Avrupa Bilim ve Teknoloji Dergisi
Sayı 31 (Ek Sayı 1), S. 890-905, Aralık 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 31 (Supp. 1), pp. 890-905, December 2021

Research Article

Multi-Camera Vision-Based Mobile Robot Control and Path Planning

Emrah Dönmez[1*], Alper Özcan[2]

[1*] Bandırma 17 Eylül University, Faculty of Engineering and Natural Sciences, Departmant of Software Engineering, Balıkesir, Turkey, (ORCID: 0000-0003-3345-8344), emrahdonmez@bandirma.edu.tr

[2] Akdeniz University, Faculty of Engineering, Department of Computer Engineering, Antalya, Turkey, (ORCID: 0000-0002-5999-1203), alperozcan@akdeniz.edu.tr

(First received 23 June 2021 and in final form 21 November 2021)

(**DOI:** 10.31590/ejosat.)

ATIF/REFERENCE: Dönmez, E. & Özcan, A. (2021). Multi-Camera Vision-Based Mobile Robot Control and Path Planning. *European Journal of Science and Technology*, (31), 890-905.

Abstract

Visual-based controlling concerns how to manage a robot by using image-based features obtained from imaging devices. There are two types of configurations; internal vision configuration – IVC (the camera is in the device) and external vision configuration – EVC (the camera is out of the device). A multi-camera process with EVC has been employed to generate input for modeling behaviors of a mobile robot in this study. Gaussian-based controller and adaptive artificial potential field (A-APF) have been utilized for control and path planning processes, respectively. Four webcams with the same specifications are operated from the same height to control the mobile robot within a specific area. Images are stitched by using identic features of intersection areas in the first stage. Then, the color-based object detection is performed on this image to detect positions and other information related to the target, robot ,and obstacles. In the next step, to obtain a feasible and safe path plan between target and robot, an adaptive potential field algorithm is performed. Then, the controller is used to model the robot motions according to the path plan. In each control iteration, only one camera is activated according to the local position obtained from the global position of the moving robot. The simulation and physical experiment results demonstrate that the multi-camera configuration provides good performance and efficiency.

Keywords: Visual-based control, Gaussian control, A-APF, Multi-camera, Image stitching.

Çok Kameralı Görü Tabanlı Mobil Robot Kontrolü ve Yol Planlaması

Öz

Görsel tabanlı kontrol, görüntüleme cihazlarından elde edilen imge-tabanlı özellikleri kullanarak bir robotun nasıl kontrol edileceği ile ilgilidir. İki tür konfigürasyon vardır; dahili görüş yapılandırması – IVC (kamera cihazdadır) ve harici görüntü yapılandırması – EVC (kamera cihazın dışındadır). Bu çalışmada bir mobil robotun davranışlarını modellemek için girdi oluşturmak için EVC ile çok kameralı bir süreç kullanılmıştır. Kontrol ve yol planlama süreçleri için sırasıyla Gauss tabanlı denetleyici ve uyarlanabilir yapay potansiyel alanı (A-APF) kullanılmıştır. Belirli bir alanda mobil robotu kontrol etmek için aynı özelliklere sahip dört web kamerası aynı yükseklikten çalıştırılır. Dört kameradan alınan görüntüler ilk aşamada kesişim alanlarının ortak özelliklerine göre dikilir. Ardından, robot, hedef ve engellerle ilgili konumları ve diğer bilgileri tespit etmek için bu dikilmiş görüntü üzerinde renk tabanlı nesne tespiti gerçekleştirilir. Bir sonraki adımda robot ve hedef arasında uygun ve güvenli bir yol planı elde etmek için uyarlanabilir potansiyel alan algoritması yürütülür. Daha sonra Gauss tabanlı mobil robot denetleyicisi, robot hareketlerini yol planına göre modellemek için kullanılır. Her kontrol yinelemesinde, hareket eden robotun global konumundan elde edilen yerel konuma göre yalnızca bir kamera etkinleştirilir. Deneysel simülasyon ve gerçek dünya sonuçları, çoklu kamera konfigürasyonunun iyi performans ve verimlilik sağladığını göstermektedir.

Anahtar Kelimeler: Görsel tabanlı kontrol, Gauss kontrolü, A-APF, Çoklu kamera, Görüntü birleştirme.

[*] Corresponding Author: emrahdonmez@bandirma.edu.tr

1. Introduction

Motions of a mobile robot is usually modeled by utilizing environment information gathered from onboard encoders, distance sensors, or imagining sensors (external and internal sensors) (Siciliano, 2008), (Dönmez, 2018). Imagining sensors capture visual information in the images. Then, visual features are obtained by employing image processing approaches. Visual-based mobile robot control is a popular researching area in robotics. The main motivation is managing the robot by using visual features obtained from its surroundings. The internal vision configuration (IVC) based system is used to detect positions of objects by utilizing visual features, distance, and depth information. The second approach is external vision system (EVC) providing to detect information about the environment. In external vision configuration (EVC), unique features of the object (target, robot etc.) can be tracked and according to these features, motion of robot is modeled.

In a visual-based control system with EVC configuration; the cameras capture visual information like low, mid, or high-level features from the working environment. The system tracks the changes in the consecutive image frames. Each tracked change may affect the robot's motion according to the control system design. The image processing and control tasks are generally performed on an external computer system rather than robot hardware in this configuration. All the sensor tasks (measuring the distance, position, etc.) are performed through the imaging sensors in visual-based control systems. Therefore, internal or external sensors to measure distance or position aren't needed in general. In terms of performance, objects are tracked with high speed by using optimized image processing algorithms in visual-based control. Path errors mainly stemming from slipping, colliding, and falling or sensor reading are highly prevented in visual-based control. Because the controller knows the exact position of the robot in each control iteration. Thus, it can be said that a visual-based control system provides a good balance between performance and cost.

The other important component of visual-based control is path planning. It is one of the basic parts of the robot control process. It simply concerns modeling a path plan between an two position configurations (Choset, 2005). A path is created by considering the obstacles (wall, door, any object, etc.) in a configuration space. The main factors about planning a path are path safety, working efficiency, and admissible path cost, (Cowan, 2002), (Dönmez, 2020). Path planning is performed according to problem structure. Two types of approaches are global and local methods. They are used to extract a path plan from a given environment (Dudek, 2010). Global approaches are divided into two categories; retraction methods and decomposition methods. The retraction methods recursively reduce the dimension of the initial problem by taking into account a certain part of the configuration space. Decomposition methods characterize the obstacle-free regions of given configuration space. In addition, distance value are mainly used in local approaches while avoiding obstacles. This distance (or the cost function gradient) generally guides the local method. To overcome complex robots, the local approaches are more efficient. Except for these two approaches, path planning can be performed with stochastic or randomized methods. These methods generally utilize building a graph and find a local minimum at each iteration. In addition to the configuration space, path planning can be considered in the trajectory space. In this space, a straight line is created between the initial and final configuration while all of the obstacles are neglected. In the next step, this path is progressively reshaped by reducing and fixing improper parts of the acquired path. Planning a path can be performed with graph-inspired approaches like Dijkstra (Tsitsiklis, 1995), (Cormen, 2001), A* (Hart, 1968), D* (Stentz, 1994), randomized methods like RRT, esiBi-RRT (Lavalle, 1998), (Lavalle, 1999), potential field inspired APF method (Rimon, 1992), partitioning, hierarchical partitioning, etc. Each method has advantages or disadvantages according to the configuration space specifications.

In section 2, related works have been given. Problem definition is defined and preliminary information is organized in section 3. We touch details of material and methods including image stitching, object detection, kinematics, potential fields, and controller design in section 4. In section 5, both simulation and real-world experiments have been performed and explained. Conclusion and future works have been given in section 6.

2. Related Studies

Visual-based robot control is investigated in a remarkable number of researches. Decreasing errors and improving speed and efficiency are generally the main focusing points in these studies. There are several configurations to implement a visual-based robot control infrastructure. Malis et al (Malis, 2000) have improved the fundamental visual servoing approaches to the use of multiple cameras tracing a number of segments of an entity. The multi-camera visual-based control has been developed as a chunk of the task cluster approach. They expressed that the design of the control and the analysis of stability has been facilitated by allowing specific selection of the task function. Lippello et al (Lippiello, 2005) has introduced a visual servoing process capitilazing a mixed eye-to-hand/in-hand multi-camera system in their study. It is claimed that depending on a modified Kalman filter, this approach utilizes the data ensured by all the cameras without “a priori” distinction, permitting real-time estimation of object position. Qiu et al (Qiu, 2006) have proposed a multi-camera configured visual servoing system. The designed system uses switching the vision system between the eye-in-device and stereo cameras with a voting process. They stated that the proposed infrastructure enables process in a more generic variety of situations than that of either eye-in-hand or stereo camera single configuration. Yoshitata et al (Yoshitata, 2007) proposed a visual control design that allows a mini-helicopter to hover under local and temporal occlusions. Two fixed and upward-looking imagining devices observe four black balls fixated to rods connected to the lower side of the helicopter. They have said that the designed structure can hold the helicopter in a resolute hover. Iwatani et al (Iwanati, 2008) proposed a visual servo control system using multi-camera for unmanned micro aerial vehicles. The cameras are placed on the floor, and they are connected through a network. They claimed that the controller is durable against occlusion, and the helicopters can move easily and freely in the field of view of the camera. Weber and Kühnlenz (Weber, 2010) have utilized triangulation of images obtained by multi-cameras pointing in different directions to manage a robot with position-based visual servoing (PBVS). The presented triangulation is deployed by a linear method which provides real-time operating and high accuracy. Kermorgant and Chaumette (Kermorgant, 2011) have introduced a sensor fusion model for positioning of a robot with

