Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

A Study on Object Detection and Tracking of a Mobile Robot Using $CIE\ L^*a^*b^*$ Color Space

Gokhan ATALI [a,*], Meltem EYUBOGLU [a]

[a] *Department of Mechatronics Engineering, Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, TURKEY*

** Corresponding author e-mail: gatali@subu.edu.tr*

DOI: 10.29130/dubited.1109850

ABSTRACT

Autonomous vehicles are increasingly used in daily life and industrial applications. Mobile robot technologies lead to autonomous architectures in these areas. The path planning methods of mobile robots contain differences in the purpose they realize. This trajectory planning from a determined starting point to the target point brings many techniques from image processing to artificial intelligence. In the study, an application with a unique design has been carried out on the tracking of circular objects with different diameters and colors by a mobile robot. Moving object is detected with $CIE\ L^*a^*b^*$ color space with RGB-D camera by utilizing the ROS server-client architecture. The mobile robot tracks the detected object at a certain distance at a constant speed. Image filtering parameters are processed by the mobile robot in the Matlab environment together with the publisher-subscriber parameters. Thus, two circular objects with different colors, detected because of image processing and determined beforehand, are continuously followed by the mobile robot at a certain speed. Experiments were carried out using different diameter, size tolerance and color parameters in the image depending on the $CIE\ L^*a^*b^*$ color space.

Keywords: *ROS, Image Processing, Object Tracking, Mobile Robot, Moving Object Detection*

$CIE\ L^*a^*b^*$ Renk Uzayı Kullanan Mobil Robotun Nesne Algılama ve Takibi Üzerine Bir Çalışma

Öz

Otonom araçlar günlük yaşamda ve endüstriyel uygulamalarda giderek daha fazla kullanılmaktadır. Mobil robot teknolojileri bu alanlarda otonom mimarilere öncülük etmektedir. Mobil robotların yol planlama yöntemleri, gerçekleştirdikleri amaca yönelik farklılıklar içermektedir. Belirlenen bir başlangıç noktasından hedef noktaya kadar olan bu yörünge planlaması temelde görüntü işlemeden yapay zekaya kadar birçok tekniği beraberinde getirmektedir. Çalışmada, farklı çap ve renklerde dairesel nesnelerin mobil bir robot tarafından takibi konusunda özgün tasarımlı bir uygulama gerçekleştirilmiştir. ROS sunucu-istemci mimarisi kullanılarak RGB-D kamera ile $CIE\ L^*a^*b^*$ renk uzayı ile hareketli nesne algılanır. Mobil robot, algılanan nesneyi belirli bir mesafede sabit bir hızla takip eder. Görüntü filtreleme parametreleri, yayıncı abone parametreleri ile Matlab ortamında mobil robot tarafından işlenir. Böylece görüntü işleme sonucunda algılanan ve önceden belirlenen farklı renklerde iki dairesel nesne, mobil robot tarafından belirli bir hızda sürekli olarak takip edilmektedir. $CIE\ L^*a^*b^*$ renk uzayına bağlı olarak görüntüde farklı çap, boyut toleransı ve renk parametreleri kullanılarak deneyler yapılmıştır.

Anahtar Kelimeler: *ROS,Görüntü İşleme, Nesne Takibi, Mobil Robot, Hareketli Nesne Algılama*

Geliş: 27/04/2022, Düzeltme: 15/07/2022, Kabul: 12/10/2022

I. INTRODUCTION

Detection of moving objects or people has been a highly studied subject in the field of image processing. Although this issue is basically done using static libraries, it is also enriched with new-generation solutions with developing algorithms and AI technologies. The item whose movement is to be perceived can be an object or a person. There are various methodologies for both of them in the literature.

