Design and implementation of shape-based feature extraction engine for vision systems using Zynq SoC

Original Scientific Paper

Navya Mohan
Department of Electronics, Cochin University of Science and Technology,
Kochi, Kerala, India
navyamohan@cusat.ac.in

James Kurian
Department of Electronics, Cochin University of Science and Technology,
Kochi, Kerala, India
james@cusat.ac.in

Abstract *– With the great impact of vision and Artificial Intelligence (AI) technology in the fields of quality control, robotic assembly and robot navigation, the hardware implementation of object detection and classification algorithms on embedded platforms has got ever-increasing attention these days. The real-time performance with optimum resource utilization of the implementation and its reliability as well as the robustness of the underlying algorithm is the overarching challenges in this field. In this work, an approach employing a fast and accurate vision-based shape-detection algorithm has been proposed and its implementation in heterogeneous System on Chip (SoC) is discussed. The proposed system determines centroid distance and its Fourier Transform for the object feature vector extraction and is realized in the Zybo Z7 development board. The ARM processor is responsible for communication with the external systems as well as for writing data to the Block RAM (BRAM), the control signals for efficient execution of the memory operations are designed and implemented using Finite State Machine (FSM) in the Programmable Logic (PL) fabric. Shape feature vector determination has been accelerated using custom modules developed in Verilog, taking full advantage of the possible parallelization and pipeline stages. Meanwhile, industry-standard Advanced Extendable Interface (AXI) buses are adopted for encapsulating standardized IP cores and building high-speed data exchange bridges between units within Zynq-7000. The developed system processes images of size 32 × 64 in real-time and can generate feature descriptors at a clock rate of 62MHz. Moreover, the method yields a shape feature vector that is computationally light, scalable and rotation invariant. The hardware design is validated using MATLAB for comparative studies.*

Keywords*: hardware-software codesign, hardware accelerators, feature extraction, vision systems, embedded systems*

1. INTRODUCTION

Machine Vision is one of the most frontiers and revolutionary technology in computer science as a plethora of industrial activities has been potentially benefitted. It ensures consistent and continuous excellence in automating monotonous chores- visual inspection in manufacturing [1-3], localization and navigation for robotic guidance [4], real-time measuring and sorting in factory floors and production lines [5-7]. Vision systems bring operational benefits by reducing human involvement in a manufacturing process and excels in quantitative analysis of structured scenes because of their speed, accuracy and repeatability.

Traditionally, visual inspection and quality control were accomplished by trained experts though it produces inconsistent results. Vision systems may effectively replace human inspection in such demanding cases, so automation has become inevitable to improve precision and reliability. Machine vision systems reckon on cameras to acquire images so that computer hardware and software can process, analyse and measure the various characteristics for decision making. Typically the initial step in such applications is to localize the object or feature of interest within the 2D images. In the past few years, various research activities have been carried out to propose intelligent systems for real-time analysis of characteristic image features like colour, texture and regions. A machine vision assisted sorting has been achieved with colour image processing [8]. Classification based on texture analysis has been investigated in [9]. The paper also discusses the advantages and disadvantages of texture image descriptors and covers the discrimination performance, computational complexity and resistance to challenges such as noise, rotation etc. Automated visual-based defect detection approaches applicable to various materials such as metals, ceramics and textiles are discussed in [11]. A survey of textural defect detection based on statistical, structural and other approaches are considered. Several efforts have been made in areas of image processing to implement algorithms in hardware. Employing dedicated hardware structures accelerates these vision modules. The work [12] discusses the description of a simple fast shape detection algorithm and its implementation in hardware structures like FPGA. The detection algorithm is based on the concepts of Hu´s moments that are invariant to similarity transformations. [13- 19] describes the realization of standard edge descriptors that are computationally intensive.

As the demand for computationally intensive algorithms for real-time industrial applications is on a steep rise, a trend has been witnessed towards utilizing Field Programmable Gate Arrays (FPGAs) or Graphics Processing Units (GPUs) for implementing the vision algorithms instead of using CPUs, which are inherently sequential processing devices. Compared with general-purpose CPUs, realizing applications in FPGAs can act as hardware accelerators that can offload the computational burden from the CPU and also help in developing a prototype system before ASIC implementation. Although modern FPGAs have excellent hardware capabilities, high development difficulty is the main disadvantage. Image processing systems realized in FPGAs involve the pipeline data processing approach, where the pixel stream passes through different computing elements. In addition, while handling significant quantities of data, resource constraints often jeopardise the real-time performance of the system. To address these challenges, modern FPGA chips usually embed microprocessor cores to achieve the convenience and flexibility of software.

