Online Object Trajectory Classification using FPGA-SoC devices

Pranjali Shinde[1], Pedro Machado[2], Filipe N.Santos[1], and T.M. McGinnity[2]

[1] INESC TEC Campus da Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, Porto, Portugal
{*pranjali.shinde*, *filipe.n.santos*}@inesctec.pt
[2] Computational Neuroscience and Cognitive Robotics Laboratory
Nottingham Trent University, Nottingham, UK
{*pedro.baptistamachado*, *martin.mcginnity*}@ntu.ac.uk

Abstract. Real time classification of objects using computer vision techniques are becoming relevant with emergence of advanced perceptions systems required by, surveillance systems, industry 4.0 robotics and agricultural robots. Conventional video surveillance basically detects and tracks moving object whereas there is no indication of whether the object is approaching or receding the camera (looming). Looming detection and classification of object movements aids in knowing the position of the object and plays a crucial role in military, vehicle traffic management, robotics, etc. To accomplish real-time object trajectory classification, a contour tracking algorithm is necessary. In this paper, an application is made to perform looming detection and to detect imminent collision on a system-on-chip field-programmable gate array (SoC- FPGA) hardware. The work presented in this paper was designed for running in Robotic platforms, Unmanned Aerial Vehicles, Advanced Driver Assistance System, etc. Due to several advantages of SoC-FPGA the proposed work is performed on the hardware. The proposed work focusses on capturing images, processing, classifying the movements of the object and issues an imminent collision warning on-the-fly. This paper details the proposed software algorithm used for the classification of the movement of the object, simulation of the results and future work.

Keywords: SoC-FPGA, computer vision, colour detection, contour tracking, trajectory detection, object tracking

1 Introduction

Image processing using computer vision techniques has gained wide importance in large scale systems and embedded platforms [1]. The conventional method of object tracking using standard computers consumes more power, requires more space for installing components than using highly integrated devices; such as the Intel (formerly Altera) Cyclone V SoC-FPGA (System-on-Chip-Field-Programmable Gate Array). This paper presents contour tracking algorithm to detect and track the object in real time and detects imminent collision. The proposed work focusses on classifying the movement of the object and a novel idea of performing looming detection on the hard processor of Intel Cyclone V SoC-FPGA which gives an idea of the object approaching or receding the camera is executed. Imminent collision is also detected which aids in knowing if the object has occupied 80% of the frame. During imminent collision the object movement cannot be distinguished. Such feature is desirable in several applications like security purposes, controlling robotic operations, etc. Thus, this system can be used to alert and take the necessary actions when the movement of the object cannot be perceived. FPGAs are fully-reconfigurable devices employed with memory, programmable logics, Input/output (I/O), routing capabilities and a matrix of resources connected via the programmable interconnects [2]. Object detection and recognition has been appealing as it involves numerous applications. Classification of the objects is performed based on size, shape, colour, etc, which aids in tracking and detecting the movement of the object which is achieved by employing computer vision technique. This technique collects the pictorial information from the real-world, processes and analyses the information and later converts it to binary values. The hard processor (dual core ARM Cortex A9), an integrated circuit on the FPGA fabric is used to further process the output and to detect the movements appropriately. The purpose of selecting FPGA over other computing devices is, it provides a robust platform for running the operating system accelerated by the custom hardware. The programmable features of the FPGA get it into the limelight as it supports parallel processing, reconfiguration of the logic elements and provides cost effectiveness for each logic element. The traits of the FPGA which includes low power consumption, faster processing speed, portability, increased memory, Intellectual property, digital signal processing (DSP) block make it tremendously a powerful computational appliance [3]. Although FPGAs deliver such flexibility, programming FPGAs is a complex task and, in this work, only the hard-processor was used. The optimisation of the tracking algorithm using the FPGA will be discussed in the Future Work section. This paper is structured as follows. Section 2 presents the related work, section 3 discusses the design methodology used, along with the hardware and software configuration and explains the detection and tracking algorithm as well. Section 4 portrays the simulation of the results and Section 5 discusses the results and the future work.

