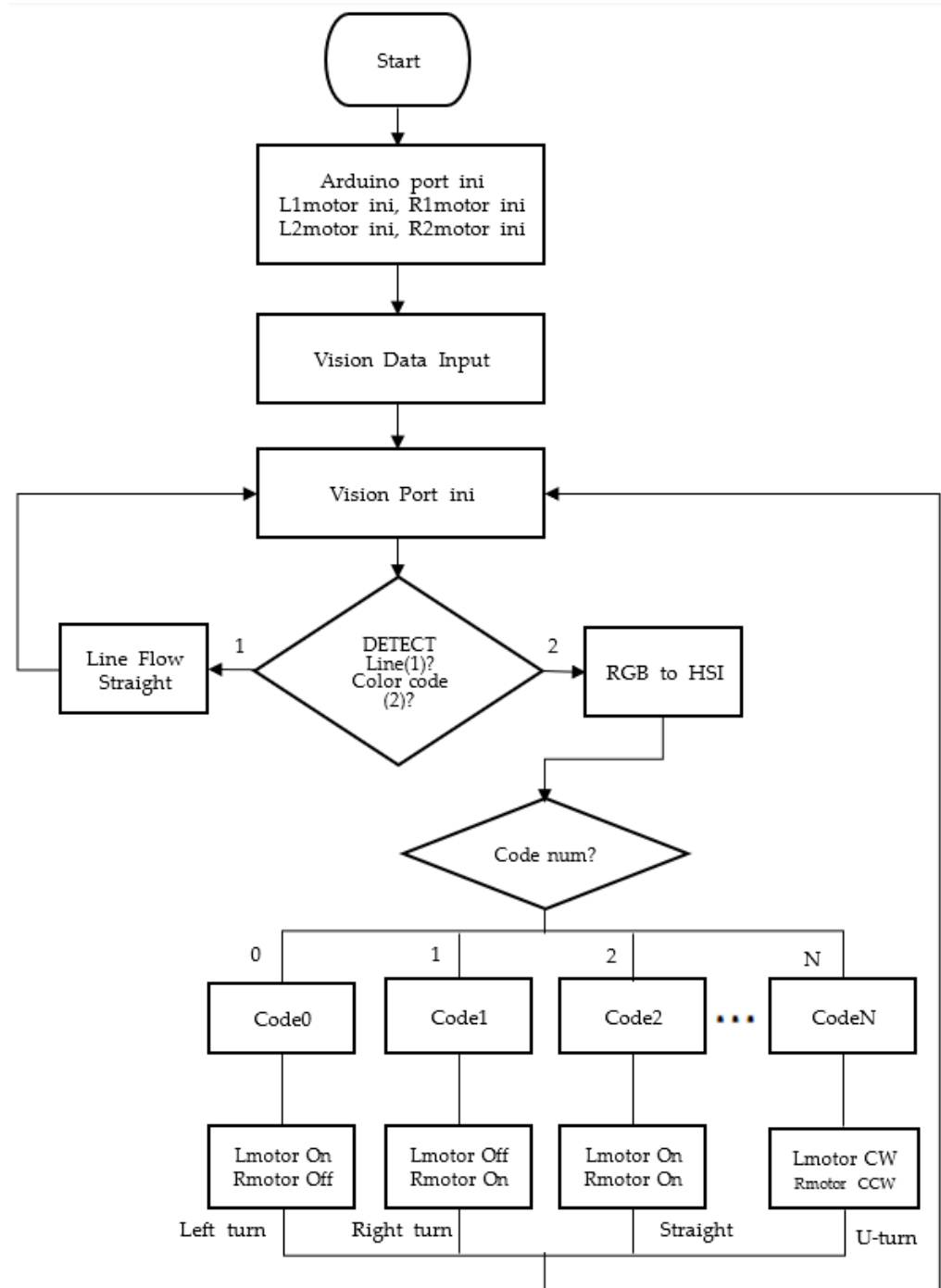


Figure 4. Operating flowchart of the AGVA-Robot.

2.4. Experimental Environment

Figure 5 shows a traveling experiment performed in a virtual layout designed to execute a traveling command in an environment where various traveling lines, composed of straight lines, curves, and plane intersections, are drawn for the traveling test of an AGVA-Robot [29,30]. The AGVA-Robot is designed to photograph, using its own camera, a traveling route consisting of a black line on a white background, to recognize lines based on the line tracking algorithm, and to execute various traveling commands based upon the color codes detected at branch points.

