

A Reconfigurable Area and Energy Efficient Hardware Accelerator of Five High-order Operators for Vision Sensor Based Robot Systems

Qianjin Wang¹, Yi Zhan¹, Bingqiang Liu¹, Jiajun Wu¹, Youhua Shi², Guoyi Yu¹, Chao Wang¹

¹ School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan, China

² Faculty of Fundamental Science and Engineering, Waseda University, Tokyo, Japan.

Abstract—This paper proposes a reconfigurable hardware accelerator design of five major high-order operators for vision sensor based robot systems. These five high-order operators include convolution, median filtering, Euclidean distance, vector inner-product and iToF, which are intensively used in robot vision algorithms. In this work, a reconfigurable hardware accelerator design method for multiple high-order operators is proposed. FPGA implementation results show that the proposed design has achieved area efficiency with 17.54% reduced LUTs and 44.02% reduced FFs against the baseline hardware design of the five high-order operators. Case studies of Laplace edge-detection and iToF benchmark demonstrate the energy efficiency of proposed design with 19.70% and 6.2% reduction in energy consumption, respectively.

Keywords—Reconfigurable hardware; vision sensor; robot SoC

I. INTRODUCTION

Vision sensors have been widely used in many intelligent robot systems including localization, path exploring, path planning, etc. After obtaining the image data from the vision sensor, robot system normally uses certain image algorithms containing pixel-level high-order operators to extract the required information [1]. Reconfigurable hardware design is preferred in mobile intelligent robot SoC systems instead of the separate hardware implementation of these operators, because the reconfigurable design method can reuse the common arithmetic operators, reduce the hardware overhead, and improve the energy efficiency [2].

This paper focuses on the reconfigurable hardware design of the five major high-order operators: convolution, median filter, iToF (Indirect Time of Flight) operators, vector inner product and Euclidean distance, which are frequently used in robot vision systems. This work proposes a reconfigurable hardware accelerator design method for multiple high-order operators. A reconfigurable hardware accelerator of these five operators is efficiently implemented in FPGA by reusing basic arithmetic operators and case studies confirm the area efficiency and energy efficiency of the proposed design method.

II. PROPOSED RECONFIGURABLE HARDWARE ACCELERATOR OF THE FIVE HIGH-ORDER OPERATORS

A. Proposed reconfigurable hardware accelerator design method for multiple high-order operators

The linear array reconfigurable hardware is suitable for structured and repetitive computing applications, as computational algorithms can be mapped as pipeline with less hardware overhead compared with the two-dimensional reconfigurable hardware [3]. In the linear array reconfigurable hardware architecture, arithmetic operators are arranged in layers, while switch networks are used to connect the arithmetic operators between two adjacent layers. The flow of proposed reconfigurable hardware design method is shown in Fig. 1.

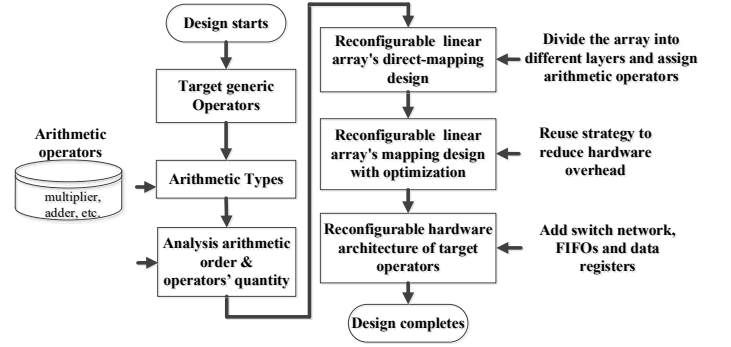


Fig. 1. Design flow of proposed method of reconfigurable hardware for multiple high-order operators for robot vision applications

B. Reconfigurable hardware accelerator design of five major high-order operators for robot vision application

The five-operator reconfigurable design is based on the following scenarios: the convolution kernel size is set as 3*3 and 5*5; the median filter template size is set as 3*3; the input vector size of Euclidean distance is 8 or 16; the vector size of vector inner product is 8 or 16; the iToF is depicted by:

$$d = k \cdot \varphi_{iToF} \cdot \varphi_{iToF} = \arctan[(A_0 - A_2)/(A_1 - A_3)] \quad (1)$$

where A_0, A_1, A_2, A_3 are the phase data acquired from iToF sensor and k is a parameter determined by sensor [4]. This study focuses fixed-point implementation of the high-order operators.

