

Control system design of duct cleaning robot capable of overcoming L and T-shaped ducts

Myeong In Seo, Woo Jin Jang, Junhwan Ha, Kyongtae Park, Dong Hwan Kim

Department of Mechanical design and robot engineering, Seoul National Univ. of Science and Technology, Korea

Article Info

Article history:

Received Dec 17, 2019

Revised Feb 13, 2020

Accepted Mar 4, 2020

Keywords:

Autonomous Driving

Duct Cleaning Robot

Image Processing

Location Identification

ABSTRACT

This study introduces the control method of duct cleaning robot that enables real-time position tracking and self-driving over L-shaped and T-shaped duct sections. The developed robot has three legs and is designed to flexibly respond to duct sizes. The position of the robot inside the duct is identified using the UWB communication module and the location estimation algorithm. Although UWB communication has relatively large distance error within the metal, the positional error was reduced by introducing appropriate filters to estimate the robot position accurately. TCP/IP communication allows commands to be sent between the PC and the robot and to receive live images of the camera attached to the robot. Using Haar-like and classifiers, the robot can recognize the type of duct that is difficult to overcome, such as L-shaped and T-shaped duct, and it moves successfully inside the duct according to the corresponding moving algorithms.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Dong Hwan Kim,

Department of Mechanical system design engineering,

Seoul National University of Science and Technology,

232 Gongneung-ro, Nowon-gu, Seoul 01811, Korea.

Email: dhkim@seoultech.ac.kr

1. INTRODUCTION

In buildings and kitchens, air is supplied or circulated through ducts with various facilities. Smoke and dust contained in the air passing through the duct in the enclosed space accumulate in the ducts and build up more in the flexions. These cause indoor air pollution and should be cleaned regularly. Especially, in case of kitchen ducts, dust and oil stain deposits prevent circulation or cause fire accidents due to high temperature air. Thermal efficiency of heating and cooling ducts for air conditioning and heating decreases when dust builds up. Duct cleaning are largely classified as dry and wet [1] methods. Since it is virtually impossible for humans to clean ducts themselves, the introduction of a duct-cleaning robot is required. The shape of the robot can be determined according to the method of cleaning the duct.

Due to the nature of ducts, especially in the closed space, wire-twisting problems occur when using a wire driven robot. Thus, it is necessary to build a wireless-driven robot and develop a robot that can move autonomously along the ducts. At the same time, the location of the robot needs to be identified while moving along complex duct structure in order to monitor the operation. Meanwhile, existing duct cleaning robots [2-4] are difficult to cope with when the duct diameters vary by location.

In general, air conditioning ducts are concealed and installed inside the ceiling. As the shape and dimensions of the air conditioner from the facility to the inlet and outlet of each chamber change, the flexions are formed, and the successful passing these flexions for the robot should be carried out even with flexions. Typical flexion is either L or T-shaped and the robot should be able to pass through this non-uniform shaped ducts.

The duct-cleaning robot introduced in this work is similar to the existing duct-cleaning robot [5-6], but it is designed with a three-legged (it looks like three tracks in appearance) robot that is more suitable for circular ducts, thereby allowing stable movement on the walls inside the curved ducts. It can also be wirelessly operated to prevent wire twisting, which occurs frequently inside long and winding ducts. Unlike conventional duct-cleaning robots, it is designed to proceed autonomously using PSD (Position Sensitive Device) sensors and cameras without being manipulated by users. The PSD sensor measures the distance to edges of L or T-shaped duct and the camera is used to distinguish between the L and T-shaped duct for self-driving. The user just controls the start button and the stop button and views the images inside the duct to check the robot's current location and cleaning status. Images of the duct inside the enclosed space are sent wirelessly to the user's PC via Wifi through the embedded PC, Raspberry Pi. In this study, UWB modules were installed on the outside of the duct to estimate the current position of the robot in the duct and the correct identification of T-shaped duct using the classifier algorithm based on the acquired images to enable autonomous driving is performed.

2. SYSTEM CONFIGURATION AND INTERFACE

When the pulley radius is r and the angular velocity of each motor for driving the leg is $\omega_i, i = 1, 2, 3$, the robot center travel velocity V_c can be obtained as the product of the pulley radius and the angular velocity. Using these variables, the robot center velocity can be determined according to the each case such as three wheels are at the same speed, one wheel stops and the other two wheels drive, only one wheel is driven, and three wheel speeds are different. Once the three wheels are at the same speed, the velocity of the robot is $V1 = V2 = V3 = V_c$. In case of different velocity on three legs, the velocity of the central body is expressed as velocity gradient seen from the pipeline side as shown in Figure 1.

Thus, the robot body velocity is determined by the average of angular velocities from three motors. Figure 2 shows the appearance of the manufactured duct cleaning robot. The front of the robot is equipped with a camera while the rear is equipped with a brush for dusting. The robot is driven by the user's starting signal from the PC, which is sent to the micro controller Arduino from the Raspberry Pi through TCP/IP communication. The PSD sensor connected to the Arduino measures the diameter of the duct, which is important information for autonomous driving in the duct. In addition, the camera connected to the Raspberry Pi continuously captures images inside the duct and sends the results to the PC. At this time, the L and T-shaped structures inside duct that are difficult to detect only by PSD sensors are distinguished through the camera's image processing technologies, and the corresponding movements of the robot according to the recognized duct structure are introduced to help autonomous driving of the robot. Users can continue to observe the inside of the duct by pressing the start button of the PC.

