2020 5th International Conference on Robotics and Automation Engineering

Interactive Learning of Mobile Robots Kinematics Using ARCore

Abhidipta Mallik, Vikram Kapila
Mechanical and Aerospace Engineering Department
NYU Tandon School of Engineering
Brooklyn, NY, USA
e-mail: {am8203, vkapila}@nyu.edu

***Abstract*—Recent years have witnessed several educational innovations to provide effective and engaging classroom instruction with the integration of immersive interactions based on augmented reality and virtual reality (AR/VR). This paper outlines the development of an ARCore-based application (app) that can impart interactive experiences for hands-on learning in engineering laboratories. The ARCore technology enables a smartphone to sense its environment and detect horizontal and vertical surfaces, thus allowing the smartphone to estimate any position in its workspace. In this mobile app, with touch-based interaction and AR feedback, the user can interact with a wheeled mobile robot and reinforce the concepts of kinematics for a differential drive mobile robot. The user experience is evaluated and system performance is validated through a user study with participants. The assessment shows that the proposed AR interface for interacting with the experimental setup is intuitive, easy to use, exciting, and recommendable.**

Keywords—ARCore, augmented reality, mobile robot kinematics, robotics education

I. Introduction

Today's smartphones come with built-in AR capabilities and with a simple cardboard one can enjoy VR experiences. Thus, over the last decade, the AR/VR technology has become quite popular and its potential in different fields is being explored. In the education sector, to cultivate immersive and stimulating learning experiences, many studies have explored the integration of AR/VR-based immersive interactions in the educational environments, especially in STEM disciplines [1]−[4]. An AR app was developed in [5] to enhance mathematics and cognitive skills with the aim to make calculus learning visual and tangible. Moreover, the AR/VR technology is being explored in post-secondary education and in job training for its potential to improve interactive learning of fundamental concepts in the laboratory in fields such as medical science [6], engineering [7]−[9], and military [10].

AR/VR technologies can also improve human robot interaction through intuitive and natural interfaces, rendering interesting use cases. Differential drive kinematics constitutes a fundamental mobile robotics concept. Traditionally students first learn the theoretical concepts in classroom and then perform laboratories experiments to improve their understanding [11]. According to [12], the AR technology has the potential to reverse the decline in the quality of education caused by the laboratories crammed with increased intake of students in engineering fields. Moreover, the cost effective AR technology can enable students to learn at their own pace, which can improve student motivation and save time and effort of teachers [12]. Integration of experiments with VR simulation can obviate safety concerns associated with real-world robots and allow wider access to experimental testbeds through resources sharing [13]. Even as AR presents opportunities and challenges to the education sector, it offers a tremendous potential to help students learn and develop skills in an interactive way [14]. To enhance students' mobile robotics learning experience, an interactive AR experiment is proposed in this paper.

Google's ARCore platform for building AR experiences consists of APIs to sense and understand environment and interact with the world by processing the information gathered [15]. ARCore uses smartphone's camera to integrate the real world with virtual content. By using motion tracking, ARCore keeps track of the smartphone's location relative to the world by identifying key feature points. The relative location of these key feature points and measurements from the smartphone's inertial sensors help ARCore compute the position and orientation of the smartphone. Environment understanding capabilities of ARCore enable it to locate vertical and horizontal surfaces as the smartphone moves through space, allowing its seamless integration with the real world. These features of ARCore obviate the need for affixing fiducial markers in the environment or on robots as in prior AR-based interactions with robots [16], [17], and use of fixed camera [18], [19].

The novelty of this AR app for mobile robotics is that no marker or identifying symbol is placed on the robot, instead a single reference image, affixed on a horizontal surface, is used for referencing the workspace environment. Note that ARCore has the capability to track up to 20 images concurrently with the tracking taking place on the smartphone. Moreover, ARCore has the capacity to store up to 1,000 reference images per image database.