Munoz-Salinas et al. developed a system that can visually detect and monitor more than one person by using a stereo camera fixed on a system at the under-head position. The researchers performed a person tracking by combining information about each person's skin color model and estimated location by using the Kalman Filter. Thus, the processing time and errors in the tracking system that occur using only location information were greatly reduced. The researchers also added the ability to transfer people who were not predefined in the image. In this way, the system can continue to monitor the people it detects while searching for more people. [1]. Treptow et al. proposed a visual-based approach to track people with a mobile robot using thermal imagery. In the study, researchers used two fast alternative methods based on particle filters [2]. In their study, Munaro et al. presented a multi-person tracking algorithm that can perform fast analysis for ROS-based mobile robots with RGB-D sensors. The proposed algorithm includes a depth-based method specially designed to detect people in clustered human populations [3]. Martin et al. introduced a human detection and tracking system by developing a probability-based algorithm with many functions using different sensory systems of their mobile interaction robot with HOROS [4]. Joshi and Thakore studied the detection of moving objects in a video surveillance system and then tracked the objects detected on the stage. They have used various methods such as Eigen background determination, statistical, alpha, and temporary contour difference methods to detect the moving object [5]. Luo et al. presented a data fusion modeling methodology to detect and track people. The researchers used Jacobian transformations during the action planning phase and obtained their results practically in their studies [6]. Jung and Sukhatme studied a powerful real-time software capable of detecting objects in outdoor environments with a single camera, and in their study, they predicted the positions of moving objects using an adaptive particle filter and Expectation-Maximization algorithm [7]. Markovic et al. developed a new method based on the unit sphere for detecting, tracking, and following objects with a 360º camera they placed on the mobile robot. In addition, the researchers used odometry-based displacement information and a spherical projection model in their studies [8]. Yokoyama and Poggio have applied the detection and tracking of objects in a study using two different methods, gradient-based optical, flow, and edge detection, using lines they detected [9]. Wu and Sun proposed a filter-based algorithm extended with covariance to solve the problem of object tracking motion in an unknown environment [10]. Badar et al. present approach to identify the target person in a crowded environment and Bellotto and Hu sought a solution to the human tracking problem with a mobile robot using sensor data fusion techniques [11, 12]. Zou and Ge have proposed two new algorithms. They have extracted the color image by using CIE Lab Space. Then they have proposed a comparison method through the collection of two image samples. Researchers have recommended their work for apple picking processes [13]. In recent years, the subject of object detection and tracking continues to be studied using deep learning, fuzzy logic, evolutionary algorithms, and hybrid algorithms [14, 15, 16, 17].

When the studies in the literature are examined, it is seen that traditional color spaces are widely used for the detection of moving objects. However, it was emphasized in the study that there was a lot of noise on the image, especially in the studies have been carried out in the HSV color space. In this context, the manipulation of a differential mobile robot working based on ROS by $CIE\ L^*a^*b^*$ color space has been presented with experimental studies.

II. MATERIAL and METHOD

A. ROS OPERATING SYSTEM and MOBILE ROBOT

ROS (Robot Operating System) is a robot operating system commonly used in experimental and industrial areas worldwide. It is specially designed for robotic platforms and provides many robotic tools, hardware simulation, and message transfer between various software nodes. Nodes in this operating system working with broadcast topology can operate independently or with one-to-many subscriber models. It also supports many network protocols such as TCP/IP and SSH.

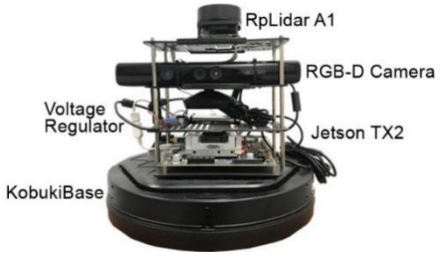

Figure 1. *Mobile robot platform used in the study*

In the study, a uniquely designed hardware system was created on the Kobuki robot platform to apply the software developed for the ROS operating system (Fig. 1). Real-time data obtained from the RGB-D camera attached to the robot platform was transferred to the motor drive unit via ROS nodes and object tracking was performed. For the object to be tracked from a certain distance at a constant speed, the ROS master publishes to speed nodes.

B. KINEMATIC MODEL

The mobility of mobile robot platforms is of great importance in limited working areas. The energy need of the robot can be determined by performing the necessary kinematic and dynamic analysis. The motion of the mechanical system is analyzed by getting the kinematic model of the mobile robot and ignoring the forces affecting the motion.

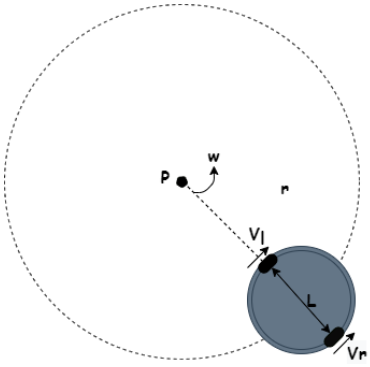

Figure 2. *Kinematic position model of the mobile robot*

The linear speed of each differential wheel affects the rotation of the mobile robot around the point shown in Fig. 2. Therefore, the linear speed of each wheel is calculated and averaged to calculate the linear speed of a mobile robot rotating around the P point shown in Fig. 2. Consequently, the equations given in Eq. 1 are used to calculate the average linear velocity. In addition, the angular velocity equation

given in Eq. 2 can be formed with the linear speeds of the wheels. The Eq. 1 and Eq. 2 of the kinematic model of the robot are given in Table 1.

$$v = \frac{v_r + v_l}{2} = r\frac{(\varphi_r + \varphi_l)}{2} \tag{1}$$

$$\omega = \frac{v_r - v_l}{2L} = r\frac{(\varphi_r - \varphi_l)}{2} \tag{2}$$

Table 1. *Table of abbreviations for the robot's kinematic model*

Symbol	Quantity	SI
v	Linear speed of the robot	m/s
ω	Angular velocity of the robot	rad
v_r	Linear speed of the right wheel	m/s
v_l	Linear speed of the left wheel	m/s
φ_r	Angular velocity of the right wheel	rad
φ_l	Angular velocity of the left wheel	rad
r	The radius of the center of rotation	m
L	Distance between right and left wheel	m
θ	Direction angle of the robot	rad
q	State matrix of the robot in the basic frame	m/s
T	Rotation matrix	-

Here, the base state matrix of the mobile robot in the base frame is given in Eq. 3 and the rotation matrix expressing the orientation of the base frame related to the motion is given in Eq. 4.