This paper presents a shape-based feature extraction system and adopts the Zynq SoC platform launched by Xilinx where it integrates the software programmability of an ARM-based processor with the hardware programmability of an FPGA, enabling hardware acceleration on a single device. Through this combination, Zynq has the advantages of having ARM and FPGA. The ARM core eases interfacing the peripherals while FPGA performs parallel processing as well as dynamic reconfiguration. The work discusses the hardware realization of the algorithm on Zynq architecture [20] so that the unit acts as a standalone shape feature extractor. Feature extraction algorithms generally employed include steps like pre-processing, feature detection and feature descriptor building. In the current work, Centroid Distance calculation and Fast Fourier Transform (FFT) are employed for feature building and the output obtained is in the form of feature descriptors. The proposed method effectively determines the shape Fourier descriptors by tackling the problems related to the hardware implementation, such as the requirement of nonlinear operations. These descriptors which uniquely characterises the object based on its shape is later utilized by classification algorithms. Although hardware realization of different features is reported, a computationally efficient algorithm is proposed, which employs custom modules developed using Verilog along with the Xilinx IP cores. However, hardware realization of the algorithm using a reconfigurable platform is a complex task that needs validation of the proposed methods in a simulation. The results of the proposed work are also validated by implementing the algorithm in MATLAB. The main contributions of the paper can be summarised as follows:

1) A vision system on Zynq heterogeneous platform is built, which covers shape-based feature detection, and process images of size 32 × 64 and generates feature descriptors/vectors at a clock rate of 62MHz
2) Shape feature using FFT offers translation, rotation and scale-invariant attributes and ensures less dedicated hardware resource utilization, thus making the robotic vision module compact.

Robotic systems with an embedded camera can make decisions based on the observations of the inputs around them. The proposed vision-enabled shape feature extraction technique integrated with robotic systems can be widely used in industrial sectors for automation by intelligent sensing of production lines. Hardware realization in FPGA ensures real-time processing of the captured images with sufficient speed and accuracy that satisfy ever-increasing production and quality requirements, consequently aiding in the development of totally automated processes. In addition, such systems can also be trained to work in extreme environments without involving human intervention.

The rest of the paper is organized as follows. Section 2 explains the overall structure of the algorithm used in the proposed shape detection. Section 3 explains the details of the proposed hardware implementation. Section 4 and 5 report the results obtained and conclude by discussing the future scope.

2. SYSTEM OVERVIEW

The proposed vision engine for shape feature determination includes the following: acceleration module for pixel stream processing and analysis, storage modules and associated interfaces. The ZYNQ-based image processing system establishes a connection with the host computer through the UART communication protocol, which enables data communication between the platform and the host computer. The feature extraction system is the core of the developed system. The working process for the entire system is described as follows: First, the robotic system is allocated the workbench where the industrial camera captures the scene with the aid of lens and associated light source. The camera image is processed and loaded into the Block RAM. Then, object boundaries, as well as the features, are extracted. Finally, these features are used for their classification. The overall architecture of the work is illustrated in Fig. 1.

2.1 ALGORITHMS FOR EFFICIENT COMPUTATION OF FEATURE DESCRIPTORS

Fourier descriptors are derived by applying Fourier Transform on a 1D Shape Signature. A shape signature z(t) is a 1D function representing 2D areas or boundaries [21]. A shape signature usually captures the perceptual feature of the shape. In the following, we assume that the shape boundary coordinates *(x(t), y(t)), t= 0,1,2,……, N-1* has been extracted in the preprocessing stage where t usually represents the arc length.

The position function of an object from its 2D view is derived from the boundary coordinates using the equation

$$z(t) = |x(t) - x_c| + i|y(t) - y_c| \qquad (1)$$

where (x_c, y_c) is the centroid of the shape, which is the mean of the boundary coordinates. For a sequence (x_1, y_1), (x_2, y_2), . . . , (x_N, y_N) of uniform contour points in order, where N refers to the sampling points on the contour, the centroid point of these contour points has to be calculated first. The centroid coordinates corresponding to the object in a binary image is the arithmetic mean of all boundary coordinates as described in the equation (2) and (3) where x_i and y_i are the x and y coordinates along the boundary of the object and N is the total number of boundary coordinates.

$$x_c = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (2)$$

and $$y_c = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad (3)$$

z(t) represents the shape boundary which is also translation invariant. Rotation causes *z(t)* circular shift and scaling of shape only introduces linear changes in *z(t)*. Employing position function (complex coordinates) as shape signature involves little computation and is calculated using Centroid Contour Distance (CCD).

The CCD function is the distance of the boundary points to the centroid (x_c, y_c) of the shape and *r(t)* is translation invariant.

$$r(t) = ((x_t - x_c)^2 + (y_t - y_c)^2)^{1/2} \qquad (4)$$

Rotation introduces circular shift while the scaling of shape only changes r(t) linearly.

The CCD feature is a distance sequence that describes the contour feature of the object by using the vector composed of the distance from the shape centroid to the contour point. The coordinates of the traversing point along the contour can be represented as a function whose period is defined by the perimeter of the shape boundary. This period function is represented by the Fourier series expansion.

This 1D signature which is derived from 2D shape boundary coordinates is subjected to discrete Fourier Transform of M points (centroid-distance points) and is given by the equation

$$F(\mathrm{u}) = \frac{1}{M}\sum_{\mathrm{x}=0}^{\mathrm{M-1}} \mathrm{f(x)e}^{-\frac{\mathrm{j2\pi ux}}{\mathrm{M}}}, \qquad (5)$$

For u=0,1,...,M-1

The results are a set of Fourier coefficients, which represents the shape using the feature vectors and these normalized Fourier coefficients are used as shape descriptors.

