

# Shopping Robot That Make Real Time Color Tracking Using Image Processing Techniques

Muhammet Fatih Aslan\*<sup>1</sup>, Akif Durdu<sup>2</sup>, Kadir Sabancı<sup>1</sup>

Accepted : 24/09/2017 Published: 30/09/2017

DOI: 10.18100/ijamec.2017331881

**Abstract:** Robots are human-designed, autonomous electro-mechanical devices to perform tasks assigned to them. It is considered that robots will work more efficiently, better quality and more economical than humans. For this reason, the use of robots in different areas is increasing day by day. This work can contribute to Industry 4.0, which is called the revolution of the future. In this study, we developed a service robot for shopping. Thanks to the designed market robot, market owners will be able to offer easier, more enjoyable and faster service to all customers, especially disabled people and elderly people who are shopping. In this work, the images recorded by the high resolution camera are processed in real time and the target tracking is performed. The study consists of two phases; software and hardware. During the hardware phase, the design and implementation of the used robot is done. In practice, omni wheels are used to increase the flexibility of motion of the robot. In the software phase, the image processing software and the designed algorithm are introduced. During the image processing phase, the C ++ software language is used with the OpenCV library. With the designed algorithm, software and hardware, real-time target tracking has been successfully implemented. The tracking of the customer in the markets can be carried out by different methods such as radio frequency (RF), wireless communication. However, this study is important in terms of the first study for a robot of this kind. In this way, new ideas will be revealed.

**Keywords:** C++, Embedded systems, Image processing, Industry 4.0, Omni Wheel, OpenCV, Real Time Target Tracking, Shopping Robot

## 1. Introduction

Technology and science are rapidly advancing and new worlds are being created thanks to new technologies developed. Thanks to digital image processing, real-world images are transferred to computers and used in many industrial applications. Using image processing techniques in the computer and digital image data, existing images can be improved or converted to different images with object recognition and object detection. Image processing is a series of mathematical operations. These processes begin with the capture of the scene, continuing with the use of different techniques to achieve the desired goal [1].

Mobile robots have to move autonomously in their work areas. For this purpose, the cameras provide a visual task for the robots. An image captured by the camera contains a large amount of information about the working area [2]. After the image obtained through the camera is processed, robot motion is provided with the aid of the obtained data. The vision sensors provide information about the environment in which the robot is located. In particular, vision sensors are a great need for robots in an unplanned, unstructured environment. For this reason, image processing is extremely important in robot technology. So far, much research has been done to develop robotic studies based on vision. However, the main difficulty in vision-based work is that traditional image processing algorithms are expensive to calculate or require a powerful processing platform. [3-4].

Digital image processing is used in many applications such as

medicine, biology, geographical science, defense industry, industrial applications, biometric recognition and security systems. In addition to these applications, image processing is now also used in applications involving our social life. Real-time image processing is one of many areas that have seen great improvements in recent years. Applications for this area range from home surveillance (security) applications to military search boats [5]. Robots such as cleaning robots, service robots, agricultural robots are just a few of them. In this work, an autonomous robot is designed to serve people. The designed autonomous shopping robot is capable of real-time image processing and tracking of moving targets.

The services provided by the robots, which can be decided according to the sensors and the algorithms designed, have increased the use of autonomous robots. The decision-making process of such robots depends on their estimation algorithms and are quite complex systems. T. Schmitt and al. performed the co-operation of vision-based robots by making a probable state estimate [6].

Tracking moving objects in the image is one of the important issues in computer vision applications. There are many different applications for motion detection and analysis. For example, the destruction of a moving target in military applications is of great importance for the development of intelligent weapons for national security. Similarly, automatic interpretation of human activities can be achieved through the establishment of a robust and robust system based on vision, with the ability to perceive and follow people [7].

