



A 28 nm, 397 μ W real-time dynamic gesture recognition chip based on RISC-V processor

Yong-Liang Zhang, Qiang Li, Hui Zhang, Wei-Zhen Wang, Jun Han^{*}, Xiao-Yang Zeng, Xu Cheng

State Key Laboratory of ASIC and System, Fudan University, Shanghai 200433, China

ARTICLE INFO

Keywords:

Dynamic gesture recognition
Programmable

ABSTRACT

This paper presents a low-power programmable dynamic gesture recognition chip based on a RISC-V processor using RGB images. The chip uses two gesture recognition algorithms to recognize the most commonly used dynamic gestures (up, down, left, and right) with any shape of a hand at the distance of 10 cm ~ 90 cm and speed of 10 cm/s ~ 150 cm/s. The chip is fabricated in a 28 nm process and occupies an area of 640 μ m x 640 μ m. The accuracy reaches 93% ~ 99% at complex backgrounds with 320x240 RGB image. It operates either in the recognition mode or in the standby mode that can be automatically switched to the other. The power consumption of the recognition mode is 397 μ W at 0.584 V supply voltage, 25 MHz clock, and 30 FPS. And the power consumption of the standby mode at room temperature is 78.3 μ W.

1. Introduction

With the rapid development of intelligent devices, IoT, AR/VR, and more, the new intelligent human-computer interaction (HCI) with characters of natural, barrage-free, contactless, fast, and convenient, has become the inevitable trend to replace the traditional non-intelligent HCI methods such as the usual mouse, keyboard, touch screen [1,2]. Gesture recognition (GR) as an import HCI has been gradually used by smartphones, IoT, wearable devices, intelligent homes, and more for its most natural, noiseless features. However, these fields have extremely strict requirements for power consumption, GR chips need to have low power consumption, high precision, low complexity characteristics [3, 4]. GR is largely dependent on the acquisition equipment and is realized by different acquisition devices and algorithms. Lamberti [5] extracts the features of color gloves for real-time GR, and the recognition accuracy reaches 97%. However, gloves are uncomfortable for wearable, and in many cases, they are not suitable to wear. With the appearance of the bracelet, GR through muscle electrical signals has become one of the trends. Ulysse [6] uses transfer learning to recognize electromyographic (EMG) for GR and achieves high recognition accuracy. At the same time, millimeter-wave radar has also become a good acquisition device for GR and is widely used in a variety of intelligent devices [7].

Another GR technology applies images as inputs. Lu [3] applies the most widely used ordinary camera to gain gesture with RGB image and realizes the dynamic and static GR chip with high energy efficiency ratio. Ba [8] realizes wake-up and dynamic gesture recognition (DGR) chip with an infrared sensor. The TOF (Time of flight) camera captures

images with 3D depth for recognizing gestures in real-time [9]. The device Kinect [10], which is composed of many kinds of acquisition devices, has been widely used in GR. The Kinect provides color images, depth images, and skeleton data with rich information, which is conducive to the needs of GR [11]. Also, the skeleton data generated by Kinect plays a role in GR. Liu [12] decouples the gesture into hand posture variations and hand movements using 3DCNN and 2DCNN based on gesture skeleton. To further reduce the power consumption, the image sensor [13,14] is used for image acquisition in GR.

RISC-V [15,16] is an open-source hardware instruction set architecture (ISA) based on the established RISC principles, which has the compelling advantages of simplicity, extendibility, flexibility, customizability, high energy efficiency and free. More and more companies and universities have joined the RISC-V Foundation, and RISC-V has been used in IoT, machine learning, wearable devices, and more. In this work, an open-source RISC-V core rocket-chip [17] is used to configure and control our chip for two DGR algorithms, which can significantly improve the flexibility and programmability of our chip, accelerate the design and reduce the design cost.

In this paper, we propose a low-power high-precision programmable DGR chip based on 320 x 240 RGB image, which is integrated, configured, and controlled by RISC-V processor [18]. Our chip operates in the complex backgrounds at the distance of 10 cm~90 cm, at the speed of 10 cm/s~150 cm/s, and gets accuracy of 93%~99%. Our chip identifies the most commonly used dynamic gestures (up, down, left, and right) with characteristics of programmable, easy to transplant,

^{*} Corresponding author.

E-mail address: junhan@fudan.edu.cn (J. Han).

<https://doi.org/10.1016/j.mejo.2021.105219>

Received 28 April 2021; Received in revised form 2 August 2021; Accepted 15 August 2021

Available online 25 August 2021

0026-2692/© 2021 Elsevier Ltd. All rights reserved.