The paper begins with Section I providing introduction and literature review. In Section II, a description of the system is presented and Section III provides the model of the system. In Section IV, a description of the app is presented and Section V describes system testing and evaluation procedure. Section VI provides discussion and concluding remarks.

978-1-7281-8981-9/20/$31.00 ©2020 IEEE

II. System Description

The system used in this study includes a differential drive robot endowed with a Bluetooth (BT) module, a smartphone (Google Pixel 3, Android v9.0 (Pie) OS, powered by Octa core, 2.5 GHz), and a reference image affixed to the floor. The smartphone camera allows the user to view the robot and its workspace. The newly developed ARCore-based app looks for the reference image on a horizontal surface. Upon detecting the reference image, the user interacts with the app to mark the center of the robot, the front of the robot, and the desired destination location of the robot. The differential drive mobile robot has two rear wheels connected to two servo motors and one castor wheel in the front. An Arduino UNO powered by a 9V battery is used as the robot's onboard controller. A motor shield interfaced with the Arduino extends its capabilities and regulates the power supply to the motors. A BT module attached to the Arduino allows it to receive data from the smartphone. The Arduino processes the data received from the smartphone and sends input to the servo motors.

III. Modelling

In a differential drive mobile robot, two wheels are attached on a common axis and each of them can be controlled independently to drive the robot. To rotate the robot, the velocity of each wheel needs to be controlled and the rotation takes place about a point that lies along the common left and right wheel axis, i.e., the instantaneous center of curvature (ICC). Considering the rate of rotation about ICC for the robot as ω, equations of motion are obtained. First, note that

$$V_r = \omega\left(r_{cc} + \frac{d}{2}\right) \tag{1}$$

$$V_l = \omega\left(r_{cc} - \frac{d}{2}\right) \tag{2}$$

where d is the distance between the centers of the two wheels, V_r, V_l are the right and left wheel velocities, and r_{cc} is the signed distance from the ICC to the midpoint between the wheels as shown in Fig. 1. From (1) and (2), ω and r_{cc} can be obtained as shown below.

$$\omega = \frac{(V_r - V_l)}{d} \tag{3}$$

$$r_{cc} = \frac{d(V_r + V_l)}{2(V_r - V_l)} \tag{4}$$

Now, using (3) and (4), if $V_r = V_l$ then motion of the robot is straight line (forward motion with $r_{cc} \to \infty$); if $V_r = -V_l$ then the robot will rotate in place (with $r_{cc} = 0$); if $V_r = 0$ then the robot will rotate around right wheel; and if V_l=0 then the robot will rotate around the left wheel. The general equations of motion of a differential drive mobile robot moving in a particular direction $\Theta(t)$ at a given wheel velocities $V_l(t)$ and $V_r(t)$ are described as follows [20], [21].

$$x(t) = \int_0^t \left(\frac{V_l(t) + V_r(t)}{2}\right)\cos[\Theta(t)])\mathrm{d}t$$

$$y(t) = \int_0^t \left(\frac{V_l(t) + V_r(t)}{2}\right)\sin[\Theta(t)])\mathrm{d}t$$

$$\Theta(t) = \int_0^t \left(\frac{V_r(t) - V_l(t)}{d}\right)\mathrm{d}t$$

These equations of motion of a differential drive mobile robot can be rewritten, for the time $t + \delta t$, as a function of wheel velocities and robot angular velocity as below.

