ORIGINAL ARTICLE

Real Time Image Processing on Object Tracking CNC

Hadiseh Babazadeh [1]*, Sobhan Javan [2]

[1]Faculty of Electrical Engineering, Urmia University of Technology, Urmia 57166-17165, Iran

[2] Faculty of Electrical Engineering, Urmia University of Technology, Urmia 57166-17165, Iran

*Corresponding author h.babazadeh@uut.ac.ir

Received: 14th December 2021
Revised: 16th May 2022
Accepted: 24st June 202

OPEN ACCESS

Abstract: Object tracking is one of the main issues in image processing. A real-time CNC is designed and implemented in which the target image is taken using the digital camera every ten milliseconds. After extracting the new position, the target's exact location is sent to the CNC; then, the CNC pen will move to that location and copy the target path. In this project, a Raspberry Pi 3 Model B+ board is used as the processor and the controlling board, a Raspberry Pi digital camera is the input unit, and the CNC motors and its pen are the output unit. The target is a small dark dot moving on a white surface with the dimensions of 4.5 cm × 4.5 cm. The camera is placed under the surface, so a tiny magnet is used on the surface, which the user can easily displace to move the target. Kinematic analysis of the proposed structure is done using Denavit-Hartenberg parameters. The CNC is programmed in Python and implemented using stepper motors and an output plane. Template Matching TM_CCOEFF_NORMED function of the OpenCV image processing library of Python is used to read and detect the target. Various test results show that the machine is successful both in repeating the patterns created by the user and reproducing the same line without deviation.

Keywords: image processing, Python, Raspberry Pi, CNC, kinematics, Denavit-Hartenberg, stepper motor, digital camera

Introduction

CNC, which stands for Computerized Numerical Control, is a mechanical framework for various applications from home workshops to high-level industrial usages. In this manufacturing machine, usually, a pre-programmed code (known as G-code) controls the movement of the end effectors. New equipment and features are added to the device every day, making CNCs more powerful and valuable in the industry. The main applications are cutting or carving tools using drilling machines [1,2], where the end effector's exact location and trajectory are totally controlled using controller boards.

Combining CNC with image processing makes it more powerful and suitable for various applications. A digital camera can capture pictures, and the CNC is supposed to act according to them. For example, image duplication of a key on a blank one can be automatically done [3], where the image of the key is taken and, after compiling to G-code, is transferred to the CNC to create the copy. Or the image processing can be used to detect faults instead of heavy and power-consuming sensors [2,4]. Since the camera does not load the system electronically and can be moved easily toward new targets [5,12], image processing has become very popular [16,17].

However, real-time code generation is of interest as well [6-8]. Tracking a moving target, for example, needs taking pictures and processing them once in a while to follow the target [9]. Surgical robots that are doing surgery based on the image sent by a micro camera in the patient's body [10], or anywhere a robotic arm is used to do tasks according to the image processing [11] are examples of this type.

Moreover, studies of mechanical structures' forward and inverse kinematics are critical to perform simulations before implementation. Using Denavit-Hartenberg parameters (DH parameters) is a famous and well-known In this paper, a target is moving on a surface by the user, and the CNC is supposed to repeat its trajectory on another surface using a pen, as shown in Fig. 1. To do this, the device is made into three separate units: input, processing and output unit. Processing and control of the motors are done using Raspberry Pi 3 Model B+ board, and a digital camera suitable for Raspberry Pi takes the real-time position of the target as the input unit. Finally, the result is transferred to the motors of the biaxial CNC as the system output. DH parameters extract the CNC forward kinematic model, and the geometrical method is used to establish the inverse kinematics analysis.

Usual CNCs work with predefined codes (G-Code); here, a CNC is presented, which can generate real-time codes for CNC using a raspberry pi board and in Python programming language, so there is no need for computers and no need for G-Codes. On the other hand, the raspberry pi can take images from its camera module, which is a very cheap and available module, making this kind of CNC implementation easier.

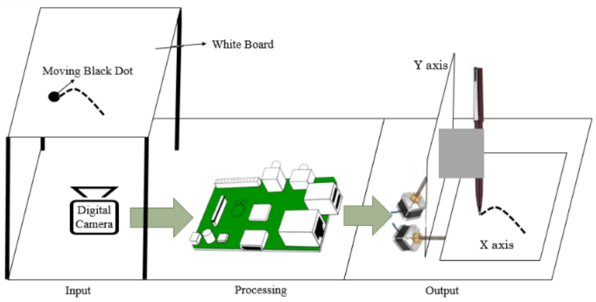


Figure 1. Block diagram of the online object tracking CNC

In the following, at first different parts of the project are described, along with the programmed code explanation, and then the kinematic model is presented. The final part includes some test results of the CNC.