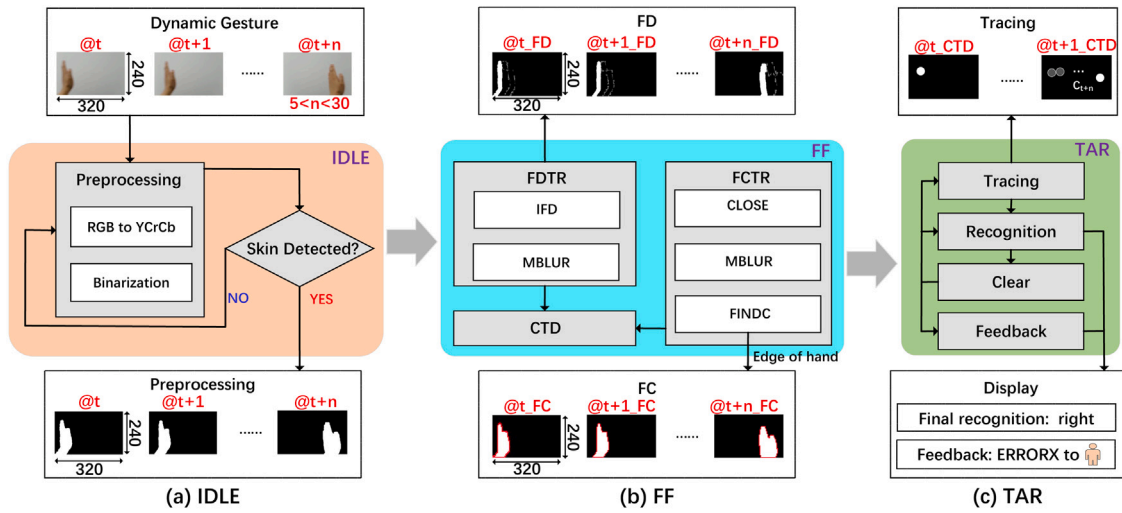


Fig. 1. The proposed DGR algorithm.

low power consumption, high precision. Besides, we use a simple way to make the chip enter the standby mode, which greatly reduces the power consumption of the chip.

2. The proposed DGR algorithms

2.1. The influencing factors of DGR

In addition to the influence of sampling equipment, the recognition effect of DGR is directly related to several factors including distance, speed, illumination, environment. However, there is no perfect algorithm and hardware for DGR in various complex backgrounds. Therefore, it is an urgent need to further improve the recognition ability, flexibility, and applicability of DGR for the environment through the algorithm and hardware.

2.2. The detecting algorithm

To meet the practical needs as much as possible, we use the ordinary RGB camera as our image acquisition equipment. The input image size is 320×240 , and the frame frequency is 30 frames per second (FPS). Fig. 1 shows our proposed DGR algorithm with low algorithm complexity, which consists of two GR methods named inter-frame difference for gesture tracking and recognition (FDTR) and finding contours for gesture tracking and recognition (FCTR).

$$Skin = \begin{cases} Cb = (-43R - 85G + 128B)/256 + 128 \\ Cr = (128R - 107G - 21B)/256 + 128 \\ Cr \in [133, 173] \\ Cb \in [77, 127] \end{cases} \quad (1)$$

When the IDLE module receives the image, the preprocessing unit (PREU) uses Eq. (1) to preprocess each pixel which separates the skin color region from the non-skin color region. The PREU includes the YCrCb conversion operation and the binarization operation. After PREU, the total amount of skins in the image is compared with the preset threshold value to determine whether there is a hand in the camera field. Also, the result of the comparison decides whether the GR algorithm enables subsequent processes, as shown in Fig. 1(a).

2.3. The FDTR/FCTR method

We design two DGR methods under the condition of reuse resources as much as possible to be suitable for DGR in two different application scenarios, as shown in Fig. 1(b). FDTR is suitable for gesture movement

with no other skin-like area's movement at complex backgrounds when FCTR is suitable for one-handed movement with no or similar skin color in the background. The switch between the two algorithms is operated by software.

Firstly, FDTR carries out an inter-frame difference (IFD) operation on the binary image to get the difference image containing the moving area of the hand. Secondly, FDTR carries out the median blur (MBLUR) operation on the difference image to filter the salt and pepper noise caused by uneven illumination. Finally, since the difference image only contains the moving area of the hand, the centroid (CTD) operation directly calculates the centroid of the moving area for the subsequent tracking and recognition process (TAR).

Similarly, FCTR firstly carries out the CLOSE operation, which includes dilation and erosion operations, on the binarization image to gain a complete hand contour. Secondly, the MBLUR is applied, which is the same as the MBLUR in FDTR. Thirdly, the find contour operation (FINDC), which is the same as OpenCV's FindContours function, finds the hand's contour and gets the edge of the hand. CTD calculates the centroid of the gesture by the edge. FCTR operation is suitable for only one hand or hand with the user's face. For background with multiple skin or skin-like tones, the number of contours exceeds 2, FCTR gives some feedback to adjust the backdrop.

2.4. TAR

After FDTR or FCTR operation, the centroid is sent to the TAR operation to obtain the direction of the dynamic gesture, as shown in Fig. 1(c). TAR tracks the centroid, calculates the accumulated historical displacement (AHD), compares AHD with the current displacement (CRD) generated by the current centroid (CURC) and the last centroid (LAC), and recognizes the direction according to the classifier of GR. TAR gives feedback based on the number of centroid in the FCTR. After sending feedback and recognition, TAR clears the AHD.

3. The processor block diagram

3.1. Overall architecture

The proposed programmable DGR chip is composed of peripherals, OV5640 Config unit (OVUCU), UART, QSPI, PREU, and DGR system, as shown in Fig. 2. The peripheral devices include the OV5640 camera that collects RGB images, a PC that interacts with the DGR chip by UART, and an external flash that saves the running program. The camera needs to be initialized by OVUCU through software or hardware