multi-sensors. To realize an image-based visual servoing task, two cameras are used in eye-in-hand and eye-out-hand configuration. It is said that the sensor data fusion design is activated by a versatile comparison. Elsheikh et al (Elsheikh, 2016) has recommended an application and practical results of dynamic path planning and robot navigation for a mobile robot by using visual based controlling in an indoor environment. It is said that power consumption is balanced, short locomotion distances for the robot are anticipated and consequently enhance the overall time. Aliakbarpour et al (Aliakbarpour, 2014) have introduced several contributions on controlling a mobile robot operating a generic camera system. It is claimed that by using a basic radial model, the recommended visual servoing technique can be operated for commonly known type of general camera, both non-central and central. Ahlin et al (Ahlin, 2016) have proposed a leaf grasping system using a robotic manipulator in an unstructured environment by using deep learning and visual servoing. They said that Monoscopic Depth Analysis (MDA) enables a random number of features in unknown geometric characteristics. Alepuz et al (Alepuz, 2016) have exhibited a visual-based controller to fulfill a robot manipulator guidance. The manipulator has an eye-in-device camera configuration and is placed to a base satellite. The base is entirely independent and floating in space without attitude control. They said that by taking into account dynamics and kinematics, the controller system permits the robot to accomplish a defined location from an starting point and implement the tracking of the desired trajectory. Dönmez et al (Dönmez, 2016) introduce a visual servoing method. They use the decision tree method and graph-based input calculation to design controller functions. They claimed that the proposed method provides efficient and robust motion modeling. Dirik et al (Dirik, 2016), (Dirik, 2020) proposed a decision tree controller taking positioning inputs according to a triangle scheme. They use an eye-out device configuration and a wheeled mobile robot. They only model the go-to-goal motion model for a mobile robot. Dönmez et all (Dönmez, 2017) introduced an adaptive artificial potential field method that dynamically manages the parameters according to the target distance and available sensor data. They use image processing techniques to detect obstacles, robots, and target positions. They compare adaptive potential field and conventional methods and emphasize the notable achievements. Dirik et al (Dirik, 2017) present a fuzzy logic-based static path planning method. They use a fuzzy-based controller to extract a path plan on acquired image space. Floor vertical camera (FVC) configuration has been used within an external computation unit.

The majority of the investigated studies have been carried out without considering the number of cameras in the configuration space in general. The stereovision systems with double cameras have been generally focused on in these studies. Multi-camera configured systems similar to this study are commonly designed by utilizing Position-Based Visual Servoing (PBVS) method. IVC studies are more researched compare to EVC studies. A simple and easy-to-use multi-camera configuration is implemented in this study. We have used four cameras to extend the configuration space. The process of image stitching is employed to generate a binary map including entities of the working environment. The position of the robot is determined in consecutive image frames by using simple object detection methods. Both simulation and real implementation results have been implemented and results are analyzed.

3. Preliminary Information

A multi-camera configured visual-based control design hosts a couple of problems in its essence. The first problem is the image stitching procedure; it is a time-consuming task because of high mathematical complexity. It is not possible or impractical to apply it in real-time visual-based control. Therefore, this task is performed once before extracting the path plan. By using common features acquired by a feature detector method, images are stitched through intersection areas where common features are dominantly evident. After the stitching process; robot, target, and obstacle positions are detected and a path plan is extracted from this new map configuration. The second issue is dividing the acquired path plan according to the camera intersection areas and aligning these paths to the related camera as local paths. The entire path has global coordinates acquired from the stitched image. However, these coordinate values cannot be used in this state. Each coordinate has to be updated according to the local coordinate system of the related camera. After the update step, the robot is controlled by taking and processing consecutive images from the related camera (The related camera is the device that sees the robot momentarily). But, this time a new problem appears; how to make the transition from one camera to another where the path continues. In other words, the camera seeing the continuation of the path has to take over the robot tracking task. The camera changing simply can be modeled according to the intersection areas. When the robot appears in these intersection areas, the other camera where the local path as next part of the global path continues, start tracking the robot and the previous camera is stopped. Then the next camera is activated, the target position is updated by taking into account the end of the path in the local area. This process continues until the mobile robot successfully reaches the target position.

Such a working mechanism may not be needed if separate image processing units are employed. But, this time such a design will dramatically increase the cost of the control system. The multi-camera system ensures several advantages in terms of robotic control systems. Therefore, in experiments, the whole images of different configuration spaces are unified with the image stitching method. In the next step, a binary map of the whole image is obtained and required objects in this image are found with image processing. Then the multi-camera-based controller is employed to control the WMR until reaching the target position. Results of control processes are utilized to compare experiments in the aspect of performance, advantages, and disadvantages.

4. Material and Method

There are three stages to construct a multi-camera-based mobile robot control infrastructure in this study. In the first stage, configuration space, camera calibration, image processing, and controller tasks are modeled. In the second stage, the path planning task is performed and the acquired path is resampled. In the third stage, multi-camera hardware is utilized to control WMR in each iteration. These stages are illustrated in Figure 1.

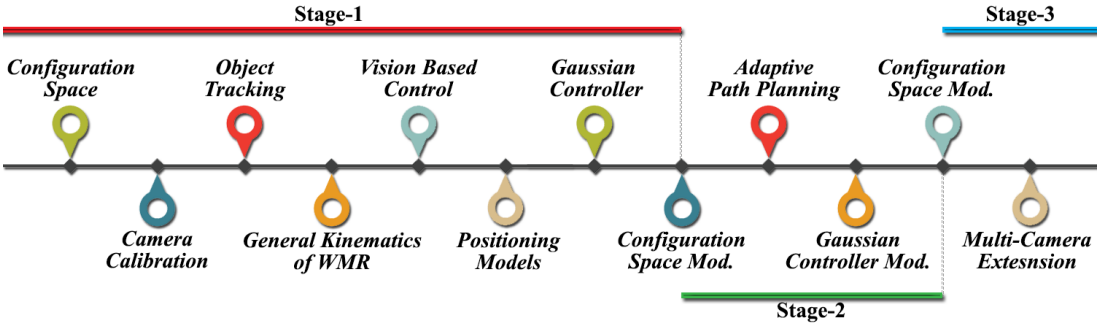

Figure 1 Stages for the multi-camera control model

4.1. Configuration Space

There are several setups for EVC infrastructure; the camera can be horizontally, vertically, or diagonally placed relative to the floor with a single or multi-camera option. In this study; four cameras have been vertically placed to the ceiling, thus the floor is seen by them as bird's eye view. Since the utilized vehicle is a wheeled mobile robot (WMR), this camera configuration ensures an applicable infrastructure.

Configuration space has been observed through four cameras that have the same specifications. All cameras have been placed to a fixed position with the same heights from the ground. Figure 2 demonstrates working space representatively. The brown shapes represent the obstacles. Cameras are expressed with C1, C2, C3, and C4. The viewing area for C1 consists of the multiplication of C1_x and C1_y. Intersections of the viewing area of two cameras are C1-C2, C1-C3, C2-C4, and C3-C4. There is one area (C1-C2-C3-C4) where the viewing areas of all cameras are intersected. This situation means that there are several areas jointly tracked by different cameras. In addition, the remaining areas are observed with only one camera. Blue objects (T1, T2) represent the target positions. Red objects (I1, I2) represent boundary positions where the robot starting to enter or exit (from) the viewing area of two cameras at the same time on the path. Rp1, Rp2, and Rp3 are reference points between two obstacles. They are only utilized for showing the shortest path (desired path) and have no functions in path planning tasks.

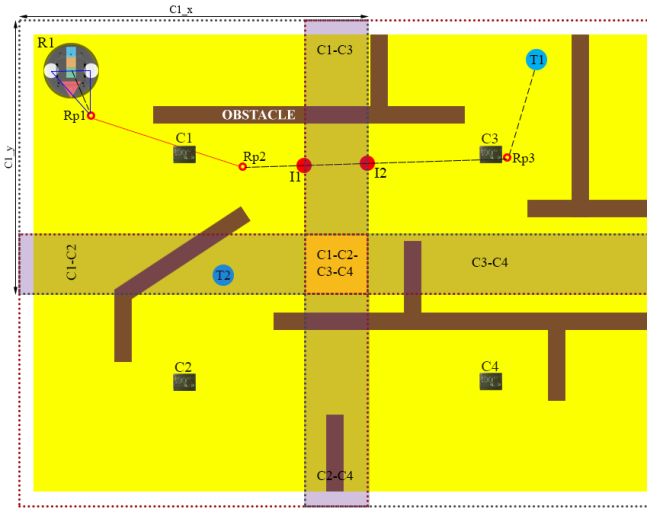

Figure 2 Working environment for the designed control system (Representative)

4.2. Image Stitching

Image Stitching is one of the particular studying fields that are commonly researched and it hosts several problems needed to be overcome. Generally, there are two basic goals; (a) superimposing the images taken from the same position and different angles and (b) integrating identic intersection points in most convenient form. The main problem with image stitching is the component size difference between the region x and x' because of the difference between angles, Figure 3 (Brown, 2003). In the equations (Brown, 2003) (1), (2), (3), and (4), the components connected by x have been shown;

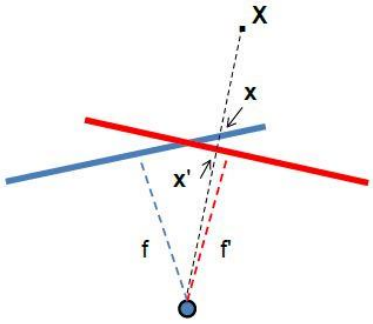

Figure 3 Changes for parameters due to angle

$$x = K[rt]X \tag{1}$$

$$x' = K'[r't']X' \tag{2}$$

$$t = t' = 0 \tag{3}$$

$$H = K'R'R^{-1}K^{-1} \text{ to be, } x' = Hx \tag{4}$$

In equations, K and K' are the matrices used for measurements (calibration). The parameter X is the real object location. x and x' represent the object's different positions stemming from another angle shooting. R and R' are matrices used for rotation procedure. t and t' are matrices for translation procedure. Figure 4 demonstares the difference of distances between the components. This difference originates from the camera angle. The red dot is common to both images. A component from the second shoot has been pointed by a green-colored dot.

