Microelectronics Journal 86 (2019) 49–56

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

A 256 pixel, 21.6 μW infrared gesture recognition processor for smart devices

Ngoc Le Ba [a,*], Sechang Oh [b], Dennis Sylvester [b], Tony Tae-Hyoung Kim [a]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore, 639798, Singapore

[b] Department of Electrical Engineering and Computer Science, University of Michigan, 1301 Beal Ave., 2417C EECS, Ann Arbor, MI 48109-2122, USA

ARTICLE INFO

Keywords:
Gesture recognition
Passive infrared

ABSTRACT

This paper presents a low-power digital signal processor for gesture detection using infrared (IR) sensors. IR compatible detecting algorithms are introduced to reduce power consumption and provide accurate detections for various types of gestures including sweeping in 8 directions, zooming in/out and a unique wake-up gesture. To further reduce the power, the proposed algorithm is capable of setting the processor into idle mode when there is no motion and resume active mode upon wake-up request. Test chips were fabricated in a 65 nm commercial TSMC process and occupy an area of 580 μm × 580 μm. Measurement results taking input from a 16 × 16 Heiman IR sensor array show that the proposed gesture recognition processor is able to detect correct gestures at 100 MHz maximum clock frequency. Under normal working condition, the processor consumes 21.6 μW in active mode and 7.2 μW in idle mode while processing 8-bit 16 × 16 input data array at 30 fps.

1. Introduction

Interaction between human and machine becomes more and more popular with fast evolution in technology. Smart devices are invented to interpret human's instructions through face or hand gestures in which many gesture recognition systems have been reported, such as remote gaming devices, remote controllers and wired gloves. These implementations are mainly realized by using vision image sensors, RF sensors, ultra sound sensors, and active infrared (IR) sensors. In the case of image sensors, high resolution images are captured and software analyzes pictures to get information about the gesture. Kinect [1] is the most popular device which was developed by Microsoft using image sensing technique. It includes an infrared camera to measure distance and reconstruct 3D structure of an object together with a VGA (640 × 480) video camera to produce 30 frames per second image stream. At relatively low cost, Kinect has become very attractive tools for gesture detection in vision based domain. In Refs. [2,3], hand gesture recognition systems using Kinect sensor were proposed to interpret 3D hand movement and language signs for Portuguese and Japanese respectively. With YCbCr or RGB color space, objects such as hand, face, or human body can be differentiated to make decision about gestures more accurately and have a wide range of complex detectable gestures [4]. However, image sensor fails to work in absence of light and it requires large memory and power consumption as for its drawbacks.

Another popular sensing technique is RF sensing. By transmitting an RF signal which travels toward the object and reflects back to the receiver, object's information can be obtained by deriving the distance from the signal's time-of-fly (TOF). Since RF signal works without light, the sensor is able to perform under no light condition. This technique is widely used for movement detection where object's appearance or object's absence is interpreted by interruptions of RF signal streams. For gesture recognition, low resolution is the biggest obstacle to utilize this technique. Nevertheless, attempts to use RF sensor for gesture sensing have increased recently. With the aid of accelerometer, a system consists of RF sensors and accelerometers [5] is able to detect 3D hand moments. In Ref. [6], a group of MIT researchers proposed a design of wearable hand-held interface using passive RFID sensor tags. The design was successfully demonstrated with sensors attached to user's hand to control a computer. Another significant progress in RF sensing is the work reported in Ref. [7], a single scalable RF transmitter/receiver cell that can be constructed into a large array for 3D imaging consumes only 1.4 W and achieves working range from 20 cm to 30 cm.

Time-Of-Fly (TOF) principle is also applied for ultrasound sensing. In Refs. [8,9], a circuit for range finder using an ultrasound sensor is able to calculate the distance from the object to the sensor by measuring the TOF of ultrasound pulses. By using an array of ultrasound range-finder

* Corresponding author.
E-mail address: leba0002@e.ntu.edu.sg (N. Le Ba).

https://doi.org/10.1016/j.mejo.2019.02.016
Received 31 October 2018; Received in revised form 12 January 2019; Accepted 22 February 2019
Available online 25 February 2019

sensors, object's shape and its movement can be detected. However, the power consumption of these systems is still quite large in sub-mW range and not suitable for portable or mobile applications. To save the power, passive infrared (PIR) sensing is an alternative option. In Ref. [10] to verify feasibility of using PIR for hand gesture recognition, a completed system integrating 16 PIR Pyreos sensors which requires 500 times less power than active elements [11] and a signal processor was implemented. The final design is able to detect hand movements in 4 directions (Up, Down, Left, Right) at 20 cm distance.

In this work, we present a hand gesture recognition processor using a 16 × 16 Heiman IR sensor array to detect a wide range of gestures. Motion History Image (MHI) technique is employed for sweeping detection in 8 directions. Special detecting algorithms are used to detect zooming actions and a unique wake-up gesture to reactivate the systems from its idle mode. With the idle mode feature and algorithms that require simple mathematic operations, our proposed SOC is able to bring the total power consumption down to the order of a few tenths microwatt and reduce the system complexity to make the design more suitable for mobile applications such as smart phones or smart watches. A smaller system (16 × 4) using our own front-end design was demonstrated in Ref. [12] to show the scalability of our proposed algorithms.

