Bulletin of Electrical Engineering and Informatics
Vol. 12, No. 2, April 2023, pp. 771~781
ISSN: 2302-9285, DOI: 10.11591/eei.v12i2.4538

Field programmable gate array based moving object tracking system for robot navigation

Hoang T. Tran[1], Dong LT. Tran[1], Quang N. Pham[1], Thanh C. Vo[1], Quan NA. Nguyen[1], Thang K. Nguyen[1], Duyen M. Ha[1], Minh T. Nguyen[2]
[1]Center of Electrical Engineering, Duy Tan University, Da Nang, Vietnam
[2]Thai Nguyen University of Technology, Thai Nguyen, Vietnam

Article Info

Article history:

Received Aug 8, 2022
Revised Oct 6, 2022
Accepted Nov 2, 2022

Keywords:

Embedded system
FPGAs
Homomorphic filter
Median filter
MicroBlaze
Robot chasing objects

ABSTRACT

This paper proposes a method in which an object tracking robot system is implemented on field programmable gate arrays (FPGAs). The OV7670 camera provides real-time object pictures to the system. To improve picture quality, images are put via the median filter phase. The item is distinguished from the backdrop based on color (red), after which it is subjected to a mathematical morphological approach of filtering to eliminate noise. To send the robot control signals, the object's (new) coordinates are found. In this method, the median filter, color separation, hardware IP cores, and morphological filter are all part of the embedded system on FPGA. Through the direct memory access (DMA) controller, these cores may communicate and perform high-speed pipeline computing at higher data rates. The entire system is executed in real-time on Xilinx's spartan-6 FPGA KIT. The results show practical and promise.

Corresponding Author:

Minh T. Nguyen
Thai Nguyen University of Technology
Thai Nguyen 240000, Vietnam
Email: nguyentuanminh@tnut.edu.vn

1. INTRODUCTION

The practice of tracking an object's movement over time using a camera is called object tracking [1]–[3]. Since the locations of the objects are always changing, tracking moving images is a crucial task in surveillance systems [4], [5]. Application areas for object recognition and tracking include autonomous robot navigation, surveillance, and vehicle navigation [6], [7]. Finding things throughout a series of frames is the process of object detection. Object tracking uses a camera to find the items that will gradually appear. A hard digital signal processing module, distributed memory, and a number of programmable logic blocks make up a field programmable gate array (FPGA), which can handle real-time objects [8], [9]. There are several real-world uses for object tracking, including security, surveillance, autonomous driving, automated traffic management, biological image analysis, and intelligent robot control [10], [11]. The goal of object tracking, like that of the majority of computer vision systems, is to identify and extract the target item from a stream of images that the camera continuously records [12]–[14]. Better object tracking is made possible by faster image processing computation. In actuality, when object tracking apps are deployed, they are often created on the open CV library [15]–[17] and operate on Windows or Linux. As a result, the graphics libraries used in image processing make the program's execution speed extremely reliant on the hardware setup, which raises the cost of design. Due to the current demand, tracking systems must be as affordable as feasible while still meeting the requirements of processing speed, well-handling, accuracy, and reaction time. Because of the quick development time and low cost, while yet meeting the needs of reaction speed and precision, FPGA is the ideal solution for developing relativistic tracking algorithms.

In our study, the system makes use of a morphological noise-removal filter, a median filter, and red color to distinguish the target item from background objects. The robot motor is given a control signal by determining the object's location. In order to boost the system's speed, the system is constructed on an FPGA utilizing a combination of hardware cores and embedded microcontrollers from MicroBlaze. Different from previous work, the contributions of this work can be pointed out shortly as; i) an object tracking robot system based on FPGA is constructed, ii) the system captures real-time photos of objects from the OV7670 camera, iii) a mathematical morphological method is utilized to remove noise from around the object, and iv) the entire system is run in real-time on the Xilinx's spartan-6 FPGA KIT.

The rest of this paper is organized as follows. Section 2 provides the proposed object tracking system including an overview of the tracking system and a description of the algorithms to be used in the system. Section 3 describes the structure diagrams of the embedded system on the KIT FPGA and details the IP cores in the system. In section 4, some results of the implementation of the tracking system on FPGA and the experimental evaluation results are provided. Finally, conclusions and future work are addressed in section 5.

2. THE PROPOSED OBJECT TRACKING SYSTEM

2.1. System overview

First, the camera will provide the system one picture frame. The picture is then subjected to object separation image filtering to enhance image quality, red color separation, and eliminate noise-free zones. Following object separation, it will ascertain the item's shape and its location with respect to the image's center before generating a motor control signal that will cause the robot to move in that direction. A process of the object tracking system is shown in Figure 1.