2 Related Work

Real time classification in computer vision has been widely studied over the last decades. The real time moving target detection on UAV using FPGA is presented in [4] which allows to perform the computer vision algorithms on-the-fly platform. The vehicle detection algorithm used in [[4] is Motion Estimation and object segmentation process and supports complex embedded video processing. The work of [5] uses camera to track the objects based on colours and the movement of the cursor is controlled by the tracked objects. The work of [5] is sceptical in the use of computer vision technique as it does not specify the method used for distinguishing the objects. The frame rate of the camera is not specified as well. Although, the work produced by [5] showcases the segmentation process but does not provide any evidence if a still image or video is captured and lacks the details about the processing. Besides, the work of [5] does not specify if the detection is performed in real-time and does not indicate the type of objects detected for controlling the cursor movement and lacks details about the operating system. Furthermore, the angle is not mentioned at which the object could track the cursor movement. A similar approach of multiple object detection in real time is described in [1] using computer vision on embedded system. This work employs the cascade classifiers used for object detection based on Haar feature and provides a prospect to cascade other complex classifiers thereby augmenting the speed of object detection. The application was developed on desktop and on the embedded system. It was observed from the results of [1] that the execution time on windows-based PC is lower than the embedded platform.

In [6] is described two cases which are considered while detecting objects in real time. They are: Detecting objects in complex background and Detecting objects with colour against a background. The second case is more challenging as it involves detecting an object of a colour same as the background. Differentiating the work of [6], [7] uses Kalman filter for motion estimation and tracking of the objects. The work of [5], [7] and [1] faced problems in implementation of the system in real time. Because of the constraints related to algorithms which dealt with white or plain background. However, the research produced by [6] has developed and implemented a system which detects and tracks the object in an unknown environment. The real time object classification is presented in [8] which aims in detecting the vehicle, classifying and tracking it in real time. Also, they have described computer vision algorithms are used to classify the objects based on the size, shape and movement of the vehicles. The work of [8] does not give a clear view of the test performed like the road and weather conditions, period of the day, etc.

Occlusion and Illumination are two big issues in object tracking. The work of [9] stresses on tracking of objects with partial or full occlusion in real time. [9] has proposed a colour-based algorithm which is used to track the objects through colours and contours. It is a challenging task to obtain real time performance on embedded vision as there is no perfect hardware which meets all the conditions of the levels of processing. Computer vision applications finds three different levels of processing: low, mid and high level. The iterative operations at pixel level is termed as the low level which includes filtering operations such as noise reduction, edge detection, etc. The Simple Instruction Multiple Data (SIMD) architectures better serve the low-level processing. The region of interest considered by the mid-level is the classification criteria such as segmentation, classification of objects, feature extraction, etc. Parallelism with mid-level can be achieved to a certain extent. The sequential processing is obtained at the high level which is responsible for decision making [10]. In addition to [10], [11] presents the embedded vision challenges regarding the hardware and software issues such as power consumption, timing etc. From, the literature, it can be observed that object recognition, detection and tracking has been demanding in real time. Various object detection and tracking algorithms are illustrated in the works though not including looming detection. Different algorithms are used to detect the objects like the Binary histogram which is based on split or merge [12], Haar classifiers, multi-object detection using grid based Histograms of Oriented Gradient (HoG) is presented in [13],colour detection, segmentation, etc., which aids in detecting the objects. Another approach of small object detection using infrared camera is shown in [14] and makes use of FPGA and Digital Signal Processing (DSP) for the computations. The infrared camera an capture 22 frames per second leading to slower processing speed in comparison to the camera used in the proposed work. Like, the work of [9], the proposed research incorporates the colour and contour tracking algorithm to classify the object trajectory and detects imminent collision in real time considering the illumination and background changes.

In this work, it is presented an object tracking system for tracking object movements and emit a warning when the object gets too close to the camera. Such algorithm can be used as part of Advanced Driver Assisted Systems (ADAS) for tracking the movement of objects that may, or may not interfere, with autonomous system. This algorithm can also be used in defence application where the enemy could be identified, and the required action could be taken.

3 Methodology

The work described in this paper was carried out on an Intel Cyclone V SoC-FPGA. The goal of this work is to implement an embedded object tracking device which can be used for general-proposed applications where object tracking is required. Such a system requires a camera connected to a SoC-FPGA with a standard Operating System (OS) running on the Hard Processor(HP) from a SD-Card. A SoC-FPGA was chosen because of its flexibility feature. A standard OS runs on the SoC device, for providing all the services and applications (e.g. network services, OpenCV library, security, etc) which is connected via a high-speed bus to the FPGA. This provides the flexibility for performing hardware acceleration (which will be discussed in the Conclusions and Future Work section). In this work, the Ubuntu Linux 16.04 LTS was chosen over other possible embedded Linux distributions. Robotic Operating System (ROS) was installed because it was desirable to (i) increase the compatibility of the system with most Robotic Platforms and (ii) ROS has the OpenCV library already integrated. The SoC-FPGA board is interfaced with camera, keyboard, mouse, monitor.