3. DESIGN OF PROPOSED VISION MODULE

3.1 ZYNQ HARDWARE PLATFORM

The proposed image processing system is realized with Xilinx fully programmable SoC chip ZYNQ-7000 7z010clg400-1 and is integrated with a high-performance dual-core ARM Cortex-A9 processor. It provides a wealth of peripheral interfaces and general expansion pins, and can generate the required processing and computing performance for high-end embedded system applications such as industrial control, machine vision, image and video processing and automotive driving assistance. The corresponding hardware block design for the module developed is shown in Fig. 2.

3.2 VISION MODULE FOR THE SHAPE FEATURE EXTRACTION

Image processor block forms the core of the architecture where real-time implementation of feature extraction algorithm of objects in 2D images is carried out. The proposed block determines the centroid as well as computes the Fourier descriptors. To compute the feature, a pipeline process is carried out for each frame and the digital machine for the feature extractor is shown in Fig. 3. In the first stage, the conversion from gray to binary is carried out and stored in Block RAM (referred to as BRAM1). In the second stage, contour coordinates (row, column indexes) of the region of interest are calculated and stored in another Block RAM (referred to as BRAM2). In the third stage, the indices are averaged to determine the x and y coordinates of the centroid (x_c, y_c). The centroid contour distance is determined using equation (4) in the fourth stage. Finally, the FFT of the centroid distance is determined which yields the shape feature descriptors. Starting from the next frame, the feature is obtained in each subsequent frame. BRAM1 is for storing the images which are captured using the cameras from the processor side. BRAM2 is for storing the indexes of the boundary pixels of the object. The feature vector can be stored in a BRAM that is subsequently accessed for classification.

Following are the main steps involved in the implementation of the image processor block:

1) Initialize row and column counters for reading each row of the binary 2D image stored in BRAM1.

2) Identify pixels (equal to 1) from each row that corresponds to the boundary; the position of row and column index is stored in BRAM2.

3) Average the row and column index to determine the centroid coordinates.

4) CCD distance is calculated by finding the square root of distances between each index position and centroid coordinates, which is the 1D shape signature of the object.

5) Determine the Fourier descriptors by applying FFT on the shape signature

The finite state machine is used to implement BRAM reading and writing, while square root and FFT operations are performed using the Xilinx IP Cores. The efficient data transfer between the BRAM IP core and ZYNQ hard core is very important as the image pixels are stored in the BRAM initially and it is configured in BRAM controller mode. The data is passed through AXI interconnect from the processor/ Processing System (PS) and read out again using custom controller module for further processing developed using Verilog HDL and the state diagram for exchange of data between BRAM is shown in Fig. 4.