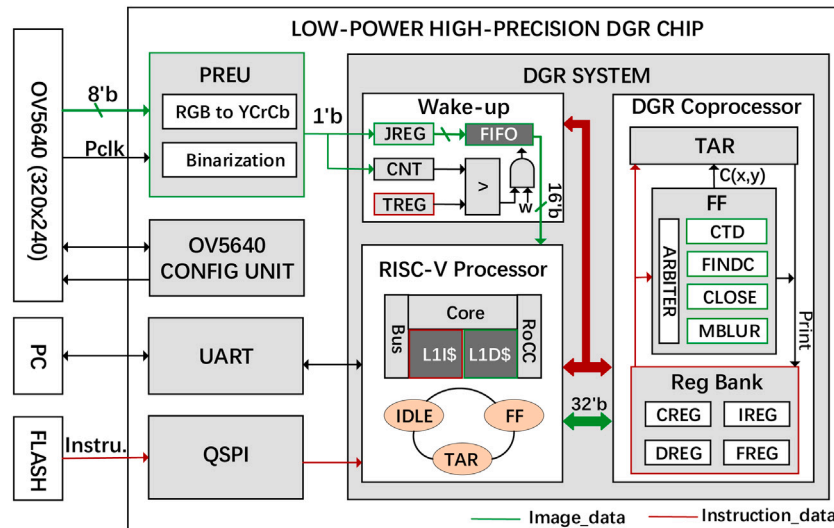


Fig. 2. The proposed low-power high-precision DGR chip.

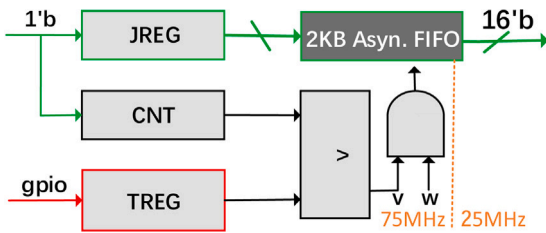


Fig. 3. The wake-up unit.

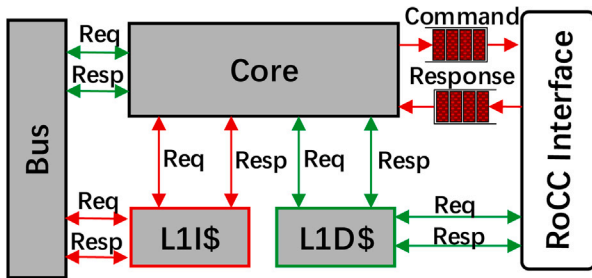


Fig. 4. The RISC-V processor.

configuration. The DGR system is developed based on RISC-V instruction set. It has the capability of simplicity, low power consumption, modularization, and extensibility. And it enhances the application of GR in embedded, IoT, intelligent devices, and more.

3.2. PREU and wake-up unit

After initializing the camera, the DGR chip gets a 320×240 RGB image stream from the camera. When PREU receives the image from the camera, it binarizes the image as proposed before and sends it to the wake-up unit. In PREU, the image is decomposed into the skin color/non-skin color. And the image size is compressed by 16 times, which reduces the power of subsequent transmission and processing.

The wake-up unit, on the one hand, uses 16-bit JREG to concatenate the pixels, on the other hand, adds up the binary image and saves the sum in 17-bit CNT register, as shown in Fig. 3. When a complete image enters the chip, the CNT value is compared with the 17-bit threshold value TREG which is configured by the RISC-V processor. The result

of the comparison determines whether the next frame is written into 2 kB asynchronous FIFO and whether the chip wakes up the subsequent recognition process. The clock of all operations before writing FIFO is 75 MHz provided by OV5640, and all subsequent operations use the given main clock of 25 MHz. According to the comparison, the standby function is realized in a simple way with lower cost and lower standby power consumption.

3.3. The RISC-V processor

We use a RISC-V processor, named rocket-chip, for configuration, data handling, control, output, etc. In contrast to most instruction sets, the RISC-V instruction set is freely used for any purpose, allowing anyone to design, manufacture, and sell RISC-V chips and software. While this is not the first open-source instruction set, it is significant. Because it is designed to work with modern computing devices, such as warehouse-scale cloud computers, high-end mobile phones, and tiny embedded systems.

In this paper, the RISC-V processor contains 4 kB L1I\$ and 24 kB L1D\$, as shown in Fig. 4. L1I\$ gets the instructions from the external flash through QSPI and the bus. L1D\$ can be directly accessed by the core and coprocessor and the 16-bit data from the FIFO is stored in the corresponding location of L1D\$. The core controls the operation of the coprocessor through the rocket custom coprocessor (RoCC) interface. The RISC-V processor effectively improves the hardware development cycle, reduces the control of hardware resources, and makes our chip has the programmable, extensible ability.

3.4. The DGR coprocessor

As shown in Fig. 5, the DGR coprocessor includes the RoCC interface that interacts with the RISC-V processor, the arbiter module that determines the priority of reading and writing L1D\$, the FF and TAR module. The coprocessor performs operations under the control of the RISC-V processor through the RoCC interface. DGR coprocessor implements the aforementioned FDTR/FCTR through the program.

