Vision-Based Control of a Line-Tracing Mobile Robot

SAMO SIMONČIČ, PRIMOŽ PODRŽAJ

Faculty of Mechanical Engineering, University of Ljubljana, 1000 Ljubljana, Slovenia

Received 5 October 2010; accepted 25 July 2011

ABSTRACT: In this article, a vision-based control of a line-tracing mobile robot is presented. The position of the robot is determined based on the analysis of the images obtained by a ceiling mounted camera. Such an approach has some advantages compared to the more common robot mounted camera approach. The robot detection algorithm uses some well-known image processing algorithms. As the PC is common these days and the mobile robot is quite easy to build or buy, the only prerequisite is a camera. The time delay of a web camera is too large so a FireWire CCD camera is preferred for better performance. The control system presented can be an ideal educational tool for an introductory level course in image processing, image analysis, or machine vision. As the line-tracing problem is well known to everyone studying mechatronics or robotics, the approach presented in this article should be of great interest to all students in these fields, because it offers an alternative solution to the line-tracing problem. Besides that this approach makes it possible to implement a more advanced predictive control algorithm. In order to make the experiment even more interesting, an obstacle avoidance algorithm is presented as well. Comput Appl Eng Educ 22:474–480, 2014; View this article online at wileyonlinelibrary.com/journal/cae; DOI 10.1002/cae.20573

Keywords: image processing; predictive control; mobile robots; obstacle avoidance

INTRODUCTION

In the last decade, the research in the field of mobile robots has been spreading into different areas. Mobile robots are increasingly used in industry [1], domestic needs [2] (lawn mowers, vacuum cleaners, toys), hazardous environments [3–5] (fire-fighting, military applications), and also for educational purposes [6–10]. The increasing trend of mobile robot applications is expected to continue in future as well. It is therefore essential for any serious student of engineering to get a firm grip in this area as soon as possible.

Robotics is an interdisciplinary field. It encompasses the areas such as electronics, computer science, physics, mathematics, mechanical engineering, etc. As it can create interesting experiments with relatively low theoretical requirements (see for example [11,12]), it is very attractive for students and teenagers. These experiments are good at convincing the students that the underlying theory needs to be learnt in order to realize more demanding (interesting) projects. They also make the underlying theory easier to understand as the teachers can give real-life examples of the theoretical concepts they are trying to explain. The most widely known type of a robot is a common mobile robot shown in Figure 1. This type of robot is used in many applications [4,11,13]. By far the most well known are the line-tracing applications [8,11–14]. The robot is typically made of a chassis with wheels and a ball caster. The wheels are driven by two electric motors. Different angular velocities of the electric motors make it possible for the mobile robot to move at different speeds on various (curved) trajectories (if the angular velocities differ from each other). As such it is an appropriate educational tool for the explanation of the feedback control. Feedback control systems, with a mobile robot as a controlled object, use different kinds of sensors for the feedback [6,13,14]. The most common ones are photo diodes, which detect the position of the robot relative to the line it should follow [9]. The other possibility is a robot mounted camera [13,14,16]. Such an architecture is usually referred to as an autonomous guided vehicle (AGV) [8,13]. This approach, due to the inherent image analysis, makes it possible to implement more advanced types of control, such as predictive control. Vision-based robot guidance and control is interesting for industrial applications as well. Many mobile robots are used to transport different items within factories. The problem with robot mounted cameras is that each robot must have a camera and a computer capable for executing the control algorithms in real time. As an alternative, a system with a ceiling mounted camera capable of monitoring several robots at the same time is proposed. This type of approach is called Intelligent Space and was first introduced by the Hashimoto Lab of the University of Tokyo in 1995 [15].

In this article, a common line-tracing robot feedback control system is presented. It differs from the previously presented comparable systems, because it uses a ceiling mounted camera for the feedback. The program used to control the motion of

Correspondence to P. Podržaj (primoz.podrzaj@fs.uni-lj.si).