$$x(t + \delta t) = \int_0^{t+\,\delta t} V(t)\cos[\theta(t)]\mathrm{d}t$$

$$= x(t) + \int_t^{t+\,\delta t} V(t)\cos[\theta(t)]\mathrm{d}t$$

$$y(t + \delta t) = \int_0^{t+\,\delta t} V(t)\sin[\theta(t)]\mathrm{d}t$$

$$= y(t) + \int_t^{t+\,\delta t} V(t)\sin[\theta(t)]\mathrm{d}t$$

$$\theta(t + \delta t) = \int_0^{t+\delta t} \omega(t)dt = \theta(t) + \int_t^{t+\delta t} \omega(t)\mathrm{d}t$$

where

$$V(t) = \big(V_r(t) + V_l(t)\big)/2$$

$$\omega(t) = \big(V_r(t) - V_l(t)\big)/d$$

From the above equations, we can now obtain

$$\dot{x}(t) = \left(\frac{V_l(t) + V_r(t)}{2}\right)\cos[\theta(t)]$$

$$\dot{y}(t) = \left(\frac{V_l(t) + V_r(t)}{2}\right)\sin[\theta(t)]$$

$$\dot{\theta} = \left(\frac{V_r(t) - V_l(t)}{d}\right)$$

that characterize the dynamic equations for translational and angular motion of the mobile robot.

In case of a mobile robot, the inverse kinematics problem concerns calculating the path and the velocity of the robot for reaching a target location from a given initial position. The robot moves to a desired goal pose in two steps. First the robot rotates in place until it is aimed towards the goal and then it moves forward towards the goal.

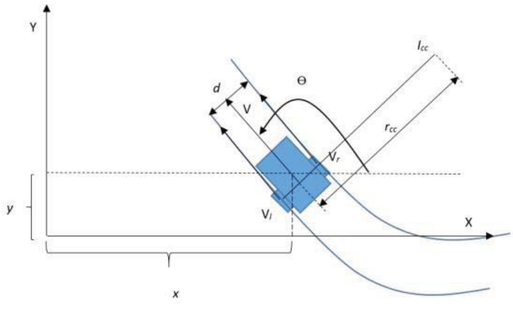


Figure 1. Kinematics of differential robot

IV. App Description

The user places the reference image on the floor and turns on the app. The image can be placed anywhere on the floor, in the workspace of the robot, and the center of the image serves as the origin of the system. The user can place the mobile robot in any orientation on the floor. Fig. 2 shows the reference image on floor with the differential drive robot.

The user turns on the Unity app on the smartphone. The app has one scene with three buttons. The “quit” button is used to close the app. The “tick mark” button allows the user to make proper selection and the “cross” button is used to cancel the prior selection. When the app is initialized, smartphone’s back camera opens and captures images. ARCore senses and understands the environment and identifies feature points in the environment. After ARCore detects the horizontal ground plane, it looks for the reference image on the ground (Fig. 3(a)). When the reference image is found, an augmented anchor resembling a 3D coordinate system is placed on it by the app (Fig. 3(b)). A reference coordinate system is attached to the image by the app with the center of the image as origin.

Next, ARCore continues to track the image while providing estimates for its position, orientation, and size. Once the reference image is detected, it is not necessary to keep it in camera view. As the smartphone collects more information, the estimates are continuously refined. ARCore uses a single camera along with the onboard IMU device. The IMU measurements combined with these estimates enable ARCore to calculate the smartphone pose. Specifically, it uses