The RoCC interface module contains the register bank that can be written and read by the RISC-V processor. CREG configured by the processor for improving DGR precision at different scenarios stores parameters. Changing parameters can achieve the optimal recognition effect at different distances, different moving speeds. IREG is the extension instructions from the processor to control the processes of the coprocessor. As shown in Table 1, the extended instructions are

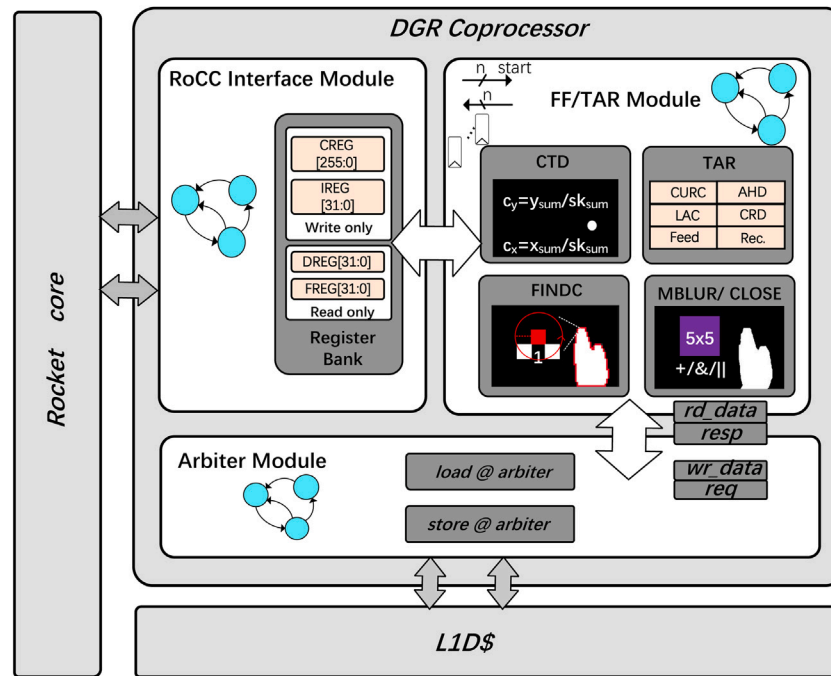


Fig. 5. The DGR coprocessor.

Table 1

The extension instruction.

Extension Ins.	Function
RoCC_CONFIG	Config CREG
RoCC_INITIAL	Initial coprocessor and clear AHD
RoCC_MBLUR	MBLUR
RoCC_CLOSE	CLOSE
RoCC_FINDC	FINDC
RoCC_CTD	CTD
RoCC_TAR	TAR
RoCC_PRINT	Print FREG and DREG

Table 2

Feedback situation.

Situation	Feedback
Tip1: CTD_Count > 2	Please make sure there are no skin-like areas in background!
Tip2: CTD_Count = 2 && Face at right	Please move your face left or tips!
Tip3: CTD_Count = 2 && Face at left	Please move your face right or tips!
Tip4: CTD_Count = 2 && Face at bottom	Please move your face up or tips!
Tip5: CTD_Count = 2 && Face at top	Please move your face down or tips!

combined for our two DGR algorithms. DREG and FREG store the output direction and feedback from TAR for print.

The FF and TAR modules are the DGR calculation modules, including MBLUR, CLOSE, FINDC, CTD, and TAR, which are controlled by instructions in IREG. Each module is independent of the other, and the computation cannot be executed in parallel, controlled by instructions. Since MBLUR, CLOSE, FINDC, and CTD all need to read and write L1D\$, the arbiter module is required to control the reading and writing of L1D\$.

The MBLUR module uses a 5×5 kernel to slide on the image to gain the sum. If the sum beyond the median value, the center of the 5×5 area is 1, otherwise 0. Similarly, the CLOSE module uses “&” and “||” instead of “+” in MBLUR. Each instruction of MBLUR and CLOSE takes effect on only one row of data, which hides calculation time in image transfer. As mentioned earlier, MBLUR eliminates the noise of salt and pepper, and CLOSE gets a better-closed hand contour.

For FINDC, firstly, it traversals the image until detecting skin pixel, and sets this coordinate as a breakpoint. Then FINDC performs the eight-neighborhood search to find the hand peripheral contour and calculates the area. Thirdly FINDC performs CTD and fills the current area with zeros. Finally, FINDC returns to the first step and continues traversing the image from the breakpoint until traversing the whole image.

For FCTR, CTD reads out all the skin pixels according to the coordinates of the peripheral contour and calculates the value of the centroid. And CTD is executed in the course of a FINDC loop. For the FDTR

module, CTD reads out the whole image and calculates the centroid of the image.

The TAR module draws the movement trajectory of the hand according to the centroid and takes the AHD and CRD as the direction judgment standard. The tracing bases on continuous growth in one direction of X/Y. When the CRD is opposite to the growth direction or the large abrupt displacement is appearing, the direction of DGR is recognized and written into DREG.

3.5. Program and feedback

The chip is controlled by the C program, which includes the camera configuration, the TREG/CREG configuration, the image transfer, and performing operations. For FDTR, the IFD process is done during the image transfer, as shown in Fig. 6.

For feedback, the core receives data from FREG and outputs the corresponding tips, instructing the user to change their background and location, as shown in Table 2.

4. Experimental results

4.1. Wake-up and recognition processing latency

As shown at the top of Fig. 7, the transmission process of the image is represented by vertical synchronization (VSYNC) and horizontal synchronization (HSYNC). And the wake-up unit judges the binary image

FCTR	FDTR
<pre> OVCU(320x240); GPIO(TREG); RoCC_CONFIG(CREG); int pic[240]; while(1) { RoCC_INITIAL; for(int i=0; i<240; i++){ STORE(pic[i]); RoCC_MBLUR(i); RoCC_CLOSE(i); } RoCC_FINDC(pic); RoCC_TAR; RoCC_PRINT(FREG, DREG); } </pre>	<pre> OVCU(320x240); GPIO(TREG); RoCC_CONFIG(CREG); int pic1[240], frame-dif[240]; while(1) { RoCC_INITIAL; for(int i=0; i<240; i++){ IFD(pic[i], image[i], frame-dif[i]); STORE(frame-dif[i]); RoCC_MBLUR(frame-dif[i]); } RoCC_CTD(frame-dif); RoCC_TAR; RoCC_PRINT(DREG); } </pre>

Fig. 6. The DGR program.

from the PREU module whether to perform wake-up or sleep operation in the next frame. If the wake-up operation is executed, which means the chip enters the recognition mode. The wake-up unit pulls up the valid signal and the chip is in recognition mode. Instead, when the recognition process ends, the valid signal is set to 0, and the chip is in standby mode. The image input time is 33 ms, and the latency of the wake-up/sleep processes is three cycles after the input process.