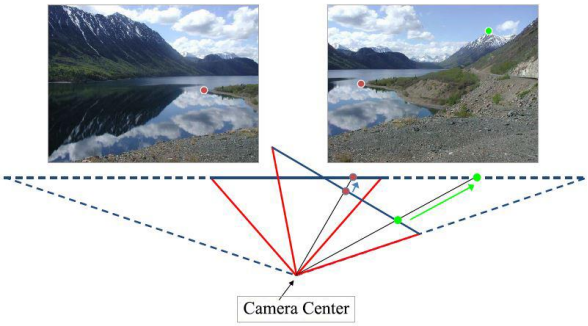

Figure 4 Photo shots taken from the same camera center and different angles (Photo: Russell J. Hewett)

In this study, the input images for the stitching process, are obtained from the four cameras that are vertically placed to the ceiling. These images are mapped horizontally or vertically relative to the camera locations by superimposing common areas with the next intersecting camera image. Although this process is similar to creating a panoramic image, it is different from each other in terms of the location in which the image is taken. Input images for a panoramic image are generally taken at different angles with a single camera from the same camera center. On the other hand, within this study, input images are obtained from multiple cameras with different camera centers and the same angle (perpendicular to the surface). Generally, creating a panoramic image with images taken from the same spot is more prone to distortions in the image. This situation stems from the difficulty of matching the intersection points of the images because of the different angles of the input images. If the matchings at these intersections are not sufficient, the stitching success at the relevant region will be low and visible distortions will occur. In images taken consecutively in the direction of both x and y coordinates from the central view, the presence of common intersection areas closest to each other due to the shooting angle is an enhancement factor.

The performance of a multi FVCs is promising since the images are captured within the identic angle. Thus, the matching ratio of intersection regions is very high, the stitching process can be initiated from any reguired image. The SURF (Bay, 2008) or SIFT (Lowe, 2004), (Gonzalez, 2002) can be utilized to detect the key points when image properties are extracted. In this study, the SURF detector has been used. The pseudo-code executed for the stitching process is as follows:

1. Input images (2) as parameters. G (n) and G (n-1).
2. Feature extraction by SURF detector.
3. Compute the matching points.
4. Implement the RANSAC to presume a transforming for homography then overlap the image spots, T (n).
5. Transform the images utilizing the homography.
6. Perform the stitching operation.
7. The whole process is repeated to stitch the following image.

Image properties are detected and matched as $G(n)$ to $G(n-1)$ – (common intersection regions). The SURF features are extracted from the black-and-white transformation of the first image. Because the images are close enough to the camera, a projective conversion is used. If pictures are farther away, an affine transformation is used. Then, in the iteration process, the SURF properties of the $G\ (n)$ image are extracted. Matching of these extracted properties between $G(n)$ and $G(n-1)$ is performed. The geometric transformation of $T(n)$ mapped from $G(n)$ to $G(n-1)$ is calculated by the RANSAC method [30] using property mappings and taking the previous property mappings as parameters. Transformations mapped from $G(n)$ into panaroma image/view $T(1)*\cdots*T(n-1)*T(n)$ are calculated. It is obtained by multiplying itself and the previous transform together. Moving from the situation that "the center of the captured scene exhibits the least distortion", a good panorama can be acquired by changing the transformations. These changes are performed by inverting the transform for the central image. Then this transform is applied to the other components. This can be neglected, however, since multi FVCs all receive images from the same vertical angle and different positions. Because angle-induced distortions hardly ever occur.

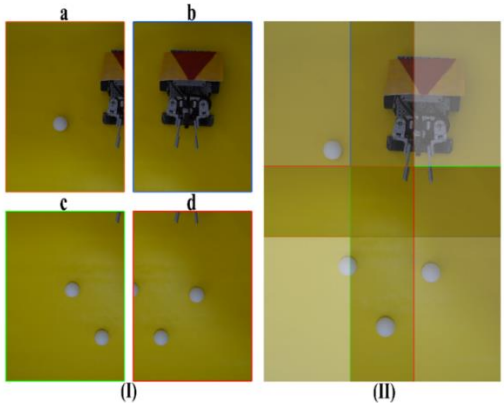

Figure 5 (I) Images obtained at the same angle from different camera positions (II) stitched the state of four-images

In Figure 5 (I), four images (a, b, c, d) taken from the different positions and same angles are stitched on common intersection points. It should be noted that all the cameras have the same specifications. (II) The opacity values have been changed so that the stitched areas in the images look better.

4.3. Image Stitching

Object detection and tracking task are performed by utilizing color-based segmentation. To improve the quality of the detection process color quantization (Mota, 2001) is used. To detect object positions, simple color labels are used. To detect orientation, a triangle-shaped red label is employed. The form of the robot positioning scheme with related components (robot and target labels) is illustrated in Figure 6 (a). Angle values in Figure 6 (b) are calculated to be used as input for the control process.

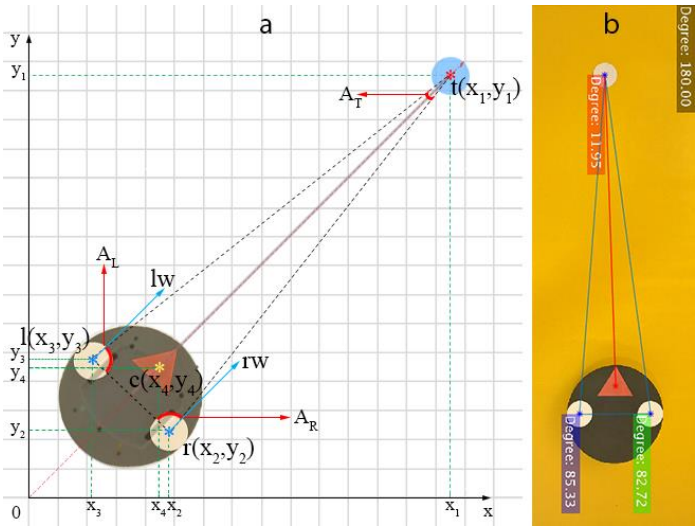

Figure 6 (a) Positioning mode for robot, (b) Physical world

The white circular labels (l, r) represent robot wheels, red triangle label (c) is utilized to acquire information about robot direction and the blue label (t) is the target. After the centroids have been computed, the regions are separated based on color. Each separated object linked corner points of the triangle where lines are intersected. These centroid and angle values are computed for each image obtained from the camera in real-time. After that, all angle and centroid parameters are assigned and new inputs for the control step are recalculated. The velocities of wheels are calculated and sent to the robot in each update iteration. The positioning model scheme for triangle form is illustrated in Fig. 6. The "c, l, r, t" asteriks are centroids of the labels mentioned previously. In addition, 'AR, AL, AT' denote the Right-Left wheel, and Target Angles, respectively. To adjust angular parameters being input to the controller, internal angles of the virtual triangle are used. Whenever the robot moves under FVC, the coordinate system in 2D (x-y) space is updated periodically using the angle changes. Ultimately, velocity parameters are accurately calculated by using these angle values.

4.4. Adaptive Artificial Potential Field Principles for Path Planning

Attractive Potential Field (P_{att}): It is an attraction field that is created by resultant of attractive forces in a region. It is used to pull the object to the target position in configuration space. The magnitude of this force generally maintains a fluctuating pattern until the robot reaches the predefined position. Repulsive Potential Field (P_{rep}): It is a repulsion field that is created by the resultant of repulsive forces in a region. It pushes the object to avoid obstacles and unseen locations. The magnitude of this force generally demonstrates a variable character according to the global positions and distance values between moving objects and obstacles. Gradient vectors are utilized to define these forces and they are specified as magnitude and direction. The mentioned forces are shown in Figure 7 for an obstacle-hosted configuration space. F_{att} is the attractive force, F_{rep} is the repulsive force and F_{total} is the resultant of these two forces.

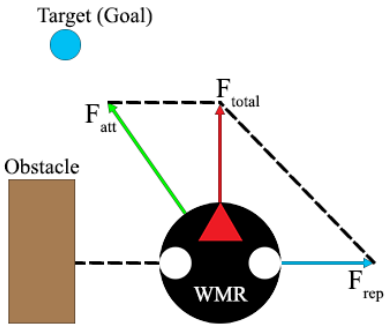

Figure 7 Potential field forces

The attractive field between robot and target is created to pull the robot to the target position. Attractive potential field (5) formed by target and robot parameters is found by the following equation;

$$P_{att}(q) = \begin{cases} \frac{1}{2}\zeta d^2(q,q_h), d(q,q_h) \le d*_h, \\ d*_h\,\zeta d(q,q_h) - \frac{1}{2}\zeta {d*_h}^2, d(q,q_h) > d*_h, \end{cases} \quad (5)$$

$d(q,q_h)$ is the current distance between the robot and the q_h target. ζ is attraction gain and $d*_h$ is the threshold being the next value of the conical form of second-order function. Repulsive potential field (6) formed by obstacle and robot parameters is found with the equation below;

$$P_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q)} - \frac{1}{Q^*}\right)^2, \rho(q) \le Q^* \\ 0, \rho(q) > Q^* \end{cases} \quad (6)$$

In this equation, η states repulsion gain, Q^* is the distance threshold creating a repulsive force on the robot for an obstacle. $q_c = (x_c, y_c)$ is a unique configuration in the nearest obstacle to q. $\rho(q) = \|q - q_c\|$ is the shortest path between the robot and the obstacle. If the distance between the robot and the obstacle is larger than Q^*; repulsive force is not applied to the robot (object). The ζ and η gain parameters and the d* and Q^* threshold parameters are adjusted empirically in general. Similarly, the repulsive force is the gradient of repulsive potential function.