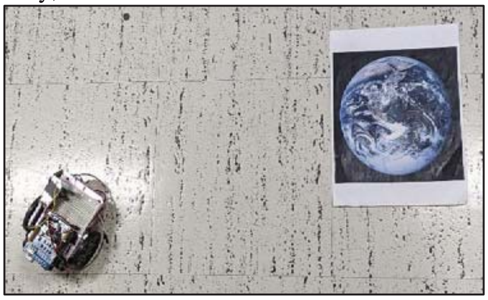
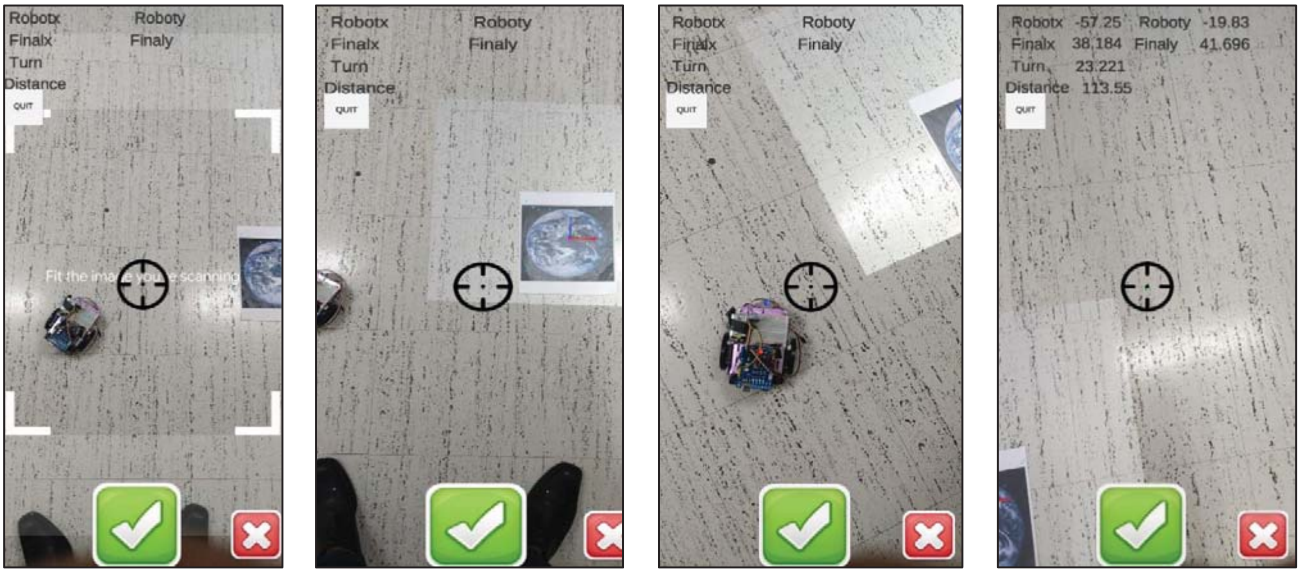


Figure 2. Differential robot and reference image on floor



(a) (b) (c) (d)

Figure 3. (a) App searches for the reference image on the horizontal surface; (b) augmented anchor is placed on the reference image; (c) user indicates the center and front of the robot; and (d) user indicates destination

the concurrent odometry and mapping (COM) algorithm [22] to build a 3D visual representation of the environment and estimate the pose of the smartphone based on feature descriptors. Data from motion sensor allows it to estimate change in position over time.

The user now needs to mark the center of the robot. To do so, the user brings the smartphone in the vicinity of the robot (Fig. 3(c)), just on top of the robot center, and taps the “tick mark” button to create an anchor in the form of small dot at that location. This point on the smartphone screen is sensed and ARCore performs coordinate transformations between the camera coordinate frame and reference image coordinate frame to find its equivalence relative to the robot. Next the user needs to indicate the front of their robot. Specifically, the user taps the “tick mark” on top of the front of the robot, creating another anchor there. After that the user is free to choose any point on the floor as destination (Fig. 3(d)). In Fig. 4, O is the origin (center of the reference image), F and C are the front and the center of the robot, respectively, and D is the destination location. In the triangle FCD, angle FCD can be found from cosine angle formula. From the coordinates of the center, front of the robot, and