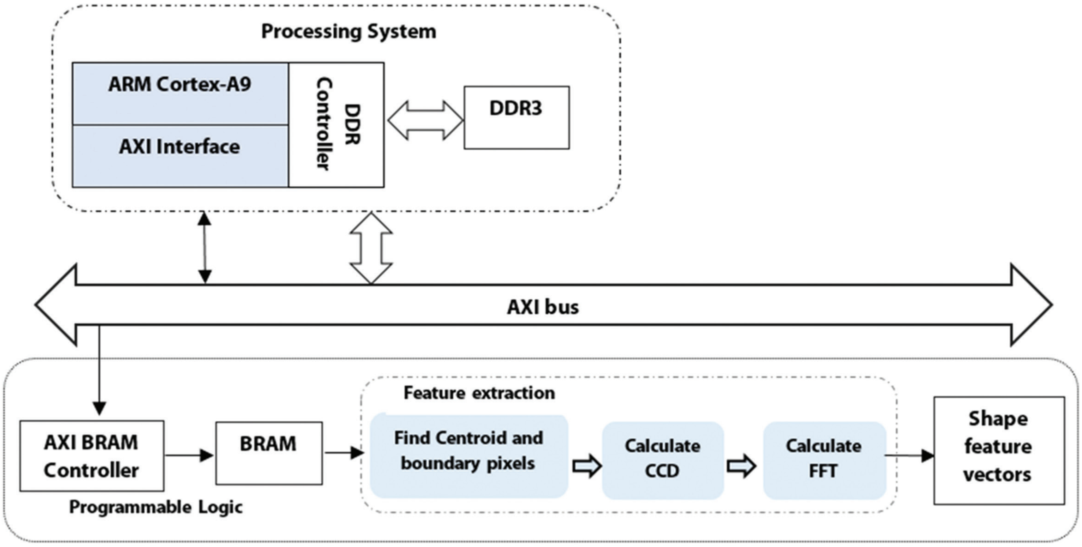


Fig. 1. The overall architecture of the proposed feature extraction module

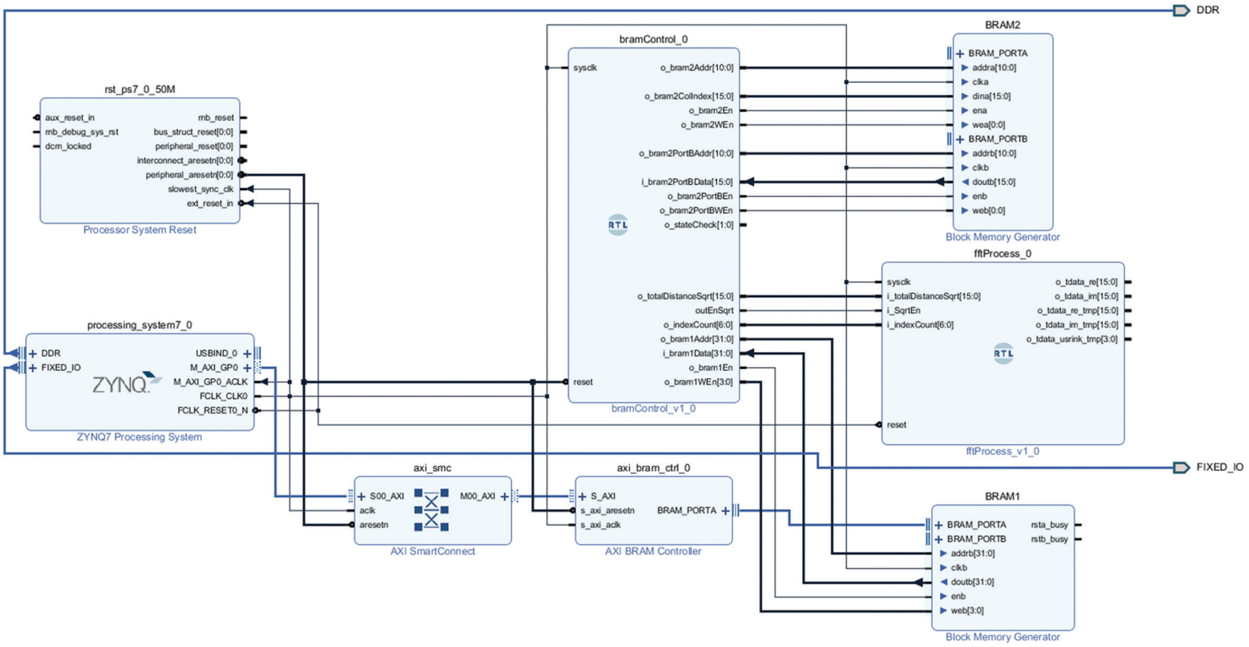


Fig. 2. Hardware Block Design of the proposed shape feature extraction module in Vivado

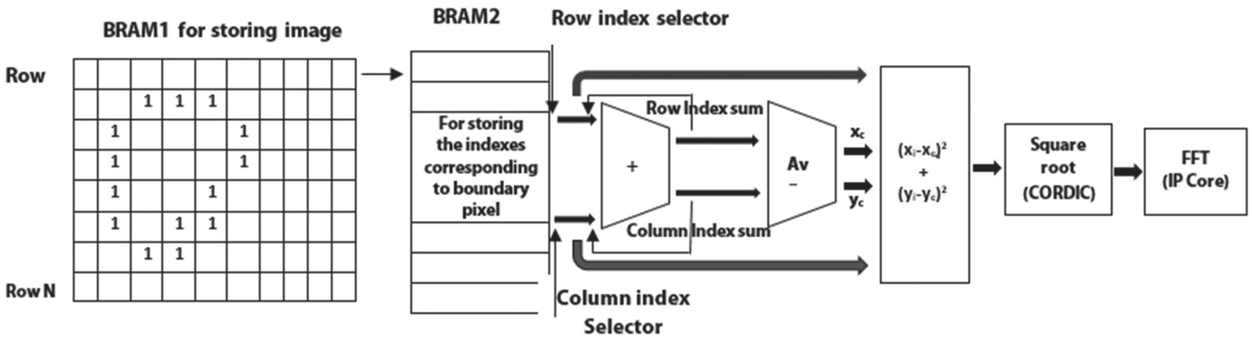


Fig. 3. Digital Machine for CCD –FD feature extractor

The image pixels are written to BRAM from the PS side and the access to the contents of the BRAM1 (configured in BRAM controller mode) is done using the Finite State Machine technique. The memory size of BRAM is initially calculated based on the standard image size from the camera interface. Four states are used to manipulate the pixels and the output of the state machine gives the position of the boundary pixels in terms of row and column index. In the **IDLE** state, all the signal initialisation for the data fetches is done. This state resets the row and column position counters for identifying the pixels on the object shape contour. The address is initialized to read the image data from the storage element. In the **RD_BRAM1** state, enable is made high while the read signal is tied low for its reading. The pixel value is checked if it is in object periphery, in such cases, its coordinates like row and column have to be stored which is done in **WR_BRAM2** state. Another BRAM2 (in standalone mode) is configured for storing the row and column index for each contour pixel. The process is repeated until the entire image frame is processed when it enters the **DONE** state. Likewise, another state machine is maintained to read the contents from the standalone BRAM for further processing.