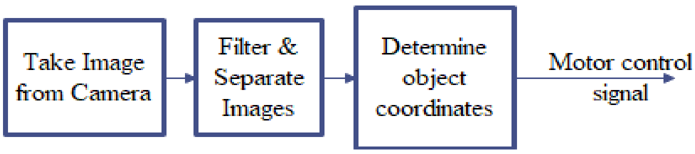

Figure 1. A process of object tracking system

2.2. Deploying algorithm steps

2.2.1. Collecting object images

The OV7670 camera continually recorded photos of objects. The camera's data is delivered by the camera in parallel 8-bit frames, and the system configures the camera using the standard serial camera control bus (SCCB) [16]. The received picture is a 320 by 240 pixel RGB565 color image [18].

2.2.2. Apply filters to separate objects

The color on red, green, and blue (RGB) images collected from the camera will be applied a median filter of size 3x3 to reduce noise. Then, the image will be color separated into a binary image in which only the black background and white pixel blocks represent the object to be processed. The binary image is then applied a morphological filter to enhance the quality to make it easier to find the contour around the object.

a. Filter the median

Boyle's median filter is a popular choice for mild noise reduction (impulse noise). The median filter minimizes noise by replacing a pixel's value with the median of the gray levels of nearby pixels because pulse noise frequently appears unique, and its gray level value differs significantly from that of its neighbors. The core principle of the median filter technique is to employ a mask that scans every pixel of the input picture sequence; typically, a mask of sizes 3x3, 5x5, and 7x7 is employed. A 3x3 median filter is shown in Figure 2. Take the value of the associated pixels in the mask area at each pixel location and arrange them in ascending or descending order. After sorting the range of pixels for the pixel value being taken into consideration for the final picture, assign the pixel to the middle (median) of the range [19].

b. Color separation of objects

The color separation block, which is a block that recognizes the object based on the color of the item compared to the picture backdrop, is used to filter the noise after obtaining the RGB image. It is important to examine the RGB values at several locations across the image in order to distinguish the color of the item. The red ball used in this article has an RGB color space value of (255, 0, 0), however the picture captured by the camera has a low resolution and is influenced by light. Light causes the obtained R, G, and B picture area's red color to vary around the values 255, 0, 0. The object's color value is equal to 1, the backdrop color returns a value of 0, and the thresholds for the color channels R, G, and B are experimentally chosen to separate the color. Figure 3 shows the binary output picture that is the end result.

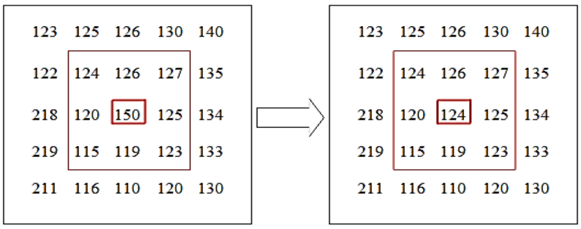

Figure 2. An illustration of the median filter: the median value of its neighboring pixels (3×3) replaced the current pixel value

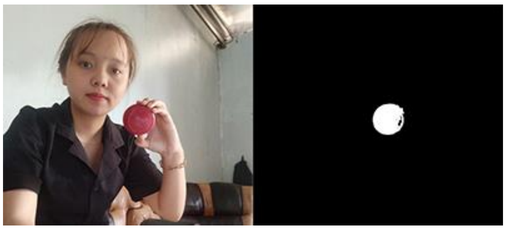

Figure 3. Image after filtering and color separation

c. Morphological filtering

Due to the aforementioned circumstances, the object that was removed from the backdrop becomes the subject of the survey as well. This includes noise and any interior spaces that were left empty due to background confusion. Therefore, the object should be refined by eliminating the noise regions that are not the object and filling the vacant spaces inside the object in order to guarantee that the best information is delivered for the following blocks of the system. Using mathematical morphology (MM) is one of the techniques used to filter items after removing them from the background [20], [21].

MM is a set theory-based method for treating geometrical structures. The basic morphological procedures used by this approach, which is based on structure and shape, allow the picture to be made simpler while preserving the key elements seen in Figure 4 of the original. In order to assess if a given basic block, or structural element, fits or misses the form in the picture, the fundamental goal of MM is to find images that contain that block. In the case of Figure 4(a), applying dilation helps to connect the dashed points of the image that increases the details of the image. Next, in Figure 4(b), the erosion removes groups of pixels that are much smaller than the size of the object in the image to remove noisy areas for more accurate object for identification. There are 4 basic morphological operations [22] in Figure 5; i) dilation: used to expand or thicken the object in the frame, ii) erosion (shrink): used to shrink or thin the object in the frame, iii) opening: opening combines an erosion and a dilation with the same structuring element, and iv) closing: closing combines a dilation and an erosion with the same structure element.