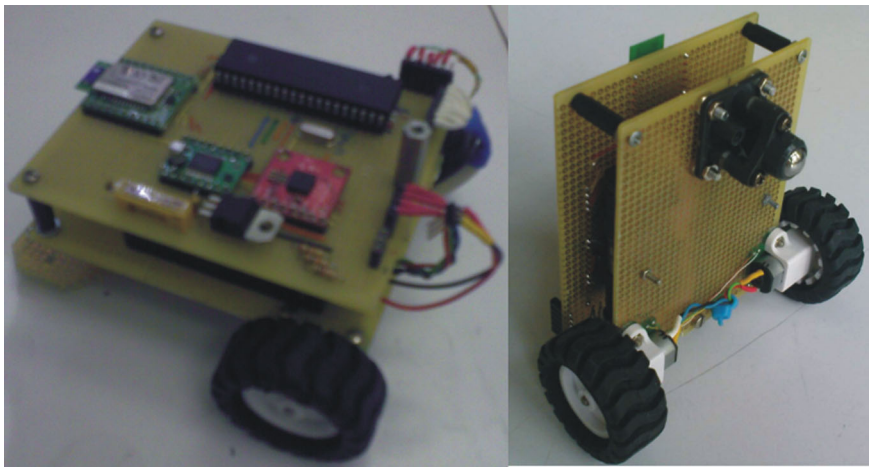

Figure 1 A common mobile robot. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the robot was written in $C_{\#}$ programming language. We used an inexpensive CCD camera in order to make the experiment feasible for educational purposes.

THE MOBILE ROBOT ARCHITECTURE AND CONTROL ALGORITHM DESCRIPTION

The mobile robot is of a rectangular shape with the following dimensions: 15 cm (length) × 10 cm (width) × 5 cm (height). The printed circuit (PC) board is used as a chassis. Two powerful DC (direct current) electric motors with transmission gears (30:1 Micro Metal Gearmotor HP from Pololu) are mounted on the chassis in order to drive the wheels. Ball caster mounted at the front end of the robot is the third point of contact with the ground (see Fig. 1). The chassis is also used to carry a 5 V battery of a significant weight. It supplies the energy needed to rotate the electric drives. The control algorithm is implemented in a PIC16F877A microcontroller. The control signals are 5 kHz pulse-width modulated (PWM) signals for each motor, respectively. The TB6612FNG Dual Motor Driver Carrier is used to convert low current signals into high current ones. Because we used the ceiling mounted camera approach, the robot must be able to communicate with the computer. In order to achieve this, a Roving Networks RN-41 bluetooth module was mounted next to the microcontroller. The bluetooth module and the microcontroller have an UART (Universal Asynchronous receiver-transmitter) hardware module, which provides connection between them. Two pins are used for the UART connection and they are TX (transmit) and RX (receive) pins. The bluetooth module can transfer 115,200 bits of data per second and enables communication between the microcontroller and the PC.

The control of the mobile robot is executed at two levels as shown in Figure 2. The low-level control of the wheel's angular velocities is implemented in a microcontroller, whereas the high-level control of the robot position/velocity is based on the images acquired from the CCD camera. Image processing algorithms are executed in a PC. The control signal is sent to the microcontroller via the bluetooth module.

The line the robot should trace is marked with a black tape on a 1.5 m × 1 m sheet of paper as shown in Figure 3. The line is 1 cm wide.

Because the control system operates in real time, a special care must be taken in order to reduce the time delay (latency) between the moment of image capturing and the moment, when the control action based on this image is executed by the electric motors. For that purpose, we analyzed the performance of the control system using two different cameras. The first one was a low-cost Philips SPC710NC/00 web camera connected to the PC via USB. The second one was a slightly more expensive Imaging Source DFM 21BF04-ML CCD color camera. It was connected to the PC via FireWire cable.

The control algorithm is written using Microsoft Visual Studio 2008 framework with $C_{\#}$ programming language as it is a perfect choice for real time applications because it provides fast and efficient execution of code. The control algorithm captures images from the color CCD camera, frame by frame. The entire algorithm works with 30 frames per second and every frame is represented as a set of three matrices (each matrix represents the Red, Green, and Blue components of the image). The resolution of each image is 640 × 480 pixels. The image

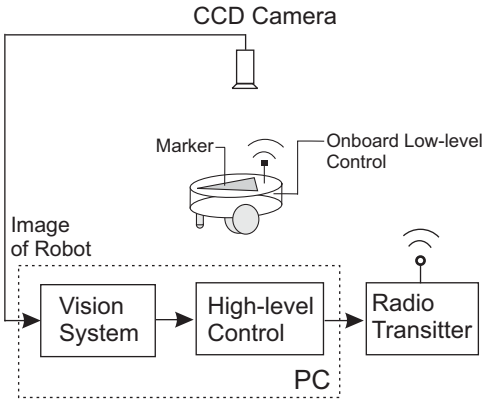

Figure 2 A schematic representation of the control system.