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{3}$$

$$T(\theta) = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The kinematic model of the robot frame is given in Eq. 5 by using the matrices given in Eq. 3 and 4.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{-L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \varphi_r \\ \varphi_l \end{bmatrix} \tag{5}$$

C. DYNAMIC MODEL

The dynamic model of the mobile robot platform with differential wheels is created by the Newton-Euler method. During the motion of the robot platform, a force acts on the front and back casters in the opposite direction of the movement. However, it is not taken into consideration as it has a very small effect. The forces acting on the mobile robot platform are shown in the force diagram in Fig. 3.

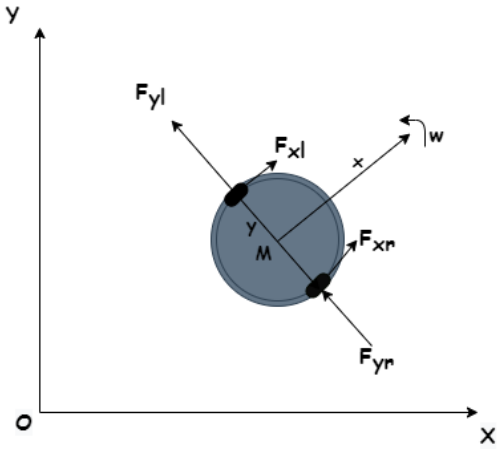

Figure 3. *Force diagram of a mobile robot*

Equations of motion for the x and y coordinates giving the connection between the velocity and coordinates required for the mobile robot are expressed in Eq. 6. The abbreviations expressing the dynamic model of the robot are given in Table 2.

$$\begin{cases} \dot{X} = v_x \cos\theta - v_y \sin\theta \\ \dot{Y} = v_x \sin\theta + v_y \cos\theta \end{cases} \tag{6}$$

Table 2. *Table of abbreviations for the dynamic model of the robot*

Symbol	Quantity	SI
v	Longitudinal Velocity Vector	m/s
ω	Lateral Velocity Vector	m/s
θ	Heading Angle	rad

A function of lateral velocity, longitudinal velocity, and heading angle velocity is shown in Eq. 7.

$$\dot{v}_y = f(v_x, \omega) \tag{7}$$

The dynamic model of the mobile robot is given in the equations in Eq. 8, since the robot may have a deviation from the longitudinal motion vector as it may be $\dot{\theta} = \omega$ considering the differences that may occur in wheel heading angles.

$$\begin{cases} \dot{X} = v_x \cos\theta - v_y \sin\theta \\ \dot{Y} = v_x \sin\theta + v_y \cos\theta \\ \dot{\theta} = \omega \\ \dot{v}_y = f(v_x, \omega) \end{cases} \tag{8}$$

D. OBJECT DETECTION COLOR SPACE

There are different color space modes to detect the objects by separating them from their environment. One of the most widely used color spaces of these modes is $CIE\ L^*a^*b^*$. When the studies using this color space were analyzed, it was seen that various algorithms were developed especially for color image segmentation [17,18]. This color space is widely used to distinguish the object that needs to be detected from other images outside the object [19, 20, 21, 22].

The object with the desired color and tone can be separated from other objects in the environment by using these separation modes in gray or RGB tones. In this study, $CIE\ L^*a^*b^*$ color space that shows smooth changes in detection and is successful in removing artifacts in the images was used. This color

space, in which all colors perceived by the human eye are defined, is used as a standard color space for various areas. The 3-dimensional color coordinates in the $CIE\ L^*a^*b^*$ color space are given in Fig. 4.