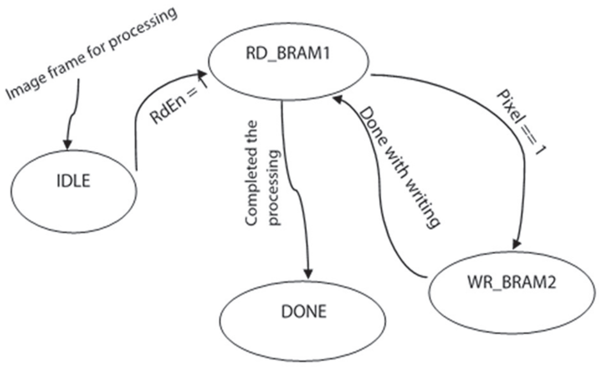


Fig. 4. State diagram showing the pixel manipulation of images

3.2.1 Centroid Calculator

It is necessary to identify the object's boundary pixels to obtain the (x_c, y_c) values for equations (2) and (3). It is determined by averaging up the row_index_sum and column_index_sum separately each time the clock changes.

3.2.2 CORDIC IP Core Configuration

Characterization of IP cores is important while concerning the timing for various configurations. This information is critical in the design flow and it affects all stages, from IP core selection to the synthesis of interfaces to calculation of latency and throughput. CORDIC IP Core is for determining the square root operation while calculating CCD distance. For the CORDIC IP core to be used as the square root calculator block parameters like pipelining mode, data format, input width and round mode has to be properly configured.

Not all configurable parameters will be available for all functions supported by the Xilinx CORDIC IP core. At the same time, some parameters cannot be configured when a particular parameter is selected. For instance, when configured for square root operation with 32-bit unsigned integer input, the output width is automatically set to 17 in the core and the architectural configuration is set to parallel. The square root of the distance between the boundary and centroid coordinates has to be computed for which CORDIC (6.0) IP core is employed and is optimized for FPGA fabrics. The data format chosen is unsigned integer, truncate rounding mode has opted.

3.2.3 FFT IP Core Mode Selection

The Xilinx FFT core (9.1) supports four architectures -Pipelined, Radix-4, Radix-2 and Radix-2 Lite and for the proposed image processing work, a pipeline architecture is adopted. The FFT core is configured to a transform length with the target of 50Millions of Floating-point operations per second (MPS) throughput. Input ports are connected to input source block through appropriate blocks keeping the data and signal integrity issues in mind. Out ports of the FFT processor are terminated, and only the output data such as real, imaginary and index values are captured. The outputs (real and imaginary components) of the FFT IP core are captured from the port m_axis_data (TDATA) by using the configuration settings. The real part of the FFT IP core is obtained by concatenating the bits {m_axis_data_tdata[20],m_tdata[14:0]}. The imaginary part is identified using the bits configuration {m_axis_data_tdata[44], m_axis_data_tdata[38:24]}. The index of the FFT Core is identified by m_axis_data_tuser[3:0].

The designed hardware architecture was functionally correct, simulation was performed with the image (rows) used for feature extraction and was evaluated to improve the simulation time of the design. Simulation of the algorithm is also done in MATLAB.Then the testbench results of the algorithm stages are compared to their counterpart implementation to ensure that the implemented hardware architecture has correct functionality.

The verilog implementation considers the image as a continuous stream of pixel values. The simulation and MATLAB results for the CCD step are almost identical even if the rounding errors associated with square root operations are considered. While performing FFT operations, suitable rearrangement of the output bits and conversion is performed to make the results comparable with software simulation. By performing proper bit alignment to get the real and imaginary components of Fourier descriptors, the implemented hardware architecture for the algorithm can be proved functioning correctly.

4. SIMULATION AND ANALYSIS RESULTS OF THE PROPOSED VISION MODULE

To evaluate the performance of the proposed shape feature extraction algorithm, it is tested on the KTH dataset [22] of hand tools and was collected using the Yumi pedestral robot platform under vision setup. Data set consists of handtools of category- hammer, plier and screwdriver that are taken under different illumination and background as in Fig. 5