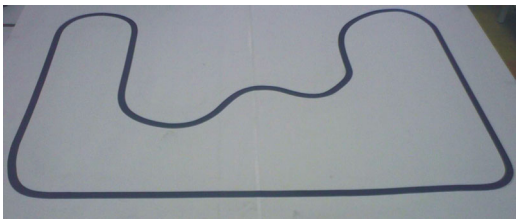

Figure 3 The reference path on a sheet of paper.

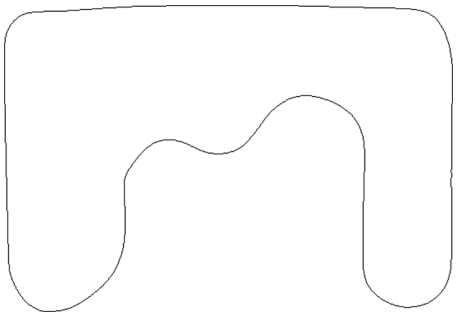

Figure 5 Image of the trajectory after the execution of the thinning algorithm.

covers approximately 1.5 m × 1 m area on the sheet of paper with the black line (see Fig. 3).

The application supports multitasking (multithreading) for fast and efficient real time execution. The program is composed of two threads which work in parallel. The first thread captures the frames from the camera and executes the image processing algorithm. The second thread sends the values of the velocities of both motors via wireless connection to the mobile robot. Personal computers or machines with multiple core processors can execute the application in multitasking structure, where tasks (threads) of the program are distributed on multiple cores. The program of this application is executed on personal computer with dual core processor.

IMAGE PROCESSING

The arbitrary trajectory marked by a black tape on a white sheet of paper (see Fig. 3) is captured as a first frame. This frame is then converted into a grayscale image. The trajectory is then segmented from the grayscale image using a threshold filter. The threshold value is obtained experimentally and depends on the contrast between the trajectory and the background. The trajectory has a width of more than 1 pixel. Therefore, a simple thinning algorithm is applied to reduce the width of the line to 1 pixel. The thinning algorithm is composed of eight D-type hit-and-miss transform masks [18,19] (Fig. 4) and each of these masks recognizes a particular situation where the central pixel is unnecessary for maintaining connectedness and can be eliminated. Elements in the mask 0, 1, and X present background (white pixel), foreground (black pixel), and either background or foreground, respectively. Masks T_1– T_8 are executed one after another on the image of the trajectory. If the mask elements match the neighboring pixels in the image, then the central pixel of the image is eliminated and vice versa. The thinning algorithm is being repeated using all eight masks until the image is not changing any more. The number of the iterations needed depends on the object(s) in the image and the structuring elements (e.g., matrices from Fig. 4) used. The final image is a skeleton (one pixel width) of the trajectory after the thinning algorithm (see Fig. 5). Such a trajectory is needed in order to apply the control algorithm presented later in the article.

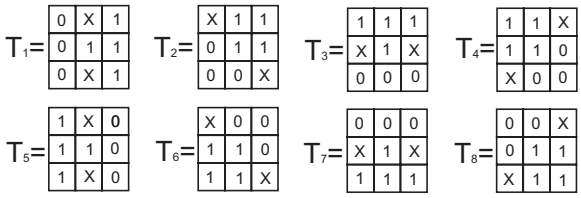

Figure 4 Eight D-type hit-and-miss transform masks.

Detection of the Mobile Robot

The next step in the image processing algorithm is the detection of the mobile robot. In order to make this step as easy as possible at this introductory level of image processing, we mounted a blue triangle on the top of the robot (see Fig. 6). Color filtering is used to detect the blue triangle marker in 24-bit deep color images from the camera.