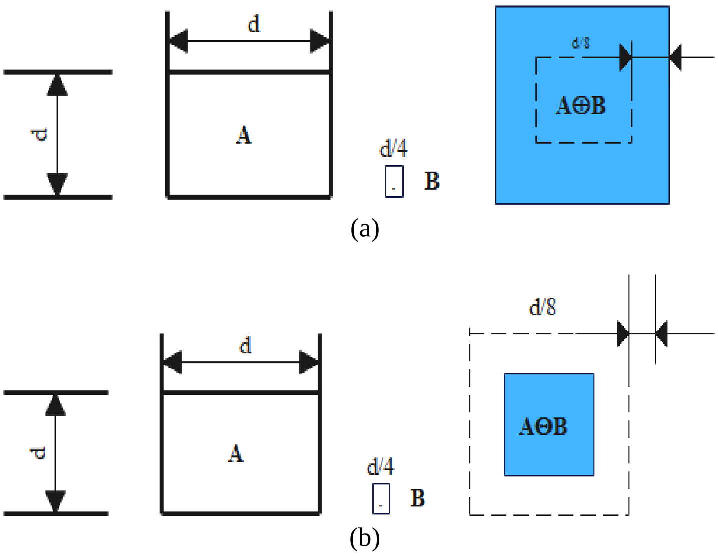

Figure 4. An example of applying two MMs on block A, (a) block A after performing the dilation and (b) block A after performing the erosion

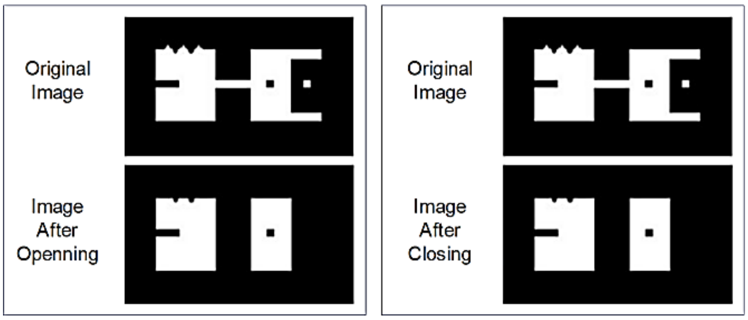

Figure 5. An example of image after performing the openning and closing

2.2.3. Determine object coordinates

In Figure 6, the first thing we can find a way around the object. The contour edges are constructed based on the minimum distance from the subject to the corresponding sides of the frame. The coordinates of the object are determined based on the center of the contour relative to the center of the image frame. From there, determine the direction of movement for the robot so that the object returns to the center of gravity.

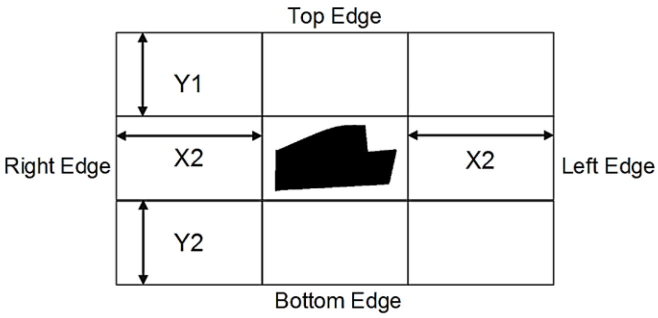

Figure 6. Find the object contour

3. FPGA EMBEDDED SYSTEM STRUCTURE

3.1. Embedded system structure diagram

Figure 7 shows a detailed built-in diagram of the embedded system on the spartan-6 FPGA SP605 evaluation KIT including:

- One 32-bit MicroBlaze processor core [23] running at 100 MHz with 32 K of data and instruction memory is connected to high-speed computer and peripheral cores through the AXI interface.
- A single UART controller with a baud rate of 128,000 that can transport from the computer to the board the picture to be processed and receive the completed image for display on the computer.
- The picture to be filtered is stored in external RAM with a maximum memory capacity of 128 MB, which is connected to a single core SDRAM controller.
- IP core for a median filter with two memory FIFOs and one median filter.
- The IP core does the dilation process, which includes the morphological filtering of 1 math filter and 2 FIFO memory. The IP core performs erosion math (performs morphological filtering), includes 2 FIFO memories and 1 math filter core erosion.
- a DMA controller core that uses Xillinx to increase data processing performance while moving data between hard-core IP cores and external RAM [24].
- The PWM core regulates pulses to operate the robot's motor.
- Two Xilinx-powered clock sources, one with a 200 Mhz oscillator (both positive and negative side) and the other with a 27 Mhz oscillator (single rib) [25].
- The system employs the AXI interface, which includes AXI4, AXI4-Lite, and AXI stream [26], to connect with MicroBlaze.