For FDTR and FCTR, their recognition process is programmatically controlled, as shown in Figs. 6 and 7. For FDTR, FD and MBLUR run at the HSYNC transfer time and the time between HSYNC and HSYNC, as shown at the middle of Fig. 7. After the transmission of a frame, CTD and TAR process are calculated. TAR outputs a recognition result in each frame, and the output direction is valid if DREG is not 0. The latency of recognition is just three cycles after CTD.

For FCTR, MBLUR and CLOSE also run at the HSYNC transfer time and the time between HSYNC and HSYNC, as shown at the bottom of Fig. 7. When the L1D\$ stores the frame, FINDC is starting to find the contours and the CTD calculates the centroid by the coordinates from FINDC. Due to the particularity of FINDC, there is multiple cyclic switching between FINDC and CTD. The subsequent TAR process is the

same as the FDTR's TAR. It is important to note that FINDC detects multiple profiles and uses this to feedback to the tester's environment.

4.2. DGR accuracy

To test the accuracy of our chip at different speeds and different distances, we used a hand mold and a sliding rail that can control the speed to conduct experiments, because the speed and motion process of the hand are difficult to control. The distance we tested is 10 cm~90 cm, and the speed we tested is 10 cm/s~150 cm/s at normal light, as shown in Fig. 8.

As shown in Fig. 8(a), for FDTR, in the distance of 10 cm~40 cm, the accuracy under 20 cm/s~30 cm/s or 100 cm/s~150 cm/s is lower than the 30 cm/s~100 cm/s. When the distance exceeds 50 cm and speed is 10 cm/s, the chip ignores the movement with small CRD which means the movement is evaluated as small jitter. When the speed exceeds 10 cm/s, the recognition accuracy is 95%~99%.

As shown in Fig. 8(b), for FCTR, in the distance of 10 cm~20 cm, the recognition accuracy is zero, because the centroid does not change when the image is filled by the hand. As the distance increases, the chip can recognize the dynamic gesture. And the recognition accuracy is 93%~99% for the FCTR algorithm.

The two algorithms in the chip achieve accuracy of 93%~99% under complex background, and the accuracy can be optimized by adjusting the configuration parameters according to the actual needs. For FDTR, the testing distance of our chip can exceed 90 cm, but FCTR needs to ensure that the face is located in the center of the image in a longer distance, otherwise, feedback will appear for adjustment. In the longer distance, the parameter of TREG needs to be adjusted to avoid the chip being stuck in standby mode.

4.3. The measured power consumption

As shown in Fig. 9(a), the DGR system consumes 397 μ W at 0.584 V supply voltage, 25 MHz clock frequency for FDTR, without compromising functionality. While for FCTR, the DGR system consumes 483 μ W at 0.580 V supply voltage, 35 MHz clock frequency. Due to the influence of the FIFO depth in the wake-up unit and the high time of FINDC, the minimum clock frequency of FDTR is 35 MHz. For the standby mode,

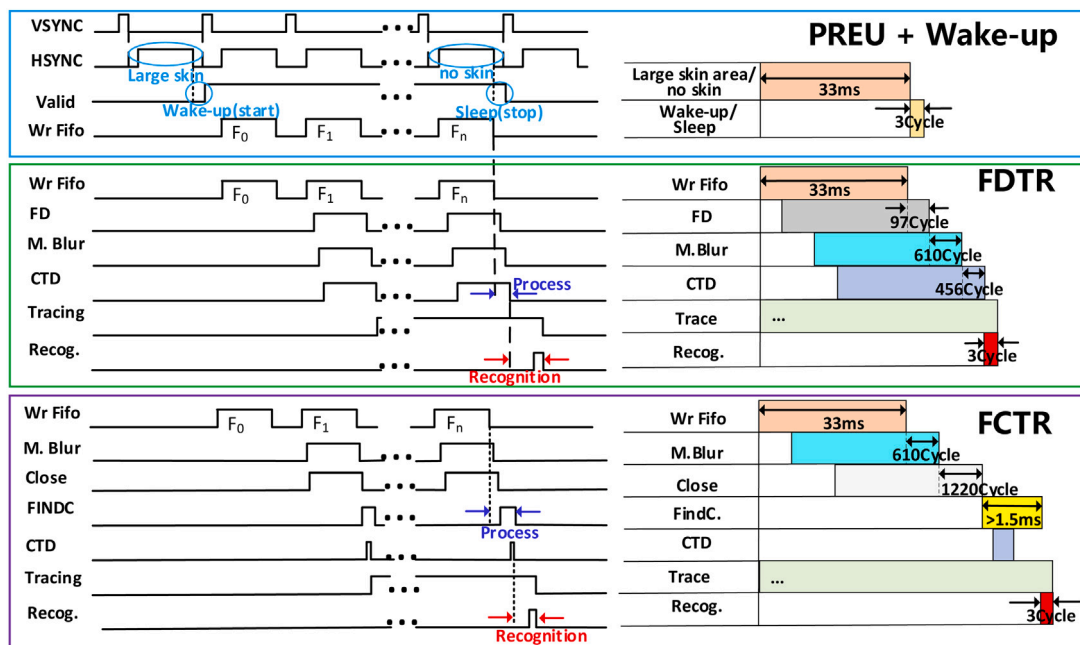


Fig. 7. The latency of wake-up and recognition.