3.1 Overall System architecture

The hardware and software configuration for the proposed system involves setting the hardware platform and developing the object tracking algorithm on the SoC-FPGA. The classification of the movement of the object and looming detection is observed on the monitor.

3.1.1 Hardware Layer The Altera Cyclone V SoC FPGA is used as the embedded platform. The feature of reconfigurability, low power processor system leverages the user as it supports flexibility in the design. The SoC kit is an integration of hard processor Cortex A9, comprising of processor, memory, peripherals tangled impeccably with FPGA through the interconnects. Also, it possesses high speed DDR3 memory, video, audio, networking capabilities, etc [3]

The Logitech camera provides a maximum resolution of 640X480 and the image sensor used is of type Complementary Metal Oxide Semiconductor(CMOS) and as well supports USB interface. This camera is made to operate on Ubuntu by running the guvcview program. A 64GB micro secure digital (SD) card is used to set up the operating system, OpenCV on the SoC-FPGA to perform the classification. Due to its huge storage capacity of 64GB it provides flexibility in setting the additional packages on the Hard Processor which include the editor, libraries, etc. The main feature of the SD card is that it reads the data at a speed of 80 MBps.

3.1.2 Software Layer The Ubuntu 16.04 LTS Linux operating system and OpenCV 3.2.0 is installed on the hard processor as well on ThinkPad laptop. The laptop has Intel i5 processor 2nd generation and operates at a processing speed of 2.5GHz. It supports a memory of 8GB and hard drive of 1TB with windows 10 operating system installed whereas Linux operating system is installed for the implementation of the work. The environment set up on the laptop is the same as on the FPGA.

The programming platform used for the proposed work is python, running on Ubuntu and uses the OpenCV library. Whereas, python does not require compilation instead needs an interpreter which generates the python executable file. The elf file is used to run on the hard processor through RS232 port [9]. The flowchart of the contour tracking algorithm used in the proposed system is as shown in Figure 1.

The conversion of the grabbed frame (RGB) to HSV plays an important role in detecting the object. RGB colour space is used in all the systems like computer, TV, etc as it is easy to implement as described in [15]. RGB is not suitable for processing the colour images. When considered the perceptual nonlinearity, machine dependency and the integration of luminance and chrominance the data obtained makes RGB incompatible for colour image processing. HSV is mainly used because it is capable of handling images influenced by illumination such as lighting, shade, contrast light [16].

The colour detection algorithm aids in recognizing and identifying the object. The video is captured, and the frame rate is calculated determining the processing speed of the frames. The extracted frame is scaled down in size because the frames with lower size can be processed at a faster rate. The raw input or the real-world frame captured is influenced by the presence of the noise and noise removal is important to process the frame. After removal of the noise the

[3] Retrieved from https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=836&PartNo=2, Last accessed: 2018-06-06

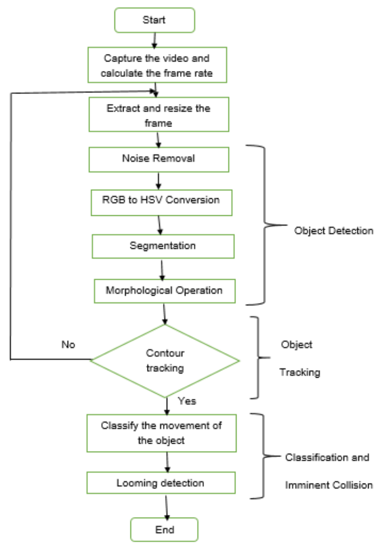

Fig. 1: Flowchart of the proposed system

frame is converted from RGB to HSV. The segmentation technique used in the work is threshold segmentation. After performing segmentation, morphological operations are performed to further process the frame. This operation filters out the noise and is useful to get the segmented frame. The HSV based contour will be applied to the object based on the area of the frame. If at least one contour is found, then it is used to compute the minimum enclosing circle and centroid.