2. The proposed gesture detection processor

2.1. Sensing technique

In our previous work [12], a completed SOC using a 16 × 4 thermopile array is used to capture infrared images. Each thermopile signal connects to an AFE path that consists of an instrumentation chopper amplifiers (ICA) and low pass filters (LPF). The four row ADCs digitize the amplified/filtered signals using time-division multiplexing and the processor analyzes the waveform to detect gestures. The gesture detection processor in this system is scaled down and limited to basic gestures (motion detection and sweeping detection) due to low resolution input. To improve sensing capability in this work, we utilize a HTPA16 × 16d infrared array sensor from Heimann Sensor GMBH to capture infrared images then our detecting algorithm is used to analyse these gestures. The images from the Heimann sensor arrays has 16 × 16 pixels as shown in Fig. 1. With better resolution image, we are able to demonstrate capabilities of our detecting algorithm for motion detection, sweeping detection, zooming detection and a special wake-up gesture to resume the system from sleeping mode and to avoid users from accidentally trigger the system.

2.2. The Processor's block diagram and detecting algorithms architecture

Our proposed design for the processor's system block diagram is shown in Fig. 2. Input integer data which represent pixels' temperature in Kelvin are taken directly from Heiman sensor array and quantized into 8 bit data (DIN). These data are written into three memories by a Write Circuit inside the Interface block. The first memory is to keep motion history image (MHI) which are the difference between two continuous frames. When a motion is detected by motion detecting algorithm the next two frames will be stored in the 2nd and 3rd memory. The write Circuit is controlled by a control circuit which is the control hub to communicate with other sensing modules. The control circuit is also responsible for multiplexing data to each of the sensing module. When a request signal is sent from a sensing module, the control circuit will fetch the data from memories to the requesting module. Analysis will be performed in each module and feedback to the control circuit if any other module needs to be activated. Finally, the control circuit will gather data from each module to make decisions about the gesture.

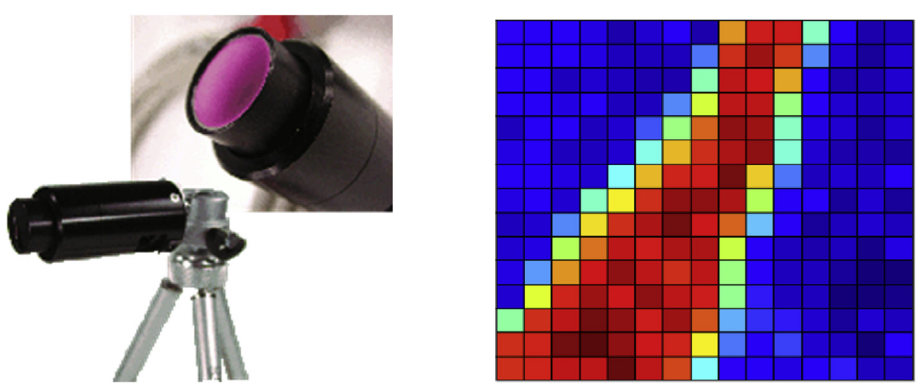

Fig. 1. A thermograph captured by the Heiman HTPA 16 × 16d sensor array.

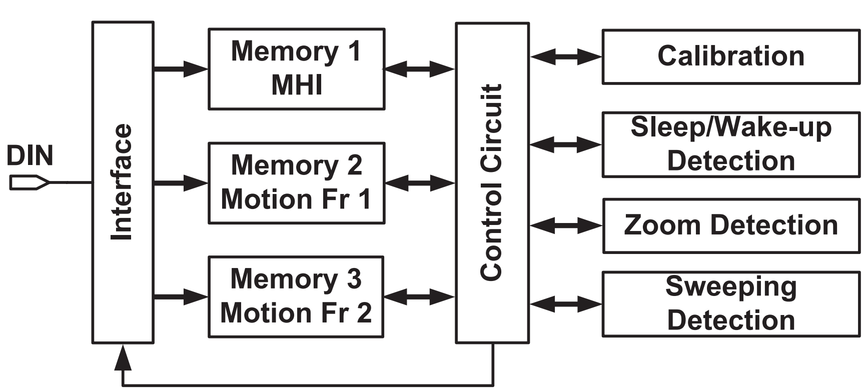

Fig. 2. Block diagram of our proposed gesture recognition processor. Interface circuit consists of write circuits which are controlled by the control circuit to write 8-bit data into the three memories.

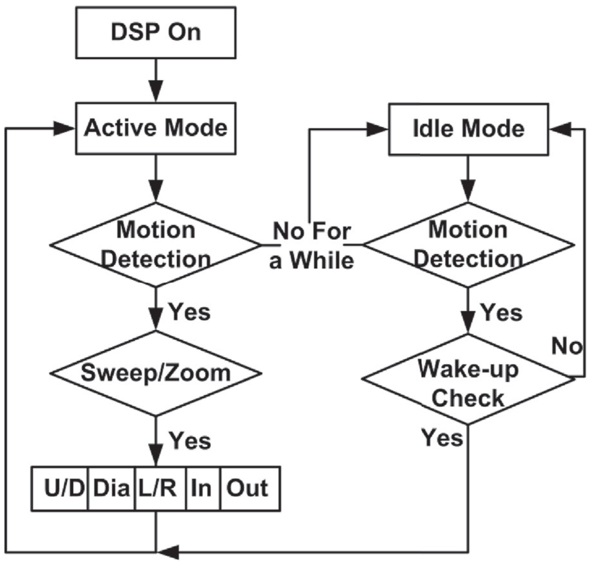

Fig. 3. Top level design of detecting algorithms. Note that there are total 8 sweeping directions: 4 movements up/down, left/right in straight directions and 4 movements in diagonal directions.