The color filter is specified by the predefined values of hue, saturation, and value parameters (HSV parameters [19]). HSV color space is a cylindrical-coordinate representation of points in RGB color model. The application of color filters is a very simple and effective image processing tool. If the ambient lighting is varying then the intensity of colors varies as well. Because of that the filter threshold has to be adjusted and sometimes this can be quite time consuming. The main advantage of HSV color space compared to RGB color space is its relative insensitivity to the ambient lighting. In our case the HSV filter is used to determine blue pixels (pixels within a specific HSV parameter range). Their color remains the same, whereas the color of all the other pixels is changed to black (see Fig. 7).

Figure 6 Blue isosceles triangle marker is attached to the mobile robot. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

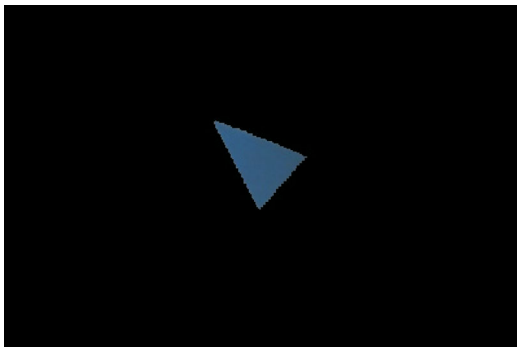

Figure 7 Detection of the blue pixels based on the color filter. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

The next step of the image processing is the determination of the borders of the blue objects (blobs) in the image [20]. The chain code is used for this purpose [19]. The basic idea of the chain code is presented in Figure 8. When the algorithm comes to the reference pixel, it looks for the first black pixel in the 8-neighborhood surrounding area. The search is made in an anticlockwise direction (sequence 0, 1, 2, 3, ...). After that the algorithm moves to that pixel and again executes the 8-neighborhood search. The whole process goes on until we return to the reference pixel. For the object/blob shown in Figure 8, the resulting sequence (the so-called chain code) is 0776665544442222111. It should be noted that besides 8-neighborhood search, sometimes 4-neighborhood search is used.

The next step in image processing is the determination of the number of vertices of the object(s) in the image. This can be achieved by the number of changes in the chain code. The blob in Figure 8 has, for example, seven changes in chain code, so it has seven vertices. In reality, this algorithm is used to find vertices of triangles or rectangles. In that case the three (or four) largest changes are defined as vertices.

As our robot is marked with a triangle we are searching for the triangle in the image. The problem is, however, that due to noise there might be some other very small object with three vertices in the image. We can filter them out based on the area of the object. The area of any blob in the image is determined based on its vertices (i_k, j_k), and $(i_0, j_0) = (i_n, j_n)$. The following formula can be used for that purpose [19]:

$$\text{area} = \frac{1}{2}\left|\sum_{k=0}^{n-1}(i_k j_{k+1} - i_{k+1} j_k)\right| \tag{1}$$

So the position of the robot is determined by the largest (largest area) blue triangle in the image.

Geometrical Model of Image Capturing. As the height of the mobile robot is 5 cm and on the top of it there is a blue triangle used for its detection, a problem arises, whenever the camera is not exactly over the robot. Whenever we see a robot from an angle, the triangle should be moved virtually from the ground plane to the plane on the top of the robot. The situation is shown in Figure 9. The coordinates in the plane on the top of the robot are x, y, b and the coordinates in the ground plane are x', y', 0. The virtual coordinates x' and y' obtained by the image acquisition can be transformed to the real coordinates x and y by the following equations:

$$x = \frac{x' \cdot \overline{ab}}{\overline{0a}};\ y = \frac{y' \cdot \overline{ab}}{\overline{0a}}.$$

Control of the Mobile Robot

The problem addressed in this subsection is the feedback control design allowing a mobile robot to track arbitrary trajectory using a visual feedback information. In our case we have tried two different control designs. The first one is the classical (nonpredictive) control and the second one is the predictive control. The application of both of them for robot control is described in the following subsection.

Classical (Nonpredictive) Control of the Mobile Robot. The implementation of the classical control is depicted in Figure 10.