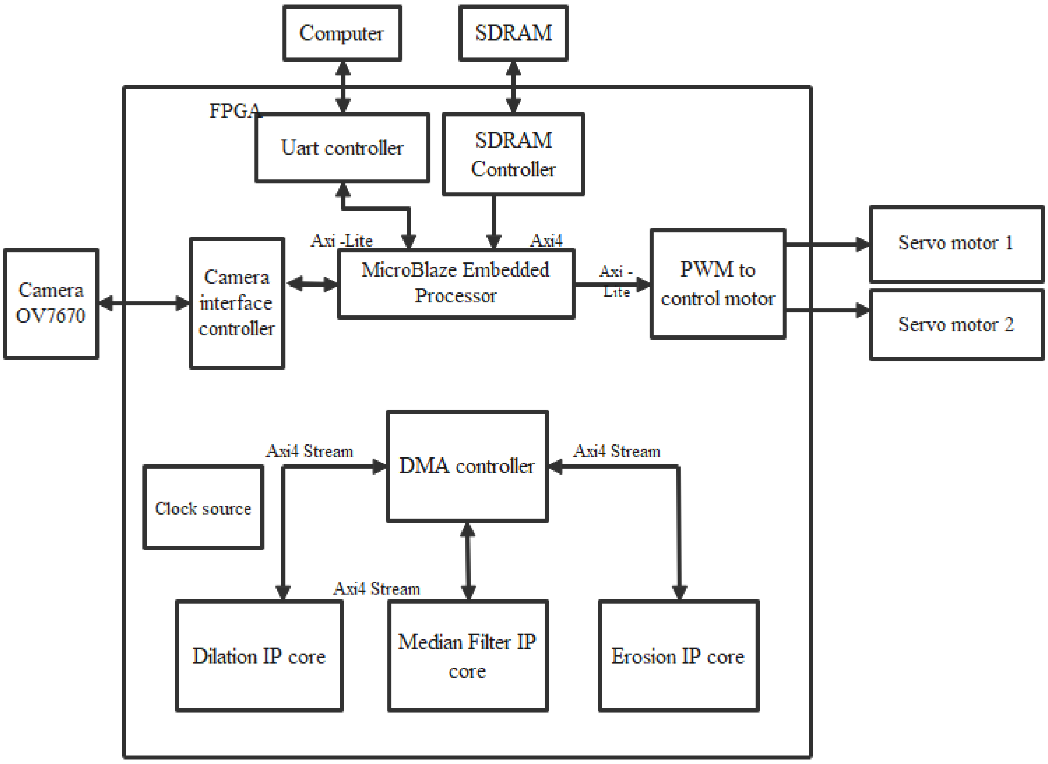

Figure 7. The embedded system structure diagram on spartan-6 FPGA SP605 evaluation KIT from Xilinx

3.2. Structure of hardware cores

3.2.1. Filter median and color separation of objects

a. Median filter element

Figure 8 shows the block diagram of the median and color separation filters including:

- The median filter and color separation comprise two FIFOs for synchronizing data between two clock domains; one data domain is obtained from an AXI stream with a frequency of 100 MHz and the other from the internal clock domain of the median filter, which operates at 30 MHz.
- The mechanism pipeline is responsible for filtering the median, and a stiff core does this.
- A control_bi block to manage synchronizing the writing of data from the median filter output to the FIFO OUT.
- The work of binaryizing each R, G, and B channel with the necessary thresholds is essentially what makes up a color separation block.
- Consider a 3x3 mask with the pixels sorted in ascending order for each row, then in descending order for each row, and lastly sorted diagonally. The median of the diagonal equals the 3x3 mask's median [27].

Figure 9 shows the hardware that was constructed using that approach. To compare two 8-bit A and B inputs and output the bigger number *H* and the smaller number *L*, use the basic node block. Based on the fundamental nodes in Figure 10, the aforementioned procedure is used to sort a block and determine the median of a 3x 3 mask.

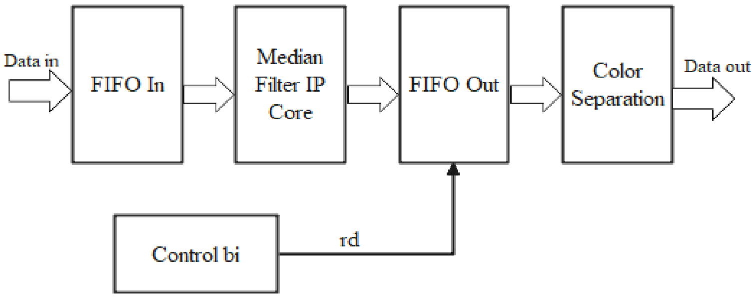

Figure 8. Block diagram of median filter and color separation

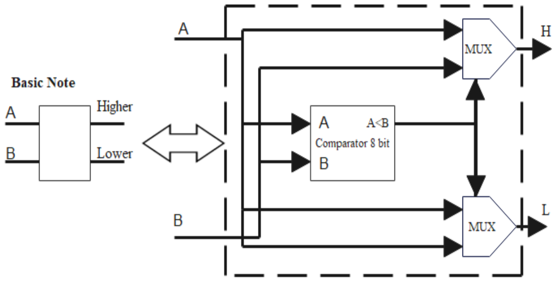

Figure 9. Hardware of basic node

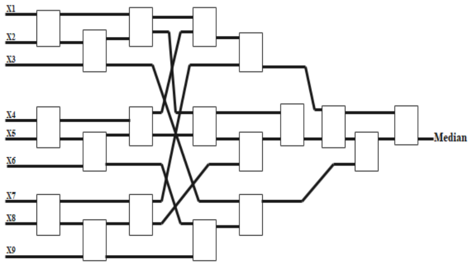

Figure 10. The block hardware calculates the 3×3 mask median from the basic node

b. Build data flow

In Figure 11, each cycle 3×4 non-overlapping block of the image will be input for median filtering, here will build a filter block by pipeline, 3×4 data blocks (12 pixels) are fed consecutively after every clock from FIFO in, after 3 cycles there will be filtered data. Each filter cycle will get 4 pixels. Figure 12 is morphological filtering IP pipeline architecture describing IP stateful filtering architecture.