Different Parts

Input Unit

The target is a moving black dot that will be discerned on a white square board. A Raspberry pi suitable digital camera (Fig. 2) takes the target image and sends it to the processing unit once every ten milliseconds. Communication between the camera and the processing unit is established through a 15cm flat cable.

The camera needs about three seconds for the first tuning, and the minimum distance for the proper focus can be trimmed using a tiny wheel on the module[18]. Here a distance of 14cm between the digital camera and the whiteboard is set, and the autofocus mode is used to make the camera to be able to focus correctly on the moving black dot and take clear pictures. No special light source is added and the pictures are taken in the ambient light. The black dot can be moved in the two-dimensional space; then, the CNC is supposed to track and draw the same pattern using a pen.

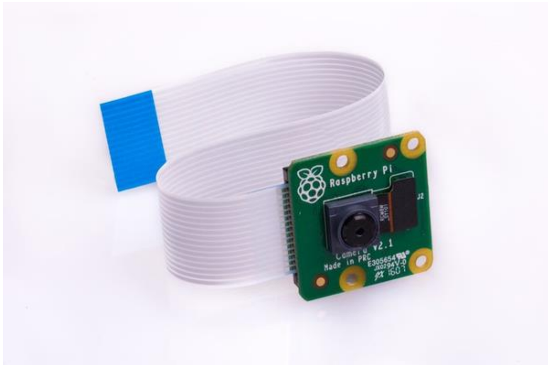


Figure 2. Raspberry Pi digital camera

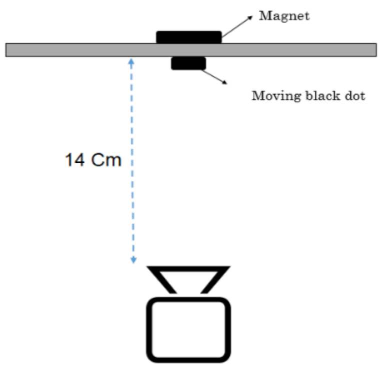


Figure 3. The platform layout of the camera and the target

To implement the idea, a whiteboard with dimensions of 4.5cm × 4.5cm is considered as the surface, where a small magnet is moved on it by the user (Fig. 3). The movement of the magnet is transferred to a small iron piece of metal which is taken as the target by the camera under the surface. Edges of the surface are divided into 270 parts; hence each pixel on the whiteboard is a square

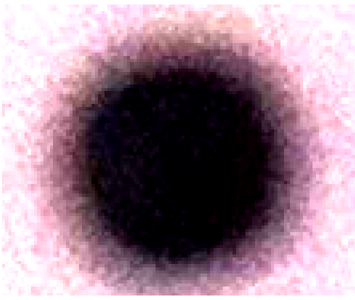


Figure 2. Closed shot of the target

with dimensions of about 0.16mm.

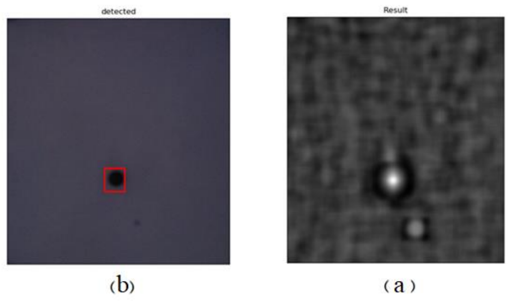


Figure 3. a) the most matched part of input image with the template, b) detected target

Processing Unit

All the process of finding coordinates of the black dot and commanding the axes of the CNC is done by a Raspberry Pi 3 B+ using Python programming language.