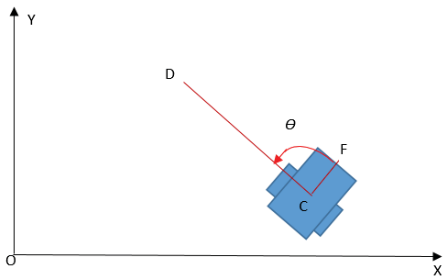


Figure 4. Inverse kinematics of differential drive robot

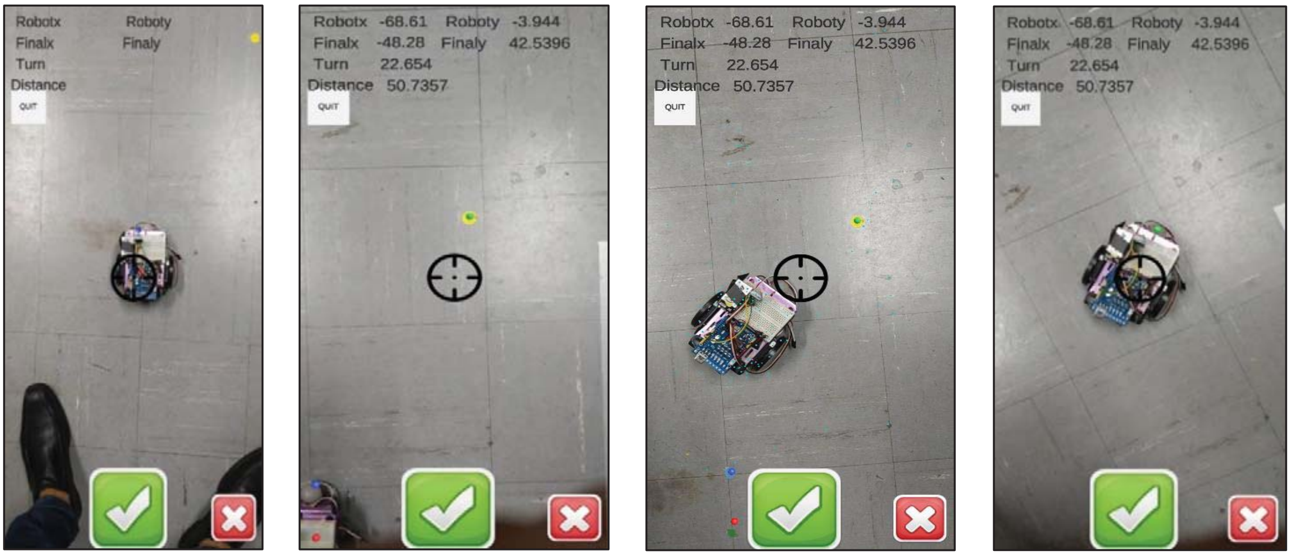
origin, graph quadrant adjustment is made with respect to the origin. As explained in section III, the robot moves to goal position in two steps. First the robot rotates in place until it is aimed towards the goal, this angle Θ is computed in (5) below. Then it moves forward towards the goal by the distance given by $|CD|$ in (6) below.

$$\Theta = \arccos\left(\frac{|CD|^2 + |FC|^2 - |FD|^2}{2 \times |FC| \times |CD|}\right) \tag{5}$$

$$|CD|^2 = (C_x - D_x)^2 + \left(C_y - D_y\right)^2 \tag{6}$$

$$|FC|^2 = (F_x - C_x)^2 + \left(F_y - C_y\right)^2 \tag{7}$$

$$|FD|^2 = (F_x - D_x)^2 + \left(F_y - D_y\right)^2 \tag{8}$$



(a) (b) (c) (d)

Figure 5. (a) User indicates the center and front of the robot during trial phase; (b) user indicates destination during evaluation accuracy testing phase; (c) robot turns towards destination and moves towards it; and (d) robot reaches destination

TABLE I: SYSTEM VALIDATION

	Center (user selected)		**Front (user selected)**		**Destination (user selected)**		**Actual robot destination**	
Coordinate, cm	***x***	***y***	***x***	***y***	***x***	***y***	***x***	*y*
Mean	−70	−2.7	−70.1	4.44	−46.7	44.7	−47.5	45.9
Std. deviation	2.06	2	2.23	1.36	2.7	2.3	2.8	3.73