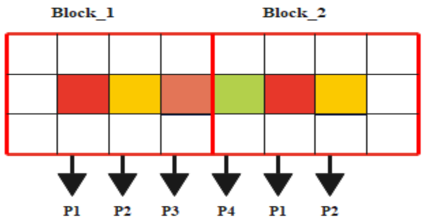

Figure 11. Median filtering IP pipeline architecture

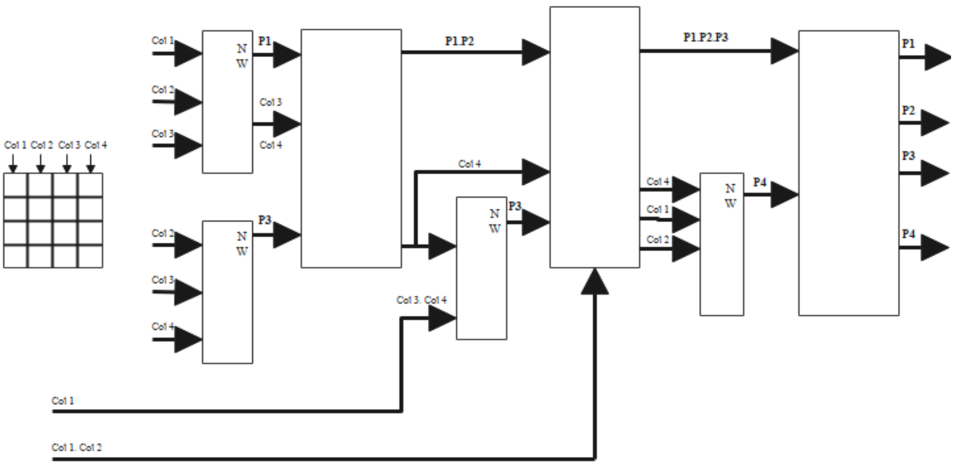

Figure 12. Morphological filtering IP pipeline architecture

3.2.2. Morphological filtering

We construct 2 IP cores, including dilation and erosion, based on the homomorphic filtering theory. The hardware architecture and I/O data flow for these two IP cores are identical. In Figure 13, the entire picture is scanned using a 9x9 structural element. The structural element will travel one pixel at a time throughout the whole image, from left to right and top to bottom. A new picture pixel will be generated for each 9x9 block of the image that corresponds to the identified 9x9 structural element. As a result, it is clear that, with an image size of 320x240 pixels, translating the structure pixel-by-pixel from top to bottom will take a very long time. Because of this, the article uses a pipeline computation with the input data stream in each 9x9 picture block in Figure 14 to shorten the execution time. After the first 9 cycles since the first data is pushed in, we will get the exact 9 bits of data from Out1 to Out9 show at Figure 15. Then every cycle we will have 9 bits of data after homomorphic filtering. Therefore, base on the pipeline architecture, the calculation speed of the system is very high.