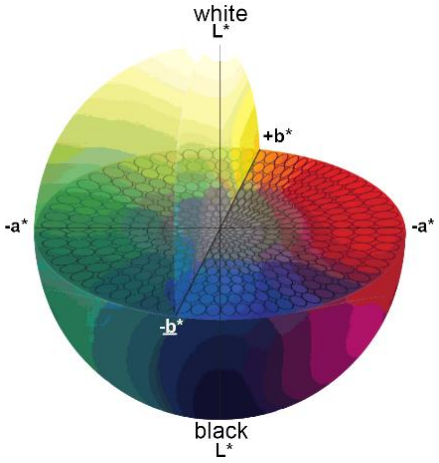

Figure 4. *$CIE\ L^*a^*b^*$ color space*

L^* - Lightness coordinate: It indicates the lightness from 0 (black) to 100 (white)
a^* - The coordinate from green to red: $-a^*$ represents green $+a^*$ represents red
b^* - The coordinate from blue to yellow: $-b^*$ represents blue $+b^*$ represents yellow

The forward transformation of the $CIE\ L^*a^*b^*$ color space is given in Eq. 9, 10 and 11. Here X_n, Y_n and Z_n are the CIE XYZ tristimulus values of the reference white point.

$$L^* = 116\, f\left(\frac{Y}{Y_n}\right) - 16 \tag{9}$$

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \tag{10}$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \tag{11}$$

The red circle in the image taken from the RGB-D camera on the robot is detected in the $CIE\ L^*a^*b^*$ color space and taken into the contour as shown in Fig. 5. The mobile robot platform performs a movement action using the diameter information of this perceived red circle.

(a) ***(b)***

Figure 5. *Object detection using $CIE\ L^*a^*b^*$* ***(a)*** *1.5m* ***(b)*** *4m*

III. EXPERIMENTAL STUDY

A. ACTION PARAMETERS

In the image-based robotic application performed in the study, there are two basic actions while tracking a 165 mm diameter red circle. The first action is detecting the moving object and the second one is tracking it at a determined constant speed at a certain distance. In the study conducted based on the client-server architecture, ROS commands are published using Matlab Robotics System Toolbox over the host computer (Master Station). These ROS commands, which are published wirelessly, are subscribed and processed by the mobile robot platform originally developed in the study. Then, the detection and tracking of the moving object is carried out using the algorithm specified in Fig. 2. The nodes used by the ROS master while performing these two actions are as follows;

- /mobile_base/commands/sound
- /mobile_base/commands/velocity
- /camera/rgb/image_raw
- /mobile_base/events/bumper
- /mobile_base/events/cliff
- /odom
- /rosout

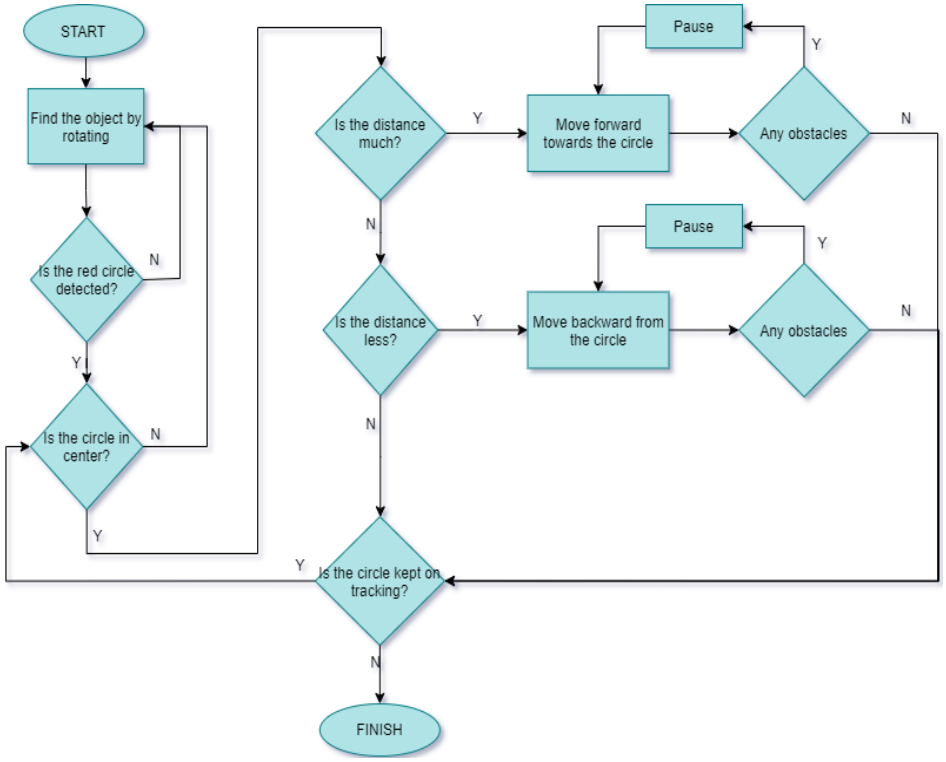

Figure 6. *Moving object detection and tracking flowchart*

The ROS node "/camera/rgb/image_raw" is subscribed with the algorithm tested and regular images are obtained from the RGB-D camera located on the original robot platform. The captured image is processed in the Matlab software in the master station and then velocity commands are sent to the mobile robot platform by the master station with the "mobile_base/commands/velocity" node publish. These operations are carried out according to the flow chart shown in Fig. 6. As it can be seen in this flow chart, first the robot platform rotates and receives images from the RGB-D camera continuously and then processes the data in these images and checks whether an object is present. This process continues