Arithmetic operation types and the quantities of arithmetic operators are analyzed as following: 3*3 convolution requires 9 multipliers and 8 adders, while 5*5 convolution requires 25 multipliers and 24 adders; median filter requires 21 comparators; the Euclidean distance with vector size of 8 requires 8 multipliers and 15 adders, while the other one with vector size of 16 requires 16 multipliers and 31 adders; similarly, one vector inner product requires 8 multipliers and 7 adders, while the other one requires 16 multipliers and 15 adders, respectively; iToF needs 1 multiplier, 1 CORDIC unit and 2 adders. As shown in Fig.2 (a), the direct-mapping design can be composed of 5 layers that are assigned with different arithmetic operators: the first layer has 16 adders for subtraction of two vectors in Euclidean distance and A_0, A_1, A_2, A_3 in iToF; the second layer contains a CORDIC unit to calculate arctan in iToF; the 25 multipliers in the third layer and 24 adders in the fourth layer are to complete the multiplications and additions in operators including Euclidean distance, convolution, vector inner product and iToF; the fifth layer needs 21 comparators to perform the median filtering. This simple-mapping hardware design can be further optimized by the proposed design method in Fig. 1.

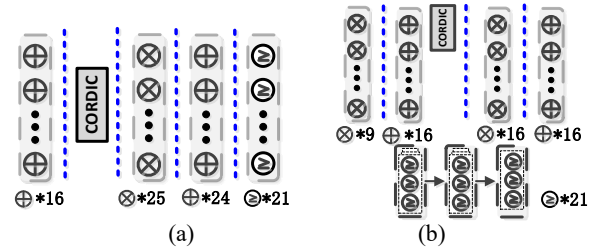


Fig. 2. (a) Direct-mapping reconfigurable hardware design; (b) reconfigurable hardware design after optimization by the proposed method.

Fig. 2 (b) presents the optimized reconfigurable design. Setting a separate layer for the comparators and CORDIC used in only one operator will increase the overhead of interconnection. The comparators are set with separate data path because they don't receive data from other arithmetic operators, while the CORDIC is combined with the former layer because it needs adders' results in the former layer. With the two optimizations, two-layer interconnection overhead can

be well reduced. Additionally, the 9 multipliers in the third layer can be extracted as the first layer alone to realize a part of multiplication in the 5*5 convolution. In this way, the adders in the second layer can be reused to complete the addition of the multiplication's results, which can reduce 8 adders in the fourth layer.

When the accelerator is configured to calculate convolution and median filtering, FIFOs are needed as line buffers to store rows of pixels, and data registers are required to store the data from the FIFOs in the form of 3*3 or 5*5 matrix. When the accelerator is configured for other high-order operators, the FIFOs are configured to store external data and the data registers are used to store temporary data. Additionally, some adders' results need to be transmitted to other adders in the same layer through interconnection to realize the pipelined adder tree. Switch network consisting of MUXs are used to connect the FIFOs and arithmetic operators. Fig. 3 shows the reconfigurable accelerator architecture after overall optimization.

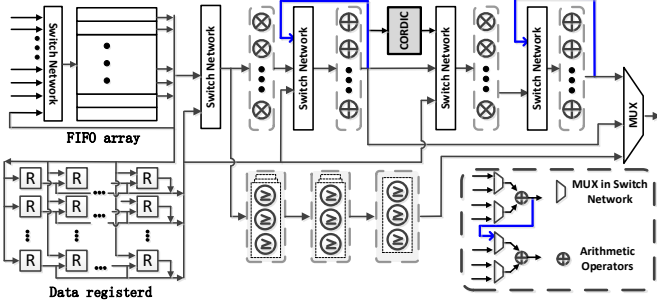


Fig. 3. Proposed reconfigurable hardware accelerator architecture of five high-order operators.

III. IMPLEMENTATION RESULTS AND DISCUSSIONS

A. Reconfigurable solution VS separate hardware implementation solution

TABLE I shows the required arithmetic operators and hardware utilization in FPGA implementation of the proposed design. The proposed reconfigurable accelerator uses significantly fewer arithmetic operators as well as reduced LUTs and FFs against the separate hardware implementation.

B. Case study of Laplacian convolution

Laplacian operator is a second derivative operator often used in edge detection, which is consisting of a convolution and a subsequent comparison between convolution results and threshold. Laplacian operator's kernel matrix is defined as

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (2)$$

A case study is performed to use the reconfigurable hardware accelerator to implement the Laplacian-based edge detection. The edge detection results are shown in Fig. 4. FPGA implementation shows that, compared with the separate hardware solution's power consumption of 0.066W, the proposed hardware solution's power consumption is 0.053W, which is reduced by 19.7%.