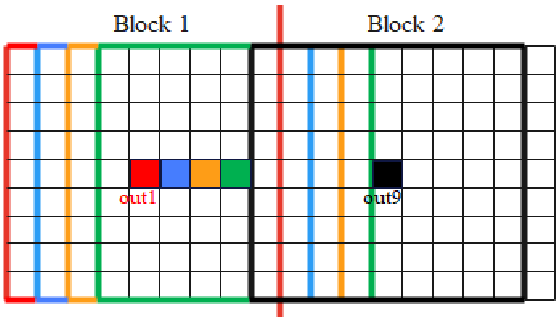

Figure 13. Morphometric filter calculation

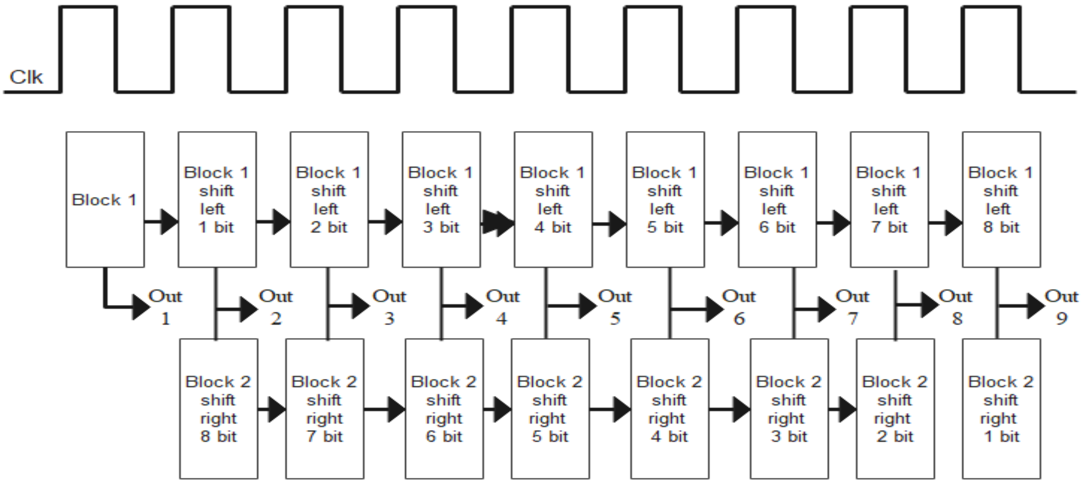

Figure 14. A process of object tracking system

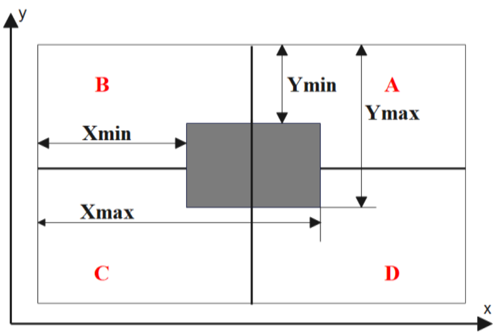

Figure 15. Object contour values

3.2.3. Engine control

The robot follows the object based on the coordinates and contour of the object as shown in Figure 15. Controlling the robot forward or backward is based on the parameters, Y_{min} and Y_{max}; i) $\beta < Y_{min} - Y_{max} \leq \alpha$: the robot is stationary, ii) $\beta \leq Y_{max} - Y_{min}$: the robot is moving backwards, and iii) $Y_{max} - Y_{min}$: the robot is moving forwards. Controlling the robot to turn left or right is based on the coordinates of the center of gravity of the object x_{ob}, y_{ob}; i) $(x_{ob}, y_{ob}) \in \{B \cup C\}$: the robot turns left, and ii) $(x_{ob}, y_{ob}) \in \{A \cup D\}$: the robot turns right.

Figure 16 depicts the IP core for motor control. The motor control core consists of: FSM module and PWM module. The FSM module receives signals (cometo, backward, right, left) from the DSP module to decide to control the moving robot to follow the object. The output data of the FSM module (driver_1, driver_2) is sent to the PWM module to control pulses (signal_1, signal_2) for the two robot motors to rotate at the right speed and in the right direction.

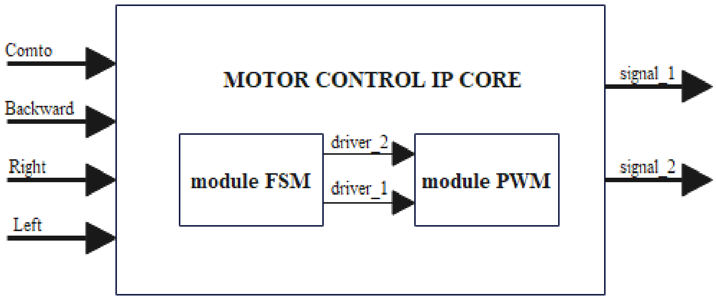

Figure 16. Motor control IP-core

4. RESULTS OF IMPLEMENTATION AND ASSESSMENT

4.1. Hardware synthesis results

The embedded system implemented on the spartan-6 KIT. It's used about 80% of the memory elements including RAM blocks, LUTs and about 20% of other logic of the spartan-6 KIT. This result shows that the design is suitable for resource-constrained systems.

4.2. Execution time results

Table 1 shows the execution time results. Following all the algorithm steps, the execution times are shown corresponding to each block name.

Table 1. Execution time results

No.	Block name	Execution time (s)
1	Collect photo frames	1.0
2	Median filter	1.5
3	Color separation	0.5
4	Morphological filter	2.0

4.3. Performance evaluation results

To evaluate the system's tracking performance, the team evaluated the system based on the good light environment and the robot's ability to follow in the right direction. Table 2 shows the performance evaluation of each movement direction.

Table 2. Performance evaluation

Light conditions	Movement direction	Result	Efficiency
Good	Forward (25 times)	23/25	88%
	Back (25 times)	22/25	
	Left (25 times)	22/25	
	Right (25 times)	21/25	

5. CONCLUSION AND FUTURE NETWORK

The real-time object tracking robot control system on Spartan®-6 FPGA SP605 evaluation KIT is proposed in this study as being low-cost and low-power. The technology is being tested on the KIT and is capable of precisely directing the robot to pursue red objects under various lighting conditions. The system is constructed using high-speed pipelined hardware cores. In order to enable the system's high speed operation, DMA is also used to transfer data in bursts back and forth between external DDR3 memory and hardware IP cores. The system's response time to the movement of the item is adequate. Our upcoming study will focus on implementing robot collaboration using FPGA. Fast real-time robot embedded systems will be crucial in assuring the effectiveness of robot collaborative work as robotics adoption in manufacturing rises via work alongside humans.