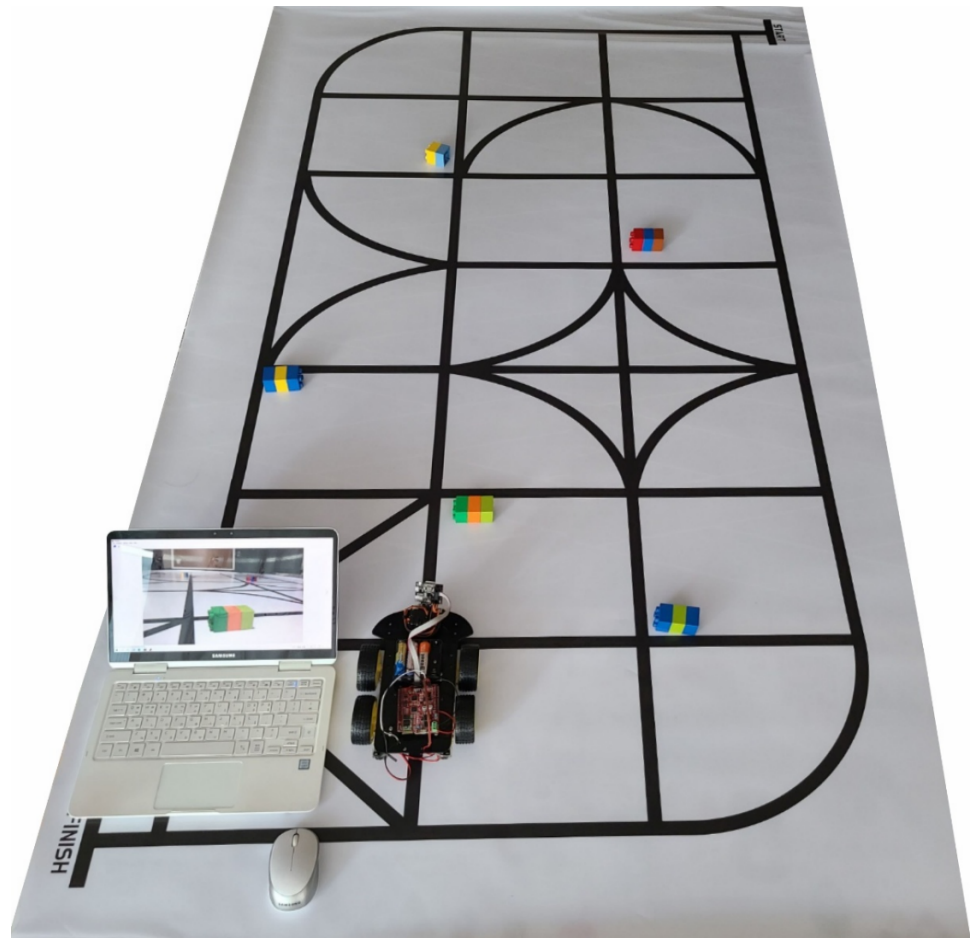


Figure 5. Virtual experimental environment.

3. Color Model

An RGB color model and an HSI color model were both used to enable the AGVA-Robot to recognize the color code from the input image using the vision sensor. The HSI model is a model that expresses colors using three characteristics: H (Hue), S (Saturation), and I (Intensity). H refers to an attribute that expresses pure colors, S refers to an attribute that expresses saturation, and I refers to an attribute that expresses the brightness value [31].

3.1. RGB to HSI

Since the HSI color model is less sensitive to illumination than the RGB color model, converting the RGB color model into the HSI color model enables it to function regardless of illumination and sudden changes in light. It is not necessary to know what percentage of blue or green is needed to generate a certain color, as dark red turns to pink simply by adjusting the saturation. Controlling contrast can brighten darker colors; for this reason, the HSI color model is being used in many applications. Contrast I and hue H can be calculated using the following formula [32].

$$I = \frac{1}{3}(R + G + B) \quad (1)$$

$$\begin{aligned}
H_1 &= \cos^{-1} \left(\frac{(P_r - W) \bullet (P - W)}{|P_r - W| |P - W|} \right) \\
&= \cos^{-1} \left(\frac{2R - G - B}{\sqrt{\frac{2}{3}} \sqrt{6(R^2 + G^2 + B^2 - RG - GB - BR)}} \right) \\
&= \cos^{-1} \left(\frac{3R - (R + G + B)}{2\sqrt{R^2 + G^2 + B^2 - RG - GB - BR}} \right) \\
&= \cos^{-1} \left(\frac{(R - G) + (R - B)}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \right)
\end{aligned} \tag{2}$$

If $B > G$, hue H is the same as Formula (3).

$$H = 360 - H_1 \tag{3}$$

Hue H , obtained from Formula (3), represents a hue between 180 and 360 degrees [33]. Saturation S can be obtained as shown in Formula (4).

$$S = 1 - 3 \min(R, G, B) / I \tag{4}$$

Figure 6 shows source codes of the C language used for programming to convert RGB values to HSI values.

```

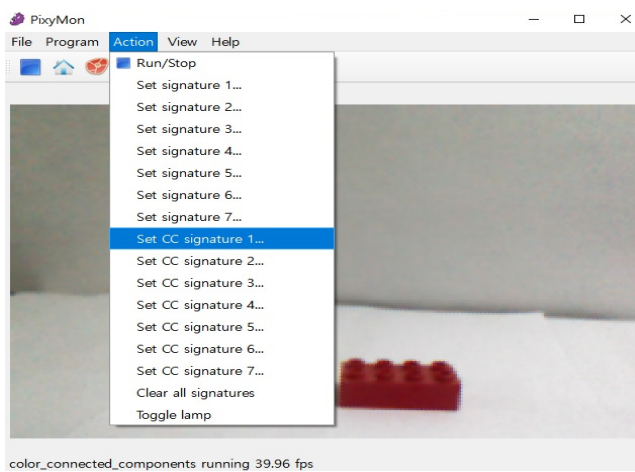
rgb_hsi(Red, Green, Blue, Hue, Satu, Inten)
float Red, Green, Blue, *Hue, *Satu, *Inten;
{
float red, green, blue, w, i, min=1.e-6;
*Inten = (i=Red+Green+Blue)/3.0;
red = Red/i, green = Green/i, blue = Blue/i;
if (Red==Green && Green==Blue)
{
*Satu=0, *Hue=0;
}
else
{
w = 0.5 * (Red-Green+Red-Blue) / sqrt( (Red-Green) * (Red-Green) + (Red-Blue) * (Green-Blue));
if(w > 1) w = 1;
if(w < -1) w = -1;
*Hue= acos(w);
if( *Hue < 0 )
{ printf("Hue<0 : %f\n",*Hue); pause(); }
if( Blue>Green ) *Hue = 2 * Pi - *Hue;
if( red<=green && red<=blue ) *Satu=1-3*red;
if( green<=red && green<=blue ) *Satu=1-3*green;
if( blue<=red && blue<=green ) *Satu=1-3*blue;
}
}