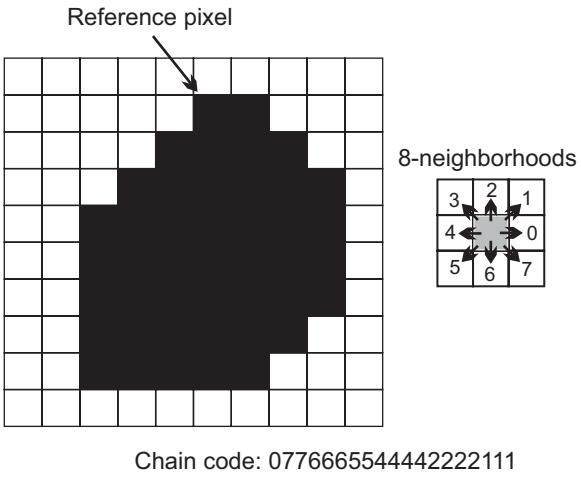

Figure 8 An example of the chain code.

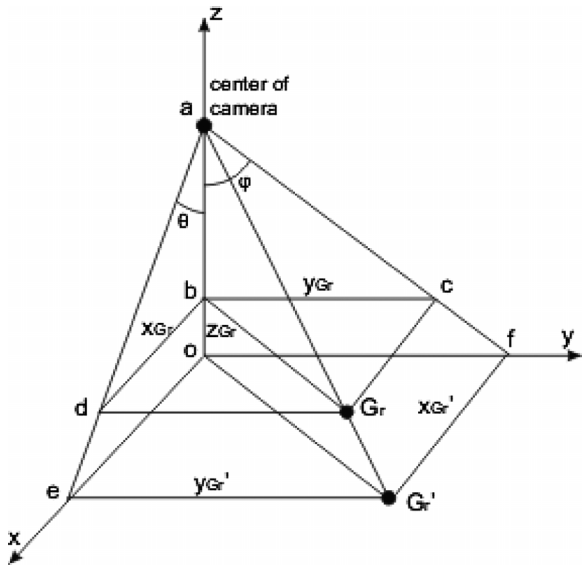

Figure 9 Geometrical model of image capturing.

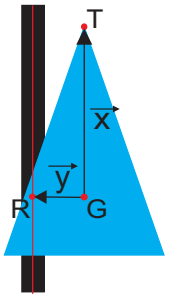

Figure 10 Classical (nonpredictive) control. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

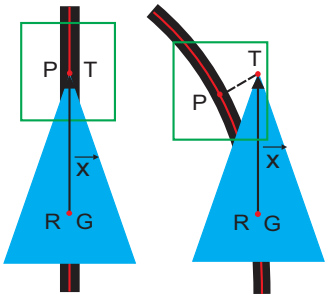

Figure 11 Predictive control. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

The first step in the classical control is the determination of the center of gravity of the mobile robot which is marked by point G on Figure 10. The point G is the center of gravity. The x and y components of the center of gravity of the blue isosceles triangle are calculated by the following equations [17,19]:

$$x_T = \frac{\sum_{i=0}^{n} x_i}{n}; \quad y_T = \frac{\sum_{i=0}^{n} y_i}{n}, \tag{2}$$

where n is the number of blue pixels which represent the mobile robot. The center of gravity is marked by a red dot in Figure 10.

The second red dot in Figure 10 is point R, which is the point on the trajectory closest to the center of gravity G.

The third red point is the point T, which is the blue point with the largest distance from the center of gravity. It can be used to determine the orientation of the robot. As we need to determine not only the distance of the robot from the desired trajectory, but also on which side of trajectory it is located, we defined vectors **x** and **y** as shown in Figure 10. The distance from the robot to the desired trajectory as well as its position relative to it (left or right side) can be determined by the cross product of vectors **x** and **y**. Its absolute value determines the distance from the center of gravity of the mobile robot to the desired trajectory, whereas its sign determines on which side of the trajectory the center of gravity is located. These two values are the inputs for the motor drive control algorithm. If the cross product is, for example, positive the robot is located on the right side of the reference trajectory. That means that the angular velocity of the right wheel has to be higher than the angular velocity of the left wheel. The difference between these two velocities depends on the absolute value of the cross product.

Predictive Control of the Mobile Robot. If the reference trajectory in front of the robot is known, it is better to use predictive control. The difference between predictive and nonpredictive control is analogous to the difference between driving a car in clear weather and in thick fog. In our case the predictive control is implemented with the assessment of the curvature of the reference trajectory in front of the mobile robot. If the trajectory is straight, the speed of the robot can be increased, whereas the speed is reduced if there is a curve in front of the mobile robot. The amount of this reduction in speed depends on the curvature of the reference trajectory. The basis for the predictive control algorithm is shown in Figure 11.