The classification is basically performed by storing the x-y coordinates of the previous frame and finding the difference with the current frame. The difference obtained is used to classify the movement of the object horizontally, vertically and diagonally. Next, the looming detection is performed by calculating the binary difference between the previous and the current frame which helps in determining the imminent collision, proceeding or receding of the object. Imminent collision is basically detecting the proximity of the object to the camera and this technique can be employed in automotive cars which would aid in avoiding accidents. Imminent collision is detected when number of 1s in the binary image occupy more than 80% of the frame size. The movement of the object cannot be classified when imminent collision is detected as the object has occupied the maximum frame size. Single object is being tracked and the distance is computed to perform the looming detection.

The algorithm used for tracking the object in the work is the contour tracking algorithm. Contours are basically the line joining the boundary of the object. When the object in the frame is detected a contour is formed. After finding the contours then the centroid of the object is found which is obtained from the image moments. The centroid coordinates search are ruled by Equation 1.

$$c_x = int\Big(\frac{M\,[m_{10}]}{M\,[m_{00}]}\Big), c_y = int\Big(\frac{M\,[m_{01}]}{M\,[m_{00}]}\Big) \tag{1}$$

M is the image moments inside the contour, M_{00} is the average of image pixel intensity, M_{01} is the pixel value of y coordinate and M_{10} is the pixel value of x coordinate.

The centroid points are used to put the bounding box(circular boundary) for the object. The dimensions of the box are calculated, and the object enclosed in the box aids in tracking the object. The difference between the x and y coordinate is used in classifying the movement of the object. If the object moves through the x coordinates, then the horizontal movement of the object is tracked. Likewise, the vertical movement of the object is tracked when the object moves over the y coordinates. The diagonal movement of the object is tracked based on the x and y coordinates. The looming detection is performed by comparing the binary matrix of the image. More number of 1s in the image determine the object is approaching. Less number of 1s in the image determine the object is receding and when the 1s occupy more than the maximum size of the image, imminent collision is determined. The algorithm for looming detection is as described in Algorithm 1. This differs from the existing algorithms as it detects the imminent collision which is not performed in any of the related works. This algorithm was designed for providing a good balance between low latency in an Embedded System and accuracy.

Algorithm 1 Classification of the object motion.

if $average > P_{MIN} \times MaxValue$ & $average < P_{MAX} \times MaxValue$ & $binSum[n] < BinSum[n-1] + Th$ **then**
    $out = Receding$
else
    if $average > P_{MIN} \times MaxValue$ & $average < P_{MAX} \times MaxValue$ & $BinSum[n] > BinSum[n-1] + Th$ **then**
        $out = Approaching$
    else
        if $average < P_{MIN} \times MaxValue$ **then**
            $out = FarAway$
        else
            if $average \geq P_{MAX} \times MaxValue$ **then**
                $out = ImminentCollision$
            else
                $out = Inconclusive$
            end if
        end if
    end if
end if

4 Results

The application was initially tested on a laptop and then ported to the SoC-FPGA. Classification of the movement of the object was performed by moving the ball in various directions. The object performs the trajectory in pink based on the movement of the ball. The contour tracking algorithm was producing an accuracy above 95% when the object had a white background and with no other objects moving in the background and the accuracy was dropped in more complex background with natural images. Also, the object tracking was not so robust when the background had colours similar to the object. The horizontal, vertical, diagonal and looming tests are reported in sections 4.1, 4.2, 4.3 and 4.4 respectively.

4.1 Horizontal movement test

If the change in the x co-ordinate was positive provided there was no change in the y co-ordinate it was determined that the object was moving towards the right (east) direction else in the left (west) direction. The classification of the horizontal movement of the object is shown in Figure 2.

Fig. 2: Object moving towards east (left image) and west (right image) directions

4.2 Vertical movement test

The movement of the object in the y co-ordinate would determine the vertical movement of the object. If the change in the y co-ordinate was positive provided there is no change in the x co-ordinate then it was determined the object is moving upwards in North direction else in South direction. The classification of the vertical movement of the object is shown in Figure 3.