```

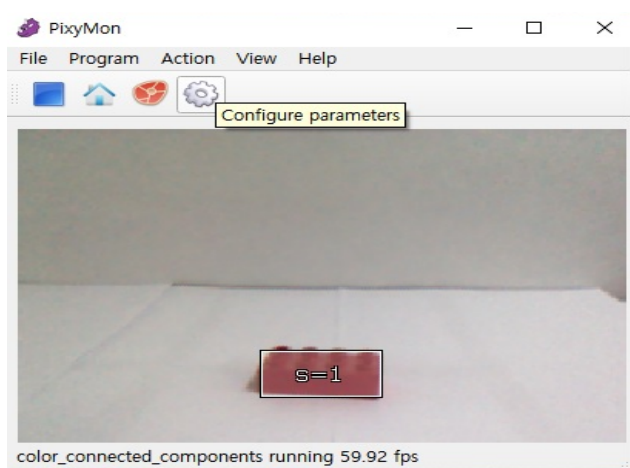
Figure 6. RGB to HSI conversion C-Code.

3.2. Color Code

The vision sensor used in this paper can index seven arbitrarily specified colors. The proposed color code is a method for recognizing adjacent specified colors as a single code. Figure 7 shows the setting screen for indexing seven colors that can be saved in the vision sensor. As shown in Figure 7, after indexing red in color 1, orange in color 2, yellow in color 3, blue in color 4, green in color 5, pink in color 6, and purple in color 7, it can be seen that adjacent blocks of seven colors are recognized as a single object.



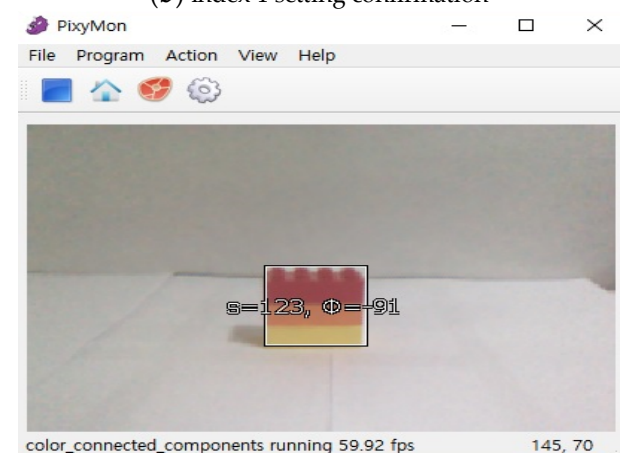
(a) index 1 setting



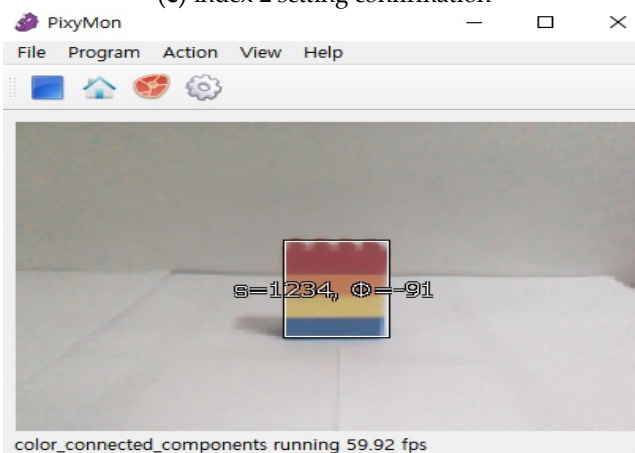
(b) index 1 setting confirmation



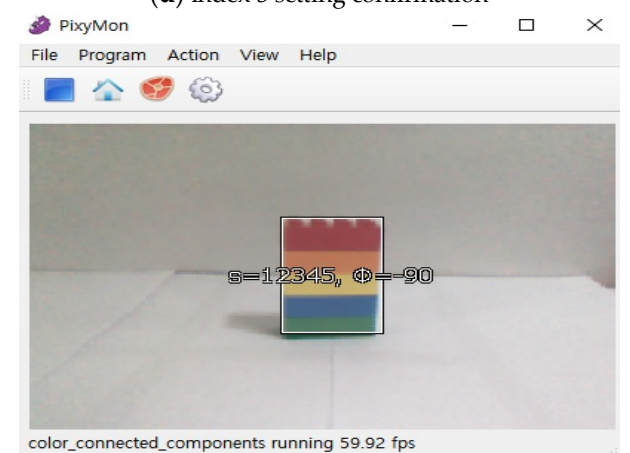
(c) index 2 setting confirmation



(d) index 3 setting confirmation

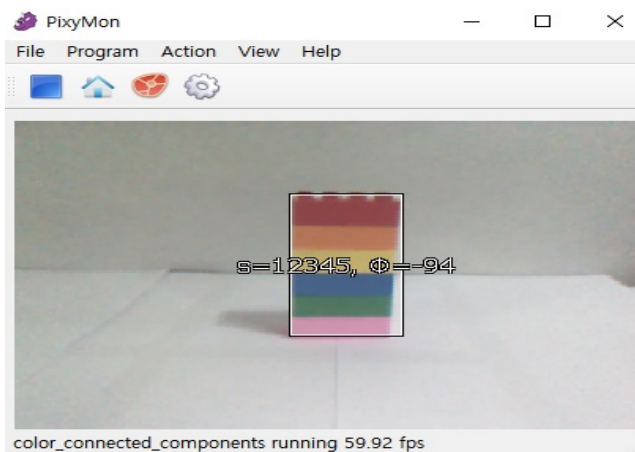


(e) index 5 setting confirmation

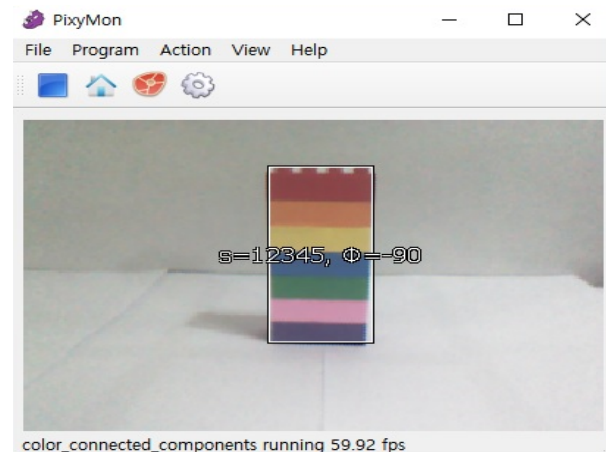


(f) index 6 setting confirmation

Figure 7. Cont.