Targeting for ultra-low power applications, algorithm complexity has to be minimized to achieve fast response time with least hardware requirements. The proposed algorithms is designed to require only addition, subtraction and shifting operations. Multiplication and division operations with constants are designed to be with power of 2 numbers which can be implemented by shift operations together with addition and subtraction. Therefore, the implementation only uses adder, subtractor and shifters while it stills ensure high detection accuracy for motion detection, sweeping and zooming detection with a distance up to 30 cm. For detecting wake-up gestures, it requires a shorter distance of below 20 cm as it needs to analyse the image's content and shorter distance also limits users from accidentally waking up the device by some unknown movements. In Fig. 3, our proposed gesture detection algorithm is implemented with two working modes. In standby mode, it disables half of its pixels and reduces the frame rate. This mode is triggered when there is no detected motion for a pre-set duration (e.g. 10 s). During stand-by, the processor only senses for motions then analyse if it is a wake-up gesture. If a wake-up gesture is found, active mode will be recovered then the processor will turn on all the pixels and increase the frame rate. In active mode, the processor is able to detect sweeping actions in 8 directions (Up, Down, Left, Right and 4 diagonal directions) and zooming actions as shown in Fig. 4.

2.3. Calibration and motion detection

Fig. 5 shows the block diagram of the calibration circuit which calculates the reference temperature value when there is no motion. This circuit is activated at power on events and calibrates for 64 frames which are required to be the background images (without any object's presence). Calibration algorithm subtracts the current input from the previous input and updates the maximum pixel's difference reading. Input noise is cancelled out by taking the difference of input therefore clearer images can be obtained. When the counter reaches 64 frames, the maximum value will be stored as a calibrated reference value as shown in Eq. (1). A pixel whose MHI's reading higher than the reference temperature is considered as having motions. The motion detecting algorithm will count these pixels. If the number of pixels having motion is greater than a preset value then the difference is significant enough to be considered as a movement. With a hand moving into the sensor's focus, the number of pixels having temperature value higher than the reference value (hot pixels) keeps increasing and reach the threshold for motion detection. The motion detection circuit employs the formula in Eq. (2) to calculate object's temperature and triggers a signal to inform the control circuit about its detected movement for further processing. In the event there is no movement for a period of time, the processor will be put into sleep mode which disables all the blocks but motion detection to save

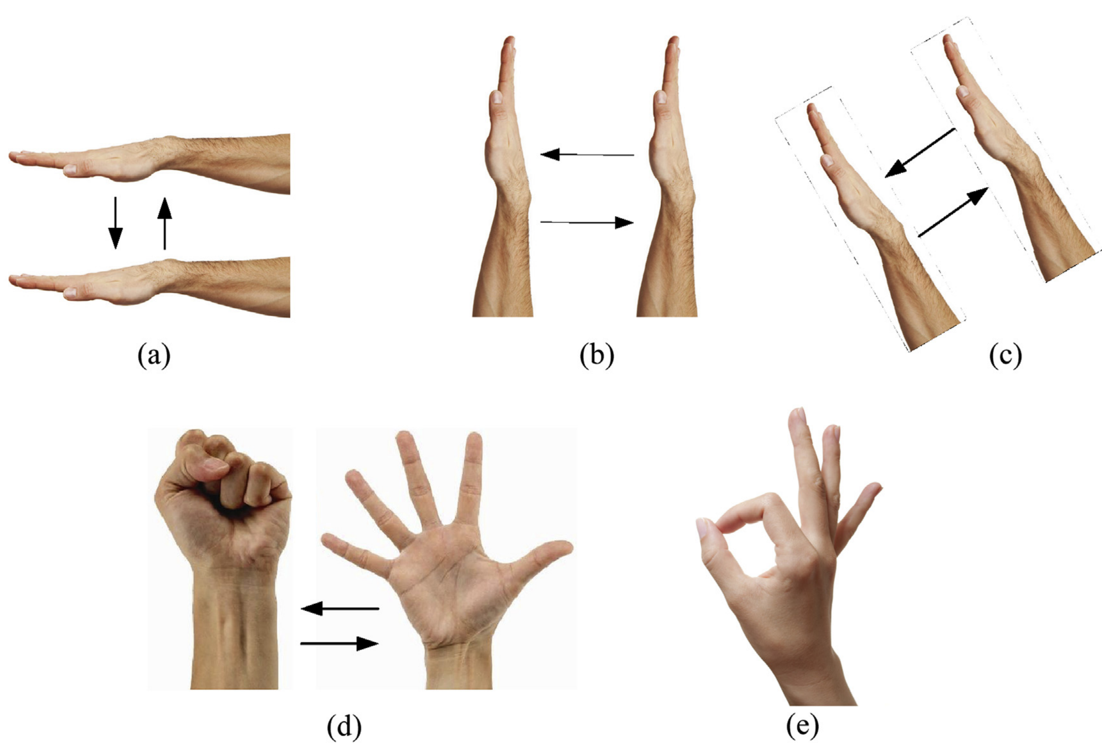

Fig. 4. The list of recognizable gestures (a) up-down (b) left-right (c) diagonal sweeping (c) zooming in/out (d) a unique wake-up gesture.