In order to know the exact location of the robot inside the duct, UWB anchors installed on the outside of the duct and a UWB tag on the robot body are used. The distance values between anchors and tag are transmitted by Bluetooth communication to the PC and the location of the robot is estimated by the trilateration algorithm on the PC.

There are three main categories of control system developed in this work. A UWB module for estimating the robot's real-time position, an Arduino Megaboard for controlling the robot, and a Raspberry Pi Zero for transmitting images via PC and TCP/IP communication to control the robot remotely. Figure 3 is the overall control system configuration of the robot used in the robot system. Arduino controls the motor and LED lights for robot driving and receives the values from the PSD sensors. The UWB module transmits the distance values to the PC via Bluetooth communication. Raspberry Pi transmits images captured by the camera mounted on the robot to the PC via TCP/IP communication and also sends command signals from the PC to Arduino.

$$V_c = \frac{V1+V2+V3}{3} = \frac{r}{3}(\omega_1 + \omega_2 + \omega_3) \quad (1)$$

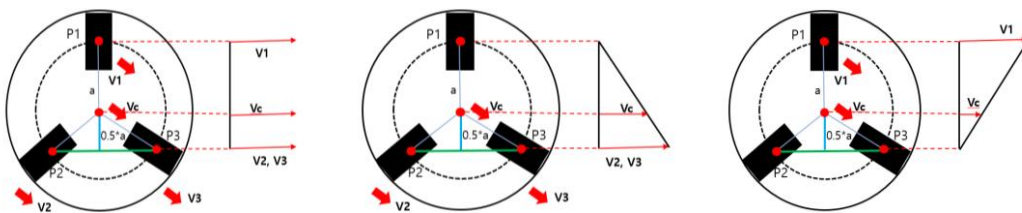


Figure 1. Velocity profile at the side view of the pipeline.

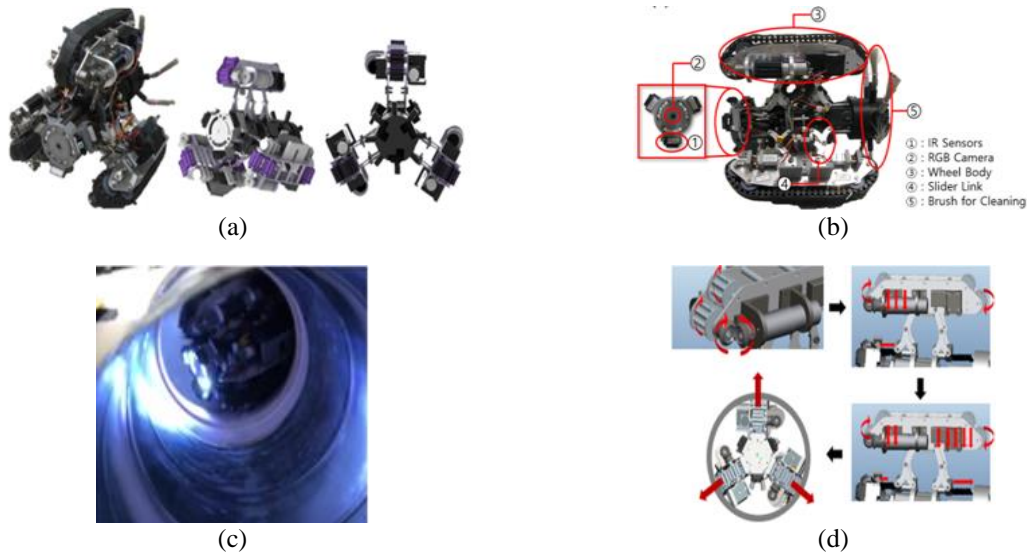


Figure 2. (a) Overall view of duct cleaning robot, (b) assembled robot, (c) and picture of robot running in a duct, (d) drawings of duct cleaning robot

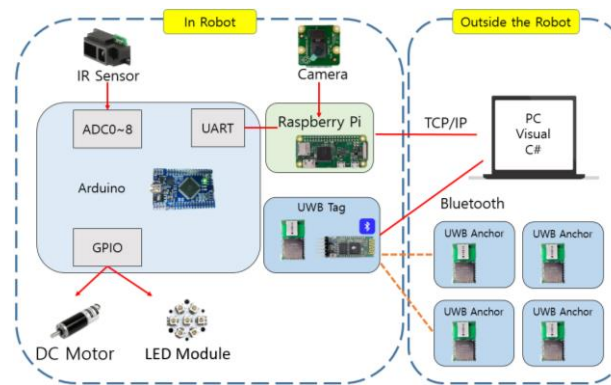


Figure 3. Control system structure

2.1. UWB module configuration

As mentioned earlier, UWB communication is used for indoor location measurement, and the DWM1000 [7] used for communication is a CMOS chip for measuring precise distances with an accuracy of 10cm. The Arduino Pro Mini is adopted to operate the UWB module. One tag on the robot and four anchors outside the duct are placed, as shown in Figure 4, and the PCB (Printed Circuit Board) is used to reduce the size of anchor and tag module so that they can be placed inside the robot. Bluetooth modules are also attached to the PC to receive distance values from UWB modules wirelessly.