until the object is detected. After the object is detected, the image center and the object are positioned on top of each other. Then, the distance of the robot to this object is determined according to the diameter of the detected circle. If it is not at the desired distance, the distance between the robot and the object is tried to be fixed by giving forward or backward speed. If an obstacle is detected between the object and the mobile robot platform, the mobile robot waits until the obstacle is removed. After the obstacle is removed, the object is detected and the circle is centered with right and left movements, and the circle is followed by giving forward and backward speed. This loop is run continuously until the tracking process is complete. The pseudo-code obtained for the mobile robot according to this working principle was created as in Table 3.

Table 3. *Pseudo code*

```
Start
Connect to robot with IP address
Determine the properties of the object
Receive images from the robot
Find the position of the ball
   For i=1:100
         Receive a new image from the robot
         Calculate the position and size of the ball

         If is empty (location) //Right and left speed control
                  Give angular velocity and search objects by rotating the robot
         Elseif If the location is on the left side of the image
                  Turn the robot left
         Elseif If the location is on the right side of the image
                  Turn the robot right
         End
         If is empty (location) //Forward and backward speed control
                  Give angular velocity and search objects by rotating the robot
         Elseif location if the ball is too small in the image
                  Move the robot forward

         Elseif location if the ball is too big in the image
                  Move the robot backward
         End
   Publish Velocity Commands
   pause( )    Wait
End
```

B. OBJECT TRACKING WITH VELOCITY COMMANDS

The velocity information is published by the ROS master to the differential wheels for tracking the detected 165 mm diameter circle. In the study, an approximately 0.62 rad/s velocity was determined for the angular approach of the mobile robot to the object and approximately 0.2 m/s for its approach in the linear plane. According to the answer of "Is the circle in the center?" in the algorithm in Fig. 6, the right or left motors are moved and this rotational action occurs continuously until the red circle settles in the center of the image (Fig. 7). Then, using the diameter information of the circle image placed in the center, the forward and backward movement of the robot in the linear plane is planned. If the measured diameter is less than the specified value, the movement is carried out in the forward direction and if it is larger, the movement is carried out in the backward direction until the robot settles at a fixed distance (Fig. 8).

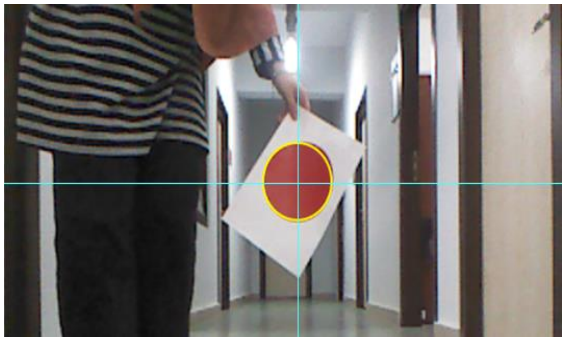

Figure 7. *Centering the red circle*

Figure 8. *Robot's approach to the perceived circle*

C. IMAGE CAPTURING PERFORMANCE TESTS

In the experiments, three different objects in red, green, and blue with different diameters were detected and the results were evaluated. Image capture times were compared as a result of experiments performed in 10 iterations. Thus, the image capture times were analyzed depending on the diameter information of the detected RGB objects. In the experiments, it has been observed that the image capture time in the first iteration had a higher margin of error compared to the results of the later iteration. The most important factor of this time difference in the first iteration is that the process of capturing the first image depends on the environmental scanning operation of the robot. To make a better comparison, image capture times were performed in 10 iterations and these periods are given in Table 4. Standard deviation was calculated using Eq. 12 for the T_{Δ} values given in Table 4, and the propagation sensitivity was found to be $\sigma = 0{,}046628$. In addition, when the T_{Δ} values were examined, the highest performance rate was obtained for 50 pixels in red. As a result of the experimental study, it has been observed that the blue color is a more advantageous color in the $CIE\ L^*a^*b^*$ color space.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{12}$$