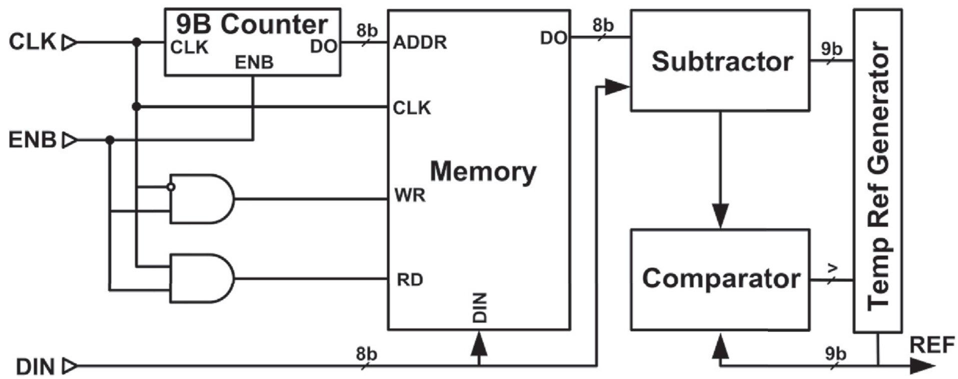

Fig. 5. Circuit block diagram for calibration module. Note that write and read signal are performed within a half clock cycle.

power until a movement is found. When there is a movement, other gesture detecting modules are activated to determine the gesture. The calibration threshold for number of having motion pixels are set based on multiple experiments and it is reconfigurable to provide users with options to adjust the sensitivity level.

$$Ref\ Value = \underset{\substack{all\ pixels \\ i=1:64}}{\text{Max}} (frame_{i+1} - frame_i) \tag{1}$$

$$Object\ Value = 0.75{*}Max\{X\}\ where\ X = \{hot\ pixels's\ value\} \tag{2}$$

$$\begin{aligned} &Zoom\ In\ if \begin{cases} x_{i,j+3} > x_{i,j+2} > x_{i,j+1} \\ y_{i,j+3} > y_{i,j+2} > y_{i,j+1} \end{cases} \\ &Zoom\ Out\ if \begin{cases} x_{i,j+3} < x_{i,j+2} < x_{i,j+1} \\ y_{i,j+3} < y_{i,j+2} < y_{i,j+1} \end{cases} \end{aligned} \tag{3}$$

where $x_{i,j+k}$ is the ith corner of frame-k, k = 1,2,3 (3).

2.4. Wake-up detection

Wake-up gesture has to be unique so that it is not easy to be accidentally triggered however it must be simple to be recognized in low resolution mode using less complex algorithms to optimize for power consumption. With these constraints, the gesture shown in Fig. 6(a) is selected to be our wake-up gesture. To detect this pattern, the algorithm looks for any closed loop created by a chain of continuous hot pixels which have pixel value higher than the reference temperature. Fig. 6(b) shows the operations to detect this wake-up gesture.

2.5. Zooming detection

When a motion is detected, there will be a sequence of object's location analysis performed by the processor as shown in Fig. 7. First, zooming detection module informs the control circuit to fetch the three most recent frames together with reference temperature value to the coordinate calculation circuit. This calculation circuit finds object's

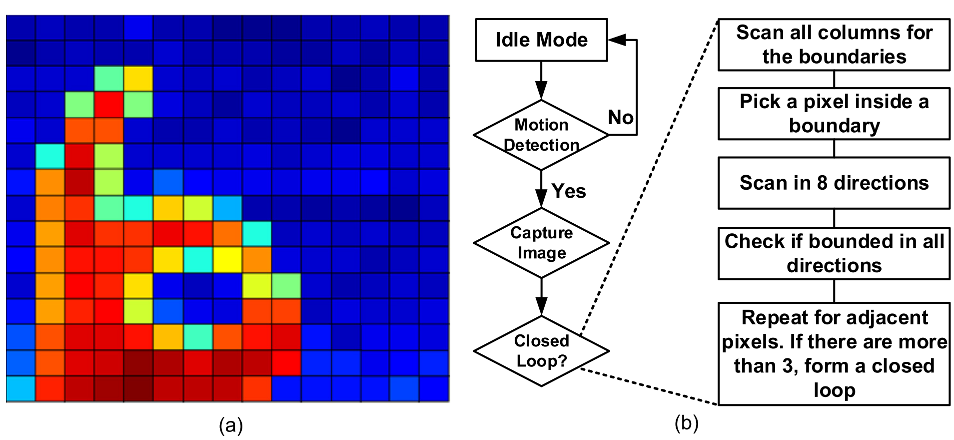

Fig. 6. (a) A unique wake-up gesture (b) Wake-up detecting mechanism.

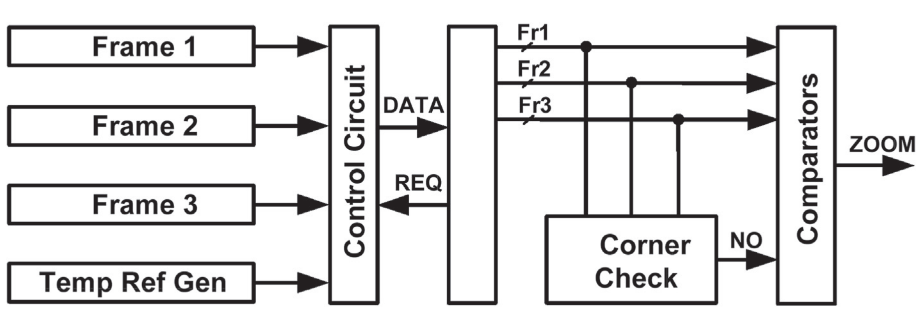

Fig. 7. Zooming detection block diagram which consists of coordinate calculation circuit, corner checking circuit and a comparator bank.