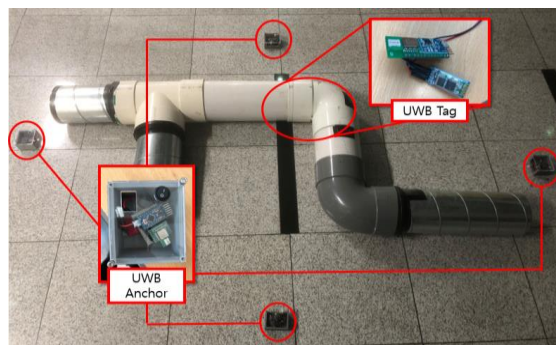


Figure 4. UWB modules placement for positioning in duct environment

2.2. Controller configuration

In this work, the ATmega2560 based Arduino Mega was used to acquire the values of the nine PSD sensors which are used to identify the internal shape of the duct. To simplify the circuit, a PCB compatible with the Arduino Mega was created. One DC motor for each wheel and two smaller DC motors for wire control were connected to the DC motor drivers.

2.3. Infrared sensor

As stated earlier, PSD sensors are used to identify the structure inside the duct which is crucial for an autonomous driving of the robot. The infrared sensors with a range of 2cm to 15cm can be used to detect structural changes inside the duct while the robot is driving. The Kalman filter can be used to more effectively handle the analog values received through the infrared sensor, since the sensor's signal itself is partly absorbed into the duct internal metal parts, potentially distorting the sensor values. The Kalman filter equations used here are as follows.

$$K_{k+1} = \frac{\sigma_k^2 + \sigma_p^2}{\sigma_M^2 + \sigma_k^2 + \sigma_p^2} \quad (2)$$

$$\hat{x}_{k+1} = K_{k+1}z_{k+1} + (1 - K_{k+1})x_k \quad (3)$$

$$\sigma_{k+1}^2 = (1 - K_{k+1})(\sigma_k^2 + \sigma_p^2) \quad (4)$$

where K_{k+1} is the Kalman gain, σ_k is the error covariance, σ_p is the process noise variance, and σ_M is the measurement noise variance. z_k is the measured state variable and x_k is the estimated state variable at k -th step, respectively.

Based on the above equations (2-4), we applied the proposed Kalman filter to the micro controller. In Figure 4, the output data before and after the filter passes for the infrared sensor signal were compared. The blue line shows the value of the data before going through the filter, and the red line shows the value of the data coming through the filter. In this case, $\sigma_k=0.25$, $\sigma_p=0.01$, $\sigma_M=0.09$ were used and $K_0=1.0$ was used as initial value. The experimental results confirmed that the value that came from the Kalman filter was more stable than original signal (Figure 5).

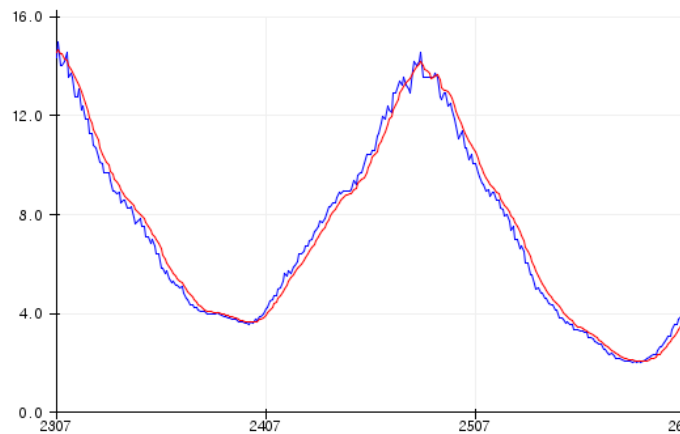


Figure 5. Distance data comparison using Kalman filter (x-axis : time [ms], y-axis : Distance [cm])

3. DISTANCE MEASUREMENT USING UWB COMMUNICATION

3.1. UWB communication

UWB (Ultra Wide Band) communication is a wireless communication technology that can guarantee transmission distance of 10m to 1km over a wide frequency range from 3.1 GHz to 10.6 GHz, and has been developed and applied for special purposes by the military, but has recently gained attention as it opened to the private sector. UWB, which transmits large amounts of information at low power, is a new technology with a very wide application range, including indoor observation, sports tracking, intruder detection system, and prevention of heavy equipment vehicle accidents. In this study, the UWB communication and Real-Time Locating Systems (RTLS) were applied to implement a system that could

automatically identify and track the position of objects. The communication modules used for UWB communication are DWM1000 from DECAWAVE and Time of Flight (TOF) method for distance measurement is a main principle of the sensor.

Figure 6 is a representation of the TOF-type distance measurement procedure, which is characterized by DS-TWR (Double-sided Two Way Ranging) [8] extending the measurement range of SS-TWR (Single Sided Two Way Ranging), in which two round-trip time measurements are used to obtain the TOF results. Errors are reduced despite significantly long response delays and the TOF is 333 ns in the relatively large UWB operating range of 100 m.