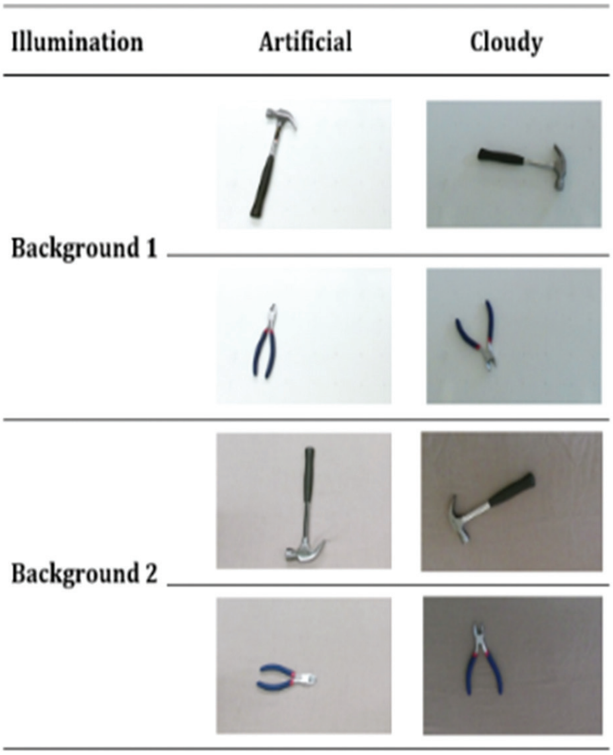


Fig. 5. Sample images of the KTH database captured under the Vision Setup

The shape feature vectors determined are tested using eight different objects: two hammer variants, three distinct pliers and three different screwdrivers. The evaluation used a total of 640 images divided into 8 different classes with 40 images per class. Each of the datasets is captured under artificial and cloudy background. The feature vectors so calculated for the hand tools of the KTH database under different illumination and background setting are translation, rotation and scale-invariant.

To test the robustness of the proposed scheme, we determine the Fourier coefficients of rotated and translated postures of a sample object. The accuracy plot is drawn to prove the feasibility and effectiveness of the proposed algorithm, as shown in Fig. 6

The accuracy was around 86% with 10 feature points (descriptors) and was chosen as the optimum number of points for the classification. Fig. 7a and 7b show the processed binary images and the extracted shape signatures respectively. Taking the Fourier Transform of each of the shifted signatures produced the same normalized Fourier coefficients and can be shown that our implementation is invariant to translation and rotation as shown in Fig. 8.

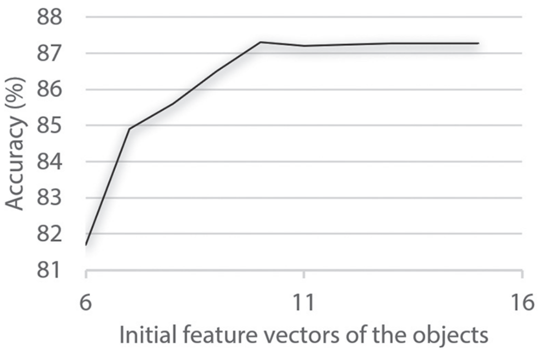
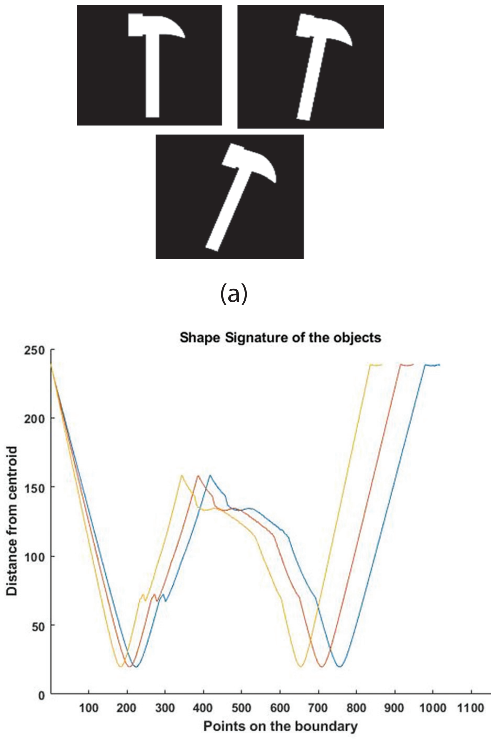


Fig. 6. Accuracy plot using SVM classifier

The hardware implementation of the proposed architecture is synthesized for the Zybo board using Xilinx Vivado Design Suite 2019.2. Simulation is done to verify the functionality of the shape-based feature extraction algorithm. The implementation was also done in MATLAB and both the results were evaluated with the same frames. The results obtained from the simulator are validated by converting the fixed-point output to integer format and are almost identical.

The operations were performed on a 32 × 64 image and the proposed hardware acceleration module computes the feature vectors within 16.13ns(62 MHz). In this context, it can be recalled that the software implementation of the technique requires 26.3 ms to extract the features. The simulator results of the feature extractor are shown in Fig. 9. The output lines u_ila_0_o_tdata_re_tmp[15:0] and u_ila_0_o_tdata_im_tmp[15:0] are the user defined signals from the ILA (Integrated Logic Analyser) corresponding to real and imaginary components of the FFT IP Core.


(a)

(b)

Fig. 7. Translation and rotation invariance of objects (a) Hand tools for extracting feature vectors (b) The shape signatures of the images with the starting points shifted.