Robotic vision systems need to be used in hard real-time applications. These systems have a certain period of time to carry out the actions they are responsible for and produce the appropriate outputs / reactions. If output takes longer to produce, output

<sup>1</sup> Faculty of Engineering, Electrical and Electronics Engineering, Karamanoglu Mehmetbey University, Karaman, 70000, Turkey

<sup>2</sup> Faculty of Engineering, Electrical and Electronics Engineering, Selcuk University, Konya, 42002, Turkey

\* Corresponding Author: Email: mfatihaslan@kmu.edu.tr

becomes useless and very critical errors can occur. Therefore, the hard real-time requirement for these systems is also valid for the algorithms to be used in these systems. Because the robot must be able to manage the motion system using data obtained by image processing or provide high-level services such as image communication between user terminals and robots. For this reason, the robot image system requires a real-time operating system on the embedded system. Embedded software forms the basis of current and future telecommunications, automotive, multimedia and industrial automation systems. Embedded systems are devices with limited resources that fulfill tasks previously defined specifically for them rather than general-purpose tasks [8-10].

In this work, the robot is initially in an unstructured environment. The robot is provided with the definition of color information in the field of view. Then, the robot is moved autonomously to the specified target. Camera image information received from the application environment is analyzed using a color tracking algorithm on the computer. Object recognition is achieved by means of color space, so that the robot has reached the target in real time.

This application is designed to help all customers, especially elderly and disabled people, who are shopping at shopping centers. Thus, at the shopping centers, the color information (RGB and HSV) of the person to be followed firstly is introduced to the system. Then, the images captured by the camera from the environment are processed. The data obtained by the image processing is evaluated by the designed algorithm and real-time target tracking is performed.

## 2. METHODS

In this study, the color tracking algorithm is used for object tracking methods which are known and widely used in the literature.

### 2.1. Color Spaces

Color spaces are mathematical models used to describe colors. These are created to represent all colors. Color spaces are designed in three dimensions. Three independent variables are required to determine the color. The locations of the colors in the color space are determined by these variables. Each color space has some standard to create its own color. When creating color spaces, it must be possible to transform into another color space with linear or nonlinear applications [9]. In the application, the RGB format image taken from the camera is converted into HSV (Hue, Saturation, Value) format. Thus, the color to be followed can be distinguished better.

#### 2.1.1. RGB Color Space

The RGB color space consists of the main colors 'Red', 'Green' and 'Blue'. It is a color space that is often used in identification processes. It is accepted that all the colors in the nature come from these three colors. All the other colors can be obtained as a result of mixing three colors at certain ratios. If all the colors are mixed at the maximum, white color is obtained. If all the colors are mixed in the minimum ratio, black color is obtained. Other colors can also be created in the ratio between black and white [9].

#### 2.1.2. HSV Color Space

The HSV or HSB color space defines colors as hue, saturation, and brightness (Value), respectively. Hue determines the dominant wave length of the color. Saturation relates to the vitality of the color. A picture with high saturation is a vivid picture. Brightness

is related to the white color ratio in the color [9].

### 2.1.3. YCbCr Color Space

In this color space, color information is recorded in YCbCr format. Each of Y, Cb and Cr values represent different parameters related to color. Y is the signal of brightness. Cb and Cr represent color information. The YCbCr color space emerged during the efforts to create a digital video standard. Y is defined in the range 16-235 of 8 bits. Cb and Cr are defined in the range of 16-240 [9].

## 2.2. Color Recognition Method

Each image represents a matrix. These matrix values depend on the color of the pixels that make up the image. A color image is usually expressed in three colors, RGB, that is, Red, Green, Blue. But in the OpenCV library, this ranking is BGR. HSV color space is the most suitable color space for color-based image segmentation used in object tracking processes. Therefore, during image processing, the image must be converted from RGB color space to HSV color space. The HSV color space is defined by Smith (1978). HSV is obtained by non-linear transformation from RGB color space [10]. The equations for this transformation are given below (see Equation (1) - Equation (4)).