As mentioned earlier, an image of the target is sent to the processor every ten milliseconds. Then the processor needs to process the image and discern the black dot; Template Matching Technique will do this. For this purpose, a closing shot of the target (Fig. 4) will be used as the reference template to seek its properties in the received image from the digital camera. After finding the part of the image that completely matches the template, it is possible to see its coordinates on the whiteboard. There are six methods of Template matching in the Python image processing library. All of the methods have been tested, so the most proper way for this purpose is TM_CCOEFF_NORMED (from the OpenCV library) [14]. This method works with the following formula:

$$R(x, y) = \sum x', y'T' \ (x', y')\ 2\ .\sum x', y'I'(x+x', y+y')\ 2\sum x', y' \ (T'(x', y').I'(x+x', y+y')) \quad (1)$$

Applying the above formula to the image, the result will be like what is shown in Fig.5-a. The whitest part of Fig.5-a is the most matched part of the input image with the template. So, the detected target in Fig.5-b is correct. To obtain the coordinates of the detected point, the function "cv2.minMaxLoc ()" is used. Using this function, the coordinates will be saved in two different variables. Fig. 6 shows the flow chart of the code.

Output Unit

A biaxial CNC is used to print the resulting pattern. Each axis is moved by a four-wired stepper motor with a 4.5cm shaft, as shown in Fig.7.

The 4.5cm is traversed through 270 steps. Thus, the length of each step will be 0.16mm. According to the library written for Raspberry Pi to turn the stepper motors, four steps will be passed each time they get commands to move. Therefore, the length of each sketched step will be about 0.66mm. The output surface is a 4.5cm × 4.5cm square containing 4489 pixels. The origin of coordinates is marked in Fig. 8.

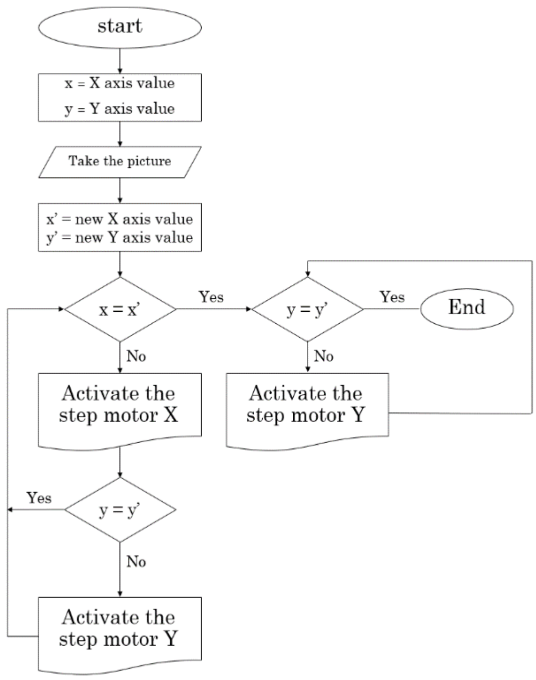


Figure 4. Program flow chart

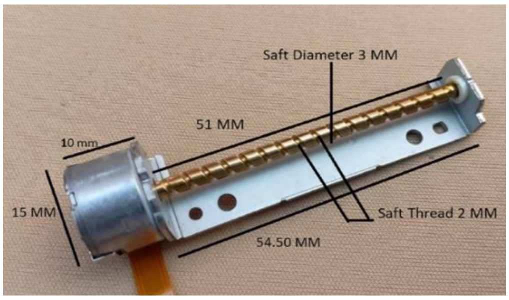


Figure 7. Four wired stepper motor with 4.5Cm shaft

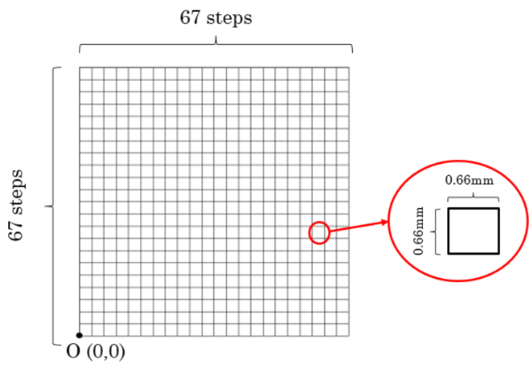
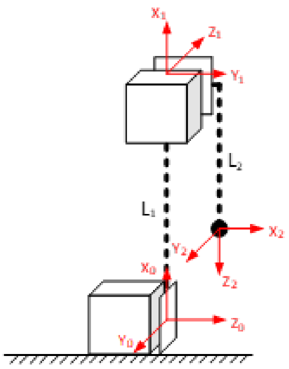


Figure 8. Output surface

 https://doi.org/10.5875/ausmt.v12i1.2386

Motor movements are done according to the commands sent from the processor. The procedure considers the new location and the current location, and the difference is calculated linearly. The steps of movement in each direction are sent to the motors.