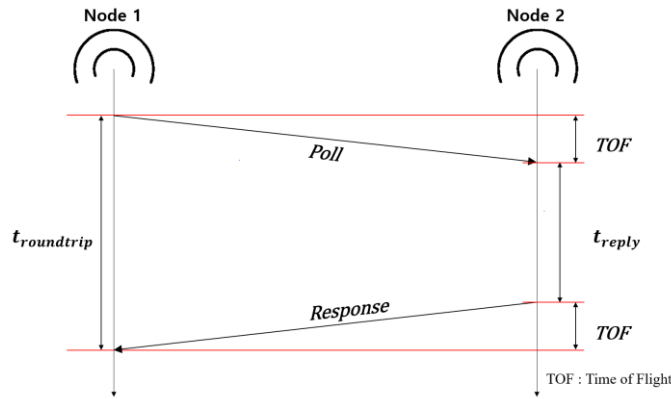


Figure 6. Sequence of TWR using TOF

3.2. Estimating the location of the robot by trilateration measurement

To identify the position of the robot inside the duct, a trilateration method was applied to estimate the center position of the robot while it moves around using anchors that are placed outside the duct and a tag on the robot body. Here, the gradient decent method is applied to reduce the error between the actual and the estimated values for the robot location. Figure 7 shows the basic environment for applying the trilateration method to obtain the robot's central position (x, y) , using the distance values from four anchors and one tag. The true value of the center position of the robot is constructed by the following formula.

$$(x_i - x)^2 + (y_i - y)^2 = r_i^2, \quad i = 1, 2, \dots, m \quad (5)$$

where m is the number of anchors, and r_i is measured distance between the anchor i and tag. Next, d_i which is the estimated distance to the robot from anchor i is expressed as follow.

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (6)$$

Error between the actual value and the estimated value between anchor i and tag is $e_i = d_i - r_i$, and the cost function C that minimizes the error at each step can be defined as;

$$C = \frac{1}{m} \sum_{i=1}^m e_i^2, \quad m \geq 3 \quad (7)$$

By using the gradient descent method, the robot's central position is updated in each step as follows.

$$X_{k+1} = X_k - \gamma \nabla C$$

$$X_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}, \quad k = 1, 2, \dots \quad (8)$$

The final update law for the estimated 2D location is determined in (9).

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \gamma \begin{bmatrix} \frac{2}{m} \sum_{i=1}^m \frac{(x_i - x_k)(d_i - r_i)}{d_i} \\ \frac{2}{m} \sum_{i=1}^m \frac{(y_i - y_k)(d_i - r_i)}{d_i} \end{bmatrix} \quad (9)$$

As shown in (9) is an equation determined by the gradient descent method that calculates spatial coordinates that minimize the distance error between the anchors and the tag. X_{k+1} , the next point to move from the current coordinate value X_k is calculated, and γ is a parameter that controls the update speed. When UWB modules are used in a duct made of metal in part, location errors occur due to the fact that the waves do not penetrate the metal parts well but reflect them [9]. To overcome this, the following methods have been tried. We applied general signal filters to incoming data values, added the number of anchors, and supplemented the shortcomings of the gradient descent method. The disadvantage of gradient descent method is that it does not escape the zero slope well and convergence is slow. To solve this problem, the commonly used method is to apply the stochastic gradient decent with momentum method [10]. In other words, the speed constant is corrected to reach the minimum value faster and the inflection point can be smoothly passed over due to the momentum.

$$v_k = \gamma v_{k-1} + \eta \nabla C \quad (10)$$

$$X_{k+1} = X_k - v_k \quad (11)$$

In (10), v_k represents velocity at k -th step, and η represents a decaying rate. In this case, $\eta=0.9$, and $\gamma=0.95$ is used respectively, which showed that convergence was about 10 times faster than before. In addition, the value of X_k is a variable to which the gradient descent method is applied, that is, a coordinate value updated by substituting the x coordinate or the y coordinate obtained through the gradient decent with momentum method can be obtained.

Figure 8 shows the results of experiments carried out in metal ducts. The red circles in Figure 7 are the results of the conventional gradient descent method and the blue circles are the results of the gradient descent method considering the acceleration. It can be seen that the error is reduced and closer to the actual positions by employing the gradient decent with momentum method than the gradient decent method. The actual errors are up to about 30 cm for both methods, which are reliable to be adopted to location estimation inside ducts.

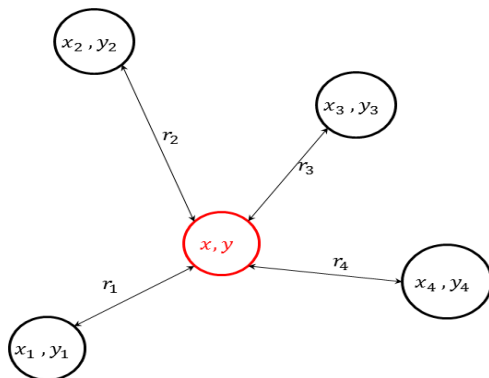


Figure 7. Distances between four anchors and one tag

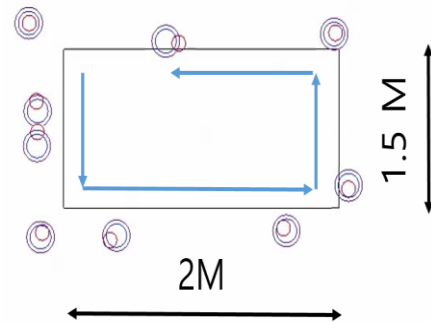


Figure 8. Experimental results of gradient descent method and considering acceleration

4. OPERATION SOFTWARE DESIGN

4.1. Autonomous driving

The robot basically moves forward through three motors. When the PC gives commands to the robot through TCP / IT communication, the robot moves or stops. When the robot starts to move, the LED and the brush motor start to work together, and at the same time, the internal shape of the duct is recognized by the distance information of the infrared sensors attached to the front and back of the robot as shown in Figure 9. Infrared sensor values usually come out with a constant value when the robot moves along straight duct. When the curved section is found, the values from the infrared sensors change. Through these values, the robot can be automatically rotated according to the left and right flexion which is called L-shaped duct, or two-way path, called T-shaped duct as shown in Figure 10. The precise schemes for passing through L or T-shaped duct will be explained in the later section.