ACKNOWLEDGEMENTS
The authors would like to thank Thai Nguyen University of Technology (TNUT), Vietnam for the support.

REFERENCES
[1] A. T. Anaraki and U. U. Sheikh, "A symbolic contour representation of shape for recognition and retrieval," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, 2021.
[2] M. T. Nguyen and H. R. Boveiri, "Energy-efficient sensing in robotic networks," *Measurement*, vol. 158, pp. 1–10, Jul. 2020, doi: 10.1016/j.measurement.2020.107708.
[3] S. V. Kothiya and K. B. Mistree, "Notice of removal: a review on real time object tracking in video sequences," in *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, Jan. 2015, pp. 1–4, doi: 10.1109/EESCO.2015.7253705.
[4] M. T. Nguyen, L. H. Truong, and T. T. H. Le, "Video surveillance processing algorithms utilizing artificial intelligent (AI) for unmanned autonomous vehicles (UAVs)," *MethodsX*, vol. 8, pp. 1–11, 2021, doi: 10.1016/j.mex.2021.101472.
[5] S. J. Gambhire, D. R. Kishore, P. S. Londhe, and S. N. Pawar, "Robust trajectory tracking with non-singular terminal sliding mode control of a mobile manipulator," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 134–142, Feb. 2022, doi: 10.11591/eei.v11i1.3457.
[6] V. C. Thanh, N. N. A. Quan, T. L. T. Dong, T. T. Hoang, and M. T. Nguyen, "Fusion of inertial and magnetic sensors for autonomous vehicle navigation and freight in distinctive environment," in *Advances in Engineering Research and Application*, Cham: Springer, 2022, pp. 431–439, doi: 10.1007/978-3-030-92574-1_45.
[7] G. Sreenu and M. A. S. Durai, "Intelligent video surveillance: a review through deep learning techniques for crowd analysis," *Journal of Big Data*, vol. 6, no. 1, pp. 1–27, Dec. 2019, doi: 10.1186/s40537-019-0212-5.
[8] R. Morris and S. Mirzaei, "Efficient FPGA implementation of parameterized real time color based object tracking," in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Oct. 2021, pp. 0102–0105, doi, 10.1109/IEMCON53756.2021.9623221.
[9] Z. Wan *et al.*, "A survey of FPGA-based robotic computing," *IEEE Circuits and Systems Magazine*, vol. 21, no. 2, pp. 48–74, 2021, doi: 10.1109/MCAS.2021.3071609.
[10] M. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sensors*, vol. 14, no. 2, pp. 2822–2859, Feb. 2014, doi: 10.3390/s140202822.
[11] H. T. Do *et al.*, "Energy-efficient unmanned aerial vehicle (UAV) surveillance utilizing artificial intelligence (AI)," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–11, Oct. 2021, doi: 10.1155/2021/8615367.
[12] G. Chandan, A. Jain, H. Jain, and Mohana, "Real time object detection and tracking using deep learning and OpenCV," in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, Jul. 2018, pp. 1305–1308, doi: 10.1109/ICIRCA.2018.8597266.
[13] H. T. Do, H. T. Hua, H. T. T. Nguyen, M. T. Nguyen, and H. T. Tran, "Cooperative tracking framework for multiple unmanned aerial vehicles (UAVs)," in *Advances in Engineering Research and Application*, Cham: Springer, 2022, pp. 276–285, doi: 10.1007/978-3-030-92574-1_29.
[14] D. Zhang and T. Yang, "Visual object tracking algorithm based on biological visual information features and few-shot learning," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–8, Mar. 2022, doi: 10.1155/2022/3422859.
[15] A. Brdjanin, N. Dardagan, D. Dzigal, and A. Akagic, "Single object trackers in OpenCV: a benchmark," in *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, Aug. 2020, pp. 1–6, doi: 10.1109/INISTA49547.2020.9194647.
[16] Mohana and H. R. Aradhya, "Object detection and tracking using deep learning and artificial intelligence for video surveillance applications," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, pp. 517–530, 2019, doi: 10.14569/IJACSA.2019.0101269.
[17] S. Satpute, H. Shende, V. Shukla, and B. Patil, "Real time object detection using deep-learning and OpenCV," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 4, pp. 3243–3246, 2020, [Online]. Available: www.irjet.net
[18] OmniVision Technologies, "OmniVision serial camera control bus (SCCB) functional specification," *Application Note*, vol. 2, pp. 1–24, 2007.
[19] M. T. N. J. Aji, S. P. Ajith, J. Amal, S. Muhammed, "Ball tracking robot using image processing and range detection," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 3, pp. 3571–3576, 2014.
[20] Y. Kimori, "Mathematical morphology-based approach to the enhancement of morphological features in medical images," *Journal of Clinical Bioinformatics*, vol. 1, no. 1, pp. 1–10, 2011, doi: 10.1186/2043-9113-1-33.
[21] P. Maragos, "Chapter 13 - morphological filtering," in *The Essential Guide to Image Processing*, Cambridge, UK: Academic Press, 2009, pp. 293–321, doi: 10.1016/B978-0-12-374457-9.00013-5.
[22] R. C. Gonzalez and R. E. Woods, *Digital image processing, 4th edition*. Chennai, India: Pearson Education, 2018.
[23] Xilinx, *SP605 hardware user guide*, vol. 526. California, USA: AMD Xilinx, 2012.
[24] Xilinx, *LogiCORE IP AXI DMA (v4.00.a)*. California, USA: AMD Xilinx, 2011.
[25] Xilinx, *LogiCORE IP clock generator (v4.01a)*. California, USA: AMD Xilinx, 2010.
[26] Xilinx, *AXI reference guide (UG761) (v14.3)*. California, USA: AMD Xilinx, 2012.
[27] P. J. C. -Caicedo, A. I. B. -Gutiérrez, E. I. G. -Hernandez, P. B. -Mendoza, J. A. P. -Medina, and H. R. -González, "An FPGA-based architecture for a latitude and longitude correction in autonomous navigation tasks," *Measurement*, vol. 182, pp. 1–11, Sep. 2021, doi: 10.1016/j.measurement.2021.109757.