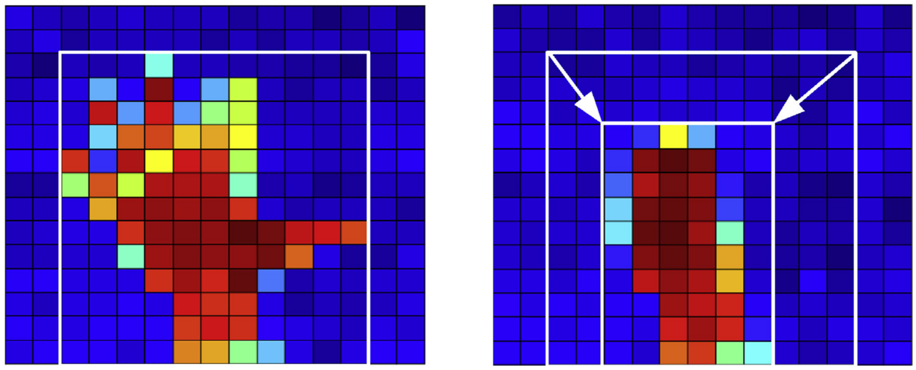

Fig. 8. A zooming action. Note that in the next frame, object shrinks inside it previous locations.

location in each frame using object temperature recorded by Eq. (2) and determines the 4 boundary coordinates of the object. It then passes the 4 sets of object's boundary coordinate to the corner checking circuit which compares if any coordinate has the value that equals to one of four corners: (0,0), (0,16), (16,0) and (16,16). If it is not at the corner then the comparator circuit starts comparing coordinates of the three frames. Once ascending or descending order is formed by the coordinates from frame 1 to frame 3 following the logic in Eg. 3 then there is a zooming in or zooming out action. Otherwise, it is another type of gesture and it will be analyzed by other modules. Fig. 8 demonstrates a zooming-in action which has the object boundary shrinking with time. Our proposed zooming detection has a limitation in interpreting zooming at the corner and when it happens the algorithm is only able to tell there is a movement at the corner but not the correct gesture. This is because when the user sweeps across the corners, the ascending or descending order will be formed and it is misinterpreted as a zooming action. Therefore, corner checking is introduced to avoid this situation however it also limits the ability to recognize gestures at corners.

2.6. Sweeping detection

When a motion is not classified as a zooming action, sweeping detection module will check if it is a sweeping action. In this detecting mode, motion history image (MHI) technique as shown in Eg.4 is employed. By taking difference of two continuous frames, MHI is able to record both object's locations in the two frames. Fig. 10 demonstrates operations of the sweep detecting algorithm. Two MHIs are obtained upon requesting data from the 3 memories then the processor takes average of all the pixels in every rows and in every columns following the formulas in Eq. (5). These numbers form two distributions for rows' average pixel values and columns' average pixel values as shown in Fig. 10. For vertical sweeps, the column pixels of MHI cancel each other

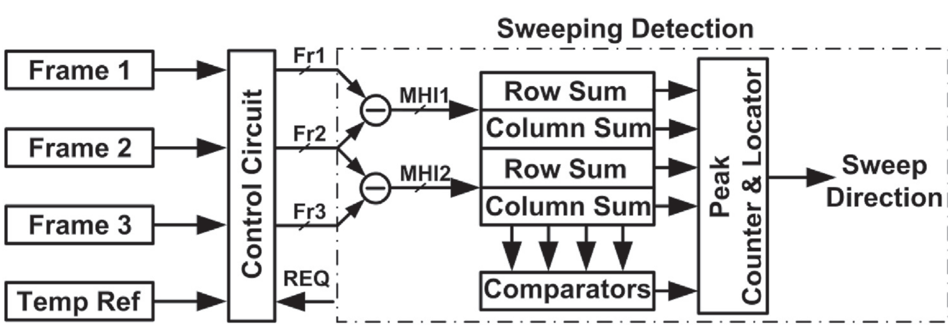

Fig. 9. Sweeping detection module. Sweeping gestures are detectable in eight directions.

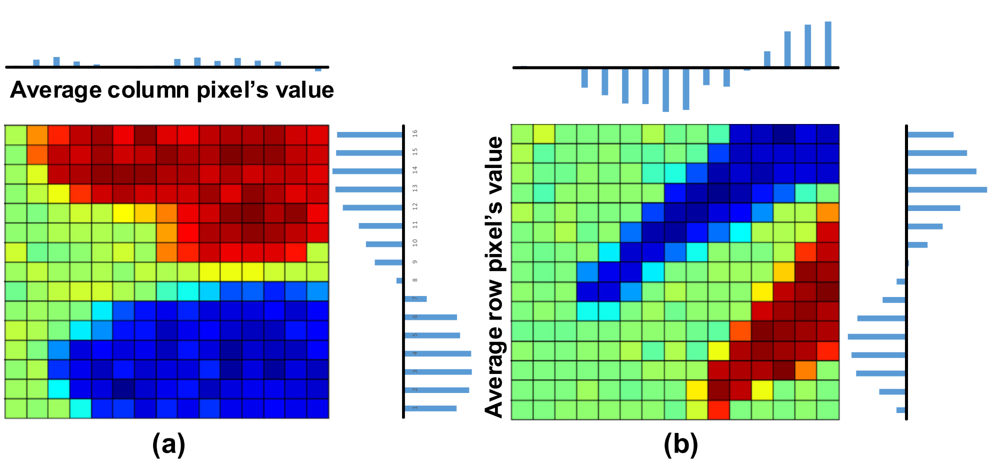

Fig. 10. (a) A vertical sweep (b) a diagonal sweep. Note that the peaks in column sum and row sums will tell the sweeping direction.

in a summation and generate no clear peak values as shown in Fig. 10(a) while the row pixels of the same MHI are magnified in the summation therefore generate two clear peak values. Similar analysis is applied for diagonal sweeps in which overlapping area in both vertical and horizontal directions is cancelled out while non-overlapping area is magnified. This results in two clear peak values in each row average and column average as shown in Fig. 10 (b). In order to make a decision about the gesture, peak counter and locator circuit is used in Fig. 9 to count the number of peaks and locate the peaks' positions in two MHI frames. By looking at the number of peak values and their relative positions to each other from the two MHI frames, direction of sweeping can be accurately concluded.