4.3 Diagonal movement test

The movement of the object in the x, y coordinate would determine the diagonal movement. When the change in the x value and y value is greater than the pixel difference it is determined that the ball is moving diagonally. If the change in

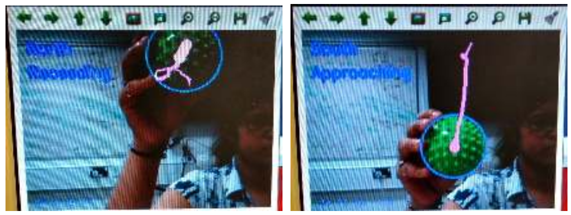

Fig. 3: Object moving towards east (left image) and west (right image) directions

the x co-ordinate is positive/negative and change in the y co-ordinate is positive/negative, then the ball is termed to move either in north-east, north-west, south-east or south-west direction which is portrayed in Figure 4.

Fig. 4: Object moving towards east (left image) and west (right image) directions

4.4 Looming movement test

This test was performed by comparing the previous and current segmented matrix to determine if the object was approaching or receding. The imminent collision was detected when the computed average of the matrix was greater than the maximum value of frame size. Imminent collision was determined when the object was at a distance approximately 15cm from the camera. The approaching and receding of the object was determined when the object was between a distance approximately of 15cm and 120cm from the camera. After, a distance of 120cm from the camera, the object to be detected is termed as far away. The approaching, receding and the imminent collision can be seen from Figure 5.

4.5 Performance results

The proposed system was implemented on the laptop and SoC-FPGA which resulted in the different execution speed which is determined from Table 1 followed by the graphical representation.

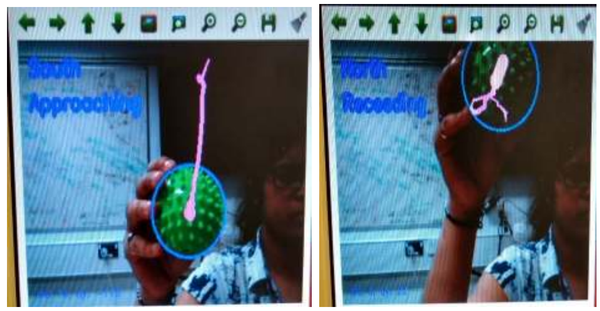

Fig. 5: Object moving towards east (left image) and west (right image) directions

Table 1: Execution time per processor type

Platform	CPU clock	Camera frame rate	execution time
Dual-core Intel i5 processor	2.5GHz	30fps	0.4×10^{-6} ms
Dual-core ARM Cortex A9	925MHz	30fps	1.8×10^{-6} ms

It can be observed from Table 1 , the processing speed of the proposed system on the laptop is six times more than the execution of the proposed system on the hard processor of FPGA. The laptop here is used as ground truth and that the goal is to improve the SoC-FPGA for achieving the same performance with low power. When compared to the energy consumption of the laptop and SoC-FPGA, the energy consumed by the SoC-FPGA is less as it operates on 12 volts supply. In addition, laptops are not suitable for robotics and real time object detection for many reasons: size, performance, robustness in comparison to SoC-FPGA. Also, it is not possible to use a laptop on a Drone or a Robotic Platform used in resilient environments.

5 Discussion and Future Work

In this paper, we have presented a system which detects the object trajectory and detects imminent collision in real time with the specified distance of the object from the camera. It can be observed from the results; the movement of the ball is displayed on the screen. When the object is at a distance of 15cm from the camera it detects imminent collision. Similarly, when the ball moves away from the camera and is at a distance of 120cm, the ball is identified, displaying far away on the monitor. The implementation of the proposed work makes it more representable in real time because of the execution of looming detection. The results show that the performance must be improved. However, the results demonstrate that the classification results were very similar to the results obtained with a powerful computer. The proposed algorithm can be improved by lowering the resolution, porting some of the filters to the SoC-FPGA and increasing the speed and performance of the system. The implementation of looming detection in the work presents new prospective which is not considered in any of the previous works. This makes the proposed system more reliable for security purposes- military, vehicle traffic system, UAVs, etc. The object tracking in the proposed system can be made more specific in terms of minute movement of the object. This can be achieved by knowing the angular movement of the object which would make the system more robust. In future, the system can be made to detect more than one object of different or same colours and shapes. The performance of the algorithm on the laptop is better than the FPGA. The variation in the processing time was not taken into consideration as the main motive of the work was to classify the movement of the object and to perform looming detection. In addition, the functionality of the present contour tracking algorithm can be increased by considering the colour of the object same as the background. The proposed system can be made to operate at a faster speed than the processor of laptop by forwarding the captured frames to the SoC-FPGA. This was not implemented due to shortage of time. By ignoring the difference of the execution time on both the platforms-laptop and DE1-SoC, the SoC makes the system portable and energy efficient which is recommended in real time robotics applications, UAVs, etc. The Terasic 8-megapixel daughter card can be employed in future to capture the video and process the video frame wise. The video from the camera can be forwarded to the FPGA through the high speed General Purpose Input Output (GPIO) interconnects. The camera consists of a Mobile Industry Processor Interface (MIPI) decoder and a module in addition to (High Definition Multimedia Interface (HDMI-TX). The captured video from the camera MIPI module is forwarded for further processing in video signal MIPI package. The MIPI decoder converts the packet from the module into 10-bit Bayer pattern. The voice coil motor (VCM) is employed in the camera to adjust the focus through the I2C interface. As the FPGA does not possess display capability, the HDMI-TX allows the video to be displayed on the monitor [4]. This interfacing would make the system to operate at a significantly higher speed and would provide drastic increase in the performance of the system.