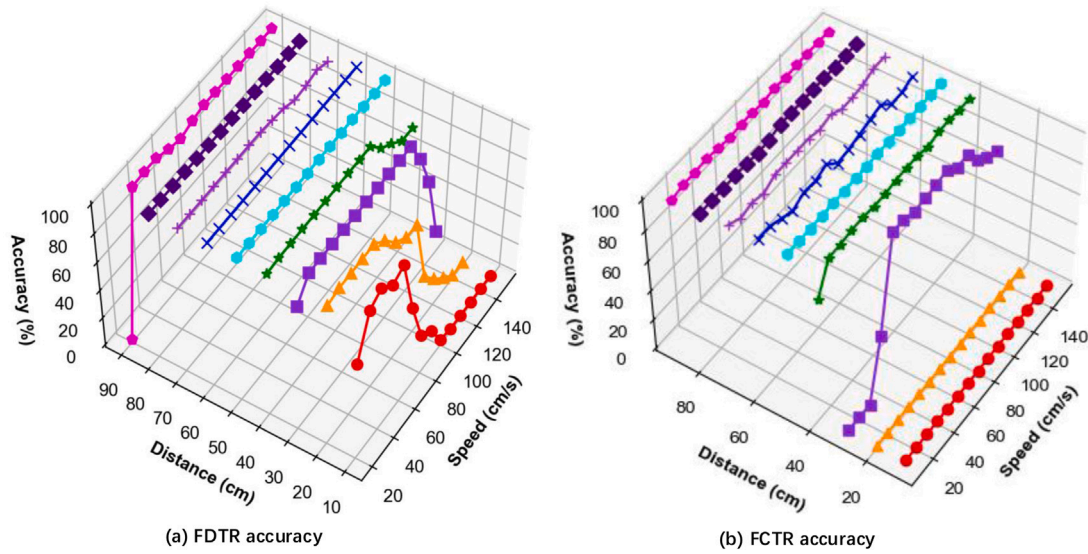


Fig. 8. Measured accuracy of DGR chip.

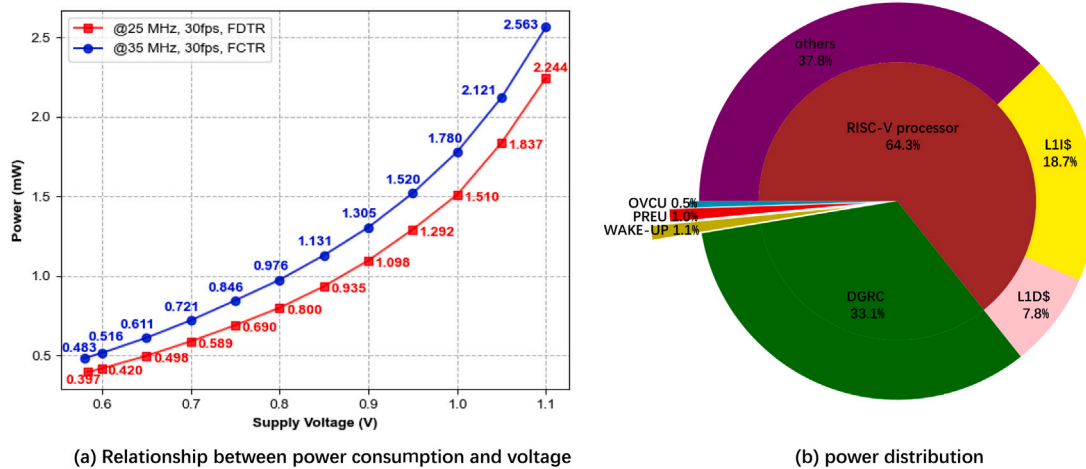


Fig. 9. Measured power distribution of our DGR chip.

the DGR system does not work, that is, the static power consumption is the standby power consumption when the master clock is turned off. The standby mode consumes 78.3 μ W for FDTR and consumes 82.5 μ W for FCTR at the minimum voltage and minimum clock.

As shown in Fig. 9(b), the power consumption ratio of CAMC, PREU and WAK-UP is 0.5%, 1.0% and 1.1%, respectively. The RISC-V processor’s power consumption ratio is 64.3%, including 7.8% for the L1D\$, 18.7% for the L1I\$, and 37.8% for the others (except storage), since RISC-V core controls all processes. DGRC power consumption accounted for 33.1%.

4.4. Chip summary and test demo

The test chip was fabricated in 28 nm CMOS process and occupies an area of 640 μ m \times 640 μ m, as shown in Fig. 10(a). The entire chip contains 36 I/O ports and 26 kB memory. With 30 frames of 320 \times 240 RGB images as input, the DGR chip consumes 397 μ W and 78.3 μ W in the recognition and standby mode at 0.584 V.