C. Case study of iToF

When the proposed hardware accelerator is configured to calculate iToF, the FIFO array stores the phase data A_0, A_1, A_2 and A_3 . Two adders in the second layer are used to calculate $A_0 - A_1$ and $A_2 - A_3$. The subtraction's results are sent to the CORDIC to calculate ϕ_{iToF} , and a multiplier in the third layer is used to calculate the $k \cdot \phi_{iToF}$. In this case study, the test data are calculated by Matlab iToF algorithm to measure distance at a range of [0.2m, 20m], and then converted into fixed-point numbers for the proposed hardware accelerator. As there is a

positive proportional relationship between d and ϕ_{iToF} , we evaluate the relative error about ϕ_{iToF} by the following

$$\lg E_r = \lg \left[\left| \phi_{iToF}^p - \phi_{iToF}^f \right| / (\phi_{iToF}^f) \right] \quad (3)$$

where ϕ_{iToF}^p is calculated by the proposed hardware accelerator and ϕ_{iToF}^f is calculated by Matlab using float-point algorithm.

Fig.5 presents the relative error of iToF implemented by the proposed reconfigurable design is at $10^{-2} \sim 10^{-4}$, which is tolerable in the short-range robot navigation application. FPGA implementation shows that, compared with separate hardware solution's power consumption of 0.065W, the proposed hardware solution's power consumption is 0.061W, which is reduced by 6.2%.

TABLE I. EVALUATION OF PROPOSED RECONFIGURABLE DESIGN

Arithmetic operator	Separate hardware implementation	This work	Reduction
Multiplier	58	25	56.90%
Adder	72	32	55.56%
CORDIC	1	1	/
Comparator	21	21	/
FPGA utilization	-	-	-
LUTs	14563	12009	17.54%
FF Registers	4846	2713	44.02%

Device: Xilinx xa7a35tcs324-11

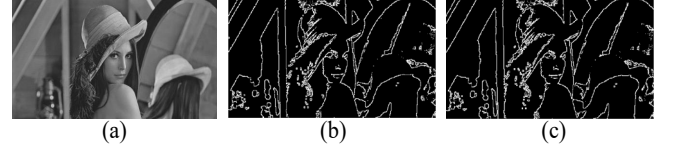


Fig. 4. Edge detection results. (a) The original image, (b) The edge detected by Laplacian algorithm model using Matlab, (c) The edge detected by the the proposed reconfigurable design.

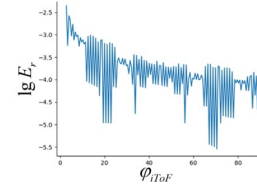


Fig. 5. The relative error of iToF results by the proposed reconfigurable design against float-point calculation results in Matlab.

IV. CONCLUSION

This paper proposes a reconfigurable hardware accelerator of five high-order operators for robot vision applications. First, a reconfigurable hardware accelerator design method for multiple high-order operators. Then, an area and energy efficient reconfigurable hardware accelerator of five high-order operators is realized. Finally, FPGA implementation results show that the proposed accelerator has significant higher area and energy efficiency than the baseline design, which provides a potential design choice for mobile intelligent robot SoC systems.

ACKNOWLEDGMENT

This research presented in the paper is partially supported by National Key Research & Development Program of China (2019YFB1310001). Corresponding email: yuguoyi@189.cn.

REFERENCES

- [1] McAndrew, et.al. "An introduction to digital image processing with matlab notes for scm2511 image processing." School of Computer Science and Mathematics, Victoria University of Technology 264.1 (2004): 1-264.
- [2] S. Purohit, et.al. "Throughput/Resource-Efficient Reconfigurable Processor for Multimedia Applications," in IEEE Trans. VLSI Syst., vol. 21, no. 7, pp. 1346-1350, July 2013.
- [3] D. Fronte, et.al., "Celator: A Multi-algorithm Cryptographic Coprocessor," 2008 Intl. Conf. Reconfig. Comput. FPGAs, Cancun, Mexico, 2008, pp. 438-443.
- [4] Z. Chen, et.al., "Calculating depth image with pixel-parallel processor for a ToF image sensing system," 2015 IEEE Sensors, Busan, South Korea, 2015, pp. 1-4.