Table 4. *Elapsed time for image capture with different diameter*

It.	Red			Green			Blue		
	Time(sec)			Time(sec)			Time(sec)		
	Radii=50	**Radii=70**	**Radii=100**	**Radii=50**	**Radii=70**	**Radii=100**	**Radii=50**	**Radii=70**	**Radii=100**
1	0.021480	0.016298	0.017597	0.020507	0.020373	0.018565	0.016105	0.016661	0.017760
2	0.014228	0.013269	0.014915	0.017983	0.012394	0.013019	0.013502	0.016269	0.013495
3	0.017966	0.014952	0.014448	0.020376	0.018624	0.016540	0.015448	0.014849	0.014104
4	0.023536	0.024912	0.014749	0.014185	0.017831	0.015817	0.015679	0.018030	0.013593
5	0.015470	0.018766	0.020868	0.014923	0.014396	0.015785	0.014012	0.019325	0.014537
6	0.012881	0.017409	0.015053	0.012784	0.011932	0.015347	0.013441	0.016662	0.014482
7	0.013962	0.016374	0.013927	0.016722	0.022311	0.012780	0.015387	0.014826	0.021548
8	0.015592	0.016767	0.012634	0.012429	0.013767	0.014222	0.017574	0.016974	0.020773
9	0.016189	0.014267	0.015635	0.014552	0.013151	0.015404	0.014145	0.016255	0.021786
10	0.016886	0.014612	0.014230	0.015928	0.015017	0.016151	0.015483	0.022815	0.015978
Mean (T_Δ)	15.078×10^{-3}	17.267×10^{-3}	16.806×10^{-3}	16.819×10^{-3}	16.763×10^{-3}	15.406×10^{-3}	16.039×10^{-3}	15.980×10^{-3}	15.363×10^{-3}

The $CIE\ L^*a^*b^*$ color space used in the study was tested under equal ambient conditions with the HSV color space to be evaluated in terms of performance. In Table 5, the image capture times obtained at the end of 10 iterations are presented comparatively for both color spaces. HSV color space is much faster than $CIE\ L^*a^*b^*$ color space in terms of image capture time. However, although this color space has provided an advantage in terms of image capture time, it is seen that the error rate of the HSV color space is high compared to the $CIE\ L^*a^*b^*$ color space when looking at Fig. 9.

Table 5. *Image capture times with $CIE\ L^*a^*b^*$ and HSV color spaces after 11 iterations*

It.	HSV Color Space	$CIE\ L^*a^*b^*$ Color Space
	Time(sec)	Time(sec)
1	1.995237	2.162119
2	0.088626	0.205195
3	0.092294	0.218634
4	0.107291	0.236095
5	0.090008	0.237199
6	0.091152	0.242487
7	0.138311	0.232046
8	0.107701	0.236549
9	0.099754	0.230907
10	0.098666	0.219314
Mean (T_Δ)	0. 290904	0.4220545

The use of HSV color space in terms of image segmentation was not considered appropriate in this study because it provides images that are fast but fuzzy. As a result of the segmentation processes, it was determined that the $CIE\ L^*a^*b^*$ color space is more applicable for the object detection.

(a)