In comparison with previous state-of-the-art gesture detections as shown in Table 3, our chip achieves 93%~ 99% recognition at distances of 10–90 cm and speeds of 10–150 cm/s under complex backgrounds. With a view to fair comparison, we normalize the energies at different technology nodes with respect to 28 nm using an energy normalization

factor. Large image inputs inevitably bring higher energy consumption. To evaluate the performance of our chip, we calculate the energy of each pixel (EOEP) to remove the impact of the input image size inconsistency. The calculation results show that our chip has the lowest EOEP of 0.1723 nJ/pixel. Our chip has higher accuracy, suitable for longer distances and complex backgrounds with the lowest EOEP.

The demo system consists of three parts, which are the DGR chip test board, the PC receiving and sending the direction, and the FPGA board for “Snake” and “Tetris” games, as shown in Fig. 10(b). The PC receives the direction of the gesture in real-time and then sends the direction to control the games. We move the hand up, down, left, and right in front of the camera. And the test board recognizes the direction and sends it to the PC. The PC sends the control instructions to the FPGA board for game control.

5. Conclusions

A low-power high-precision DGR chip is proposed for smartphones, IoT, wearable devices, and etc. The chip integrates a small RISC-V processor as the controller, which has the ability of programmable and portable. With RGB images as inputs, the chip performs real-time calculations and identifies the most commonly used up, down, left, and right dynamic gestures with an accuracy of 93%~99%. Besides, the chip runs

Table 3
Comparison with other state-of-the-art gesture recognition works.

Specifications	This work	[19] VLSI2020	[4] SSCL2019	[14] ISSCC2018	[20] ISSCC2016
Process	28 nm	65 nm	65 nm	65 nm	65 nm
Sensor type	Camera	Image sensor	Image sensor	3D Camera	Camera
Resolution	320 × 240 Image	32 × 32 Image	32 × 32 Image	3D Image	2 × 320 × 240 Image
Area	640 μm × 640 μm	1.78 mm ²	680 μm × 590 μm	4 mm × 4 mm	4 mm × 4 mm
Frame rate (FPS)	30	30	30	33.3	30
Voltage supply	0.580 V–1.100 V	0.8 V	0.46 V	0.85 V	1.2 V
Power	397 μW	137 μW	213.7 μW	9.02 mW	126.1 mW
Energy	397 μJ	137 μJ	213.7 μJ	9.02 mJ	126.1 mJ
EOEP	0.1723 nJ/pixel	18.35 μJ ^a	63.0426 μJ ^a	778.426 μJ ^a	5.460 μJ ^a
Target distance	10–90 cm	0.597 nJ/pixel ^a	2.052 nJ/pixel ^a	NA ^a	1.184 nJ/pixel ^a
Accuracy	93%~99%	≤60 cm	10–40 cm	20–40 cm	20–30 cm
Algorithm	FDTR or FCTR	90.6%	85%	4.3 mm	95%~99%
Background Programmable	Complex	Simple	Simple	Simple	Simple
	Yes	No	No	No	No

^aEnergy normalization factor is $Factor_{Energy} = (28_{nm}/65_{nm})^2 \times (V_{28}/V_{65})^2$.

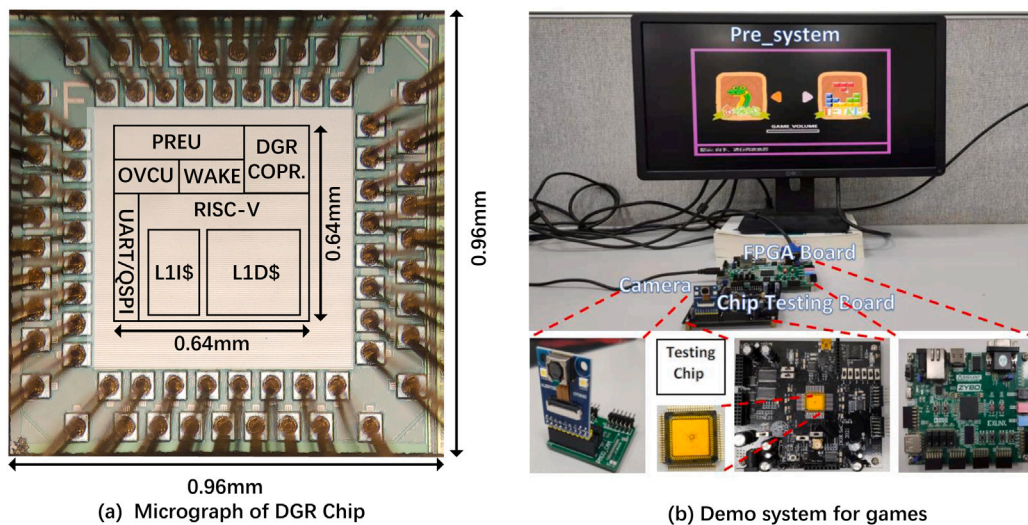


Fig. 10. Micrograph of DGR chip and test demo.

in both recognition and standby mode. In standby mode, the chip can further reduce power consumption and adopt a simple method to wake up to recognition mode. The chip realizes FDTR and FCTR algorithms which be applied to DGR under a variety of complex scenarios. The chip is fabricated in 28 nm CMOS process and demonstrates successful operations with real-time input from the OV5640 camera for a game controller. The power consumption of the chip in the recognition state is 397 μW and the power consumption of the chip in the standby state is 78.3 μW for 320 × 240 image at 0.584 V, the main clock frequency of 25 MHz and a frame rate of 30 FPS. The chip can be used in smartphones, IoT, wearable devices, game controllers with the features of high accuracy, low power, programmable, and portable.