The first step in the determination of the curvature of the reference trajectory in front of the mobile robot is the task of finding the point P. It is the point on the trajectory closest to the point T on the triangle. The point P defines the center of 40×40 pixel square (marked by green square in Fig. 11), which is the area where the estimation of the curvature is made. The curvature of the trajectory is estimated by the Hough transformation [18,21] which is an efficient procedure for detecting lines and their curvature.

Obstacle Avoidance Algorithm

Another advantage of the visual information based line-tracing robot control is a relatively simple further development of the control algorithm. With only software modification it is possible to include an obstacle avoidance algorithm as well. In the previous section control algorithm for accurate navigation of mobile robot is presented. We firmly believe that this addition makes the proposed experiment even more interesting for students.

There are many different control algorithms for obstacle avoidance in a complex terrain [22,23]. In our case it is important that the obstacle avoidance algorithm is very fast and consequently not too complex. The algorithm assumes the obstacles are convex shaped. The obstacles were of a rectangular shape and blue color as shown in Figure 12.

As already mentioned, it is possible to detect the vertices of the object based on the chain code. It is then easy to determine the center of gravity (point G in Fig. 12) and draw an imaginary circular path around it. The inner circle shown in Figure 12 should be replaced by a circle with a larger diameter. The increase in this diameter depends on the width of the robot. This new (larger) circle intersects the line the robot should trace

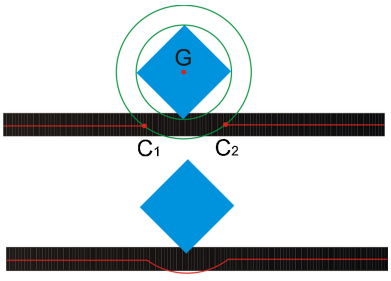

Figure 12 Obstacle avoidance. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

at points C_1 and C_2 (see Fig. 12). These two points divide the circle into two segments. The tracing line is now modified to include the shorter of the two segments as shown in the lower part of Figure 12.

MOBILE ROBOT MOTION CONTROLLER

The sign of the cross product $\mathbf{x} \times \mathbf{y}$ and length of vector $\mathbf{y}$ are used to determine the corresponding PWM voltage signals which drive the electric motors. The voltage level is calculated by the following equations:

$$V_R = \begin{cases} \alpha \cdot 1 & \text{If} \|y\| \cdot \text{sgn}(x \times y) > 0 \\ \alpha \cdot 2^{-\|y\|} & \text{else} \end{cases} \quad (3)$$

$$V_L = \begin{cases} \alpha \cdot 2^{-\|y\|} & \text{If} \|y\| \cdot \text{sgn}(x \times y) > 0 \\ \alpha \cdot 1 & \text{else} \end{cases}, \quad (4)$$

where α is the proportional gain (see Fig. 13). V_L and V_R are input voltages for the left and right motors, respectively. Both voltages are defined by the product between the normalized voltage $2^{-||y||}$ and the proportional gain α. If the proportional gain α has the maximum value and $||y|| = 0$, then the voltages V_L and V_R will be maximum. The value of the normalized voltage depends on the length of the vector $\mathbf{y}$. The value of the second term (the proportional gain α) is constant if classical (nonpredictive) control algorithm is used. The predictive control algorithm, however, changes the value of this parameter depending on the curvature of the reference trajectory in front of the mobile robot.

EXPERIMENTAL RESULTS

Both types of control algorithms (nonpredictive and predictive one) were tested using the reference trajectory shown in Figure 3. The motion of the robot using both the algorithms can be seen on YouTube [24]. We used the Imaging Source CCD camera with 640 × 480 pixels resolution and the recording rate of 30 frames per second. The data acquisition of the images obtained by the CCD camera was performed by a PC with 2.80 GHz Intel Core Duo CPU processor and 4.00 GB RAM.