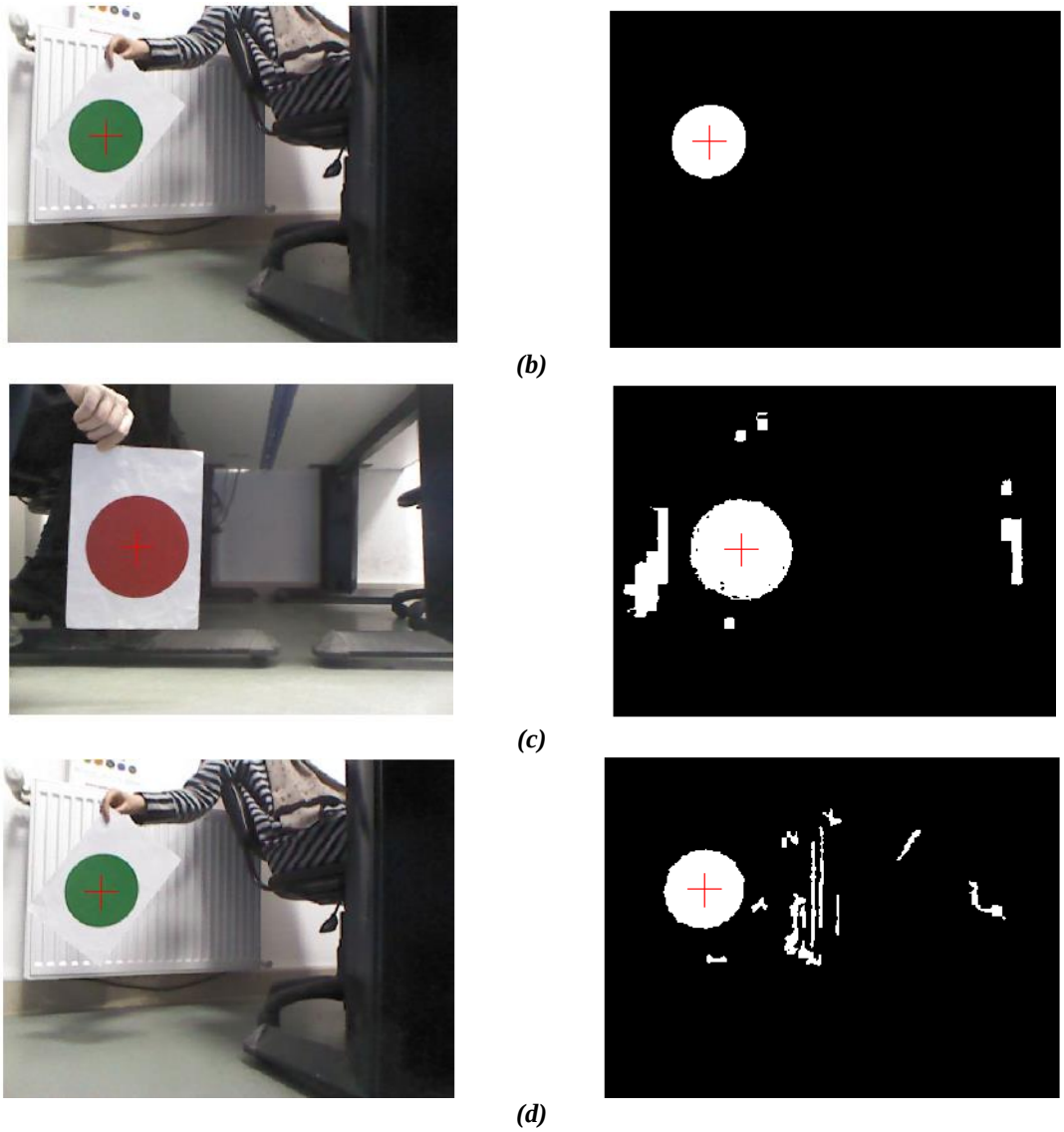

Figure 9. *$CIE\ L^*a^*b^*$ color space binary transformation;* ***(a)*** *red circle,* ***(b)*** *green circle; HSV color space binary transformation* ***(c)*** *red circle,* ***(d)*** *green circle*

It has been observed that size tolerance, which is another factor in performing tracking operations of a moving object, has no effect on the image capture time. Considering the mean (T_Δ) values of image capture times given in Table 6, the difference between the size tolerance value was observed in the acceptable value range in terms of time.

Table 6. *Image capture time at different tolerances in the detection of the object.*

It.	Size Tolerance=5(pixels)	Size Tolerance=20(pixels)
	Time(sec)	Time(sec)
1	1.995237	2.162119
2	0.088626	0.205195
3	0.092294	0.218634
4	0.107291	0.236095
5	0.090008	0.237199
6	0.091152	0.242487
7	0.138311	0.232046
8	0.107701	0.236549
9	0.099754	0.230907
10	0.098666	0.219314
11	0.240140	0.291724

12	0.222898	0.227055
13	0.238374	0.215719
14	0.247309	0.200608
15	0.235898	0.232947
Mean (T_{Δ})	0.366375	0.363777

IV. CONCLUSION

The original image processing-based software developed on the basis of the ROS architecture and performed in the study includes the effective tracking of a specific object. In this software, object detection was performed by taking the $CIE\ L^*a^*b^*$ color space as reference. Later, the environmental noise was removed from the detected object and segmentation process was carried out. After this process, the tracking of the image by maintaining a certain speed and position was carried out by the mobile robot developed in the study. Tests and experiments of robotic study were carried out with a 64bit computer with i7-10750H CPU 2.60 GHz processor and 16 GB RAM.

In the experimental study, tracking of circular objects of two different colors using RGB-D cameras was carried out by a uniquely designed mobile robot. The object detection and tracking process of the mobile robot was carried out using a server-client architecture with an algorithm developed in the ROS environment. While performing these processes, $CIE\ L^*a^*b^*$ color space was used and its advantages over HSV color space were investigated.

In this context, it has been determined that the $CIE\ L^*a^*b^*$ color space gives a clearer result from the noise in the image compared to the HSV color space. However, the HSV color space has been found to be more advantageous than the $CIE\ L^*a^*b^*$ color space in terms of image capture and response time. At the end of the tests, the results of the $CIE\ L^*a^*b^*$ color space has been evaluated according to the image capture and response times on objects of different diameter, size tolerance and color. As a result, it has been determined that the $CIE\ L^*a^*b^*$ color space is superior to detect the object in the image while performing the detection and tracking process of the moving object. In addition, this software, developed with a unique mobile robot platform, has been integrated and object detection and tracking operations has been carried out.

ACKNOWLEDGEMENT: We would like to thank the Robot Technologies and Smart Systems Application and Research Centre (ROTASAM) for providing every opportunity to carry out this study. This study was also supported by Sakarya University of Applied Sciences Scientific Researches Coordinatorship with project number 010-2020.