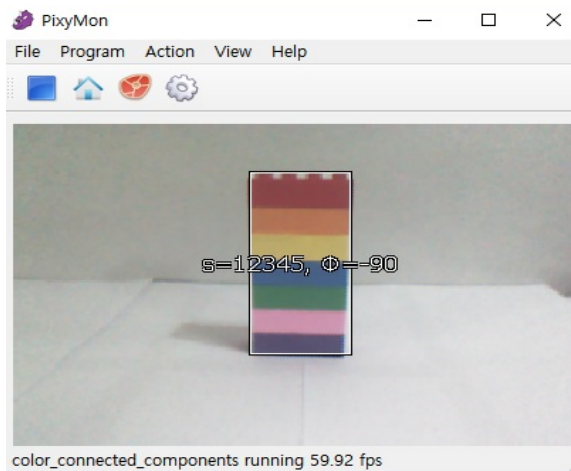
(g) index 6 setting confirmation



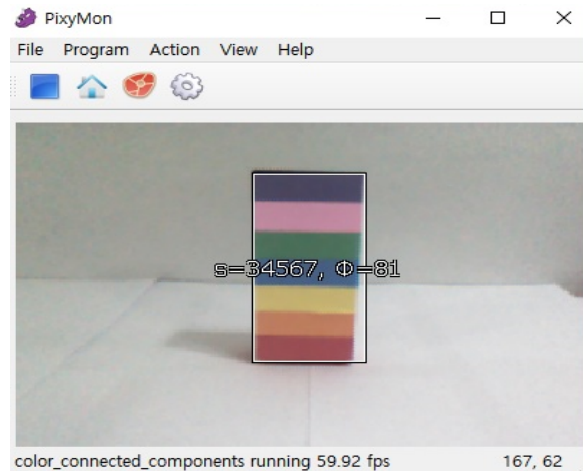
(h) index 7 setting confirmation

Figure 7. Color index settings screen.

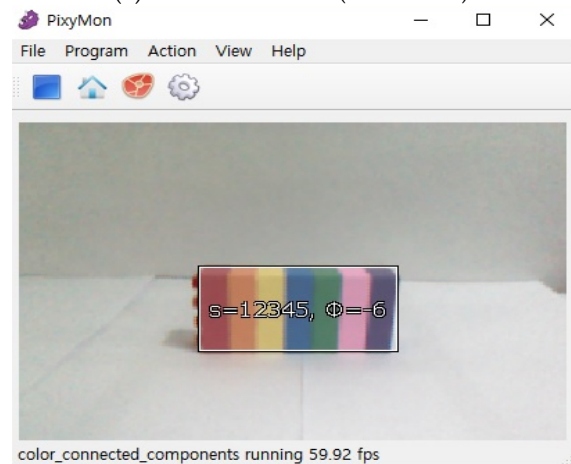
Figure 8 shows the recognition results after the indexed color code changes in various ways. For example, as shown in Figure 8a–d, if the order of the color codes changes in an up-and-down or left-and-right manner, while the colors remain adjacent in the order of the indexed color codes, it is recognized as the same code. However, as shown in Figure 8e–h, if the order of the color code is changed, the number of color codes is changed, or the color code is duplicated, it can be seen that it is recognized as a different color code.



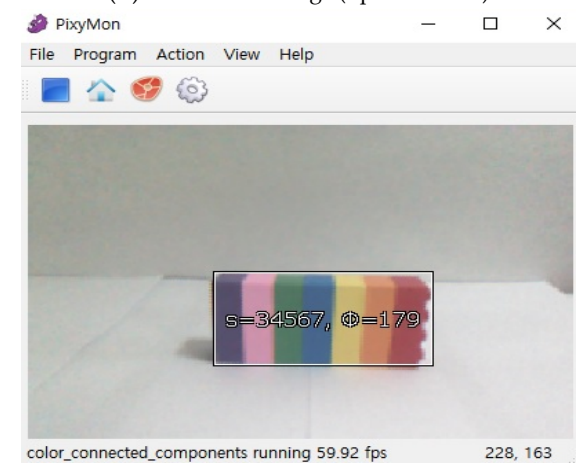
(a) code confirmation (seven color)



(b) code order change (up and down)



(c) code order change (left turn)



(d) code order change (right turn)

Figure 8. Cont.

