



# Implementation of Augmented Reality using Reconfigurable Hardware

**Praveenkumar Babu**, Department of Electronics and Communication Engineering, College of Engineering and Technology, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Kanchipuram, Chennai, Tamil Nadu, India. Authors E-mail: praveenb2@srmist.edu.in

**Eswaran Parthasarathy**, Department of Electronics and Communication Engineering, College of Engineering and Technology, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Kanchipuram, Chennai, Tamil Nadu, India. Authors E-mail: eswaranp@srmist.edu.in

**Abstract-** Augmented Reality (AR) is a sensational paradigm where real world experiences are enhanced by virtual environment based on computer vision recognition algorithms for physical objects. The real-time image and video processing is challenging because of its computation and complex functions. Field Programmable Gate Arrays (FPGAs) with their reconfigurable architectures provide flexibility, better performance and high levels of parallelism. This paper proposes the implementation of AR using reconfigurable hardware such as Xilinx Zynq SoC (System on a Chip) based on SURF (Speeded-Up Robust Feature) algorithm. The integration of AR and reconfigurable hardware for real-time image processing applications such as detection and feature extraction of images is presented. The implementation results achieve better resource utilization of approximately about 4% of flip-flops, 12% of Look-up Tables (LUTs), 3% of DSP slices and 38% of Block RAMs (BRAMs) for object recognition images with 640x480 resolutions.

**Keywords:** FPGAs, Augmented Reality, SoC, reconfigurable hardware and parallelism, Squeeze Det, Corner Net, SIFT

## I. INTRODUCTION

Technology is rapidly improving better and much smaller. Augmented Reality (AR) is gaining an impact on modern era by integrating real world with the virtual environment. The drawback is that the processing task for AR is executed on software algorithms which are not the best possible solution to meet the requirements. AR will demand the support from reconfigurable hardware such as FPGAs which will render flexibility for real-time processing. Reconfigurable computing is a technology incorporating hardware and software resources providing parallelism and flexibility. The implementation of AR on reconfigurable hardware can deliver wide range of applications in the fields of healthcare planning, industrial design, social interaction, navigation, tourism and sightseeing, emergency management and so on.

Most of the features of AR inspired the advantages of reconfigurable hardware [1]. AR involved in real-time processing which gives high computational complexity since it is interacting repeatedly with real-time users. For this reason, reconfiguration plays a major role in real-time image processing which provides improved performance and reduced costs, size and power consumption [2]. Object detection and feature extraction have made deep learning techniques more interesting as they renders variety of applications [3]. The comprehensive survey of object detection with deep learning techniques have been presented in [4].

This paper presents the implementation of AR using reconfigurable FPGA such as Xilinx Zynq® XC7Z020-1CLG484C device and the results shown the hardware resources utilization [5]. In this work, feature extraction and object detection techniques for real-time image processing is implemented.

## II. APPROACH FOR OBJECT DETECTION

Based on the high attraction for AR applications, various object detection and tracking algorithms were followed such as for colour segmentation [6], implementation of artificial neural network (ANN) [7], for hand tracking [8], hand gesture recognition [9] and FPGA implementation for increased pixel rate [10]. There are different algorithms for object detection such as YOLO (You Look Only Once)[11], Viola-Jones, Features from Accelerated Segment Test (FAST) R-CNN [12], support vector machines (SVM), HOG (Histogram of Gradients) [13], Squeeze Det, Corner Net, SIFT (Scale Invariant Feature Transform) and so on. The generic real-time object detection methodology is shown in Figure 1. The input image is taken from webcam (640 x 480 resolutions) and it is matched by using Speeded-Up Robust Feature (SURF)

algorithm. It involves in the approximation of Gaussian smoothing, blob detector (Hessian matrix) and descriptor (Haar-wavelet responses). While comparing descriptors from two images, matching pairs can be obtained as shown in Figure 2. Then the target image is selected by the user to match features pixel by pixel with the input image. This traditional approach has more advantages when compared with other techniques.

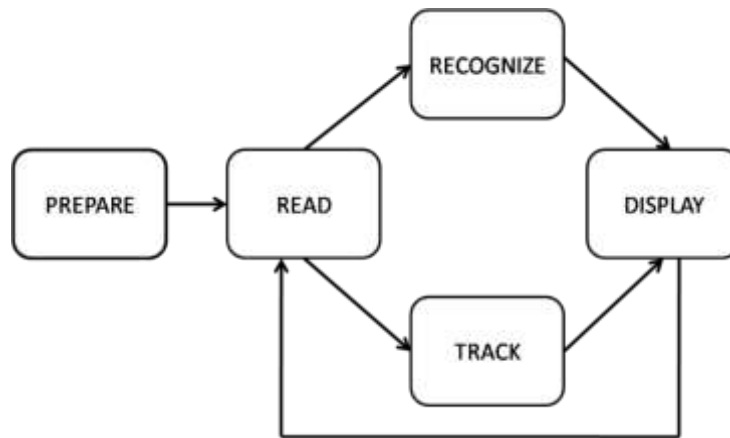


Figure 1. Basic block diagram for object recognition

For the detection of matching points, SURF uses approximation of Hessian blob detectors using integral image  $I_{\Sigma}(x)$  at location  $x = (x, y)^T$  which represents the sum of all pixels in the input image (I) as given in equation (1).