In this proposed sweeps detecting algorithm, MHI helps to zero out the intrinsic noise in each pixel and magnify the differences when there is a movement. Row averaging and column averaging technique additionally amplify the difference between the previous object's position and its current position. Therefore, applying row and column averaging on MHI frames generates clear detecting signals and accurate results while re-

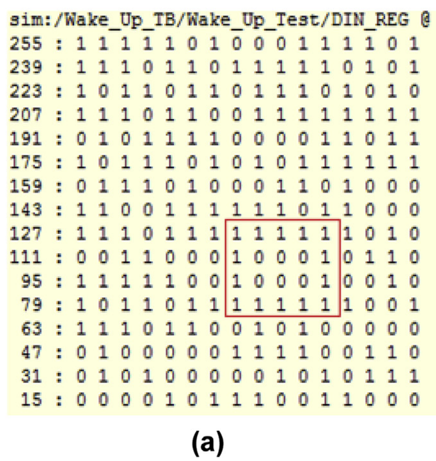

Fig. 11. (a) Test image for a wake-up gesture with a closed loop highlighted in red (b) output detection with 3 adjacent detected pixels to indicate a closed loop.

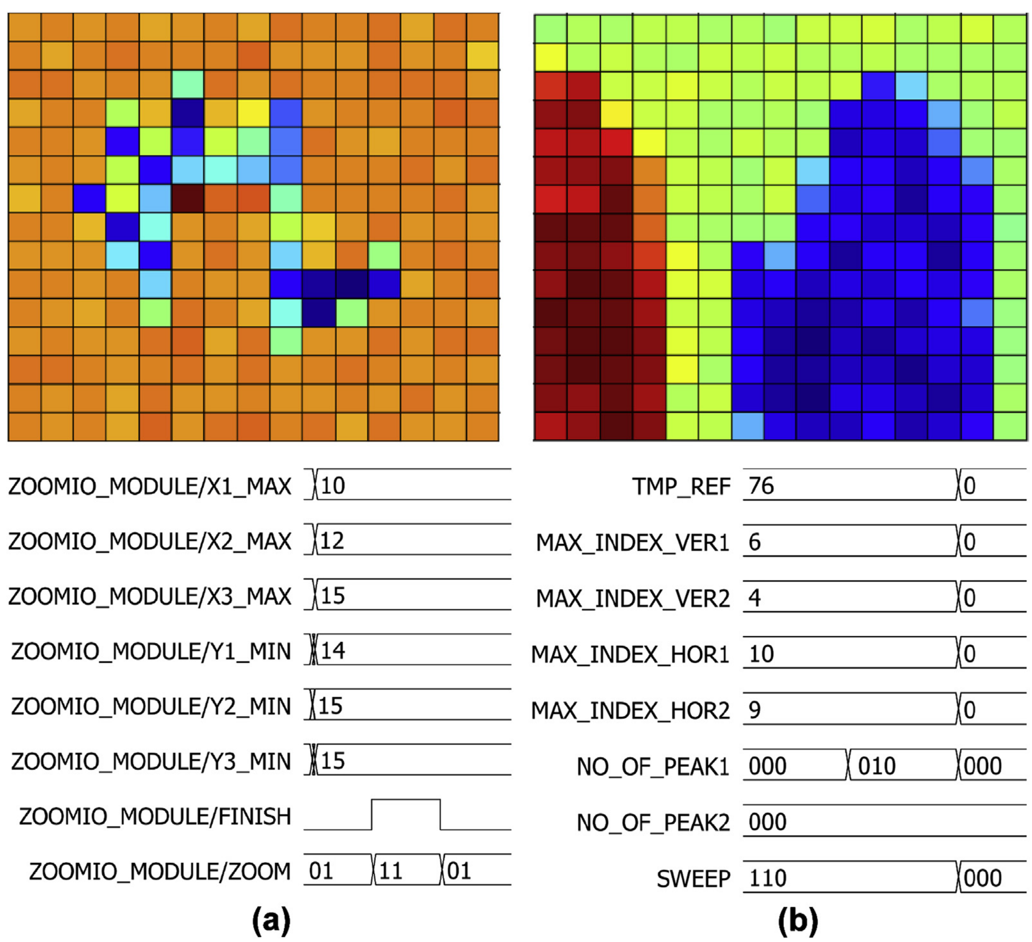

Fig. 12. (a) Output test results for zooming detection, zoom = 11 indicates zooming out (b) output test results for sweeping detection, sweep = 110 indicate a right to left sweep with 2 peaks in total. Note that in MHI image, blue color indicates object's previous position and red color indicates object's current position.

quires minimum hardware cost. The circuit is implemented in Verilog with only addition, subtraction and comparators to reduce the complexity in order to increase processing speed and reduce power consumption. From simulations, the circuit is able to work reliably at maximum clock frequency of 100 MHz.

$$MHI(f,x,y) = Fr(f+1,x,y) - Fr(f,x,y) \tag{4}$$

where $MHI(f,x,y)$ denotes MHI value of the pixel located at (x,y) in frame f and $Fr(f,x,y)$ is the actual value of the pixel (x,y) in frame f.

$$Avg_i = \frac{\sum_{j=1}^{N} pixel(i,j)}{N}, \quad Avg_j = \frac{\sum_{i=1}^{N} pixel(i,j)}{N} \tag{5}$$

where Avg_i is the average pixel values in the ith row, Avg_j is the average pixel values in the jth column and N is the number of pixel in each row or column.