There may be a significant number of obstacles in the configuration space. The total repulsive potential is the sum of the repulsive potential fields stemming from all these obstacles. The total potential field is simply expressed with the following equality (7);

$$P(q) = P_{att}(q) + \sum_{i=1}^{n} P_{rep}(q) \quad (7)$$

In this equation, n express the number of obstacles. The total artificial force can be formed with the following term (8);

$$F(q) = F_{att}(q) + \sum_{i=1}^{n} F_{rep}(q) \quad (8)$$

To adjust scaling factors of artificial potential field functions following (9), (10) equations are used. $d(q,q_o)_{d_{af}}$ express the distance values acquired from virtual sensors. These virtual sensors are located at the front, front-left (between front and front-left-diagonal), and front-right (between front and front-right-diagonal). $d(q,q_h)$ represents the distance between robot and target. The repulsion force scaling factor 'η' is increased with decreasing of the d_o distance (distance to an obstacle) in configuration space. Similarly, the attraction force scaling factor 'ζ' is increased with decreasing of the d_t distance (distance to the target) in configuration space.

$$\eta = \eta + \eta * \gamma \left(\frac{1}{\sqrt{d(q,q_o)_{d_{af}}}} \right) \Rightarrow d_o \downarrow \quad (9)$$

$$\zeta = \zeta + \zeta * \delta \left(\frac{1}{\sqrt{d(q,q_h)}} \right) \Rightarrow d_t \downarrow \quad (10)$$

4.5. Gaussian Controller Method

The mobile robot control method is designed by inspiring from the basic Gaussian function. A cost-efficient vision-based control system is modeled with it. The function is utilized to determine parameter changes for the robot motion controller. The Gaussian simply ensures characterizing the error elimination model for a mobile robot navigation infrastructure. Therefore, this function has been selected because of providing efficient and highly accurate outputs. Position and orientation parameters generate control inputs that are used for modeling wheel dynamics within the controller. The general single-dimensional Gaussian function is given in (11).

$$f_G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (11)$$

$$x = \left| \frac{A_L - A_R}{\varphi} \right| \quad (12)$$

$$S_A = S_{max} * (1 - f_G) \quad (13)$$

$$S_L = \begin{cases} P_{max} * \tau + S_A, & A_L < A_R \\ P_{max} * \tau - S_A, & A_L > A_R \end{cases} \quad (14)$$

$$S_R = \begin{cases} P_{max} * \tau + S_A, & A_L > A_R \\ P_{max} * \tau - S_A, & A_L < A_R \end{cases} \quad (15)$$

$$S_{L,R} = P_{max} * \tau + S_A, \quad A_L \cong A_R \quad (16)$$

The input parameter x (12) is the difference of angle parameters ($A_L - A_R$). It corresponds to the θ angle in general WMR kinematics. The error can be estimated by smoothing the data with a large Gaussian kernel (σ = 10). The overall standard deviation is fixed to '0.41' in the range of 0 – 1. This value has been found empirically for the designed controller model.

If the difference of input parameters approaches zero ($A_L - A_R \Rightarrow 0$), then it means that robot direction is straight toward the target and error is almost equal to zero. φ value is utilized as a scaling factor for the input value. If x value approaches '0', then f_G converges to '1' and if x value approaches to '∞', then f_G converges to '0'. This situation is inversely proportional to the desired result. S_A – "Speed Additive" is used to weight the velocity parameters and it is computed by using (13). We have used (14), (15) and (16) equations to determine S_L – 'Speed of Left Wheel' and S_R – 'Speed of Right Wheel' of the mobile robot. P_{max} is the maximum pulse value adjusted for the velocity calculation of the robot. 'τ' is used as a constant scaling factor.

4.6. Robot Control in Multi-Camera Configuration Space

All designed controllers and positioning models are generally modeled with a fixed single FVC configuration. In this study, to create a scalable configuration space; four cameras have been used. These cameras acquire image frames from configuration space as shown in Figure 8. These input images are stitched to get a single image of the area. After this stage, object detection is performed on this single image, so that the binary map is obtained. Next step, the path plan is extracted from this binary map with A-APF (Adaptive Artificial Potential Field). Then, the controller method drives WMR on this path in real-time until the robot reaches the target.

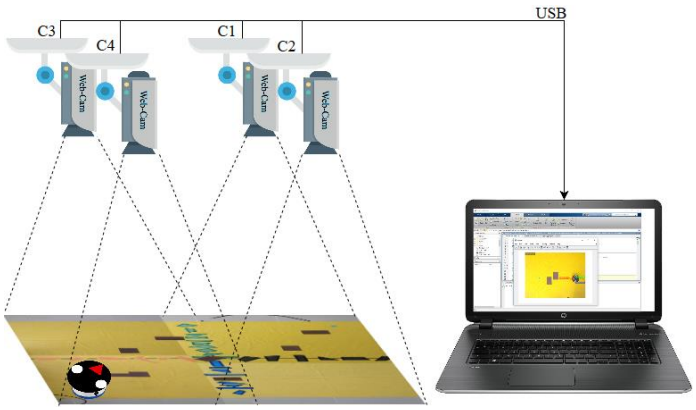

Figure 8 Multi-camera – Computer connection and configuration space

In the first stage, a path is acquired from the stitched image. It is stated with the term S_{path}. This path parameter includes path coordinates obtained from the A-APF process. At the next stage, this variable sequence is divided (17) according to the number of minimum required cameras. $S_{x_{path}}$ exemplifies a piece of path within the viewing area of camera x (C_x). The acquired path is partitioned according to the camera viewing areas where the path passes. For example; if the path plan is in the two cameras' field of view, it is divided into two pieces. The ends of these path fragments are used to model switching between the cameras. These endpoints are called intermediate targets. Only the end of the last path fragment (the main target) is not considered for this switching process. The robot starts to move according to the path fragment; firstly it tries to reach an intermediate target from its initial (starting position) position. When it arrives to the intermediate target, its available position is designated as starting position and the target is assigned as intermediate or main according to the available path fragments. Divided path coordinates are dispersed to the regions where path is hosted with the rescaling process by using (18). x_{new} and y_{new} are the new coordinates (x, y) that are required to compute in undistorted space of image. The width and height are X_{und} and Y_{und} in the undistorted image. The path coordinates are x_{old} and y_{old} in stitched image. The width and height of the distorted image are X_{dis} and Y_{dis}. By using (19) and (20) conditional statements, the path is controlled if the mobile robot appears in the intersected area. I_{nm} expresses the intersected region; this parameter receives 1 or 0, the value 1 signifies that the robot is

in I_{nm} intersected region. C_{act} expresses the active imaging device; it has to be equal to one of the camera ($C_{n/m}$) in the environment. If the robot is not in the intersected region then the currently working camera (C_{pre}) is assigned to the C_{act}. I_{all} exemplifies the intersected region where all imaging device can mutually track. t_{x_i,y_i} represents the target coordinates in the 2D image space.

$$S_{path} \Rightarrow S_{1_{path}}, S_{2_{path}}, \dots, S_{n_{path}} \wedge 1 \leq n \leq 4 \wedge n \in \mathbb{Z} \quad (17)$$

$$x_{new} = \frac{x_{old} * X_{und}}{X_{dis}} \wedge y_{new} = \frac{y_{old} * Y_{und}}{Y_{dis}}, \forall (x_i, y_i) \wedge (x_i, y_i) \in S_{path} \quad (18)$$

$$I_{nm} = 1 \; ? \; C_{act} = C_{n/m} : C_{act} = C_{pre} \quad (19)$$

$$I_{all} = 1 \; ? \; C_{act} = C_n \Leftrightarrow \exists \, t_{x_i,y_i} \in C_n : C_{act} = C_{pre} \quad (20)$$

The whole multi-camera-based control process flow is illustrated in Figure 9. After acquiring the path plan, the controller triggers the motion of the mobile robot. The robot is tracked with the related camera where it is in the field of view. When the robot moves towards the field of view of another camera; firstly, the control process is preceded until the robot fully appears in the intersection region. According to related cameras viewing this intersection region, the related FVC is activated concerning the view field of the other camera where the next part of the extracted path is located. When this second camera is activated, the first camera is deactivated instantly. This means that only one camera is activated at a time. This process continues until the mobile robot reaches the position of the main target. At each intersection point, the next camera takes over the tracking process, and the control task proceeds in this local configuration space. The next camera is the device covering the continuation of the path part. The extracted path from the whole map can be in one camera field of view or more. This situation changes according to the distance of the target, obstacle positions, and robot position and direction. In each camera region, the controller is operated without any additional workload. The WMR is actuated by the controller when there is an intermediate or main target. The summary of the multi-FVC based control system is illustrated in Figure 10.