V. SYSTEM EVALUATION

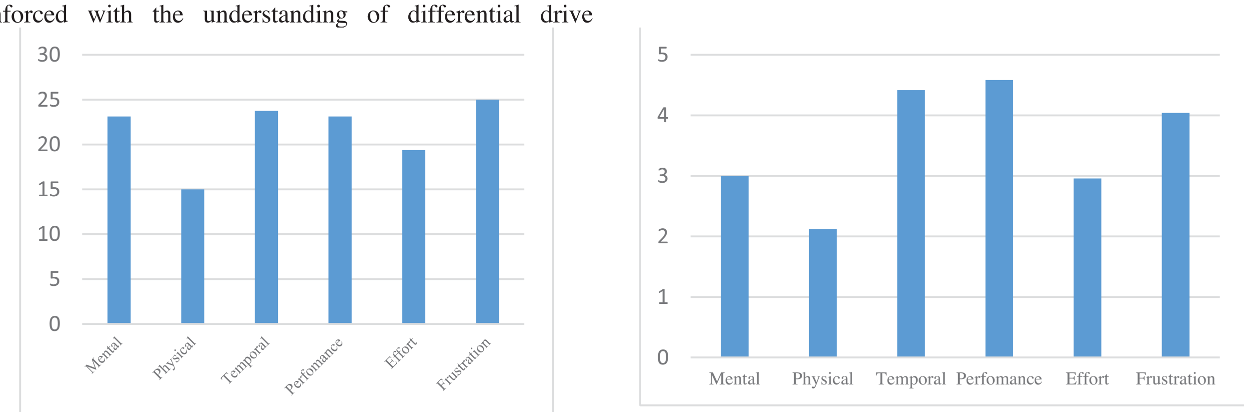
The system is tested for its usability and performance with four female and 12 male engineering students (users). First the app is described to users. Those not familiar with mobile robotics kinematics are provided the basic concepts. Then the users first perform a trial experiment by placing the robot at an arbitrary location and an arbitrary angle and choose a random destination point. After this trial, in the next task, we test the accuracy of the system and accuracy with which the user is able to communicate the correct locations. Specifically, in this task, the robot initial and front locations are fixed relative to the center of reference image with respective coordinates as (−69cm, −2cm) and (−69cm, 4cm). The desired destination location (−48cm, 44cm) is marked with a yellow sticker. The user marks the center and front selection through the app (Fig. 5(a)) and then selects the destination (yellow sticker) shown in Fig. 5(b). Fig. 5(c) shows the robot turn towards the destination and move towards it. Fig. 5(d) shows that the robot reaches the destination. Table 1 shows the average and standard deviation for the x and y coordinates of the user-selected points and the achieved destination of the robot. This results show that the system performance is acceptable with minimal difference between the intended and actual selections.

A usability questionnaire was developed, taking inputs from [23] and [24], to assess user experience. The poststudy system usability questionnaire had eight items, six positively directed and two negatively directed. Each question had five options: strongly agree (5), agree (4), neither agree nor disagree (3), disagree (2), and strongly disagree (1). The questions with average response scores are given in Table 2.

The NASA TLX [25] requires the users to indicate each of the subscales of workload, i.e., mental, physical, and temporal demands and performance, effort, and frustration levels on a 100-points scale and to make 15 pairwise comparisons of these subscales based on their perceived importance. In Fig. 6(a), the bar chart shows the average of each subscale. Fig. 6(b) shows the workload score of each subscale.

VI. Conclusion and Future Work

This paper presented the development of an AR interface with ARCore capabilities to provide immersive interactive experiences and perform mobile robot kinematics. The developed system allows users to teleoperate a mobile robot with a smartphone as the concepts of coordinate system are reinforced with the understanding of differential drive mobile robot kinematics. The user response to NASA TLX shows that that each of the subscales contributes a relatively low score and thus the overall workload is low (average of 21.12). From the usability questionnaire response, it is evident that the AR interface is deemed simple and exciting to use, and users do not need technical support. Performing such experiments will not be sufficient to grasp mobile robotics kinematic concepts.