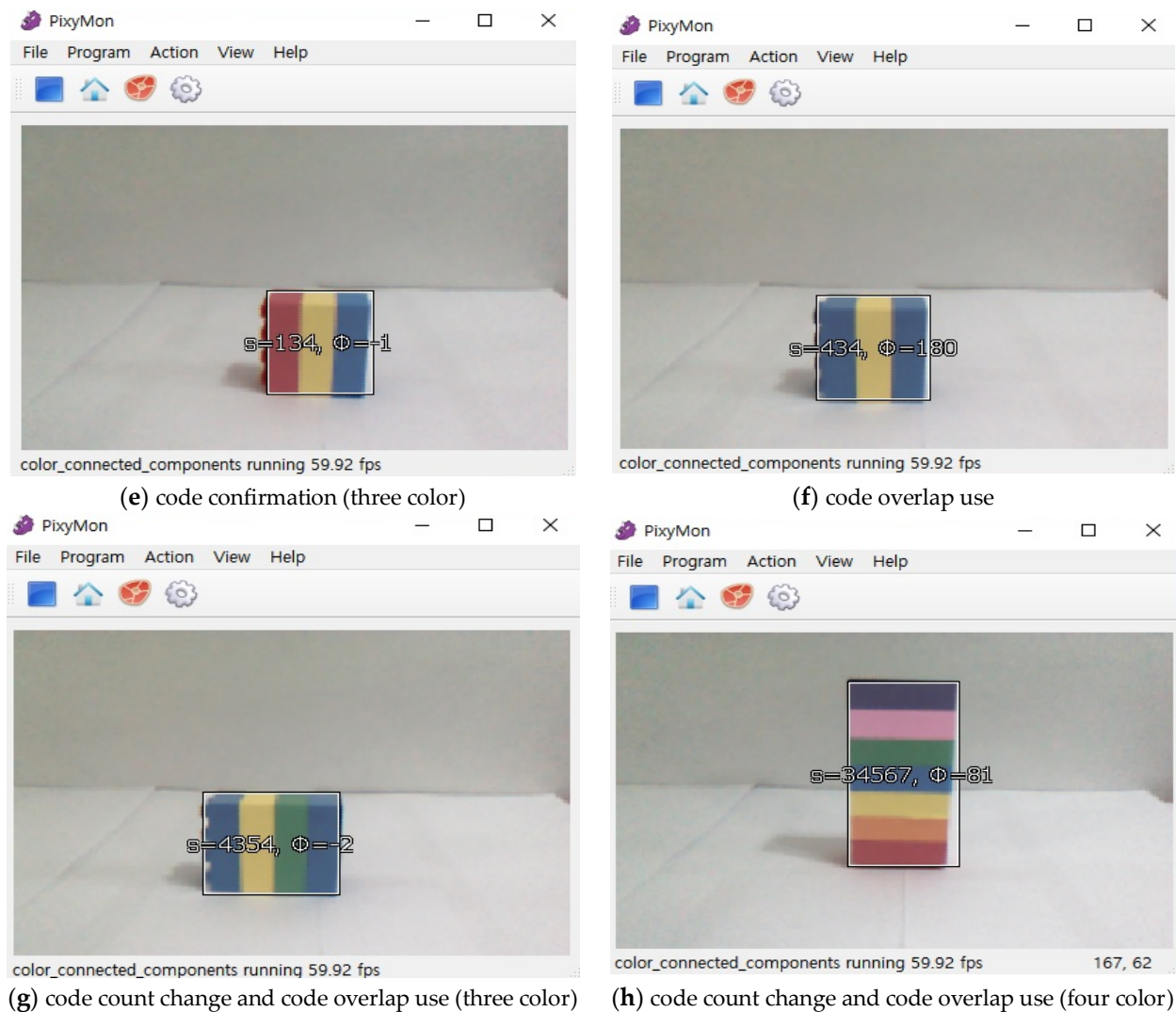


Figure 8. Color code recognition test.

The proposed color codes can vary by using seven arbitrarily designated colors. For example, if a color code in which multiple colors are adjacent is generated after indexing an arbitrary color, as shown in Table 1, 42 color codes (7×6) can be generated with color codes in which 2 colors are adjacent, and 252 color codes ($7 \times 6 \times 6$) can be generated with color codes in which 3 colors are adjacent. In this way, innumerable color codes can be generated depending on the number of adjacent colors. Furthermore, among the generated color codes, those with the same left and right values are recognized as the same code. Therefore, if duplicate codes are removed, 12 codes can be generated using 2 color codes, and 133 codes can be generated using 3 color codes. By using color codes generated in this way, it is possible to allow the AGVA-Robot to execute the traveling command related to the direction at junctions and the desired command based on branch points. For example, several different traveling instructions may be implemented, up to the number of color codes generated, in a variety of work environments, such as going straight for one meter from a branch point with a color code and then placing the object on the left, or turning left at a branch point and then placing the object on the next branch point.

Table 1. Index color code.

Index Number	Color 0	Color 1	Color 2	...	Color N
Index 1	red	red	Gold	...	random
Index 2	orange	blue	Aqua	...	random
Index 3	yellow	green	Brown	...	random
Index 4	blue	orange	Gray	...	random
Index 5	green	purple	Magenta	...	Random
Index 6	pink	yellow	Navy	...	random
Index 7	purple	pink	Pick	...	random

4. Experimental Results

The test of the AGVA-Robot was conducted in consideration of straight lines, curved lines, intersections, branch points, and confluences of the guidance line. Figure 9 shows the results of the test on the traveling algorithm of the AGVA-Robot. The following figures show the test result values at each point: Figure 9a at the three-way T-shaped intersection; Figure 9b at the intersection with a curve; Figure 9c at a plane intersection; and Figure 9d on the road with the angle.