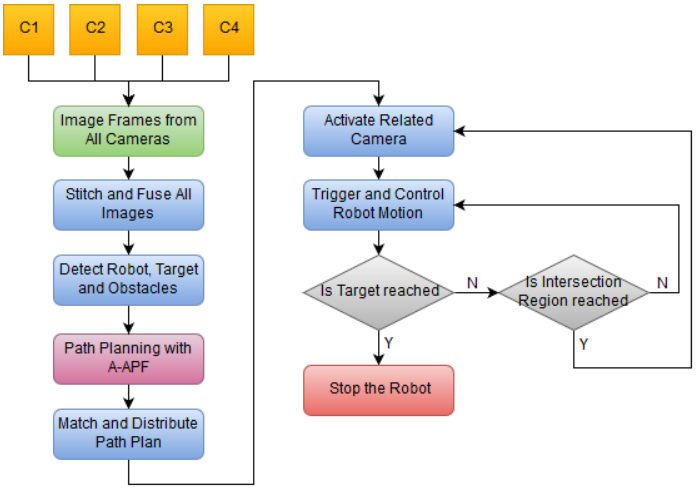

Figure 9 Multi-camera based control process flow of the designed system

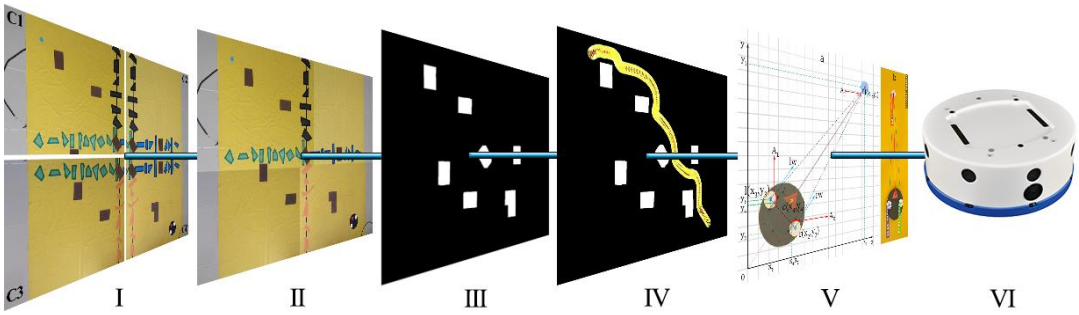

Figure 10 Summary of the multi-camera based control system: (I) Simultaneously acquired images from all cameras (II) Stitched image (III) Detected obstacles (IV) Extracted path plan between robot and target (V) Calculation of controller inputs (VI) Robot implementation

4.7. Configuration Space Properties

A multi-camera experiment has been performed with four webcam cameras. In our single camera-based previous work, a CCD camera is used for visual-based control. CCD camera provides more color depth compared to a webcam, but it is very expensive. Therefore, this cost is needed to be reduced for a multi-camera configuration. On the other hand, webcams are easy to use, platform-independent in terms of software and hardware, adaptable to any environment with proper configuration. Of course, CCD cameras have their specific advantages. However, such properties are not required for this configuration.

Each webcam is adjusted to Super-VGA (SVGA) resolution. Imaging devices (or cameras) have been placed to the ceiling in a way that their position perpendicular to the ground. Cameras have been placed about 210 cm from the floor. According to these camera positions each camera covers about 3.05 m2, so in total 12.20 m2 area may be processed for this configuration. It should be noted that all the webcams are the same models with the same specifications. Each imaging device is separately connected to the computer through USB ports. This physical configuration has been demonstrated in Figure 11 (I).

Several distinct shaped and colored labels have been emplaced on the floor as different properties for the utilized feature detector. The utilized SURF detector searches the distinctive properties and matches these properties in an image according to the given input image. The shape size and colors

have been randomly determined. The only important factor is the position of these labels. Each camera covers a single plain area including two axes labels according to camera position as shown in Figure 11 (II).

Figure 11 (I) Multi-camera configured real operating environment (II) Colored and randomly shaped labels on the operating floor

The webcam (C920) that is utilized for multi-camera configuration is demonsrated in Figure 12. It has 3.2MP maximum video resolution and 15MP image resolution. However, SVGA (800x600) is used for visual servoing tasks. Therefore, even more, basic webcams will be enough to deal with the indicated resolution. In terms of image quality and noise reduction, it is a good choice. On the other hand, it supports a 30fps frame capturing speed. They are fixed to aluminum support brackets with plastic clamps.

Figure 12 The webcam used in the multi-camera configuration

The positions of imaging devices are demonstared in Figure 13. IDs are given as C1 to C4 to the devices respectively. The camera viewing area has been indicated with the blurred area. An example viewing area is C1-X and C1-Y area for the C1 camera. Other cameras have similar viewing areas according to their positions. Red and blue regions corresponds to the common intersected areas for two webcams. The middle square area body forths the intersected region of all devices. It should be said that the black lines are the guidelines; they are not constraints for viewing area of tha cams.

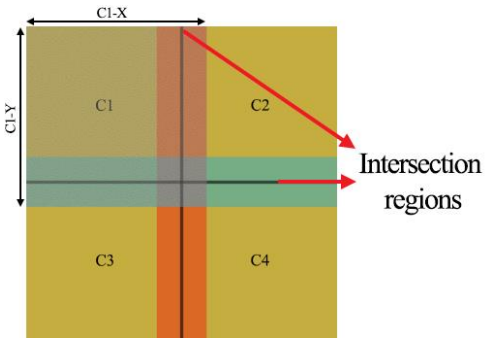

Figure 13 Camera positions and camera intersection regions

5. Experiments and Observations

5.1. Experiment (Conf-1)

Images captured via the imaging devices with FVC configuration are demonstared in Figure 14. The grayish areas on the left or right edges in the images are the real floor texture of the experiment environment. The ball caster wheels are too small and wheels of the mobile robot are quite thin. Thus, the plastic-material-based layer is utilized to prevent the wheel from jamming to the suture area of the floor tiles. On the other hand, the yellow color is one of the most sensitive colors against the illumination changes. The blue circular label is the target and the brown rectangle labels are obstacles. The remaining colored objects on the floor are utilized as the common intersection properties for the stitching process.

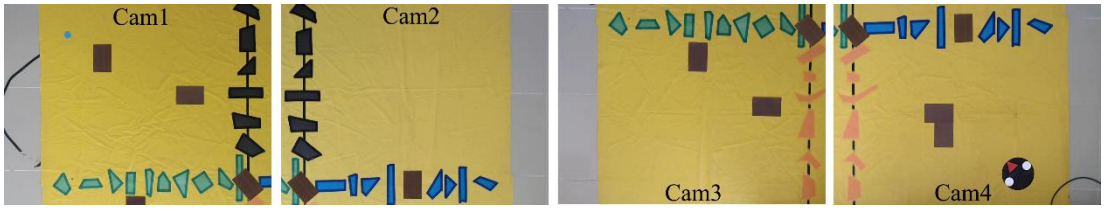

Figure 14 Real acquired areas covered by the cameras

The stitched image(s) are shown in Figure 15 (I) as the first configuration (Conf-1). The images are simply superimposed and re-scaled. To increase the accuracy of the utilized detector, color-based labels have been used. Hereby, images are precisely stitched to each other. The robot is in the C4 (Cam4) viewing area and the target is emplaced under the viewing area of C1 (Cam1). There may be a different level of shadowing in the images taken from cameras. Because of such differences, several negligible inconsistencies in stitched objects may emerge. They are negligible because all these errors are too small to be effective on path planning and visual servoing tasks.

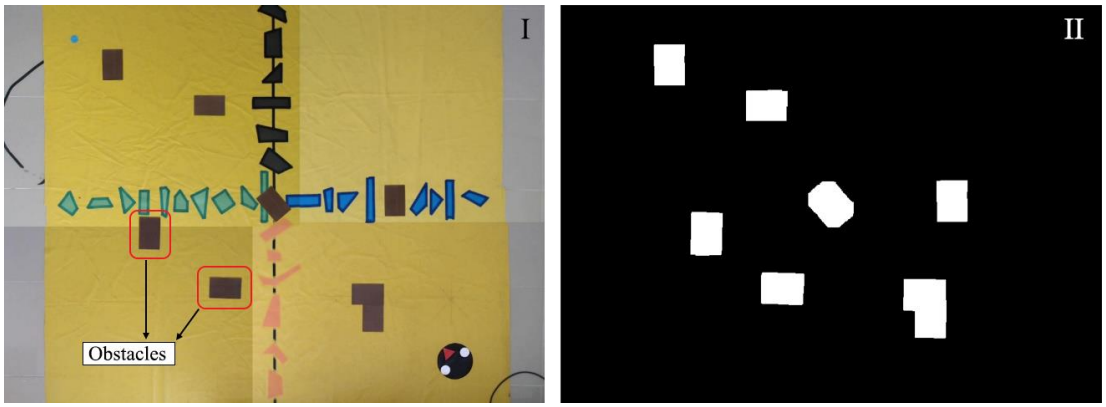

Figure 15 (I) The stitched image to acquire Configuration-1/Conf-1 (II) Obstacle map acquired from the stitched image

After acquiring stitched environment space, an obstacle detection task is executed on this space to extract the obstacle map, Figure 15 (II). Obstacles are detected and the environment is converted to the binary map. The value '1' is assigned to the obstacles and the value '0' is assigned to the remaining area. This task is known as 'Binary Image Acquisition'. Robot and target positions are also detected and stored for further processes. To increase path safety, object dilation is used on detected obstacles.

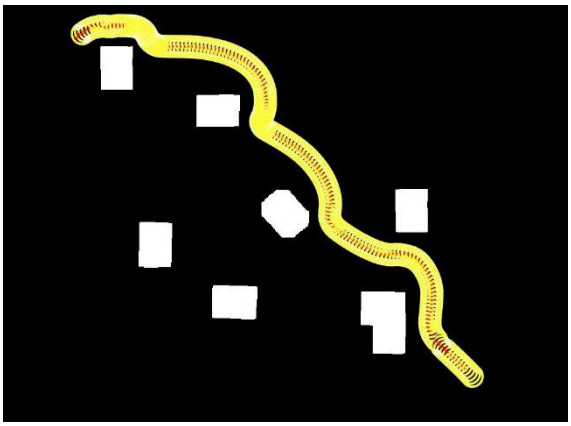

Figure 16 Simulation path with A-APF

The adaptive artificial potential field (A-APF) generates the path planning on the acquired map. 192 frames have been acquired in simulation, so 192 different position sampling is taken on the acquired path. These positions are used as input to a visual-based control process. Simulation has taken about 11,2s, so 17,142 frames per second are acquired. The path cost of simulation is calculated as 1037,53 px. The robot tries to approach the target position in each iteration by calculating the next suitable position. In Figure 16, the formed path by A-APF has been given.

The Attractive potential field (A-PF), repulsive potential field (R-PF), and total potential field (T-PF) forces against the processed frame numbers are shown in Figure 17 (I). Attractive potential field force has increased for a while from the initial state, then it has decreased until the robot arrived to the target position. On the other hand, repulsive forces show variable patterns until the task is finished. Total potential force parameters are computed by combining repulsive and attractive forces. As it can be seen, total forces are quite similar to the opposite direction values of the attractive forces. Therefore, attractive forces are the main influencers on the total force.