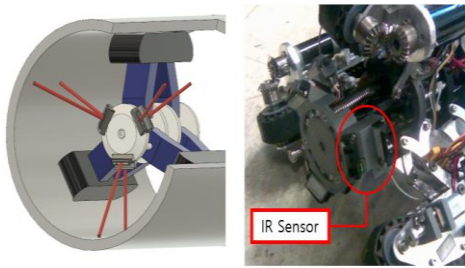


Figure 9. IR Sensors placement in robot

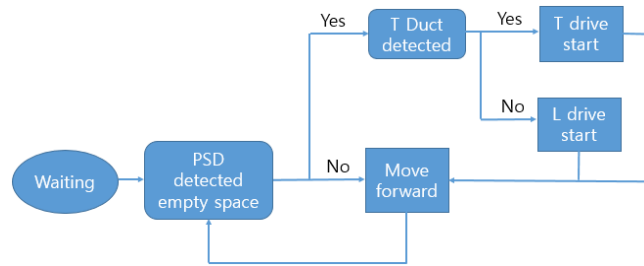


Figure 10. Robot autonomous driving algorithm using IR sensors only

4.2. Left and right driving in the L-shaped duct

When the robot meets the curved section, it operates in the order as shown in Figure 11. First, check the left and right sides through the infrared sensor. When the robot encounters a curved section while moving forward, one of the two sensors has a distance of more than certain distance (here, 12.5cm) as shown in Figure 10, making the robot detect empty space. Then, the robot stops the wheel that becomes the pivot axis and moves the rest of the wheel so that it can turn without heavy resistance inside the duct. After the robot turns, if the value of one infrared sensor is less than about the preset distance (here, for instance 12.5cm), the robot will start moving forward again. In this way, the left and right drive in the L-shaped duct can be automatically carried out.

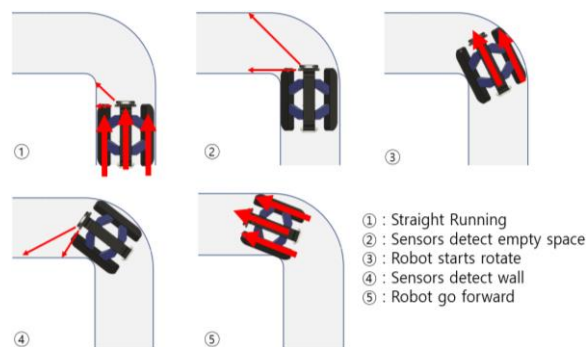


Figure 11. Sequence of L-shaped duct running

4.3. Overcoming T-shaped duct with Timers

In the T-shaped duct, the robot can move forward, left and right. In order for the robot to turn in the T-shaped duct, it must be turned in a different way than the L-shaped duct movement mentioned earlier. When turning on a T-shaped duct, the wheel body will unfold due to the robot's spring structure in the open space. In other words, because one side is open and the wheel cannot reach the wall and the robot's wheels can get stuck. Therefore, for this purpose, the robot must be controlled to escape the section by giving it a reverse turn. In case of using infrared sensors for self-driving, as shown in Figure 12, when the robot meets a T-shaped duct it can detect a side empty space, just like the L-shaped duct, but not a front empty space. Therefore, infrared sensors alone cause the robot to move in the same way as L-shaped duct, which lets the robot get stuck in the bottom of the T-shaped duct. To solve this problem, the accurate recognition of the T-shaped duct should be established. If T-shaped duct is detected through image processing without the use of distance sensors, the robot recognizes that there is a T-shaped duct in front. The detail algorithm to find the T-shape duct by image processing is explained in the following section.

When an infrared sensor detects empty space, T1 timer inside MCU starts. And while driving the geared motor, the wire is wound to close the front of the three legs. The robot moves forward until the timer reaches a predetermined time. At this time, the T2 timer (Figure 12) starts, rotating the wheel on the axis of turn reversely, as shown in (4). When the T2 timer reaches a preset time, the T3 timer starts and rotates the robot's top wheel in reverse order to turn the robot. When the T3 timer reaches a predetermined time, all the wheels of the robot move forward and starts self-driving again on straight section. Through

the experiments, the times of each timer was found to be about 4.8 seconds, 6.4 seconds and 8 seconds respectively, and it was proved the robot could automatically overcome the T-shaped duct successfully. Based on the experimental results, the setting times of the timers can be adjusted according to the conditions of the duct.

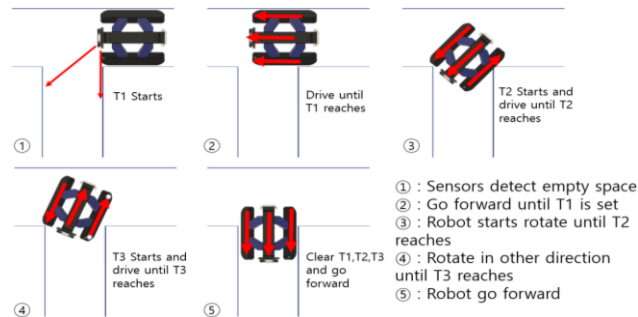


Figure 12. Sequence of T-shaped duct running using timers

4.4. Self-driving in T-shaped duct using image processing

As mentioned earlier, the duct types that robot needs to overcome are L-shaped ducts and T-shaped ducts, which are different in the driving control method to overcome the two types. Therefore, the robot's ultimate goal is to accurately recognize and properly overcome the flexions. The use of distance sensors alone may result in significant errors due to misperception at the flexion, resulting in the wrong choice of how to overcome the shape, which may cause it to fall into certain sections or become stuck there. If image information through a camera is supplemented with identification algorithms, it is certainly helpful for the robot to accurately recognize the flexion whether it is L or T-shaped and to choose the right method to overcome the duct shape.