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (1)$$

SURF operates Hessian matrix  $\mathcal{H}(x, \sigma)$  for blob detection which is accurate and provides high performance where  $L_{xx}(x, \sigma)$  and  $L_{yy}(x, \sigma)$  are the convolution of Gaussian second order derivative at point  $x$  and  $y$  respectively is given by,

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2)$$

Based on the Hare-wavelet descriptors, the orientation of the images can be computed. For example, based on the object detection algorithm, the input image is matched with target image and matched points are highlighted in Figure 2. If the target image is not matched, then the desired output is not recognized. In this approach, scaling transform has been used to change the dimensions of the target image. The sample algorithm for feature extraction is given below.

Sample Algorithm for feature extraction

```

img1 = snapshot (cam Obj); img2 = rgb2gray (sceneImage1);
Box Points = detect SURF Features (box Image);
Scene Points = detect SURF Features (img2);
[Box Features, box Points] = extract Features (box Image, box Points);
[Scene Features, scene Points] = extract Features (img2, scene Points);
Box Pairs = match Features (box Features, scene Features);
Matched Box Points = box Points (box Pairs (:, 1), :);
Matched Scene Points = scene Points (box Pairs (:, 2), :);
  
```

Scaling transform is used to compress or expand the dimensions of a given object. The scaling factor  $S_x, S_y$  scales the object in X and Y direction respectively and represented in matrix form as given in equation (3),

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \text{ or } I' = S.I \quad (3)$$

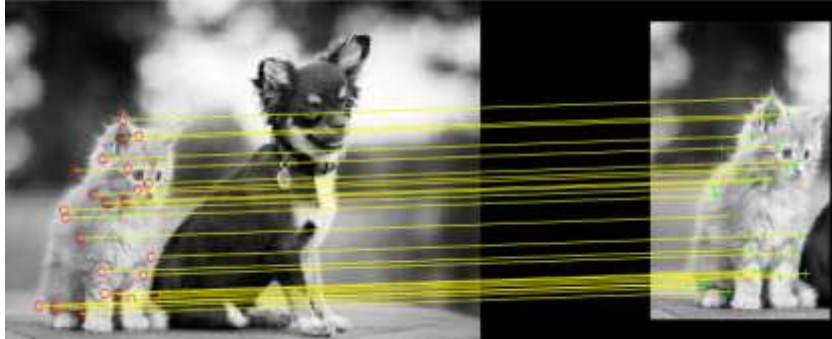


Figure 2. Object recognition using matching features.

### III. EXPERIMENTAL SETUP

The real-time object detection and recognition algorithm is implemented on Xilinx Zynq® XC7Z020-1CLG484C device [14]. It comprises Dual ARM Cortex® A9 processors, Advanced Microcontroller Bus Architecture (AMBA) Interconnects, DDR3 component memory, UART, Ethernet, QUAD SPI flash memory along with Xilinx 7000 Programmable Logic (PL). It can allow the usage of hardware resources easily along with software libraries present in Vivado® High-level Synthesis (HLS) [15]. Advanced Extensible Interface (AXI) Interconnects are used to interconnect the processing system with the hardware components. Using Vivado HLS tool, C/C++ or MATLAB code can be transformed to RTL code for simulation. These tools have various libraries for image and video processing applications [16]. The experimental setup includes Logitech C100 Web cam (640x480 resolutions) connected to the Xilinx FPGA. The input image can be acquired from webcam and it is processed and the desired output image can be seen on display screen. The desired output frame may be an image or a video file chosen by the user. Figure 3 shows the Experimental setup for object detection using Zynq hardware.

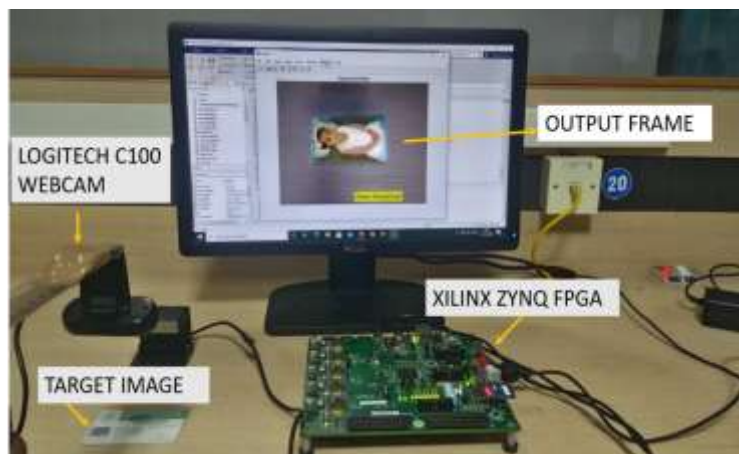


Figure 3. Experimental setup for object detection using Zynq hardware.