V. REFERENCES

[1] R. Munoz-Salinas, E. Aguirre and M. Garcia-Silvente, "People detection and tracking using stereo vision and color," *Image and Vision Computing*, pp. 995-1007, 2007.

[2] A. Treptow, G. Cielniak and T. Duckett, "Real-time people tracking for mobile robots using thermal vision," *Robotics and Autonomous Systems*, pp. 729-739, 2006.

[3] M. Munaro, F. Basso and E. Menegatt, "Tracking people within groups with RGB-D data," International Conference on Intelligent Robots and Systems, Vilamoura, Algarve, Portugal, 2012.

[4] C. Martin, E. Schaffernicht and H. M. Gross, "Multi-modal sensor fusion using a probabilistic aggregation scheme forpeople detection and tracking," *Robotics and Autonomous Systems*, pp. 721-728, 2006.

[5] K. A. Joshi and D. G. Thakore, “A Survey on Moving Object Detection and Tracking in Video Surveillance System,” *International Journal of Soft Computing and Engineering (IJSCE)*, pp. 44-48, 2012.

[6] R. C. L. Fellow, Y. J. Chen, C. T. Liao and A. C. Tsai, “Mobile Robot Based Human Detection and Tracking Using Range and Intensity Data Fusion,” 2007 IEEE Workshop on Advanced Robotics and Its Social Impacts, Taiwan, 2007.

[7] B. Jung and G. S. Sukhatme, “Detecting Moving Objects using a Single Camera on a Mobile Robot in an Outdoor Environment,” In the 8th Conference on Intelligent Autonomous Systems, Amsterdam, The Netherlands, 2004.

[8] I. Markovic, F. Chaumette and I. Petrovic, “Moving object detection, tracking and following using an omnidirectional camera on a mobile robot,” IEEE International Conference on Robotics & Automation (ICRA), Hong Kong, China, 2014.

[9] M. Yokoyama and T. Poggio, “A Contour-Based Moving Object Detection and Tracking,” Proceedings 2nd Joint IEEE International Workshop on VS-PETS, Beijing, 2005.

[10] M. Wu and J.-Y. Sun, “Moving Object Detecting and Tracking with Mobile Robot Based on Extended Kalman Filter in Unknown Environment,” International Conference on Machine Vision and Human-machine Interface, 2010.

[11] A. Badar, F. I. Khawaja, A. Yasar and M. Naveed, “Human detection and following by a mobile robot using 3D features,” International Conference on Mechatronics and Autonomation, Takamatsu, Japan, 2013.

[12] N. Bellotto and H. Hu, “Multisensor-Based Human Detection and Tracking for Mobile Service Robots,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 167-181, 2013.

[13] X. Zou and B. Ge, “The Image Recognition of Mobile Robot Based on CIE Lab Space,” I.J. *Information Technology and Computer Science*, pp. 29-35, 2014.

[14] S. Pal, A. Pramanik, J. Maiti, and P. Mitra, “Deep learning in multi-object detection and tracking: state of the art,” *Applied Intelligence*, 2021.

[15] M. Elhoseny, “Multi-object Detection and Tracking (MODT) Machine Learning Model for Real-Time Video Surveillance Systems,” *Circuits, Systems, and Signal Processing*, 2019.

[16] A. Pramanik, S. Pal, J. Maiti and P. Mitra, “Granulated RCNN and Multi-Class Deep SORT for Multi-Object Detection and Tracking,” IEEE Transactions On Emerging Topics In Computational Intelligence, pp. 1-11, 2021.

[17] P. R. Narkhede and A. V. Gokhale, “Color Particle Filter Based Object Tracking using Frame Segmentation in CIELab* and HSV Color Spaces,” IEEE ICCSP 2015 conference, 2015.

[18] A. Mondal, A. Ghosh and S. Ghosh, “Partially Camouflaged Object Tracking using Modified Probabilistic Neural Network and Fuzzy Energy based Active Contour,” *International Journal of Computer Vision*, p. 116–148, 2017.

[19] Y.-C. Li and S.-Y. Huang, “Fast-Moving Object Tracking in Air Hockey,” International Conference on Mechatronics and Automation, Takamatsu, Japan, 2017.

[20] Q. Peng, S.-d. Zhong and L.-f. Tu, “Cast shadow detection for moving objects based on binocular stereo vision,” Journal of Central South University, pp. 651–658, 2014.

[21] S. Sankarasrinivasan, E. Balasubramanian, F. Y. Hsiao and L. J. Yang, “Robust Target Tracking Algorithm for MAV Navigation System,” International Conference on Industrial Instrumentation and Control (ICIC), India, 2015.

[22] H. Kim, W. Chung and Y. Yoo, “Detection and tracking of human legs for a mobile service robot,” International Conference on Advanced Intelligent Mechatronics, Montreal, Canada, 2010.