If the image information is used, the shapes of the two ducts will be distinguished as shown in Figure 13. Distance sensor values can be similar for both shapes, and in fact, three wheels touch the wall in the L-duct, but in the T-duct, one wheel must be run without touching the wall. Due to the problem of wheel and not being able to operate, additional wire motor was to be used as shown in Figure 14.



Figure 13. Image difference between (a) L-shaped and (b) T-shaped duct

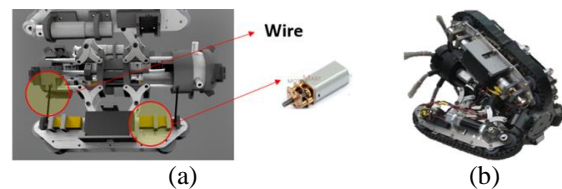


Figure 14. Robot wire attachment drawing and picture, (a) wire and wire-driving motor, (b) shape of closed front part of robot

There are six wires, two for each leg. These wires are used to fold the front and rear parts of a leg when moving in a T or L -shaped duct. Before the robot enters the duct, the geared motors are activated to fold the legs by winding the wires. In order to distinguish L-shaped duct from T-shaped duct by image, the outline of T-shaped duct edge was selected as characteristic point. It was determined that the two types could be recognized differently depending on whether the outline shown in Figure 15 was detected.

At first, we tried to detect the outline, but there were the following problems.

- There were too many contours detected.
- It was difficult to use the desired pixel coordinates.
- There was poor real-time performance because it took a long time to calculate.

Therefore, we decided to use the camera image itself rather than the pixel coordinates of the outline, and we used template matching [11]. Template matching has the advantage of obtaining coordinates of matching image from reference image, which can be verified in real time due to shorter computational time,

and the degree of matching with the threshold value can be adjusted. However, there were disadvantages of requiring proper reference images in advance, changing the threshold value every time depending on the environment, and being very susceptible to brightness changes and geometrical rotations.

Therefore, the Harr-like features [12] method that does not require a reference image beforehand was used to detect the outline of the image. Using Harr-like features as shown in Figure 16, an object is recognized by combining elementary features that are constructed based on pixel brightness differences in the image. The outline to be detected was found to be suitable for the key point recognition of the T-shape duct because it basically retains the geometric information as shown in Figure 15, and the area has a distinct brightness difference. In order to extract the feature to classify the input image, the difference in brightness of all pixels must be calculated and compared with the thresholds of the elementary features.

If we calculate the brightness differences for all pixels, the computational volume becomes very large, so integrated images [13] can be used to quickly calculate the pixel brightness differences in the desired area. This allows the brightness of all areas to be calculated in only four calculations, which enables rapid computation. If the feature of the T-shaped duct is extracted due to the difference in brightness, a classifier is made and compared with each input image to detect the T-shaped duct. For this operation, the functions provided by OpenCV were used with the elementary features and threshold values. To create the classifier, an xml file [14] was made using the Cascade Classifier provided inside the OpenCV. Figure 17 shows the process of creating an xml file as a flow chart and represents the image data used. For the Positive samples, the samples were constructed based on the shapes that required recognition within the duct, and the points to be used as a feature were marked. For the Negative samples, images that had nothing to do with the duct were selected, i.e. backgrounds, animals, or cars.

In the early stages of training, the number of samples was tested using 50 Positive and 200 Negative samples respectively, and found that non-T-shaped ducts such as L-shaped ducts or straight ducts were sometimes judged to be a T-shaped duct. To solve this problem, the number of samples for Positive and Negative were increased to 300 and 900, respectively, and the image size was modified so that pixel brightness could be clearly distinguished. Later, after checking the real-time detection, it was confirmed that the T-branch section was relatively accurately detected and overcome the errors of the T-shaped structure recognition by accurately detecting the T-shaped duct, and confirmed that it was recognized not only in the existing environment but also in the new environment. Table 1 shows the results of the experiments according to the number of samples.



Figure 15. Detected T-branch section

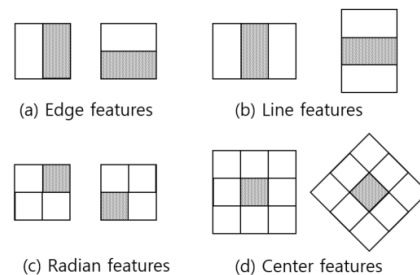


Figure 16. Example of elementary features

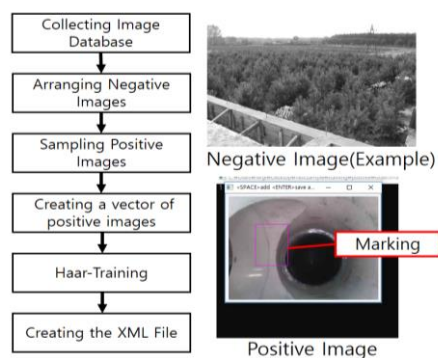
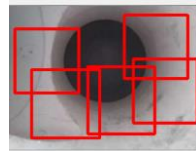