3. Implementation and experimental results

To verify the DSP's functionality, the proposed gesture recognition algorithms were firstly verified in Matlab using the captured gesture frame data from Heiman sensors. Various basic hand gestures were recorded and the proposed algorithms in Matlab analyse the frame data and generate recognition results. Upon successfully verifying the algorithms, we implemented them in a hardware form using Verilog. For hardware testing, the previously captured gesture data are stored in a logic analyzer and transferred to the implemented gesture recognition DSP. The output of the DSP is sent back to the logic analyzer and compared with the original gestures for accuracy checking.

To test our proposed gesture detection processor's operations, a wake-up pattern is sent to the input. Fig. 11(a) shows a wake-up gesture with a closed loop highlighted in red and output response is shown in Fig. 11(b). The response image taken from processor's memory has three adjacent low temperature pixels surrounded by a closed loop of high temperature pixels indicating the wake-up pattern's location which is correctly detected by the algorithm. After putting the system into active mode, sweeping and zooming gestures are performed in front of Heimann sensors to obtain digital input for the detecting algorithms. Fig. 12(a) demonstrates a zooming in detection which has an output response of “11”, for zooming out the response is “00” and “01” is the default mode. This digital pattern is chosen so that circuit only needs to toggle one bit for any detection. For sweeping detection, a three bit output and number of peaks are used to indicate the directions. In Fig. 12(b), a horizontal sweep is performed therefore there are total 2 peaks and sweep output is “110” which is encoded for right to left direction.

The recognition rate is then evaluated for sweeping actions. Fig. 13 (a) presents a MHI code generated from a Right-Left sweep, the code clearly shows two peaks in the MHI frame as expected. A series of Right-Left sweeps were performed at different speeds and distances from the sensors. All the movements are recorded in video format and processed with frame-cutting software provided by Heimann. The number of frames having Right-Left sweep is counted and their indexes are recorded to compare with the results obtained from the processor. Fig. 13 (b) shows the detection rate of our proposed algorithm. At very near distance, 10 cm from the sensor it is more suitable for slow motions as the object become bigger than itself at further distance. From 20 cm to 30 cm, the gesture detection processor has an accuracy rate of 96% at its optimal speed which generates 12 frames across the covered area of the sensor's lens. For very fast motions which produced less than 4 frames crossing the sensor's focus, there is not enough data to process therefore it does not generate any recognition results and we excluded them from the counting.

The test chip was fabricated in 65 nm TSMC CMOS process and occupies an area of 580 µm × 580 µm. The micrograph of our proposed design is shown in Fig. 14. In this tape-out, we fabricated two versions of the processor for 16 × 4 input data and 16 × 16 input data. Measurement results show that processor is able to recognize gestures correctly. With 12fps data input taken from 16 × 16 Heimann sensor array, the processor consumes 21.6 µW and 7.2 µW in active and idle mode respectively at V_{DD} = 0.5 V. In comparison with previous state-of-the art gesture detections as shown in Table 1, this work presents the first SOC for gesture sensing applications using passive IR sensors. With low power supply and

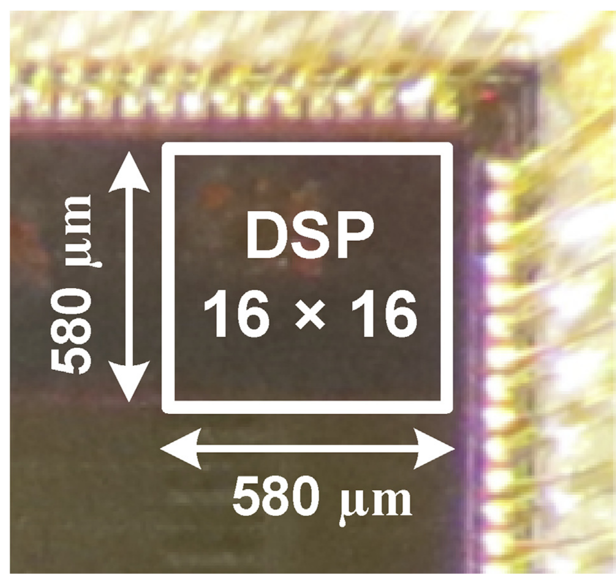

Fig. 14. Micrograph of the gesture recognition SOC. Note that PROCESSOR area contains two version for 16 × 4 input data and 16 × 16 input data.

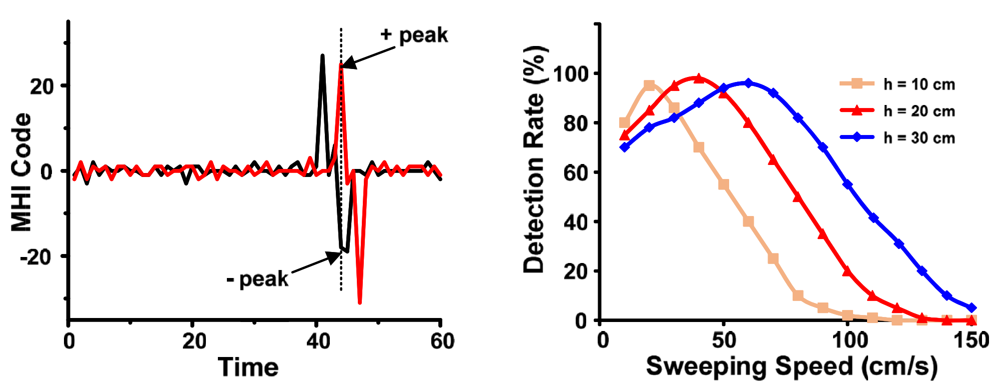

Fig. 13. (a) MHI code generated from a Right-Left Sweep (b) detection accuracy for Right-Left Sweeps at different sweeping speed measure by number of frames passing the focus area and at different object's distance to the sensor.