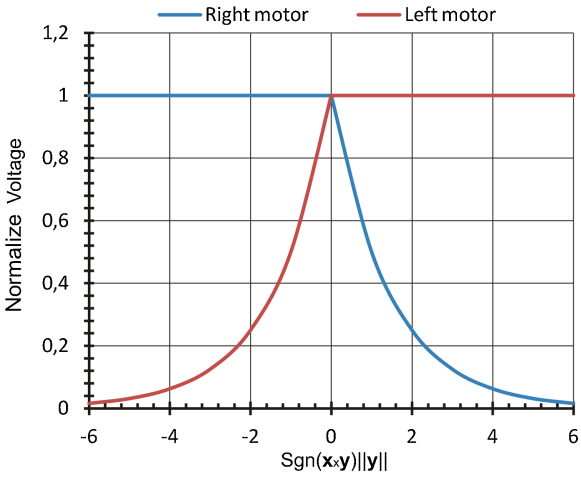

Figure 13 Normalized functions for the motion controller. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In the video it can be seen that the mobile robot is able to track the reference trajectory accurately and with minimal oscillation. The varying speed of the robot depending on the curvature of the reference trajectory in the case of the predictive control can also be observed.

The main problem of this type of control is the latency of the camera. The time delay of the whole system is around 60 ms. In order to estimate the latency of different cameras, we wrote a simple program, which shows a different rectangle on the screen every second. The camera is located perpendicular to the screen so that it records the scene on the screen. The program calculates the coordinates of the center of gravity of the rectangle: once for the rectangle drawn on the screen and once for the one obtained from the image processing algorithms. The difference in time between these two values is an indication of the time delay. The large time delay was the main reason for the purchase of the CCD camera. The time delays associated with the common web camera are simply too large for real time control purposes. If the proportional gain α is very low, the motion can be controlled accurately. If the proportional gain α is however large, the robot oscillates too much. So the selection of the camera should focus primarily on a small time delay and not so much on high resolution.

The experiment with one or more obstacles on the path can also be seen online [25].

CONCLUSION

The article shows that it is possible to control the motion of a simple mobile robot with an inexpensive CCD camera mounted on the ceiling. Web cameras are currently not good enough because the time delay associated with their application is too large. As CCD cameras can nowadays be purchased at low prices as well, this should not be such a big problem. The article describes the sequence of the image processing algorithms needed to guide the line-tracing robot. This approach makes it possible to implement the predictive control as well. The experiment clearly showed the advantage of the predictive control. Image processing based robot guidance enables us to easily implement the obstacle avoidance as well. The algorithm needed in order to fulfill this task is presented as well. It should be emphasized that if classical approach to this problem is chosen (phototransistors or buried wires), the design of the robot has to be changed significantly in order to implement these tasks.

The experiment presented in the article is very interesting for high school or university level courses in image processing and/or mechatronics. All the necessary equipment is inexpensive. We think that it is of great didactic value because a wide variety of tasks can be performed by simply modifying the algorithm. So it is up to the instructor to judge the capability of the students and choose the tasks he thinks they can comprehend. Later during lectrures, when they learn all the prerequisites, they can return to the same object (robot) and implement another task. We think that such an approach can be motivating for the students, because they can implement their new knowledge on the object they already know and because on this knowledge they can extract knew performance from the same device.

REFERENCES

[1] A. Sánchez, X. Hernández, O. Torres, and A. Toriz P., Mobile robots navigation in industrial environments, Int. Conf. on Computer Science, Mexico, 2009, pp 155–166.

[2] J. Forlizzi and C. DiSalvo, Service robots in the domestic environment: A study of the roomba vacuum in the home, Proc. First ACM SIGCHI/SIGART Conf. Human–Robot Interaction, New York, 2006, pp 258–265.

[3] N. Houshangi and T. Lippitt, Omnibot mobile base for hazardous environment, IEEE Int. Conf. Electrical and Computer Engineering, Canadian, 1999, pp 1357–1361.

[4] D. J. Pack, R. Avanzato, D. J. Ahlgren, and I. M. Verner, Fire-fighting mobile robotics and interdisciplinary design-comparative perspectives, IEEE Trans Educ 47 (2004), 369–376.

[5] P. Podržaj and H. Hashimoto, Intelligent space as a framework for fire detection and evacuation, Fire Technol 44 (2008), 65–76.

[6] M. P. Cuéllar and M. C. Pegalajar, Design and implementation of intelligent systems with LEGO Mindstorms for undergraduate computer engineers, Comput Appl Eng Educ (2011), Published online in Wiley Online Library; DOI: 10.1002/cae.20541.