References

1. S. Guennouni, A. Ahaitouf, and A. Mansouri. Multiple object detection using opencv on an embedded platform. In *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, pages 374–377, Oct 2014.
2. Intel fpga devices. https://www.intel.co.uk/content/www/uk/en/fpga/devices.html. Accessed: 2018-06-06.
3. Pedro Machado, John Wade, and T M McGinnity. Si elegans: Modeling the C. elegans Nematode Nervous System Using High Performance FPGAS. In Rita Ana Londral and Pedro Encarnação, editors, *Advances in Neurotechnology, Electronics*

[4] Retrieved from: http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1051, last accessed: 2018-06-06

and Informatics, chapter Si elegans, pages 31–45. Springer International Publishing, 12 edition, 2016.
4. Jia Wei Tang, Nasir Shaikh-Husin, Usman Ullah Sheikh, and M. N. Marsono. FPGA-Based Real-Time Moving Target Detection System for Unmanned Aerial Vehicle Application. *International Journal of Reconfigurable Computing*, 2016.
5. Dion Firmanda and Dadet Pramadihanto. Computer Vision Based Analysis for Cursor Control Using Object Tracking and Color Detection. In *2014 Seventh International Symposium on Computational Intelligence and Design*, 2014.
6. S. Prasad and S. Sinha. Real-time object detection and tracking in an unknown environment. In *2011 World Congress on Information and Communication Technologies*, pages 1056–1061, Dec 2011.
7. N.J. Uke and P.R. Futane. Efficient method for detecting and tracking moving objects in video. In *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology, ICAECCT 2016*, 2017.
8. R. H. Pea-Gonzlez and M. A. Nuo-Maganda. Computer vision based real-time vehicle tracking and classification system. In *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 679–682, Aug 2014.
9. S. D. Gajbhiye and P. P. Gundewar. A real-time color-based object tracking and occlusion handling using arm cortex-a7. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–6, Dec 2015.
10. Marcos Nieto, Oihana Otaegui, Gorka Vélez, Juan Diego Ortega, and Andoni Cortés. On creating vision-based advanced driver assistance systems. *IET Intelligent Transport Systems*, 2015.
11. Fridtjof Stein. The challenge of putting vision algorithms into a car. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012.
12. K. Appiah, H. Meng, A. Hunter, and P. Dickinson. Binary histogram based split/merge object detection using fpgas. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 45–52, June 2010.
13. A. Chayeb, N. Ouadah, Z. Tobal, M. Lakrouf, and O. Azouaoui. HOG based multi-object detection for urban navigation. In *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, 2014.
14. Zhao Wang, Haitao Song, Han Xiao, Wenhao He, Jiaojiao Gu, and Kui Yuan. A real-time small moving object detection system based on infrared image, 2014.
15. G. Saravanan, G. Yamuna, and S. Nandhini. Real time implementation of rgb to hsv/hsi/hsl and its reverse color space models. In *2016 International Conference on Communication and Signal Processing (ICCSP)*, pages 0462–0466, April 2016.
16. Tingting Xue, Yanjiang Wang, and Yujuan Qi. Multi-feature fusion based GMM for moving object and shadow detection. In *International Conference on Signal Processing Proceedings, ICSP*, 2012.