These signals correspond to o_tdata_re_tmp[15:0] and o_tdata_im_tmp[15:0] (of the block fftProcess_0 of the hardware block design). u_ila_0_o_tdata_usink_tmp[3:0] in the simulated waveform provides the index of the Fourier coefficients. The hardware utilization after the place & route of the implemented algorithm is given in Table 1. The FPGA essentially consists of hardware resources such as memory, slice registers, slice LUTs, LUT flip flop pairs and DSP blocks. A full post-synthesis and post-routing test-bench simulation are performed to measure the frame processing time. Here, it is shown that 21.08% of LUTs and 25% of available BRAM is used and this utilisation includes image processing, centroid determination and feature extractor.

Table 1. Resource Utilisation of the proposed shape feature extraction architecture

Resource	Utilisation	Available	Utilization (%)
LUT	3710	17600	21.08
LUTRAM	605	6000	10.08
FF	5025	35200	14.28
BRAM	15	60	25.0
DSP	6	80	7.50
BUFG	2	32	6.25

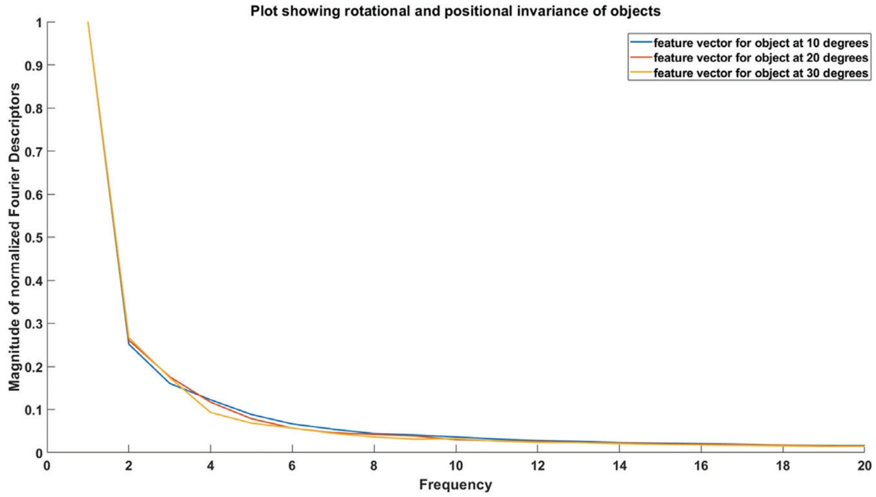
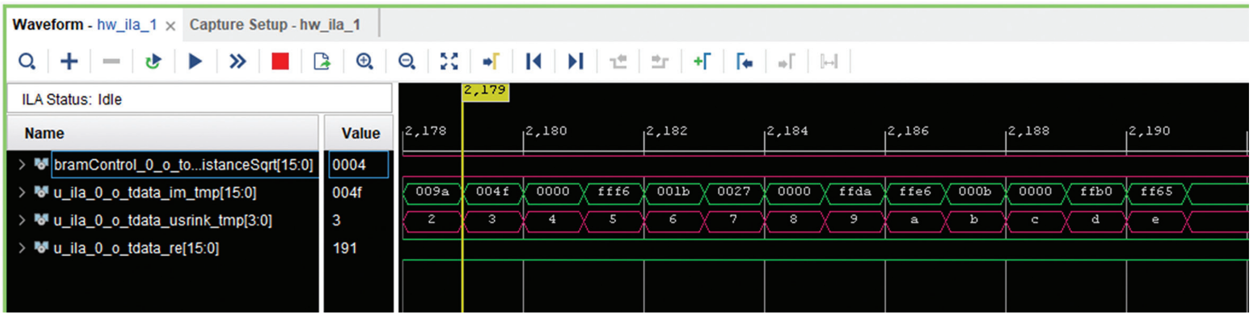


Fig. 8. The initial normalized Fourier coefficients of the shifted shape signatures


Waveform - hw_ila_1 × Capture Setup - hw_ila_1

ILA Status: Idle

Name	Value
bramControl_0_o_to...istanceSqrt[15:0]	0004
u_ila_0_o_tdata_im_tmp[15:0]	004f
u_ila_0_o_tdata_usrink_tmp[3:0]	3
u_ila_0_o_tdata_re[15:0]	191

2,179
2,178
2,180
2,182
2,184
2,186
2,188
2,190
009a 004f 0000 fff6 001b 0027 0000 ffda ffe6 000b 0000 ffb0 ff65
2 3 4 5 6 7 8 9 a b c d e

Fig. 9. Simulated results of the proposed Shape Fourier descriptor

5. CONCLUSION

This paper has presented a real-time shape feature extraction algorithm for embedded platforms. This algorithm combines the centroid distance and Fast Fourier Transform for determining the feature descriptors. These are accelerated by using Field Programmable Gate Arrays, which satisfies the real-time performance of the algorithm in embedded platforms. Algorithms for the shape feature extraction have been implemented and tested by hardware modules developed using Verilog language.

Hardware implementation of shape feature vector has been demonstrated using a sample image where processing was done at a speed of 16.13ns as compared to software implementation. The interaction using standard AXI4 interfaces allows different modules in the system to be exchanged or configured easily. Direct Memory Access (DMA) will be investigated for carrying out image transfer for future study.