Table 1
Comparison with other state-of-the-art gesture recognition works.

Specifications	This work	JSSC '13 [7]	Pyreos [11]	Robio '17 [13]	JSSC'14 [14]	ISSCC'18 [15]
System Type	SOC	Only Sensor	PCB	FPGA	Only Sensor	Only Sensor
Sensor type	Passive IR	RF	Passive IR	Image	Image sensor	Stereo Camera
Area	580 μm × 580 μm	4 mm × 5 mm	1.5 mm × 1.5 mm	N.A	9.8 mm × 8.4 mm	4 mm × 4 mm
Resolution	16 × 16	400 × 384	2 × 2	640 × 480	256 × 256	
Frame rate (fps)	10–30	200	72	60	1000	33.3
Voltage Supply	1.2 V–1.8 V	1.8 V	3.2 V	N.A	1.8 V	1.2 V
Power Consumption	21.6 μW	140 mW	46.2 μW (only for sensors)	N.A	630 mW	9.02 mW

low power consumption features, the design is the best fit for emerging smart devices and mobile applications.

4. Conclusions

A low-power hand gesture recognition processor is presented for mobile and smart device applications. Using IR sensor array as input, the processor is able to recognize a wide range of gestures which include sweeping in 8 directions, zooming in/out and a special wake-up gesture. By integrating different modes (active and idle) of operations and efficient detecting algorithms, the proposed design is able to operate at low power, low voltage conditions which are very attractive for emerging smart devices such as mobile phones or smart watches. The processor was fabricated in 65 nm CMOS process and test chips demonstrated successful operations with real time input from a customized analog design as well as pre-processed digital input from a commercial Heimann IR sensor array. The system successfully demonstrates the ability to recognize sweeping, zooming and wake-up gestures using 21.6 μW for 16 × 16 IR images at 1.8 V and a frame rate of 12 fps. This feature makes the design very suitable for emerging smart devices such as wearable devices, smartwatches, etc.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.mejo.2019.02.016.

References

[1] Microsoft. (2014). Available: http://www.xbox.com/en-sg/.
[2] P. Trindade, J. Lobo, and J. P. Barreto, "Hand gesture recognition using color and depth images enhanced with hand angular pose data," presented at the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI).
[3] L. Jaemin, H. Takimoto, H. Yamauchi, A. Kanagawa, Y. Mitsukura, A robust gesture recognition based on depth data, presented at the, in: The 19th Korea -Japan Joint Workshop on Frontiers of Computer Vision, 2013.
[4] J. Suarez, R.R. Murphy, Hand gesture recognition with depth images: a review, in: IEEE International Symposium on Robot and Human Interactive Communication, 2012.
[5] B. Mrazovac, M. Z. Bjelica, D. Simic, S. Tikvic, and I. Papp, "Gesture based hardware interface for RF lighting control," presented at the IEEE 9th International Symposium on Intelligent Systems and Informatics.
[6] R. Bainbridge, J.A. Paradiso, Wireless Hand Gesture Capture through Wearable Passive Tag Sensing, 2011.
[7] A. Arbabian, S. Callender, S. Kang, M. Rangwala, A.M. Niknejad, A 94 GHz mm-Wave-to-Baseband pulsed-radar transceiver with applications in imaging and gesture recognition, IEEE J. Solid State Circuits 48 (2013).
[8] R.J. Przybyla, S.E. Shelton, A. Guedes, R. Krigel, D.A. Horsley, B.E. Boser, In-air rangefinding with an AlN piezoelectric micromachined ultrasound transducer, IEEE Sens. J. 11 (July 2011).
[9] R. Przybyla, A. Flynn, V. Jain, S. Shelton, A. Guedes, I. Izyumin, et al., Micromechanical ultrasonic distance sensor with >1 meter range, presented at the, in: Proc. Solid-State Sensors, Actuators and Microsystems Conference, 2011.
[10] P. Wojtczuk, A. Armitagey, T.D. Binnie, T. Chamberlainz, PIR sensor array for hand motion recognition, presented at the, in: The Second International Conference on Sensor Device Technologies and Applications, 2011.
[11] Pyreos Ltd: Touchless Gesture Control. Available: http://www.pyreos.com/.
[12] S. Oh, N.L. Ba, S. Bang, J. Jeong, D. Blaauw, T. Kim, et al., A 260μW infrared gesture recognition system-on-chip for smart devices. Presented at the Symposium on VLSI Circuits (SOVC), June 2016.
[13] W. Zhou, C. Lyu, X. Jiang, Real-time implementation of vision-based unmarked static hand gesture recognition with neural networks based on FPGAs, presented at the, in: IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, China, 2017.
[14] C. Shi, et al., A 1000 fps vision chip based on a dynamically reconfigurable hybrid architecture comprising a PE array processor and self-organizing map neural network, IEEE J. Solid State Circuits 49 (9) (Sept. 2014) 2067–2082.
[15] S. Choi, J. Lee, K. Lee, H.-J. Yoo, A 9.02mW CNN-Stereo-Based real-time 3D hand-gesture recognition processor for smart mobile devices, in: ISSCC dig. Tech. Papers, 2018, pp. 220–221.