### IV. RESULTS AND DISCUSSION

Synthesis and simulation results have been generated from Xilinx Vivado HLS tools. These tools provide the flexibility of integrating hardware and software components with better resource utilization. The proposed implementation occupies about 4% of flip-flops, 12% of Look-up Tables (LUTs), 3% of DSP slices and 38% of Block RAMs (BRAMs) for object recognition images with 640x480 resolutions, Table I. Resource Utilization for implementation.

**Table I:** Resource Utilization for implementation

| Entity            | Available Resources | Utilized Resources |
|-------------------|---------------------|--------------------|
| DSP Slices        | 220                 | 3 %                |
| Flip-flops        | 106,400             | 4 %                |
| Block RAMs(BRAMs) | 140                 | 38 %               |
| LUTs              | 53,200              | 12 %               |

## V. CONCLUSION AND FUTURE WORK

Acceleration of augmented reality on reconfigurable Zynq platform is implemented which will be suitable for real-time image processing, computer vision, wearable computing, multimedia and graphics. Reconfigurable Computing provide better flexibility and utilization of resources for the development of AR applications. Vivado HLS tools provide best possible way for the integration of hardware and software modules for the proposed implementation. The results shown the better utilization of resources and primary constraint is the efficiency and accuracy of overall process. One of the limitations in object detection is the high computation for the entire task to run in real-time with desired memory and resources. Furthermore, this work also proposes the implementation approach of dynamic reconfiguration for different types of AR applications in the fields of deep learning techniques.

## REFERENCES

1. W. Luk, K.Lee, R. Rice, N. Shirazi and Y. Cheun, "Reconfigurable computing for augmented reality," 7<sup>th</sup> Annual IEEE Symposium on Field-Programmable Custom Computing Machines, pp. 136-145, 1999.
2. A. Kaur, "A Survey on FPGA Implementations in Embedded Augmented Reality Applications", 2018 6<sup>th</sup> Edition of International Conference on Wireless Networks & Embedded Systems (WECON), Rajpura, India, pp. 23-26, 2018.
3. C.B. Murthy et. al., "Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and Embedded Platforms—A Comprehensive Review", Applied Sciences, 10, 3280, 2020.
4. Li Liu et.al., "Deep Learning for Generic Object Detection: A Survey", International Journal of Computer Vision, vol. 128, pp. 261-318, 2019.
5. Xiongwei Wu, Doyen Sahoo and Steven C.H. Hoi, "Recent advances in deep learning for object detection", Neurocomputing, Elsevier, vol. 396, pp. 39-64, 2020.
6. R. Garcia, J. Battle, & R. Bischoff, "Architecture of an Object-Based Tracking System Using Colour Segmentation", Proc. of International conference on Image and Signal Processing, Elsevier, Manchester, UK pp. 299-302, 1996.
7. C. Johnston, K. Gribbon & D. Bailey, "FPGA Based Remote Object Tracking for Real-Time Control", Proc. of International Conference on Sensing Technology, Palmerston North, New Zealand, pp. 66-72, 2005.
8. M. Krips, J. Velten, & A. Kummert, "FPGA Based Real Time Hand Detection by Means of Colour Segmentation", DOKLADY of Belarussian State University of Informatics and Radioelectronics, pp. 156-162, 2003.
9. P. In, K. Jung, & H. Kwang, "An Implementation of an FPGA-Based Embedded Gesture Recognizer Using a Data Glove", Proc. of the 2nd International Conference on Ubiquitous Information Management and Communication, ACM, Rennes, France, pp. 496-500, 2008.
10. Z. He, H. Huang, M. Jiang, Y. Bai and G. Luo, "FPGA-Based Real-Time Super-Resolution System for Ultra High Definition Videos," 2018 IEEE 26<sup>th</sup> Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Boulder, CO, pp. 181-188, 2018.
11. Huang Y-Q et. al., "Optimized YOLOv3 algorithm and its application in traffic flow detections," Appl. Sci. 10, 3079, 2020.
12. A. Sharma et. al., "Implementation of CNN on Zynq based FPGA for Real-time Object Detection," 2019 10<sup>th</sup> International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, pp. 1-7, 2019.
13. Jens Rettkowskiet.al., "HW/SW Co-Design of the HOG algorithm on a Xilinx Zynq SoC", Journal of Parallel and Distributed Computing, vol. 109, pp. 50-62, 2017.

14. Xilinx Inc. Zynq SoC overview datasheet. Accessed on 18th October, 2020.
15. Xilinx Inc. Vivado Design Suite Tutorial High Level Synthesis, UG871 (v.2014.1) May 6, 2014. Accessed on 9th October, 2020.
16. Babu, P., Parthasarathy, E, "Reconfigurable FPGA Architectures: A Survey and Applications," J. Inst. Eng. India Ser. B, 2020. <https://doi.org/10.1007/s40031-020-00508-y>.