$$I_{max} = \max(r, g, b), \quad I_{min} = \min(r, g, b) \quad (1) \quad S = \begin{cases} 0, & I_{max} = 0 \\ \frac{I_{max}-I_{min}}{I_{max}}, & I_{max} \neq 0 \end{cases} \quad (2)$$

$$V = I_{max} \quad (3)$$

$$H = \begin{cases} 0^\circ & I_{max} = I_{min} \\ \frac{g-b}{I_{max}-I_{min}} \times 60^\circ + 0^\circ & I_{max} = r, g \geq b \\ \frac{g-b}{I_{max}-I_{min}} \times 60^\circ + 360^\circ & I_{max} = r, g < b \\ \frac{b-r}{I_{max}-I_{min}} \times 60^\circ + 120^\circ & I_{max} = g \\ \frac{r-g}{I_{max}-I_{min}} \times 60^\circ + 240^\circ & I_{max} = b \end{cases} \quad (4)$$

After obtaining the HSV space, the mask of the target is extracted by determining the HSV range of the target. Thus, coordinate values are obtained and tracking process is performed.

## 3. General Application Design

In this study, mechanical equipment (omni wheel, robot body, camera, etc.) to accomplish the target tracking task is completed first. Then the direction controls of the robot were made. The required direction and speed (PWM) information for the motors was transferred to the engines via the Arduino Mega 2560 card. The communication required for the transfer of the information is done via Bluetooth. The images captured by the camera were transferred to the computer in real time and image processing operations were performed on these images.

The robot's body includes a Bluetooth module, a mechanical holder (mica), four reducer DC motors, four Omni wheels, two motor driver cards (L298), a battery and an Arduino card. In addition, image transfer was done by placing a camera on the robot. In addition to tracking the target, it is also aimed to add other robotic features. At the same time, the robot can be moved manually, if necessary, via a phone with an android operating system.

This work consists of two phases. During the hardware phase, the design and implementation of the robot is explained. In the software phase, algorithms suitable for real-time application are mentioned.

### 3.1. Hardware Design

The direct movement of the omni wheels to the target makes the robot easier to navigate. In this way, the robot moved to the target in a short time. Thanks to omni wheels, the robot is programmed with four different movements as forward, right, left and rotation. Depending on the different applications, more orientation can be added to the robot by changing the movement angle. Other directions for the robot are not used because these directions are sufficient in the project.

Encoder motors used have a working voltage of 12V and have a torque of 10 kg / cm. This brush motor with 37 mm diameter has a gear ratio of 56: 1 and a rotation speed of 175 RPM. The reason for choosing motors with this RPM is to prevent sudden, rapid movement changes of the robot. If it is slow, the target can disappear. If it is fast, the robot will show an unstable attitude because the robot can not stop at the desired point. Robot speed control is performed to achieve the desired efficiency. For this, the Arduino Mega 2560, which is used as a microcontroller, generates a PWM signal so that the speed control of the motors is provided as an analogue.

The current from the Arduino pins is not enough to drive the motors. Therefore, L298 motor driver cards are used. With the L298 motor driver card, two DC motor can be managed. A small fan has been added to the system to prevent overheating during prolonged use. In situations where warming can occur, the fan is operated and thus overheating of the motor drivers is prevented.

During the communication phase, HC-05 FC-114 Bluetooth module is used. For communication speed, Baud rate has set as 9600. When the module is connected to the computer, the default name and password of the module have been changed. So that, unnecessary, different signals that could potentially come from robots are blocked.

3S 2650 mAh 20-30C LIPO battery is used as the battery. The reason for choosing this battery type is that it is long-lasting, can meet high current requirements and allows long-term use.

For robotic body design, 3D drawing of the body was performed in the Autocad program (see Figure 1). Mica with 3 different thicknesses was used in the robotic body. Mica is preferred because it is strong, durable and lightweight. For the robotic body, the bottom layer is designed as 8 mm, the middle layer as 5 mm and the top layer as 3 mm. Because the load of each layer will be different. In addition, economic savings will be achieved. The robotic body was designed as shown in Fig. 1. The cut of the mica was made using laser cutting. 37 mm motor holders are used to hold the body and motors together.