[7] M. Farrokhsiar, D. Krys, and H. Najjaran, A teaching tool for the state-of-the-art probabilistic methods used in localization of mobile robots, Comput Appl Eng Educ (2010), Published online in Wiley Online Library; DOI: 10.1002/cae.20443.

[8] N. Yilmaz and S. Sagiroglu, Real-time line tracking based on web robot vision, Comput Appl Eng Educ (2009), Published online in Wiley Online Library; DOI: 10.1002/cae.20367.

[9] D. Ibrahim and T. Alshanableh, An undergraduate fuzzy logic control lab using a line following robot, Comput Appl Eng Educ (2009), Published online in Wiley Online Library; DOI: 10.1002/cae.20347.

[10] J. F. Martin and L. Chiang, Low cost vision system for an educational platform in artificial intelligence and robotics, Comput Appl Eng Educ 10 (2002), 238–248.

[11] J. M. Gómez-de-Gabriel, A. Mandow, J. Fernández-Lozano, and A. García-Cerezo, Using LEGO NXT mobile robots with LabVIEW for undergraduate courses on mechatronics, IEEE Trans Educ 54 (2011), 41–47.

[12] C.-S. Lee, J.-H. Su, K.-E. Lin, J.-H. Chang, and G.-H. Lin, A project-based laboratory for learning embedded system design with industry support, IEEE Trans Educ 53 (2010), 173–181.

[13] A. H. Ismail, H. R. Ramli, M. H. Ahmad, and M. H. Marhaban, Vision-based system for line following mobile robot, IEEE Symposium on Industrial Electronics and Applications, Malaysia, 2009, pp 642–645.

[14] N. Chen, A vision-guided autonomous vehicle: An alternative micromouse competition, IEEE Trans Educ 40 (1997), 253–258.

[15] J.-H. Lee and H. Hashimoto, Intelligent space, its past and future, The 25th Annual Conf. of the IEEE Industrial Electronics Society, vol 1, 1999, pp 126–131.

[16] J.-B. Coulaud, G. Campion, G. Bastin, and M. De Wan, Stability analysis of a vision-based control design for an autonomous mobile robot, IEEE Trans Robotics 22 (2006), 1062–1069.

[17] A. K. Jain, Fundamentals of digital image processing, 1st ed., Colleen Brosnan Prentice Hall, Englewood Cliffs, NJ, 1989.

[18] R. C. Gonzalez and R. E. Woods, Digital image processing, 3rd ed., Pearson Education Prentice Hall, Upper Saddle River, NJ, 2008.

[19] M. Sonka, V. Hlavac, and R. Boyle, Image processing, analysis and machine vision, 3rd ed., Thomson Learning, Australia, 2008.

[20] M. Sonka, E. L. Dove, and S. M. Collins, Image systems engineering education in an electronic classroom, IEEE Trans Educ 41 (1998), 263–272.

[21] R. O. Duda and P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Int J Graph Image Process 15 (1972), 11–15.

[22] C.-W. Hun, H.-C. Chen, and M.-F. Hwang, On-line algorithm for optimal 3D path planning, Comput Appl Eng Educ (2010), Published online in Wiley Online Library; DOI: 10.1002/cae.20441.

[23] E. S. Conkur, RoboKol: A computer program for path planning for redundant and mobile robots, Comput Appl Eng Educ 14 (2006), 198–210.

[24] http://www.youtube.com/watch?v=esUIeAMood8&feature=more_related, the YouTube website, [accessed on September, 2010].

[25] http://www.youtube.com/watch?v=ZScY_1k3D98, the YouTube website [accessed on June, 2011].

BIOGRAPHIES

Samo Simončič received the BSc degree in mechanical engineering from the University of Ljubljana, Ljubljana, Slovenia, in 2009. He is an assistant with the Faculty of Mechanical Engineering in Slovenia. His postgraduate research are vision controlled mobile robots and manipulators.

Primož Podržaj received the MSc degree and the PhD degree in automatic control from the University of Ljubljana, Ljubljana, Slovenia, in 2000 and 2004, respectively. He is an Assistant Professor with the Faculty of Mechanical Engineering, University of Ljubljana. His research interests include control systems, control system design, and industrial automation.