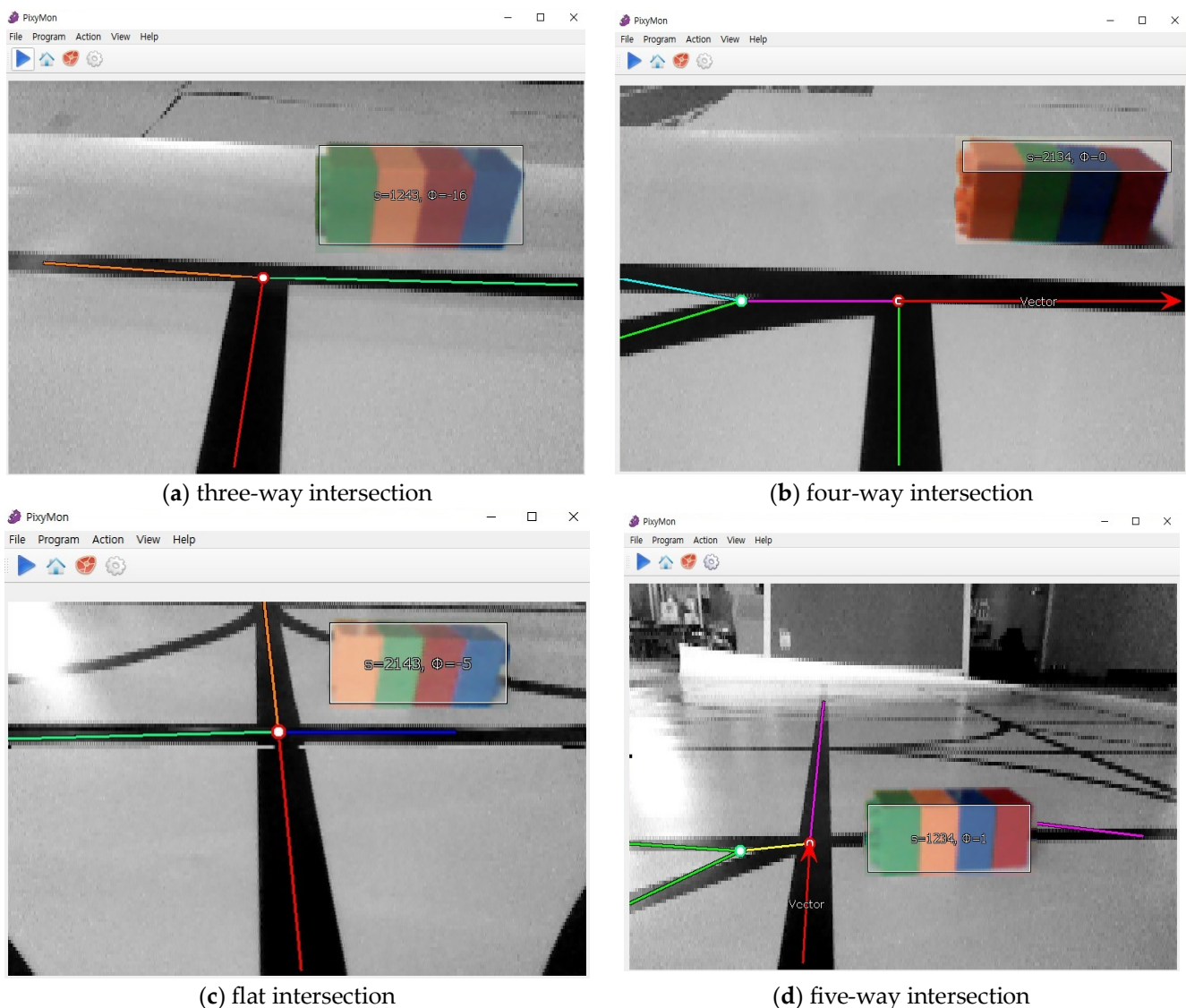
**Figure 9.** Experimental result of the driving algorithm of the AGVA-Robot.

Figure 10 shows the test results from the traveling algorithm of the AGV using a color code. Figure 10a–d shows the results of the test in which color codes were typically generated with three, four, five, and seven colors. Figure 10a'–d' shows the results of the test when the color code generated using the duplicate code is recognized. This paper proposed an AGV traveling system that uses color codes based on color recognition by a vision sensor. The proposed system extracted images by receiving images via an inexpensive, high-speed vision sensor. The algorithm for recognizing lines in the extracted image used the Otsu method, and to recognize the color code, RGB to HSI conversion was used to obtain an H value that was robust in color, facilitating recognition. After indexing the recognized colors on the vision sensor board, we generated color codes in accordance with the order of adjacent colors of the indexed colors and used them for traveling commands. As a result of the test manufactured to test the algorithm of the proposed traveling system using the AGVA-Robot, it was confirmed that the proposed traveling algorithm of the AGVA-Robot works normally.