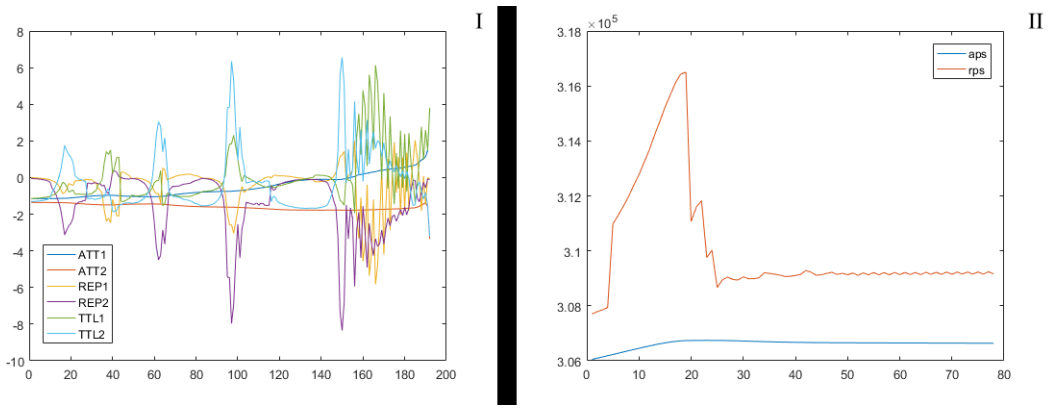

Figure 17 (I) Potential force and (II) Potential scaling factor parameter changes

The attractive and repulsive potential scaling factor (aps and rps) variations are shown in Figure 17 (II). The APS accretes by starting from the initial point, then it lessens with small steps as iteration continues. The RPS accretes aggressively firstly, then it lessens in a vertical manner. It approaches more stable states with some little fluctuations after a while. On the other hand, the potential calculating threshold value shows small changes and the minimum calculating threshold value shows no changes.

The acquired path plan has been used as a reference path which has to be followed by the mobile robot. The robot controller is triggered to make motions according to the reference path in real-time. A_R, A_L and A_T values are calculated as 73.81º, 69.19º, and 37.0º respectively at the first starting frame according to the intermediate target. At the end of the control task, these values were calculated as 59.29º, 61.41º, and 59.29º respectively. The robot has successfully reached the pre-defined target of about 10.5s. Starting/finishing positions of the robot are shown in Figure 18.

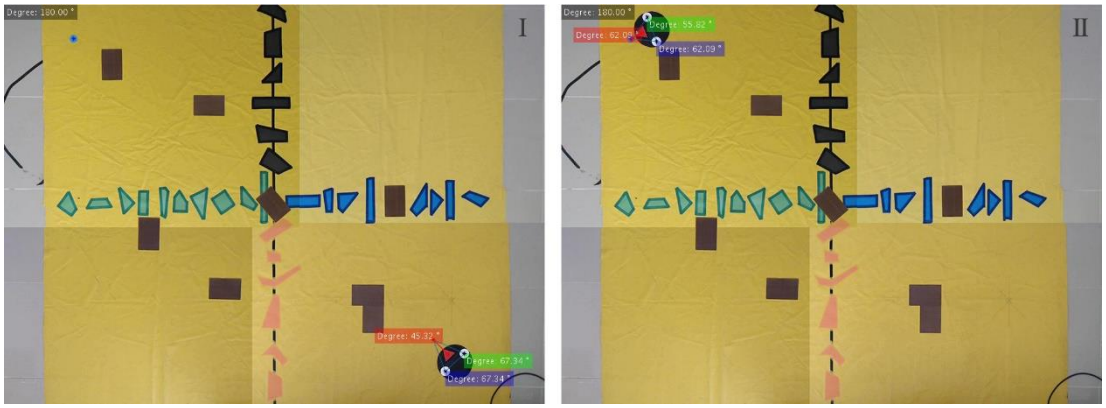

Figure 18 (I) Starting position and (II) finishing position of the mobile robot

Sample frames from the working environment, while the visual control process is performed, are exhibited in Figure 19. The 'f1, f2 … f8' frames represent distinct positions of the robot as time passes. In total 102 frames have been processed, so 9.714 FPS is achieved until the task is completed. Half of the total 102 frames (51 frames) are neglected and not stored in physical disk storage to keep performance at a stable level. (Even number frames are processed but not stored, only odd number frames are stored.)

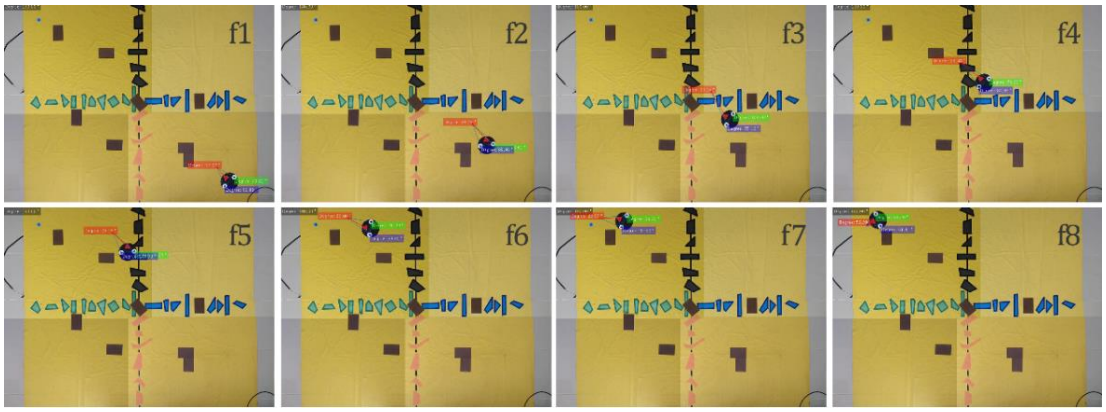

Figure 19 Sample frames from visual-based control task

The simulated path that has to be tracked by the robot (I) and real robot motions in several selected frames (II) are given in the following Figure 20. The mobile robot has tracked the path plan with the control system. The path created by robot motions until the target position is determined a little lesser than the simulated path. The primary reason behind this situation is the dynamic changing of local targets. The local target is determined according to the cartesian coordinates of the simulated path according a pre-defined threshold value until reaching the main/final target.

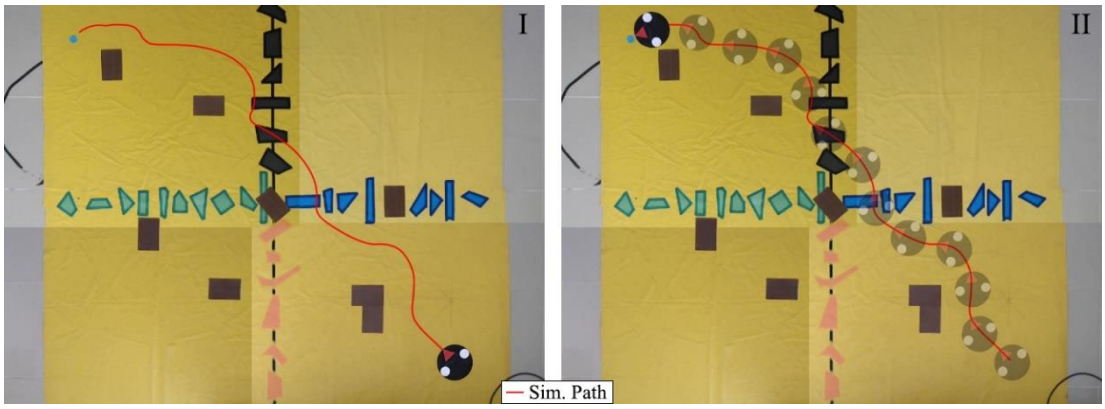

Figure 20 (I) Simulated path and starting position of robot (II) Simulation path and mobile robot motions in several frames

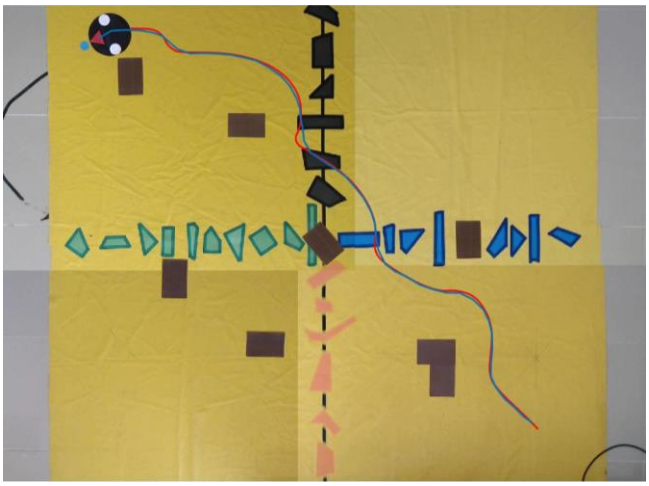

Figure 21 Simulation path (red) and Real path (blue)

Except for the start and finish positions, several intermediate positions of the robot are given in the figure. Eventually, the mobile robot has smoothly tracked the input path plan. Some error may appeared between the actual and simulation paths. However, this error is too small to affect cost, thus, this error can be ignored. The phycical/real path formed by the mobile robot has been given in Figure 21. As seen, the real path emerges a little less costly than the simulation path. The path lenght is detemined about 995,16px. Eventually, a 4% difference emerges between simulation and real paths.

The angle parameter changes of the control points (mobile robot wheels) and target are shown in Figure 22 (I). The intermediate destination position is continually inspected in control iterations. If it is required than, the target position is updated. The angle value have significantly rised when the mobile robot begins to perform rotation motions. The A_T, A_L and A_R angle parameters get close to the each other at the end of the control process.