(a) (b)

Figure 6. (a) Average of the participants' response to TLX subscale and (b) workload score from participant responses

Table II: Usability Questionnaire with Average Score

Questions	Average
1. The user interface was simple and exciting to use.	4.50
2. It took a long time to learn the app.	1.87
3. I needed assistance of a technical person to interact with the mobile robot.	2.07
4. The mobile robot went to the intended destination.	4.86
5. It was easy to select the center position, front position, and destination position in the robot workspace.	4.86
6. The system helped me to understand kinematics concepts.	4.43
7. I was easily able to use this app to clearly communicate my intentions to the mobile robot.	4.71
8. I recommend this system to people of different age groups to interact with mobile robots.	4.57

After performing this experiment, students need to study on their own, fill out worksheets, and complete assignments as they do in traditional classrooms. We need to conduct further studies to assess the educational potential of the proposed system in teaching fundamental mobile robotics concepts by gauging its efficacy in terms of student learning. In future, we will develop other testbeds along similar lines.

The novelty of this AR app for mobile robotics is that it is not restricted by a fixed camera and it obviates the placement of markers on the robots. Instead, a single reference image affixed on the horizontal surface is used for referencing the robot workspace. The user is free to move around while communicating with the robot because the AR application continuously tracks the location of the smartphone. The app provides pose of any object of interest in the view of the camera, even as the user moves around with the smartphone. Another advantage of the AR app is that the reference image needs to be in the view of the smartphone only during initialization. The user is able to indicate any location in the robot's workspace by a simple touch input on the smartphone screen thus simplifying the user effort. Such a testbed can give respite from the traditional laboratories that get crowded due to increased intake of students in engineering fields. It can allow students to learn on their own, thereby saving time and effort of teachers. Such testbeds offer cost-effective solutions that can enhance student concentration and motivation. Moreover, such testbeds may enable remote learning that is especially relevant in a pandemic situation such as Covid-19.

Acknowledgment

The authors thank the students who took part in the user-study experiments. This work is supported in part by the National Science Foundation grants ITEST DRL: 1614085, DRK-12 DRL: 1417769, and RET Site EEC: 1542286.

References

[1] M. Billinghurst and A. Dünser, "Augmented reality in the classroom," *Computer (Long. Beach. Calif).*, vol. 45, no. 7, pp. 56–63, 2012, doi: 10.1109/MC.2012.111.

[2] F. Liarokapis et al., "Web3d and augmented reality to support engineering education," *World Trans. Eng. Technol. Educ.*, vol. 2, no. 1, pp. 11–14, 2004.

[3] P. Maier, G. Klinker, and M. Tonnis, "Augmented reality for teaching spatial relations," 2009.

[4] H. Kaufmann and D. Schmalstieg, "Mathematics and geometry education with collaborative augmented reality," *Comput. Graph.*, vol. 27, no. 3, pp. 339–345, 2003.

[5] E. Quintero, P. Salinas, E. González-Mendívil, and H. Ramírez, "Augmented reality app for calculus: a proposal for the development of spatial visualization," *Procedia Comput. Sci.*, vol. 75, no. Vare, pp. 301–305, 2015, doi: 10.1016/j.procs.2015.12.251.

[6] A. G. Gallagher and C. U. Cates, "Virtual reality training for the operating room and cardiac catheterisation laboratory," *Lancet*, vol. 364, no. 9444, pp. 1538–1540, 2004, doi: 10.1016/S0140-6736(04)17278-4.

[7] A.-H. G. Abulrub, A. N. Attridge, and M. A. Williams, "Virtual reality in engineering education: The future of creative learning," *Int. J. Emerg. Technol. Learn.*, pp. 751–757, 2011, doi: 10.1109/EDUCON.2011.5773223.

[8] J. A. Frank and V. Kapila, "Towards teleoperation-based interactive learning of robot kinematics using a mobile augmented reality interface on a tablet," *2016 Indian Control Conf. Proc.*, pp. 385–392, 2016, doi: 10.1109/INDIANCC.2016.7441163.