BIOGRAPHIES OF AUTHORS

Hoang T. Tran received an engineering degree in system measurement and control from Danang University of Science and Technology in 1998, received a master's degree in network and electrical systems in 2009 from the University of Danang; Defending the PhD thesis majoring in Electronics and Communication Technology in 2016 at the University of Technology, Vietnam National University, Hanoi. A guide to synthesizing data sensors, wireless network transformation and the fields of control automation. He is currently the Director of The Center of Electrical Engineering, Duy Tan University, Da Nang, Vietnam, and the Lecture of the Faculty of Electronic Engineering, Duy Tan University, Da Nang, Vietnam. He can be contacted at email: tranthuanhoang@duytan.edu.vn.

Dong LT. Tran received the B.S. degree from the Duy Tan University, Da Nang, Vietnam, in 2009, the M.S. degree from the Duy Tan University, Da Nang, Vietnam, in 2012, and became a PhD student at Thai Nguyen University from 2022 to the present. He is currently the Vice Director of The Center of Electrical Engineering, Duy Tan University, Da Nang, Vietnam, and the Lecture of the Faculty of Electronic Engineering, Duy Tan University, Da Nang, Vietnam. He can be contacted at email: tranthangdong@duytan.edu.vn.

Quang N. Pham had a mechatronic engineering degree in 2016 from Danang University of Technology, Da Nang, Vietnam and this is also the place where he obtained his master's degree in Mechanical Engineering in 2018. At present, Mr. Quang is a researcher in mechanical design and automation products, especially in implementing robot in many practical applications. He can be contacted at email: phamngocquang3@dtu.edu.vn.

Thanh C. Vo received a degree in Mechatronic Engineering from Danang University of Technology, Vietnam in 2013. Engineer Thanh is currently an expert at Center for Electrical and Electronic Engineering, Duy Tan University, Vietnam (CEE). He has interest and expertise in research topics in the field of embedded programming, automatic control. He can be contacted at email: vochithanh.cdt@gmail.com.

Quan NA. Nguyen received a degree in automatic electrical engineering from Duy Tan University in 2019, Da Nang, Vietnam. Eng. Quan is currently an expert at Center for Electrical and Electronic Engineering, Duy Tan University, Vietnam (CEE). He has interest and expertise in research topics in the field of automatic control. He can be contacted at email: nguyenquan97@hotmail.com.

Thang K. Nguyen received an engineering degree in electronic engineering from Duy Tan University, Da Nang, Vietnam in 2013. Eng. Thang is currently an expert at the Center for Electrical and Electronics Engineering, Duy Tan University, Vietnam (CEE). He has interest and expertise in research topics in the field of power electronics. He can be contacted at email: kimthang91@gmail.com.

Duyen M. Ha received a degree in embedded systems engineering from Duy Tan University in 2019, Da Nang, Vietnam. Eng. Duyen is currently an expert at Center for Electrical and Electronics Engineering, Duy Tan University, Vietnam (CEE). She has interest and expertise in research topics in the field of image processing. She can be contacted at email: hamyduyen@dtu.edu.vn.

Minh T. Nguyen Dr. Minh Nguyen is currently the director of international training and cooperation center at Thai Nguyen University of Technology, Vietnam, and also the director of advanced wireless communication networks (AWCN) lab. He has interest and expertise in a variety of research topics in the communications, networking, and signal processing areas, especially compressive sensing, and wireless/mobile sensor networks. He serves as technical reviewers for several prestigious journals and international conferences. He also serves as an editor for wireless communication and mobile computing journal and an editor in chief for ICSES transactions on computer networks and communications. He can be contacted at email: nguyentuanminh@tnut.edu.vn.