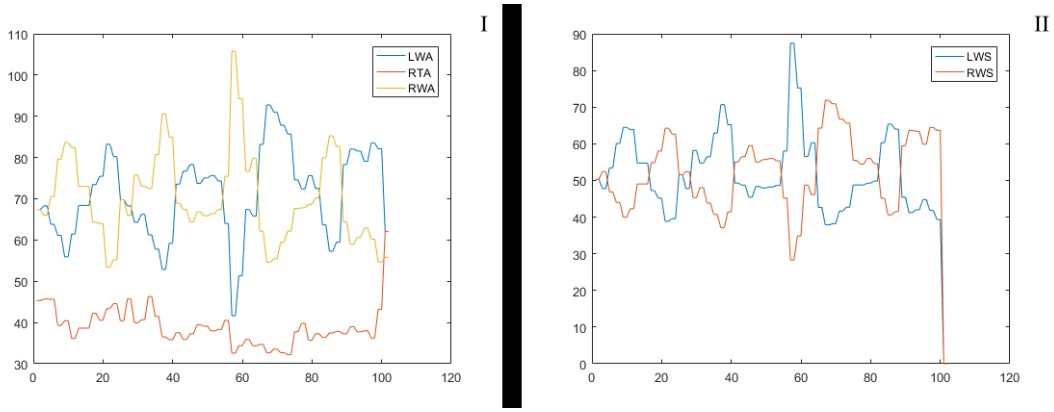

Figure 22 (I) The control point's angle changes (II) Left and Right velocity changes of WMR wheels

Velocity parameters of the right and left wheels are shown in Figure 22 (II). The changes in velocities similar to the A_L and A_R angle parameter changes with different amplitude. The primary reason is that the angle parameters affect the velocities of mobile robot wheels directly. The values of velocity and angle parameters are a bit erratic, since, two tasks are slowing down the system. These tasks are the controller sensitivity and the sample image frame storing to the disk.

5.2. Experiments with Different Configurations (Conf-2/3)

Experiments have been performed on two different operating environment configurations besides the previous configuration (Conf-1). The experiment environment and acquired path plan with the A-APF method are given in Figure 23 for Conf-2 and Figure 24 for Conf-3. Superimposing and rescaling processes are performed at different levels compared to the previous configuration. This situation generally stems from

the stitch order of captured images. Therefore, it can be said that the stitch order of images directly affects the final image output.

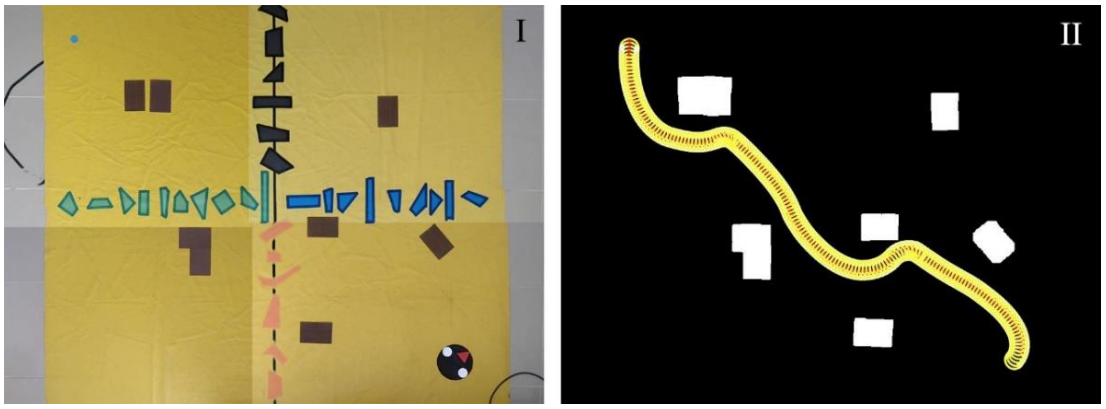

Figure 23 (I) Configuration-2 (Conf-2) and (II) simulated path plan

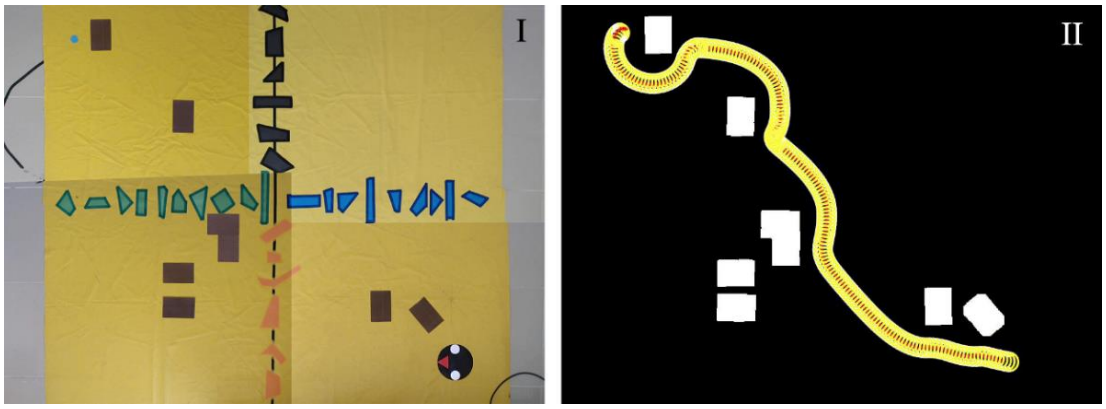

Figure 24 (I) Configuration-3 (Conf-3) and (II) simulated path plan

The mobile robot has successfully reached the pre-defined target under configurations with different obstacle alignments in simulation experiments. In each configuration images taken from the camera can be superimposed differently. The important principle is fusing common intersection areas with a high degree of precision. The starting and finishing positions with angle values by the mobile robot are given in Figure 25 for Conf-2 and Figure 26 for Conf-3. In Conf-2, the WMR has started (I) with $A_T = 14.94°$, $A_L = 142.30°$ and $A_R = 22.76°$ and reached to (II) the target with $A_T = 57.34°$, $A_L = 61.33°$ and $A_R = 61.33°$ angle values. On the other hand, in Conf-3, the WMR has started (I) with $A_T = 32.47°$, $A_L = 65.88°$ and $A_R = 81.65°$ and reached to (II) the target with $A_T = 57.03°$, $A_L = 55.80°$ and $A_R = 67.17°$ angle values.

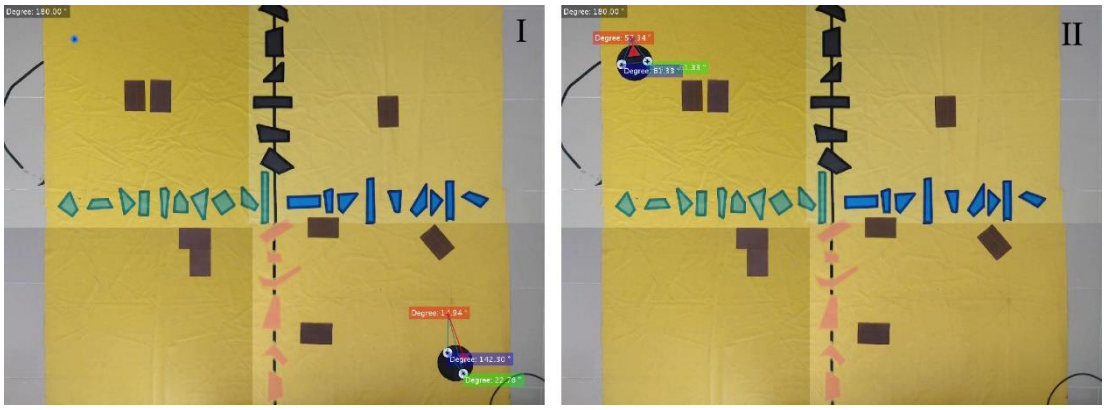

Figure 25 (I) starting position and (II) finishing position for Conf-2

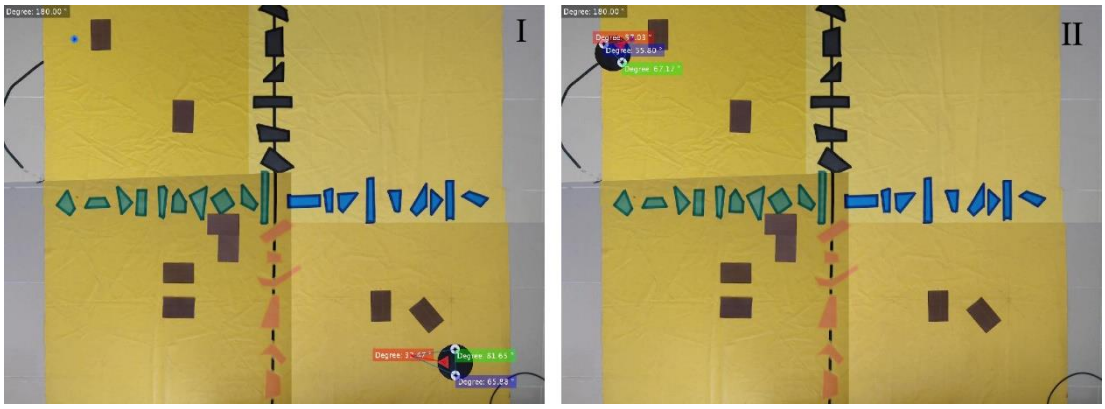

Figure 26 (I) starting position and (II) finishing position for Conf-3

The paths formed by the motion of the mobile robot from the initial position to the target position are given in Figure 27. In each configuration, the object dilation to the obstacles has been implemented to increase path safety. Similar to the first configuration, the real paths are measured a bit shorter than the simulated path.