[9] E. Manou, G.-C. Vosniakos, and E. Matsas, "Understanding industrial robot programming by aid of a virtual reality environment," *Int. J. Mech. Eng. Educ.*, vol. 47, no. 2, pp. 135–155, 2019.

[10] T. Kot and P. Novák, "Application of virtual reality in teleoperation of the military mobile robotic system TAROS," *Int. J. Adv. Robot. Syst.*, vol. 15, no. 1, Jan. 2018, doi: 10.1177/1729881417751545.

[11] R. Huq, H. Lacheray, C. Fulford, D. Wight, and J. Apkarian, "Qbot: An educational mobile robot controlled in MATLAB simulink environment," *Can. Conf. Electr. Comput. Eng.*, no. May, pp. 350–353, 2009, doi: 10.1109/CCECE.2009.5090152.

[12] J. Martín-Gutiérrez, P. Fabiani, W. Benesova, M. D. Meneses, and C. E. Mora, "Augmented reality to promote collaborative and autonomous learning in higher education," *Comput. Human Behav.*, vol. 51, pp. 752–761, 2015, doi: 10.1016/j.chb.2014.11.093.

[13] X. Xu et al., "Robotic kinematics teaching system with virtual reality, remote control and an on–site laboratory," *Int. J. Mech. Eng. Educ.*, vol. 48, no. 3, pp. 197–220, 2020.

[14] H. K. Wu, S. W. Y. Lee, H. Y. Chang, and J. C. Liang, "Current status, opportunities and challenges of augmented reality in education," *Comput. Educ.*, vol. 62, pp. 41–49, 2013, doi: 10.1016/j.compedu.2012.10.024.

[15] "*Android API reference for ARCore | Google Developers.*" [Online]. Available: https://developers.google.com/ar/reference/java. [Accessed: 12-Nov-2019].

[16] J. A. Frank, S. P. Krishnamoorthy, and V. Kapila, "Toward mobile mixed-reality interaction with multi-robot systems," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 1901–1908, Oct. 2017, doi: 10.1109/LRA.2017.2714128.

[17] K. Ohmori and K. Sakamoto, "Automatic mobile robot control and indication method using augmented reality technology," in *International Conference on Entertainment Computing*, 2010, vol. 6243, pp. 464–467, doi: 10.1007/978-3-642-15399-0_61.

[18] M. Fiala, "Artoolkit applied to panoramic vision for robot navigation," *Proc. Vis. Interface*, pp. 119--127, 2003.

[19] W. Honig, C. Milanes, L. Scaria, T. Phan, M. Bolas, and N. Ayanian, "Mixed reality for robotics," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2015-Decem, pp. 5382–5387, 2015, doi: 10.1109/IROS.2015.7354138.

[20] P. Corke, *Robotics,Vision and Control: Fundamental Algorithms in MATLAB*. Heidelberg, Germany: Springer, 2011.

[21] R. Siegwart, I.R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. Cambridge, MA: MIT Press, 2011.

[22] E. Nerurkar, S. Lynen, and S. Zhao, "System and method for concurrent odometry and mapping," 15/595,617, 2017.

[23] N. Malhotra, "Questionnaire design and scale development," in *The Handbook of Marketing Research: Uses, Misuses, and Future Advances*, London. UK: Sage Publications, 2006.

[24] J. A. Frank, M. Moorhead, and V. Kapila, "Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks," *25th IEEE Int. Symp. Robot Hum. Interact. Commun. RO-MAN 2016*, pp. 302–307, 2016, doi: 10.1109/ROMAN.2016.7745146.

[25] S. G. Hart, "Nasa-task load index (NASA-TLX); 20 years later," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 50, no. 9, pp. 904–908, Oct. 2006, doi: 10.1177/154193120605000909.