Figure 17. Sequence of making xml file and image data

Table 1. Success rate based on number of samples

| | | |
|--------------|--|---|
| Positive | 50 | 300 |
| Negative | 200 | 900 |
| Results |  |  |
| Success Rate | 0% | 95% |

5. EXPERIMENTS AND RESULTS

After the robot was manufactured, a lot of tests were conducted for each section to ensure normal driving was possible on straight duct, L-shaped and T-shaped ducts. For straight and L-shaped ducts, robot driving was smooth with image capturing, and brushing with the help of distance sensors and algorithms. In case of T-shaped ducts, the robot could move normally without falling into the duct when it was driven with the front parts of the wheels folded by wires. In some cases, the movement failed in the T-shaped duct, mainly if the wires were not sufficiently tight or image processing was not secured. Table 2 represents the success rate for each duct segment.

Using the UWB, many experiments have been conducted to identify the location of moving robot inside the duct made of metal or and plastic depending on the section. For the plastic section, the trilateration algorithm based on gradient descent method guaranteed great accuracy with an error of less than 5cm. However, with the application of trilateration algorithm for metal section, the TOF between anchors and tags varied greatly, preventing accurate positioning. With the introduction of momentum term shown in (9-10) and increase the number of anchors from three to four, the error could be reduced to 30 cm. Table 3 is the UWB position measurement errors according to the material of the duct.

Table 4 shows the performance to transmit real time image from the robot's camera to PC. Due to the performance limitations of Raspberry Pi Zero, the image frame rate had to be lowered. In addition, the video could be instantly viewed by streaming it live on internet. However, due to the internet environment or performance problems with Raspberry Pi, there were slight delay in viewing the video. On the other hand, the camera used in the robot was a small size model with Pi Camera, so it had a low resolution. When using C# program to identify T-shaped duct by image processing stated previously, the identification success rate for T-shape duct was around 95%.

Some of the reasons for the failure of the misrecognition of T-shaped duct were that when transparent ducts were used, external light was transmitted into the duct, which resulted in failure of image processing due to difference in illumination inside the duct, and a delay in image transmission due to performance of wireless communication modules.

In Figure 18 the images show that the robot operates in the vertical duct that is more difficult than the horizontal duct. Tests showed that the robot was climbing up using the friction force between the belt connected to the driveshaft and inner walls of the duct. Figure 19 shows the images that the robot moves through the L-shaped duct and the robot movements in the T-shaped duct are shown in Figure 20. All moving performance for three types of duct are compared in Table 5.

Finally, Table 5 is a measure of driving speed measured according to shape of ducts. The robot's driving speed was moderate enough to clean it up with a brush. As we guess, the robot speed is slowest in the T-shaped duct because it takes longer to recognize the T-shape duct through image processing and to safely overcome it through subsequent timer operations and wire operation.

Table 2. Success rates for moving in duct in accordance with duct shapes

| Duct Types | Straight Duct | L-shaped | T-shaped |
|--------------|---------------|----------|----------|
| Success Rate | 100% | 100% | 80% |

Table 3. Location errors of robot inside duct in accordance with different materials

| UWB Positioning Error | |
|-----------------------|------------|
| Empty Space | 5 ~ 10 cm |
| Plastic Duct | 5 ~ 10 cm |
| Metallic Duct | 30 ~ 50 cm |

Table 4. Image processing and performance figures

| Video Frame | Video Transmission Delay | Resolution (Pixel) | Image Processing Success Rate |
|-------------|--------------------------|--------------------|-------------------------------|
| 20 FPS | 0 ~ 1.5 sec | 320 x 240 | 95% |

Table 5. Robot average speed in several circumstances

| Duct Type | Straight | L-shaped | T-shaped |
|-------------|----------|----------|----------|
| Speed(mm/s) | 20 | 12 | 5 |

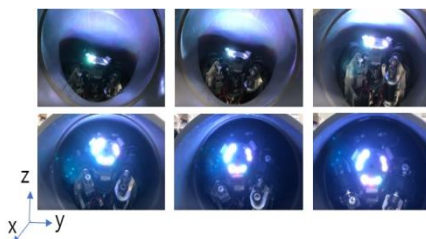


Figure 18. Moving scenes of a robot in a vertical duct

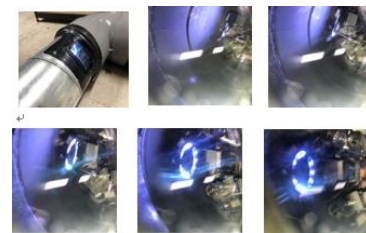


Figure 19. Moving scenes of a robot in a L-shaped duct

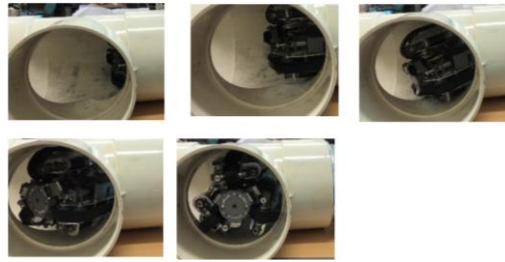


Figure 20. Moving scenes of a robot in a T-shaped duct

6. CONCLUSION

In this work, we developed a new type of duct cleaning robot that can solve the passing problem of various types of duct shapes. First, the control of the three-legged robot, which can adapt to the duct diameter change, was carried out. Next, the real-time location estimation of the robot was established by applying UWB communication and trilateration scheme so that user could easily identify where the robot is located inside the duct. In addition, the self-driving mode, which used to be a problem for existing robots, was solved by using sensors and image processing to make it easier for ordinary people to manipulate them. Finally, to conveniently identify the overall cleaning situation and location of the robot, the interface was built on the user's PC so that the control of the robot and the monitoring of current situation could be achieved at once.