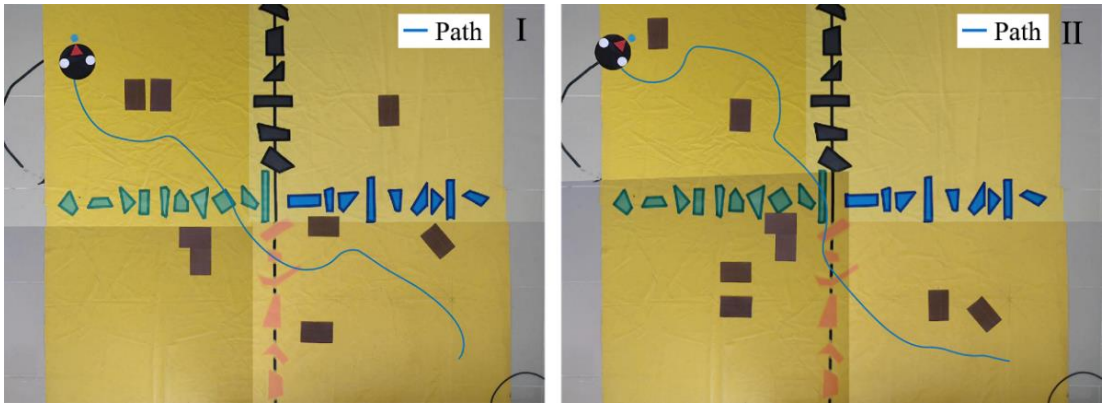

Figure 27 (I) path formed in Conf-2 (II) path formed in Conf-3

5.3. Experiment Comparisons

İdentified path cost and time variables are shown in Table *1*. Except for the first experiment (Conf-1) simulation has been carried out faster. Compare to the simulation path costs, the real path costs are generally found as less costly. The average difference between simulation and real path costs has emerged about 3,656% for all experiments. On the other hand, it should be indicated that both simulation and real paths are not perfect paths that minimize path cost and increase path safety with the best values. However, it can be stated that obtained paths are closer to the best path option and provide an efficient solution balance between safety and cost.

Table 1 The cost and acquired time parameters for different configurations

Experiment	Time (s) Simulation	Time (s) Implementation	Cost (px) Simulation Path	Cost (px) Real Path
Conf-1	11,2	10,5	1037,53	995,16
Conf-2	11,4	11,9	1088,65	1055,42
Conf-3	12,5	13,8	1143,08	1099,27

5.4. General Observations

In terms of scalability, multi-camera configurations provide advantages and disadvantages. The first advantage is eye-in-device equipment is not required. Thus, the the system cost can be decreased. The another advantage is that all the robots can be controlled from one system (actually system may be saturated to an upper limit in terms of robot number). The disadvantage is the number of required cameras may be high for large interior environments. So, the flexibility of the system will reduce. To overcome this issue, cameras with a wide-angle lens or high-resolution image levels can be used. Image stitching is a time-consuming task. However, this process is only implemented once before triggering the real control task. The instant glare is a problem for this configuration as well. Besides, shadowing can be a problematic issue. Although all the cameras have the same specifications and have been placed to the same height, images taken from these cameras can have a different level of shadowing and illumination. Therefore, these two problems should be researched for the visual-based control systems and these types of issues should be focused on as separate studies.

5.5. The Main Influencers for Control Models

Multi-camera-based visual control allows the robot to be operated for large areas in interior spaces. The camera specifications should be the same to acquire better efficiency and performance. Another factor is the number of cameras; an external computer can cope with a certain number of cameras. Therefore, as the number of the camera increases the number of required computers will also increase. This situation is the main drawback of such systems. Illumination and shadowing are big factors for multi-camera configuration as well. Each camera may be exposed to a different level of illumination. This difference may cause an improper stitching process if it is not handled properly.

6. Conclusion and Future Works

In this study, a model is proposed for multi-camera configuration for the visual servoing task. Multi-camera provides a scalable platform for visual servoing systems. Unlike the stereovision systems, depth data is not utilized. Besides, the depth information does not ensure any notable advantage for the eye-out-device configuration. This configuration resembles a global positioning system in a way that each camera helps to determine the exact location of the robot. The robot has been arrived the target position in each configuration, successfully. The simulation and real paths may be a little different from each other. However, the robot exhibits compatible motions to the extracted path plan. In the future, we aim to develop the visual control without stitching process. Ultimately, an efficient planning to create feasible path is a critical issue to overcome.

References

Siciliano B. and Khatib O., (2008). *Handsbook of Robotics*. 1st ed. Berlin. Springer-Verlag.

Dönmez, E., Kocamaz, A.F. and Dirik, M. A (2018). Vision-Based Real-Time Mobile Robot Controller Design Based on Gaussian Function for Indoor Environment. A. J. Sci. Eng. 43, 7127–7142.

Choset H. M., (2005). Principles of Robot Motion: Theory, Algorithms, and Implementation.

Cowan N. J., Weingarten J. D., and Koditschek D. E., (2002). Visual servoing via navigation functions. IEEE Trans. Robot. Autom, 18(4), 521–533.

Dönmez, E., Kocamaz, A.F. (2020). Design of Mobile Robot Control Infrastructure Based on Decision Trees and Adaptive Potential Area Methods. I.J. Sci. Tech. Trans. Electr. Eng., 44, 431–448.

Dudek G. and Jenkin M., (2010). Computational Principles of Mobile Robotics. 2nd ed. New York: Cambridge University Press.

Tsitsiklis J. N., (1995). Efficient algorithms for globally optimal trajectories. IEEE Trans. Automat. Contr., 40(9), 1528–1538.

Cormen T. H., Rivest R. L., and Leiserson C. E., (2001). Introduction to algorithms. MIT Press.

Hart P., Nilsson N., and Raphael B., (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Trans. Syst. Sci. Cybern., 4(2), 00–107.

Stentz A., (1994). Optimal and efficient path planning for partially-known environments. in Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 3310–3317.

Lavalle S. M., (1998) Rapidly-Exploring Random Trees: A New Tool for Path Planning.

LaValle S. M. and Kuffner J. J., (1999). Randomized kinodynamic planning. in Proceedings 1999 IEEE International Conference on Robotics and Automation, 1, 473–479.

Rimon E. and Koditschek D. E., (1992). Exact robot navigation using artificial potential functions. IEEE Trans. Robot. Autom., 8(5), 501–518.

Malis E., Chaumette F., and Boudet S., (2000). Multi-cameras visual servoing. in *Proceedings ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, 4, 3183–3188.

Lippiello V., Siciliano B., and Villani L., (2005) Eye-in-Hand/Eye-to-Hand Multi-Camera Visual Servoing. in Proceedings of the 44th IEEE Conference on Decision and Control, 5354–5359.

Qiu L., Song Q., Lei J., Yu Y., and Ge Y., (2006). Multi-Camera Based Robot Visual Servoing System. in 2006 International Conference on Mechatronics and Automation, 1509–1514.

Yoshihata Y., Watanabe K., and Iwatani Y., (2007). Multi-camera visual servoing of a micro helicopter under occlusions. in 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2615–2620.

Iwatani, K. Y., and Hashimoto K., (2008). Multi-camera visual servoing of multiple micro helicopters. in 2008 SICE Annual Conference, 2432–2435.

Weber B. and Kuhnlenz K., (2010). Visual servoing using triangulation with an omnidirectional multi-camera system. in 2010 11th International Conference on Control Automation Robotics & Vision, 1440–1445.

Kermorgant O. and Chaumette F., (2011). Multi-sensor data fusion in sensor-based control: Application to multi-camera visual servoing. in 2011 IEEE International Conference on Robotics and Automation, 4518–4523.

Elsheikh E. A., El-Bardini M. A., and Fkirin M. A., (2016). Dynamic path planning and decentralized FLC path following implementation for WMR based on visual servoing. in 2016 3rd MEC International Conference on Big Data and Smart City, 1–7.

Aliakbarpour H., Tahri O., and Araujo H., (2014). Visual servoing of mobile robots using non-central catadioptric cameras. Rob. Auton. Syst., 62(11), 1613–1622.

Ahlin K., Joffe B., Hu A. P., McMurray G., and Sadegh N., (2016). Autonomous Leaf Picking Using Deep Learning and Visual-Servoing. IFAC-PapersOnLine, 49(16), 177–183.

Alepuz J. P., Emami M. R., and Pomares J., (2016). Direct image-based visual servoing of free-floating space manipulators. Aerosp. Sci. Technol., 55, 1–9.

Donmez E., Kocamaz A. F., and Dirik M., (2016). Robot control with graph based edge measure in real time image frames. in 24th Signal Processing and Communication Application Conference (SIU), 1789–1792.

Dirik M., Kocamaz A. F., and Donmez E., (2016). Vision-based decision tree controller design method sensorless application by using angle knowledge. 24th Signal Processing and Communication Application Conference (SIU), 1849–1852.

Dirik M., Kocamaz A. F., and Donmez E., (2020). Visual servoing based control methods for nonholonomic mobile robot. Journal of Engineering Research, 8(2), 95–113.

Donmez E., Kocamaz A. F., and Dirik M., (2017). Visual based path planning with adaptive artificial potential field. 25th Signal Processing and Communications Applications

Conference (SIU), 1–4.
Dirik M., Kocamaz A. F., and Donmez E., (2017). Static path planning based on visual servoing via fuzzy logic. 25th Signal Processing and Communications Applications Conference (SIU), 1–4.
Brown and Lowe, (2003). Recognising panoramas. in Proceedings Ninth IEEE International Conference on Computer Vision, 2, 1218–1225.
Bay H., Ess A., Tuytelaars T., and Van Gool L., (2008). Speeded-Up Robust Features (SURF)," Comput. Vis. Image Underst., 110(3) 346–359.
Lowe G., (2004). SIFT - The Scale Invariant Feature Transform," Int. J., 2, 91–110.
Gonzalez R. and Woods R., (2002). Digital image processing.
Mota C., Gomes J., and Cavalcante M. I. A., (2001). Optimal image quantization, perception and the median cut algorithm," An. Acad. Bras. Cienc., 73(3), 2001.