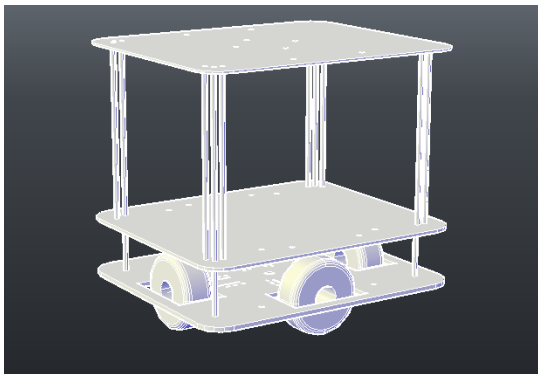


Fig 1. Robot 3D Drawing

The materials placed on the robot body are shown in Fig. 2.

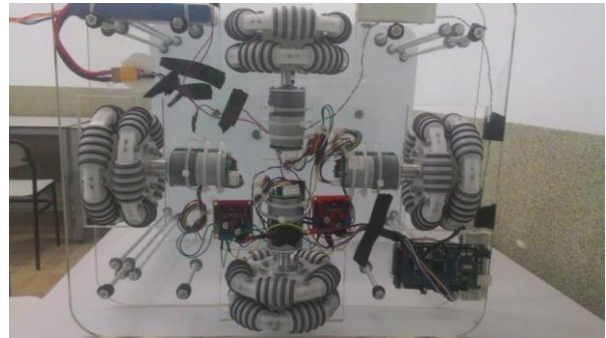


Fig 2. Demonstration of the materials used in the robot

### 3.2. Software Design

In the robot image processing section, Matlab program was used initially. This program can not achieve sufficient speed for simultaneous data transfer. So, the real-time tracking system did not work at the desired level. Therefore, OpenCV library is used with C++. Thus, more efficient results have been achieved in terms of speed. The application has been completed using the color tracking method and various filters.



Fig 3. Camera used in application

In the software section, during the image processing phase, the BGR values on the target are firstly converted into HSV values and the target is determined according to the entered Hue, Saturation and Value values. The images are in the coordinate system of 640x480. The location of the target is determined and coordinates are assigned to a variable. The algorithm is designed so that the coordinates of the target will be centered at the exact center of the image we have taken. The camera used (see Fig 3) has a resolution of 16 MP, 1920x1080. The camera takes 30 frames per second. In the software used, comparison of these frames is done and the target location is determined. After this step, the robot could move towards the target. The general algorithm of the designed system is shown below in the flow diagram, Fig. 4. The results obtained from the algorithm are sent to the Arduino card on the robot via bluetooth. The same steps are repeated for each frame. Thus, real time target tracking is carried out.

Fig 4. General algorithm

## 4. Results and Discussion

As a result of the work, the application of the image processing and the real time color tracking shopping robot has been successfully completed. In this study, firstly, the colors on the target have been introduced in order to determine the target. Then, each frame received from the camera was converted from RGB to HSV. Median and Gaussian filters were applied on the target in order to distinguish the colors in the defined range from the background. Thus, noise on the image is eliminated. Subsequently, the target was obtained with a threshold. As seen in the computer screen of



Fig. 5., the target is completely separated from the background. Then, location information of the target was obtained. After this step, image processing operations are completed. In the next phase, the designed algorithm is applied. After precise detection of the target, the coordinate values of the target were analyzed using the algorithm written in the C ++ language. As a result of the analysis, the direction and the distance of the target are determined according to the robot. Afterwards, the required direction and speed information was transmitted to the Arduino card via the Bluetooth module. The orientation of the robot is determined from the position of the camera placed on the robot according to the target. The front and side view of the designed robot is shown in Fig. 5.