CRediT authorship contribution statement

Yong-Liang Zhang: Conceptualization, Methodology, Software, Hardware design, Data analysis, Writing - original draft. **Qiang Li:** Chip back-end design, PCB, Chip test. **Hui Zhang:** Preprocessing module design, FPGA verification. **Wei-Zhen Wang:** Technical guidance, Project management. **Jun Han:** Funding acquisition, Supervision, Project management, Writing - review & editing. **Xiao-Yang Zeng:** Supervision, Project administration. **Xu Cheng:** Technical guidance, Project management.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2018YFB2202400) and the National Natural Science Foundation of China under Grant 61934002.

References

- [1] M. Yasen, S. Jusoh, A systematic review on hand gesture recognition techniques, challenges and applications, *PeerJ Comput. Sci.* 5 (2019) e218.
- [2] Z. Wang, Y. Hou, K. Jiang, W. Dou, C. Zhang, Z. Huang, Y. Guo, Hand gesture recognition based on active ultrasonic sensing of smartphone: a survey, *IEEE Access* 7 (2019) 111897–111922.
- [3] Y. Lu, T.T.-H. Kim, et al., A 184 μW real-time hand-gesture recognition system with hybrid tiny classifiers for smart wearable devices, in: *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, 2021, pp. 156–158.
- [4] T. Yoo, J.E. Kim, K.-H. Baek, T.T.-H. Kim, et al., A 213.7-μW gesture sensing system-on-chip with self-adaptive motion detection and noise-tolerant outermost-edge-based feature extraction in 65 nm, *IEEE Solid-State Circuits Lett.* 2 (2019) 123–126.

- [5] L. Lamberti, F. Camastra, Real-time hand gesture recognition using a color glove, in: International Conference on Image Analysis and Processing, 2011, pp. 365–373.
- [6] U. Côté-Allard, C.L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, B. Gosselin, Deep learning for electromyographic hand gesture signal classification using transfer learning, *IEEE Trans. Neural Syst. Rehabil. Eng.* 27 (2019) 760–771.
- [7] C. Liu, Y. Li, D. Ao, H. Tian, Spectrum-based hand gesture recognition using millimeter-wave radar parameter measurements, *IEEE Access* 7 (2019) 79147–79158.
- [8] N. Le Ba, S. Oh, D. Sylvester, T.T.-H. Kim, A 256 pixel, 21.6 μ W infrared gesture recognition processor for smart devices, *Microelectron. J.* 86 (2019) 49–56.
- [9] M. Van den Bergh, L. Van Gool, Combining RGB and ToF cameras for real-time 3D hand gesture interaction, in: 2011 IEEE Workshop on Applications of Computer Vision (WACV), 2011, pp. 66–72.
- [10] Z. Zhang, Microsoft kinect sensor and its effect, *IEEE Multimedia* 19 (2012) 4–10.
- [11] D.-S. Tran, N.-H. Ho, H.-J. Yang, S.-H. Kim, G.S. Lee, Real-time virtual mouse system using RGB-D images and fingertip detection, *Multimedia Tools Appl.* 80 (2021) 10473–10490.
- [12] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, C. Pan, Decoupled representation learning for skeleton-based gesture recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5751–5760.
- [13] T. Yoo, J.E. Kim, N. Le Ba, K.-H. Baek, T.T. Kim, et al., A 137- μ W area-efficient real-time gesture recognition system for smart wearable devices, in: 2018 IEEE Asian Solid-State Circuits Conference (a-SSCC), 2018, pp. 277–280.
- [14] S. Choi, J. Lee, K. Lee, H.-J. Yoo, A 9.02 mW CNN-stereo-based real-time 3D hand-gesture recognition processor for smart mobile devices, in: 2018 IEEE International Solid-State Circuits Conference-(ISSCC), 2018, pp. 220–222.
- [15] A. Waterman, K. Asanovi, The RISC-V Instruction Set Manual. Volume I: User-Level ISA, Document Version 20191213, RISC-V Foundation, 2019.
- [16] A. Waterman, K. Asanovi, The RISC-V Instruction Set Manual. Volume II: User-Level ISA, Document Version 20190608-Priv-MSU-Ratified, RISC-V Foundation, 2019.
- [17] K. Asanovic, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, et al., The Rocket Chip Generator, Tech. Rep. UCB/EECS-2016-17, EECS Department, University of California, Berkeley, 2016.
- [18] Y. Lee, B. Zimmer, A. Waterman, A. Puggelli, J. Kwak, R. Jevtic, B. Keller, S. Bailey, M. Blagojevic, P.-F. Chiu, et al., Raven: A 28nm RISC-V vector processor with integrated switched-capacitor DC-DC converters and adaptive clocking, in: 2015 IEEE Hot Chips 27 Symposium (HCS), 2015, pp. 1–45.
- [19] T. Yoo, J.E. Kim, N. Le Ba, K.-H. Baek, T.T.-H. Kim, et al., A 137- μ W 1.78- mm^2 30-frames/s real-time gesture recognition SoC for smart devices, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 28 (2020) 1909–1919.
- [20] S. Park, S. Choi, J. Lee, M. Kim, J. Park, H.-J. Yoo, A 126.1 mW real-time natural UI/UX processor with embedded deep-learning core for low-power smart glasses, in: 2016 IEEE International Solid-State Circuits Conference (ISSCC), 2016, pp. 254–255.