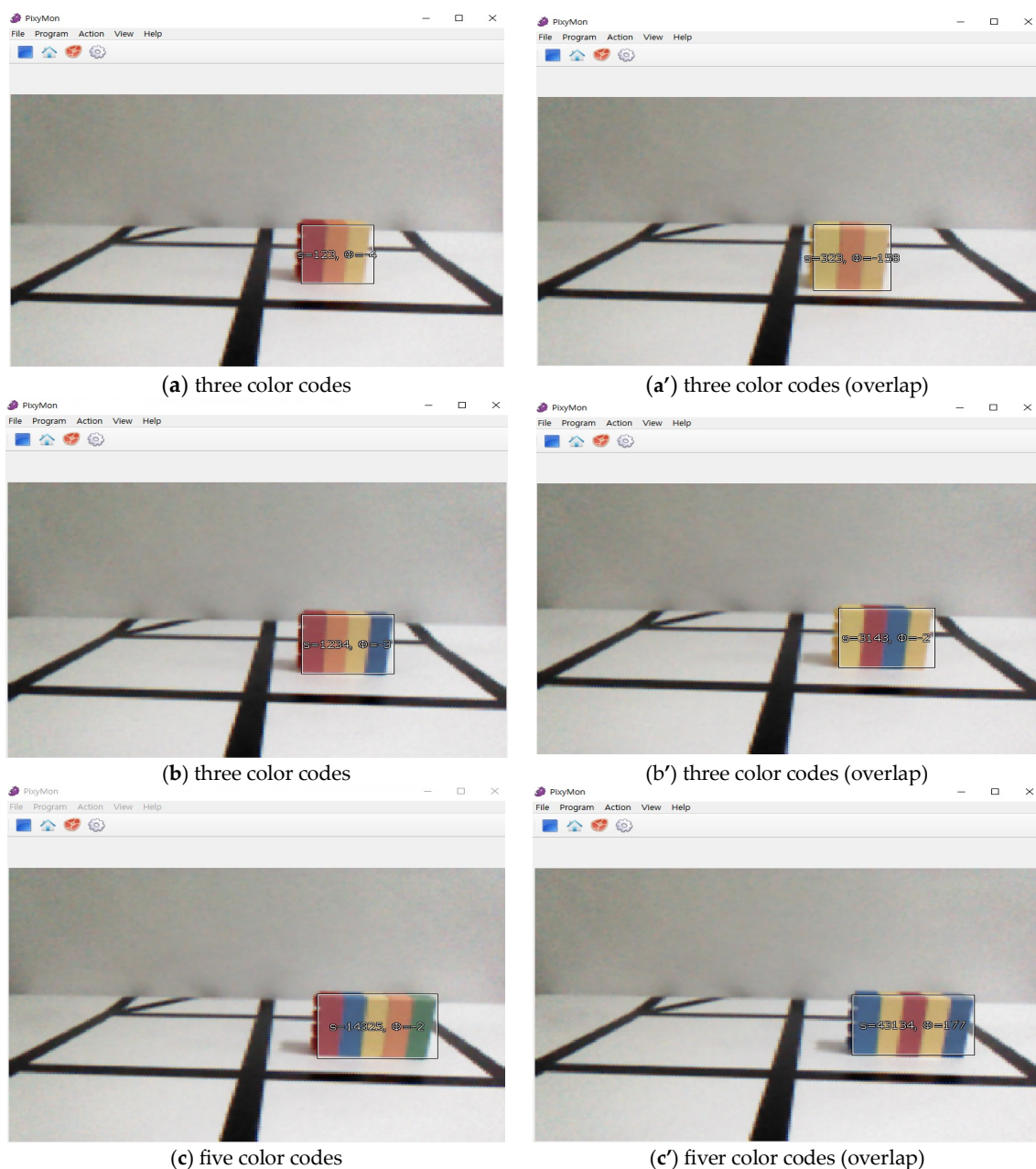


Figure 10. Cont.



Figure 10. AGVA-Robot driving command using color codes.

5. Conclusions

The facts we learned by applying the traveling algorithm using the proposed color code to the AGVA-Robot are as follows: Conventional AGVs recognize lines mainly by using guidance lines that use magnetic fields, while the AGVA-Robot applied with a proposed algorithm recognizes lines using a vision sensor. As a result, conventional AGVs travel only along designated guidance lines, and when the direction of travel at the branch point needs to be changed, the guidance line itself needs to be changed, causing time delays and cost overrun. However, since the proposed AGVA-Robot recognizes traveling commands by using color codes photographed by a vision sensor, traveling commands can be easily changed at each branch point by changing the color codes at the branch point. In addition, the color code can be used not only to inform the direction of travel at branch points but also to execute various commands in addition to following the direction of travel at branch points.

Improving the recognition rate of color codes in various environments and the recognition rate of color codes by the travel speed of the AGV by applying it to industrial sites will be a task to be addressed in future research. It is necessary to study an algorithm that can increase the recognition rate by checking the accuracy of color code recognition with regard to various lighting conditions in the actual field. In addition, research on an algorithm that can adjust the driving speed based on how much the color code recognition rate changes according to the driving speed of the AGV should be further conducted.