Link	a_i	α_i	d_i	θ_i
1	L_1	90°	$d1^*$	0
2	L_2	90°	$d2^*$	90°

Figure 9. Kinematic model and DH parameters

Kinematic Model

In Fig. 9, the kinematic model of the proposed CNC is presented. Coordinates are assigned to joints, and DH parameters are brought to the table. The homogeneous transformation matrix to conform two coordinates can be calculated as shown in Fig. 9 [15]:

$$A_i = Rot_{z,\theta}.Trans_{z,d}.Trans_{x,a}.Rot_{x,\alpha} \quad (2)$$

$$A_i = \begin{bmatrix} C_\theta & -S_\theta C_\alpha & S_\theta S_\alpha & a_i C_\theta \\ S_\theta & C_\theta C_\alpha & -C_\theta S_\alpha & a_i S_\theta \\ 0 & S_\alpha & C_\alpha & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Which Roti,j denotes rotation about i axis for the amount of j radians, and Transi,j is displacement along i axis for j meters. Since L1 and L2, lengths of the first and second link, are 10 and 8 centimeters, respectively, the homogeneous transformation matrices will be as follows:

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_1^* \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$A_2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0.08 \\ 0 & 0 & 1 & d_2^* \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$T_2^0 = A_1.A_2 = \begin{bmatrix} 0 & -1 & 0 & 0.1 \\ 0 & 0 & -1 & -d_2^* \\ 1 & 0 & 0 & 0.08 + d_1^* \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

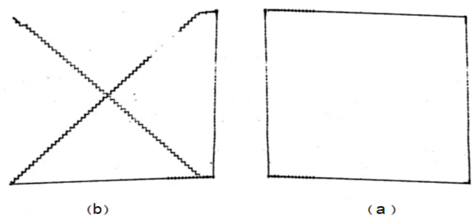


Figure 10. Geometrical model

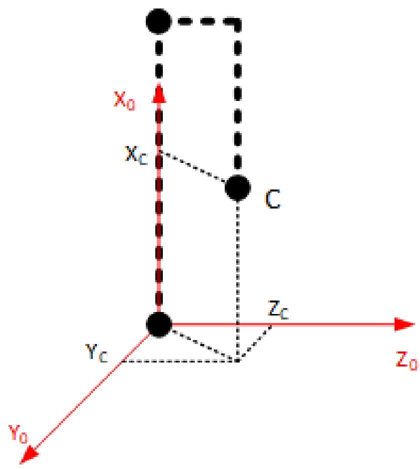


Figure 11. Edges and diagonals of a square

The last matrix shows the transformation from the base to the end effector. For inverse kinematics, the geometrical method can be used where the variable amounts of the joints are specified based on the end-effector position. The model is resketched in Fig. 10; according to the model and the location of the joints, it's clear that:

$$d_1 = Z_C, \ d_2 = Y_C \quad (7)$$

Test Results

The system is tested several times with different patterns to check its performance. The purpose of the whole machine is to follow the pattern sketched by the user correctly, so the tests are divided into two groups, one of them being shapes of different lines, e.g., edges or diagonals of a square. The other type of test is repeating a specific line or some parallel lines to examine the repeatability of the CNC. These tests show important properties, like ability to control motors and sketch edges in real time which means right after the user draws a design. Also experiments verity that the output line will have no deviation in the case of repeating the same path.

Fig. 11 shows the shape of a square and its diagonals. In Fig. 12, two parallel 'a' and 'b' lines are plotted. Also, in this test, line b is repeated twice to check for any deviation or error in the plot, and because of that, the line is highlighted. Fig. 13 shows test with more parallel lines.

Figure 5. Parallel and repeated line test

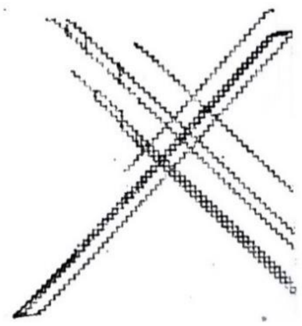


Figure 6. Parallel lines test

Conclusion

A real-time object tracking CNC machine is presented, which captures an image of the target every ten milliseconds using a digital camera and copies the trajectory. A Raspberry Pi board is programmed in Python to process the image and calculate the target's new position. The new location is sent to stepper motors to move the CNC pen and repeat the target path.

For future works, the wireless version of the project can be considered along with the parallel processing for commanding motors simultaneously.