6. REFERENCES

[1] M. Lu, C. L. Chen, "Detection and Classification of Bearing Surface Defects Based on Machine Vision", Applied Sciences, Vol.11, No.4, 2021.

[2] Y.J Chen, J. C. Tsai, Y. C. Hsu, "A real-time surface inspection system for precision steel balls based on machine vision", Measurement Science and Technology, Vol. 27, No. 7, 2016, pp. 74010-74019.

[3] Z.Ren, F. Fang, N. Yan, Y. Wu, "State of the Art in Defect Detection Based on Machine Vision", International Journal of Precision Engineering and Manufacturing-Green Technology, 2021.

[4] L. Purez, I. Rodriguez, N. Rodriguez, R. Usamentiaga, D. F. Garcia, ”Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review”, Sensors, Vol.16, No.3.2016

[5] G. Maier, F. Florian, M. Wagner, C. Pieper,R. Gruna, B. Noack, H.K. Emden, T. Langle, U. D. Hanebeck, S. Wirtz, V. Scherer, J. Beyerer, “Real-time multitarget tracking for sensor-based sorting”, Journal of Real-Time Image Processing, Vol. 16, No. 6, 2019, pp. 2261-2272.

[6] M Schluter, C Niebuhr, J Lehr, J Kruger, “Vision-based Identification Service for Remanufacturing Sorting”, Proceedings of the 15th Global Conference on Sustainable Manufacturing, Haifa, Israel, 25-27 September 2017, 2018, pp.384-391.

[7] K. Xia, Z. Weng, "Workpieces sorting system based on industrial robot of machine vision", Proceedings of the 3rd International Conference on Systems and Informatics, Shanghai, China, 19-21 November 2016, pp. 422-426.

[8] A. A. Eissa, A. A. Khalik, ”Understanding color image processing by machine vision for biological materials”, Structure and Function of Food Engineering, InTech Publishers, 2012.

[9] Q. Luo, X. Fang, L. Liu, C. Yang, Y. Sun, "Automated Visual Defect Detection for Flat Steel Surface: A Survey", IEEE Transactions on Instrumentation and Measurement, Vol. 69, No. 3, 2020 pp. 626-644.

[10] L. Armi, S. F.-Ershad, “Texture image analysis and texture classification methods - A review”, International Online Journal of Image Processing and Pattern Recognition, Vol. 2, No.1, 2019, pp. 1-29.

[11] T. Czimmermann, G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C. M. Oddo, P. Dario “Visual-Based Defect Detection and Classification Approaches for Industrial Applications—A Survey”, Sensors, Vol. 20, No.6, 2020.

[12] M. I. Al Ali, K. M. Mhaidat, I. A. Aljarrah, "Implementing image processing algorithms in FPGA hardware", Proceedings of the IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, Amman, Jordan, 3-5 December 2013, pp. 1-5.

[13] N. Nausheen, A. Seal, P. Khanna, S. Halder, “A FPGA based implementation of Sobel edge detection”, Microprocessors and Microsystems, Vol.56, 2018, pp. 84-91.

[14] R. Maini, H. Aggarwal, "Study and Comparison of Various Image Edge Detection Techniques", International Journal of Image Processing, Vol. 3, No. 1, 2009, pp. 1-11.

[15] A. B. Amara, E. Pissaloux, M. Atri, "Sobel edge detection system design and integration on an FPGA based HD video streaming architecture", Proceedings of the 11th International Design & Test Symposium, Hammamet, Tunisia, 18-20 December 2016, pp. 160-164.

[16] Y. Zheng, “The Design of Sobel Edge Extraction System on FPGA”, Proceedings of the 7th International Conference on Information Science and Technology, Washington, USA, 16-19 April 2017.

[17] G. B. Reddy, K. Anusudha, "Implementation of image edge detection on FPGA using XSG", Proceedings of the International Conference on Circuit, Power and Computing Technologies, Nagercoil India, 18-19 March 2016, pp. 1-5.

[18] S. Taslimi, R. Faraji, A. Aghasi, H. R. Naji, "Adaptive Edge Detection Technique Implemented on FPGA", Iranian Journal of Science and Technology – Transactions of Electrical Engineering, Vol. 44 No.4, 2020, pp. 1571-1582.

[19] A. G. Mahalle, A. M. Shah, "An Efficient Design for Canny Edge Detection Algorithm Using Xilinx System Generator", Proceedings of the 3rd International Conference on Research in Intelligent and Computing in Engineering, Universidad Don Bosco, El Salvador, 22-24 August 2018, pp. 1-4.

[20] Zybo Z7 Board Reference Manual https://reference.digilentinc.com/programmable-logic/zybo-z7/reference-manual (accessed: 2021)

[21] D. S. Zhang, G. Lu, "Study and evaluation of different Fourier methods for image retrieval", Image and Vision Computing, Vol. 23, No.1, 2005, pp. 33-49.

[22] M. Mancini, H. Karaoguz, E. Ricci, P. Jensfelt, B. Caputo, "Kitting in the Wild through Online Domain Adaptation", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 1-5 October 2018, pp. 1103-1109.