Author Contributions: Conceptualization, J.-Y.J., S.-J.Y. and C.-H.L.; methodology, J.-Y.J., S.-J.Y. and C.-H.L.; software, J.-Y.J. and S.-J.Y.; hardware, J.-Y.J. and S.-J.Y.; validation, J.-Y.J., S.-J.Y. and C.-H.L.; formal analysis, J.-Y.J., S.-J.Y. and C.-H.L.; writing—original draft preparation, J.-Y.J.; writing—review and editing, J.-Y.J., S.-J.Y. and C.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Vis, I.A. Survey of research in the design and control of automated guided vehicle systems. *Eur. J. Oper. Res.* **2006**, *170*, 677–709. [\[CrossRef\]](#)
- Le-Anh, T.; De Koster, M.B.M. A review of design and control of automated guided vehicle system. *Eur. J. Oper. Res.* **2006**, *171*, 1–23. [\[CrossRef\]](#)
- Qui, L.; Hsu, W.J.; Huang, S.-Y.; Wang, H. Scheduling and routing algorithms of AGVs: A survey. *Int. J. Prod. Res.* **2002**, *40*, 745–760. [\[CrossRef\]](#)
- Ho, T.-C. A dynamic-zone strategy for vehicle collision prevention and load balancing in an AGV system with a single-loop guide path. *Comput. Ind.* **2000**, *42*, 159–176. [\[CrossRef\]](#)
- Kim, S.H.; Hwang, H.; Kim, Y.-D.; Hahn, K.H. Development of operating rules for automated guided vehicle systems in heterarchical manufacturing system. *J. Korean Inst. Ind. Eng.* **1997**, *2*, 343–357.
- Moorthy, R.L.; Hock-Guan, W.; Wing-Cheong, N.; Chung-Piaw, T. Cyclic deadlock prediction and avoidance for zone-controlled AGV system. *Int. J. Prod. Econ.* **2003**, *83*, 309–324. [\[CrossRef\]](#)
- Borenstein, J. The OmniMate: A guidewire and beacon-free AGV for highly reconfigurable applications. *Int. J. Prod. Res.* **2000**, *38*, 1993–2010. [\[CrossRef\]](#)
- Caruso, M.J.; Smith, C.H.; Bratland, T.; Schneider, R. A new perspective on magnetic field sensing. *Sensors* **1998**, *15*, 34–46.
- Chan, C.-Y.; Tan, H.-S. *Evaluation of Magnetic as a Position Reference System for Ground Vehicle Guidance and Control*; California PATH Research Report, UCB-ITS-PRR-2003-8; Institute of Transportation Studies, UC Berkeley: Berkeley, CA, USA, 2003.
- Jing, L.; Yang, P. A Localization Algorithm for Mobile Robots in RFID System. In Proceedings of the 2007 International Conference on Wireless Communications, Shanghai, China, 21–25 September 2007; pp. 2109–2112. [\[CrossRef\]](#)
- Want, R.; Hopper, A.; Falcao, V.; Gibbons, J. The active badge location system. *ACM Trans. Inf. Sys.* **1992**, *10*, 91–102. [\[CrossRef\]](#)
- Priyantha, N.B.; Chakraborty, A.; Balakrishnan, H. The cricket location-support system. In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston, MA, USA, 6–11 August 2000; pp. 32–43. [\[CrossRef\]](#)
- Gezici, S.; Tian, Z.; Giannakis, G.B.; Kobayashi, H.; Molisch, A.F.; Poor, H.V.; Sahinoglu, Z. Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks. *IEEE Signal Process. Mag.* **2005**, *22*, 70–84. [\[CrossRef\]](#)
- Jung, K.-H.; Kim, J.-M.; Park, J.-J.; Kim, S.-S.; Bae, S.-I. Line tracking method of AGV using Sensor Fusion. *J. Korean Inst. Intell. Syst.* **2010**, *20*, 54–59. [\[CrossRef\]](#)
- Heo, S.W.; Park, T.-H. Localization system for AGVs using laser scanner and marker sensor. *J. Inst. Control Robot. Syst.* **2017**, *23*, 866–872. [\[CrossRef\]](#)
- Yang, K.-M.; Gwak, D.-G.; Han, J.-B.; Hahm, J.H.; Seo, K.-H. A study on position estimation of movable marker for localization and environment visualization. *J. Korea Robot. Soc.* **2020**, *15*, 357–364. [\[CrossRef\]](#)
- Choi, B.-H.; Kim, B.-S.; Kim, E.-T. Location estimation and obstacle tracking using laser scanner for indoor mobile robots. *J. Korean Inst. Intell. Syst.* **2011**, *21*, 329–334. [\[CrossRef\]](#)
- Kawano, T.; Hara, M.; Sugisaka, M. Generating target path for tracing a line before missing the traced line of dead angle of camera. In Proceedings of the 2006 SICE-ICASE International Joint Conference, Busan, Republic of Korea, 18–21 October 2006; pp. 5286–5289. [\[CrossRef\]](#)
- Lee, J.-H.; Jung, K.-H.; Kim, J.-M.; Kim, S.-S. Sensor fusion of localization using Unscented Kalman Filter. *J. Korean Inst. Intell. Syst.* **2011**, *21*, 667–672. [\[CrossRef\]](#)
- Beccari, G.; Caselli, S.; Zanichelli, F.; Calafiore, A. Vision-based line tracking and navigation in structured environments. In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', Monterey, CA, USA, 10–11 July 1997; pp. 406–411. [\[CrossRef\]](#)
- Man, Z.G.; Ye, W.H.; Zhao, P.; Lou, P.H.; Wu, T.J. Research on RFID and vision based AGV navigation. *Adv. Mat. Res.* **2010**, *136*, 298–302. [\[CrossRef\]](#)
- Jang, J.Y.; In, C.H. Design and Implementation of AGV-UNO-CAR Using a Line Scan Algorithm. *J. Korean Inst. Commun. Inf. Sci.* **2021**, *46*, 1346–1354. [\[CrossRef\]](#)
- Kim, S.H.; Lee, H.G. Implementation of Pattern Recognition Algorithm Using Line Scan Camera for Recognition of Path and Location of AGV. *J. Korea Indust. Inf. Sci.* **2018**, *23*, 13–21. [\[CrossRef\]](#)
- Lee, G.W.; Lee, H.; Cheong, H.W. Object Detection of AGV in Manufacturing Plants using Deep Learning. *J. Korea Inst. Commun. Eng. Sci.* **2021**, *25*, 36–43. [\[CrossRef\]](#)
- Kim, C.M.; Cho, H.Y.; Yun, T.S.; Shin, H.J.; Park, H.K. RFID-based Shortest Time Algorithm linetracer. *J. Korea Inst. Elec. Commun. Sci.* **2022**, *17*, 1221–1228. [\[CrossRef\]](#)
- Arduino.cc. Available online: <https://docs.arduino.cc/hardware/uno-rev3> (accessed on 14 August 2021).
- Pixy2 Camera. Available online: <https://dronebotworkshop.com/pixy2-camera/> (accessed on 23 July 2021).
- Motor Driver Module-L298N. Available online: http://wiki.sunfounder.cc/index.php?title=Motor_Driver_Module-L298N (accessed on 23 July 2021).
- Byun, S.; Kim, M. A vision based guideline interpretation technique for AGV navigation. *J. Korea Multimed. Soc.* **2012**, *15*, 1319–1329. [\[CrossRef\]](#)

30. Kim, M.H.; Byun, S. A guideline tracing technique based on a virtual tracing wheel for effective navigation of vision-based AGVs. *J. Korea Multimed. Soc.* **2016**, *19*, 539–547. [[CrossRef](#)]
31. Gonzales, R.C.; Woods, R.E.; Eddins, S.L. *Digital Image Processing Using MATLAB*; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2004.
32. Cheng, H.D.; Jiang, X.H.; Sun, Y.; Wang, J. Color image segmentation: Advances and prospects. *Pattern Recognit.* **2001**, *34*, 2259–2281. [[CrossRef](#)]
33. Lee, J.S. Velocity measurement of fast moving object for traffic information acquisition. *J. Korean Inst. Commun. Inf. Sci.* **2004**, *29*, 1527–1540.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.