References

[1] T. Yu, T. Liu, W. Wang, S. Yang, "Application of Image Processing Technology in INC," IEEE 6th World Congress on Intelligent Control and Automation, Vol. 2, pp. 6203-6207, 2006.

[2] R. G. Lins, B. Guerreiro, P. R. M. De Araujo, R. Schmitt, "In-process tool wear measurement system based on image analysis for CNC drilling machines," IEEE Transactions on Instrumentation and Measurement, 69(8), 5579-5588, 2019. https://doi.org/10.1109/TIM.2019.2961572

[3] V. Itthisupornrat, S. Kiatsunthorn, "Duplicate a blank key using image processing," IEEE SICE-ICASE International Joint Conference, pp. 5711-5714, Oct. 2006. https://doi.org/10.1109/SICE.2006.314640

[4] A. Siddhpura, R. Paurobally, "A review of flank wear prediction methods for tool condition monitoring in a turning process," International Journal of Advanced Manufacturing Technology, 65(1-4), 371-393, 2013. https://doi.org/10.1007/s00170-012-4177-1

[5] L. Gomes, S. Bogosyan, "Current trends in remote laboratories," IEEE Transactions on industrial electronics, 56(12), pp. 4744-4756, 2009. https://doi.org/10.1109/TIE.2009.2033293

[6] F. Bakhshande, H. D. Taghirad, "Visual Tracking using Kernel Projected Measurement and Log-Polar Transformation," International Journal of Robotics, Vol. 4, No. 1, pp. 1-11, 2015.

[7] K. Horak, L. Zalud, "Image processing on raspberry pi for mobile robotics," International Journal of Signal Processing Systems, 4(2), 1-5, 2016.

[8] W. Chantara, J. H. Mun, D. W. Shin, Y. S. Ho, "Object tracking using adaptive template matching," IEIE Transactions on Smart Processing and Computing, 4(1), 1-9, 2015 https://doi.org/10.5573/IEIESPC.2015.4.1.001

[9] M. Sharikmaslat, R. Sidhaye, A. Narkar, "Image Processing Based Human Pursuing Robot," IEEE 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) pp. 702-704, Jun. 2019. https://doi.org/10.1109/ICECA.2019.8821861

[10] S. Tully, H. Choset, "A filtering approach for image-guided surgery with a highly articulated surgical snake robot," IEEE Transactions on Biomedical Engineering, 63(2), 392-402, 2015. https://doi.org/10.1109/TBME.2015.2461531

[11] R. Szabo, A. Gontean, "Industrial robotic automation with Raspberry PI using image processing," IEEE International Conference on Applied Electronics (AE), pp. 265-268, Sep. 2016. https://doi.org/10.1109/AE.2016.7577287

[12] S. Shirmohammadi, A. Ferrero, "Camera as the instrument: The rising trend of vision-based measurement," IEEE Instrumentation & Measurement Magazine, 17(3), pp. 41-47, 2014 https://doi.org/10.1109/MIM.2014.6825388

[13] S. H. Ahmed, N. Razzaq, Z. Malik, U. Qadeer, I. Sarfraz, A. Sharif, "Design & fabrication of MATLAB based solar powered CNC machine," IEEE 3rd International Conference on Control Science and Systems Engineering (ICCSSE), pp. 265-268, Aug. 2017. https://doi.org/10.1109/CCSSE.2017.8087937

[14] C. R. Severance, S. Blumenberg, Python for Everybody, Create Space Independent, 2016.

[15] P. Corke, Robotics Vision and Control, Springer, 2017. https://doi.org/10.1007/978-3-319-54413-7

[16] Y. Chou, W. Liao, M. Chang, P. Lin, "An Intelligent Image Processing System for Real-Time Detection of Surface Flaws," International Journal of Automation and Smart Technology, Vol. 6, No.2, 2016. https://doi.org/10.5875/ausmt.v6i1.1006

[17] J. Tang, C. Tsai, F. Tai, "Object Recognition and Cooperative Pick-and-Place of a Single Spherical Wheeled Mobile Robot with Dual Arms," International Journal of Automation and Smart Technology, Vol. 8, No.1, 2018. https://doi.org/10.5875/ausmt.v8i1.1574

[18] Gareth Halfacree, The Official Raspberry Pi Beginner's Guide, 3rd edition, Raspberry Pi Press, 2019

Publisher: Chinese Institute of Automation Engineers (CIAE)

ISSN: 2223-9766 (Online)