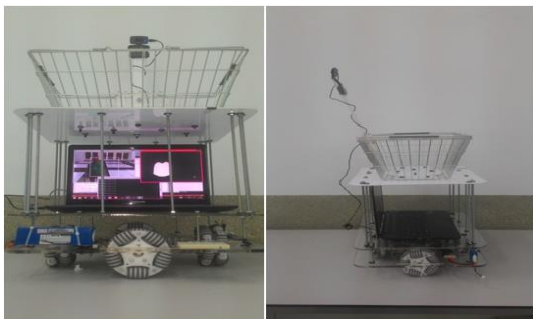


Fig 5. Robot general view



Fig 6. Using the designed robot in practice

In this work, the images recorded by the high resolution camera are processed as real time and target tracking is performed. In practice, omni wheels are used to increase the flexibility of motion of the robot. At this point, the robot can directly direct to the target, that is, the robot reaches the target in a short time. As image processing software, the C ++ language is used with the OpenCV library. Real time target tracking has been successfully implemented using the designed algorithm, software and hardware. The first prototype produced was successfully implemented in a market (see Figure 6). The current state of the project is suitable for implementation. However, it is planned to add new features to the project in the future. For this purpose, the creation of a wireless charging station is planned. The reason is that the energy required for the movement of the waiting shopping robots is provided wirelessly. Thus, those who want to use the robot will receive the robot from the wireless charging unit. Then they will be able to use it directly by activating the robot. Also, in later versions of the project, the Wi-fi module will be used instead of the Bluetooth module, so the Wireless Sensor Network area will be studied. Thanks to the Wi-fi module, the working area will be extended, the working frequency will be expanded and faster communication will be provided.

Therefore, all robots can be tracked from a single center. In the present case, color recognition and tracking are used in the project. In the future, it is planned to develop image processing software so that it can make better object tracking. For this purpose, it is considered to increase the performance of the system by using different tracking methods. It is also planned to remove the computer. Instead of the computer, the cards (Raspberry pi, Beagleboard xm, Pandaboard etc.) that will meet the operating

system requirements for the software will be used.

## References

- [1] G. Samtaş, M. Gülesin, "Sayısal Görüntü İşleme Farklı Alanlardaki Uygulamaları," *Electronic Journal of Vocational Colleges*, vol.2 no. 1, May 2011.
- [2] T. Ebata, M. Ito and M. Shibata, "Visual feedback control method of a wheeled mobile robot using a pan camera preferentially," *IEEE 13th International Workshop on Advanced Motion Control (AMC)*, pp. 149-154, 2014.
- [3] K. Hashimoto, "A review on vision-based control of robot manipulators," *Advanced Robotics*, vol. 17, no. 1, pp. 969–991, 2003
- [4] M. Clark, D. Feldpausch and G. S. Tewolde, "Microsoft kinect sensor for real-time color tracking robot," *IEEE International Conference on Electro/Information Technology*, pp. 416-421, 2014
- [5] N. Masud *et al.*, "FPGA based Multiprocessor Embedded System for Real-Time Image Processing," *15th International Conference on Control, Automation and Systems (ICCAS)*, pp. 436-438, 2015
- [6] T. Schmitt *et al.* "Cooperative Probabilistic State Estimation for Vision-Based Autonomous Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 18 no. 5, pp. 670–684, 2002.
- [7] M. Peker, A. Zengin, "Gerçek Zamanlı Harekete Duyarlı Bir Görüntü Tanıma Sistemi," *6th International Advanced Technologies Symposium (IATS'11)*, 2011.
- [8] S. Erdogdu, "Karmasık Algoritmaların Gerçek Zamanlı Gömülü Sistemlerde Gerçeklenmesi," *ASELSAN AŞ., Savunma Sistem Teknolojileri (SST) Grubu, Ankara, 4. Ulusal Yazılım Mühendisliği Sempozyumu*, 2009.
- [9] H. Hagiwara, K. Asami and M. Komori, "Real-time image processing system by using FPGA for service robots," *1st IEEE Global Conference on Consumer Electronics*, pp. 720-723, 2012
- [10] S. M. A. Shah *et al.* "Robustness testing of embedded software systems: an industrial interview study," *IEEE Access*, vol. 4, pp. 1859-1871, 2016.
- [11] [Online]. Available: <http://matlabgoruntuisleme.blogspot.com.tr/2012/09/rgbhsybcbr-renk-uzaylar-md2.html>
- [12] İ. Başlar, "Tüm Gökyüzü Kamerasından Alınan Görüntüler Kullanılarak Bulutluluk Ölçümü Yapılması," *Türkiye Bilimsel Ve Teknolojik Araştırma Kurumu*, 2012