After the robot detected the L-shaped duct in front using infrared sensors, it was possible for the robot to move freely in the L-shaped duct by stopping one leg of three legs (tracks) while the other two legs moving. The robot was also built with wires to pull the robot's front and rear parts of the tracks inwards, allowing the robot to move without falling into the open area even in the T-shaped section. In addition, the T-shaped duct was surly recognized by adopting Harr-like features method and cascade classifier using the camera images.

Using the three legs (tracks) mechanism along with the robot's slider link structure and belts with great friction, the robot was able to drive various sections, including L-shaped, T-shaped and vertical duct, and succeeded in autonomous driving using infrared sensors and image processing schemes. The robot worked well inside the circular duct, but there was a disadvantage that it could not operate on the rectangular duct because it is difficult to make the robot fit in inner surface. Nevertheless, the robot's focus is to clean the inside of the duct, but it can also be used for exploration in pipes with similar environments to round ducts.

ACKNOWLEDGEMENTS

This study was supported by the Research Program funded by the SeoulTech (Seoul National University of Science and Technology).

REFERENCES

- [1] Woo Tea Jung, "Underground History Duct Pollution and Cleaning Technology," *The Society of Air-Conditioning And Refrigerating Engineers of Korea*, vol. 41, no. 4, pp. 34-41, 2012.
- [2] Seung Woo Jeon et al, "Design of an Intelligent Duct Cleaning Robot with Force Compliant Brush," *2012 12th International Conference on Control, Automation and Systems*, JeJu Island, pp. 2033-2037, 2012.
- [3] Young Sik Kwon et al, "Design and Motion Planning of a Two-Module Collaborative Indoor Pipeline Inspection Robot," in *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 681-696, June 2012.
- [4] Ya Wang et al, "Autonomous Air Duct Cleaning Robot System," *2006 49th IEEE International Midwest Symposium on Circuits and Systems*, San Juan, pp. 510-513, 2006.
- [5] Atsushi Kakogawa et al, "An in-pipe robot with underactuated parallelogram crawler modules," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, pp. 1687-1692, 2014.
- [6] Atsushi Kakogawa et al, "Design of an underactuated parallelogram crawler module for an in-pipe robot," *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Shenzhen, pp. 1324-1329, 2013.
- [7] Kim, K. H, "UWB RADAR trend of technology," *The Proceedings of the Korea Electromagnetic Engineering Society*, vol. 13, no. 3, pp. 52-62, 2002.
- [8] Jung-ha Kim et al, "A study on indoor navigation system using localization based on wireless communication," *Journal of the Korean Society of Marine Engineering*, vol. 37, no. 1, pp. 114-120, January 2013.
- [9] Joon Hyun Park et al, "A Study on Interference by Obstacles for UWB based Distance Calculation," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 984-985, 2018.

- [10] Ning Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, January 1999.
- [11] Wisarut Chantara et al, "Object Tracking using Adaptive Template Matching," *IEIE Transactions on Smart Processing & Computing*, vol. 4, no. 1, pp. 1-9, February 2015.
- [12] Tae Joon Park et al, "Detection of License Plates Using A Cascade or Boosted Classifiers Based on Haar-like Features," *Control, Automation and Systems Domestic Conference Symposium*, pp. 631 – 633, 2008.
- [13] Ki Yeong Park et al, "An Improved Normalization Method for Haar-like Features for Real-time Object Detection," *The Journal of The Korean Institute of Communication Sciences*, vol. 36, no. 8, pp. 505-515, August 2011.
- [14] Mahdi Rezaei, "Creating a Cascade of Haar-Like Classifiers: Step by Step," *ResearchGate Publishing*, pp. 1-8, 2013.

BIOGRAPHIES OF AUTHORS



Myeong In Seo graduated from Seoul National Univ. of Science and Technology in 2019 and he has been studying graduate course at the same university. His major research areas are image processign, and robotics control.



Woo Jin Jang graduated from Seoul National Univ. of Science and Technology in 2019 and he has been studying graduate course at the same university. His major research areas are robot mechanism design and analysis, and robotics control.



Junhwan Ha graduated from Seoul National Univ. of Science and Technology in 2018 and he has been working at Samsung SDI. His major research areas robotics, and programming.



Kyongtae Park graduated from Seoul National Univ. of Science and Technology in 2018 and he has been working at Hyundai Motors. His major research areas robotics, and design.



Dong Hwan Kim graduated from Seoul National University in 1986, and finished his master degree at the same university in 1988. He had a Ph.D degree from Georgia Tech in 1995. He had been working Dusan Infracore as junior researcher from 1988 through 1991. After Ph.D he had been worked at Korea Institute of Industrial Technology in 1997-1998. He joined Seoul National University of Science and Technology as Professor in 1998. His major research areas are Mechatronics, Robotics, and Automation.