An FPGA Implementation of Intelligent Visual Based Fall Detection

Peng Shen Ong, Yoong Choon Chang, Chee Pun Ooi, Ettikan K. Karuppiah, and Shahirina Mohd Tahir

Abstract**—**Falling has been one of the major concerns and threats to the independence of the elderly in their daily lives. With the worldwide significant growth of the aging population, it is essential to have a promising solution of fall detection which is able to operate at high accuracy in real-time and supports large scale implementation using multiple cameras. Field Programmable Gate Array (FPGA) is a highly promising tool to be used as a hardware accelerator in many emerging embedded vision based system. Thus, it is the main objective of this paper to present an FPGA-based solution of visual based fall detection to meet stringent real-time requirements with high accuracy. The hardware architecture of visual based fall detection which utilizes the pixel locality to reduce memory accesses is proposed. By exploiting the parallel and pipeline architecture of FPGA, our hardware implementation of visual based fall detection using FGPA is able to achieve a performance of 60fps for a series of video analytical functions at VGA resolutions (640x480). The results of this work show that FPGA has great potentials and impacts in enabling large scale vision system in the future healthcare industry due to its flexibility and scalability.

Keywords**—**Fall detection, FPGA, hardware implementation.

I. Introduction

FALLING is one of the greatest obstacles for the elderly to live independently. It is also one of the primary causes to the senior's injury. It is reported that 28-35% from the group of elderly age 65-75 years and 32-42% from the group of elderly age 75 years and above falls at least once a year [1]. With the fast growing senior population in the world, including Malaysia, which is predicted to increase to 10% in the year 2020 [2], the impact of the elderly population on the healthcare industry is causing for concerns. The elderly have nine times greater chance of suffering fall-related injury as compared to those less than 65 years of age [3]. Other than the physical aspect, falling leaves severe psychological effect on the elderly such as loss of confidence and curtailment of their daily activities. As a result, it is crucial to have an automated fall detection system to provide immediate help to the victims so that post-fall injury and fatal cases due to delayed assistance can be avoided. Besides that, automated fall detector is more cost efficient compared to deploying full-time medical personnel or caregiver in monitoring the elderly's activities of daily living.

Over the years, while different types of fall detecting system had been introduced in the commercial market, visual based fall detection has slowly grown in favorability due to its non-intrusive behavior as compared to the others worn-based fall detector. Moreover, low cost surveillance camera can now be easily obtained. Multiple deployments of video camera are able to monitor a large area of compound like nursing home and hospital and solve the problem of large quantity of sensors required if worn-based fall detector is being used. Additionally, it overcomes the weaknesses of worn-based fall detector such as the elderly might tend to forget wearing it and false alarms might be generated when abrupt acceleration occurred due to the accelerometer based sensor. With the relatively equal level of accuracy, visual based fall detection has shown numerous advantages over the conventional fall detector system.

This paper is organized as follows. Section II discusses the related work of visual based fall detection while Section III discusses the fall detection algorithm implemented in this work. The hardware implementation will be discussed in Section IV. Lastly, the results and conclusions are presented in Section V and VI respectively.

II. Related Work

Various works have been done on visual based fall detection to improve the accuracy and to reduce the complexity of the algorithm. The most common technique used in visual based fall detection is the bounding box method [4, 5] which has low complexity. Somehow, there are limitations whereby it can only work with cameras that are placed sideways and it cannot differentiate between genuine fall and fall-like activities. Successfully, the eclipse method is introduced [6, 7] next to solve the problems of bounding box technique and it demonstrates significant improvements over the bounding box technique. However, activities like sitting down brutally and squatting down brutally are still being recognized as falls, in addition to its higher computational complexity. In [8], Andy et al. introduced a 3-points human shape representation to detect fall of the human object. The proposed solution achieved to reduce the complexity while maintaining a comparable accuracy to the prior works.

Visual based fall detection requires robust moving object detection and tracking which are often computationally expensive and requires high processing power to perform in real-time. According to the best understanding of the authors,

Peng ShenOng is withFaculty of Engineering, Multimedia University, 63100Cyberjaya, Malaysia (email: ps.ong87@gmail.com).

YoongChoon Chang is withFaculty of Engineering, Multimedia University, 63100Cyberjaya, Malaysia (email: ycchang@mmu.edu.my).

Chee Pun Ooi is with Faculty of Engineering, Multimedia University, 63100Cyberjaya, Malaysia (email: cpooi@mmu.edu.my).

Ettikan KKaruppiah is with Information and Communication Technology, MIMOS Bhd, Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia (email: ettikan.karuppiah@mimos.my).

ShahirinaMohdTahiris with Information and Communication Technology, MIMOS Bhd, Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia (email: shahirina.mtahir@mimos.my).

visual based fall detection in the literature [4-8] is implemented in software based solution using programming language like C++, openCV and Matlab which runs on general x86 or x64 processors. While these approaches offer easy prototyping, the performance is limited due to the nature of the sequential behavior of the system. The processing power of the general purpose processor that the application runs on is entirely occupied for a single video analytic function. As a result, it is difficult to handle additional tasks. In [9], a performance comparison has shown that FPGA (Field Programmable Gate Array) and GPGPU (General Purpose Graphical Processing Unit) has advantage over general purpose CPU (Central Processing Unit). Thus, it is beneficial to use FPGA or GPGPU as hardware accelerators to enhance the performance of visual based fall detection in the current research work.

In this work, it is believed that FPGA is more suitable to be used due to its flexibility, portability and its potential to migrate to ASIC design over a short period of time. Thus, novel hardware architecture of visual based fall detection using FPGA is proposed in this work.

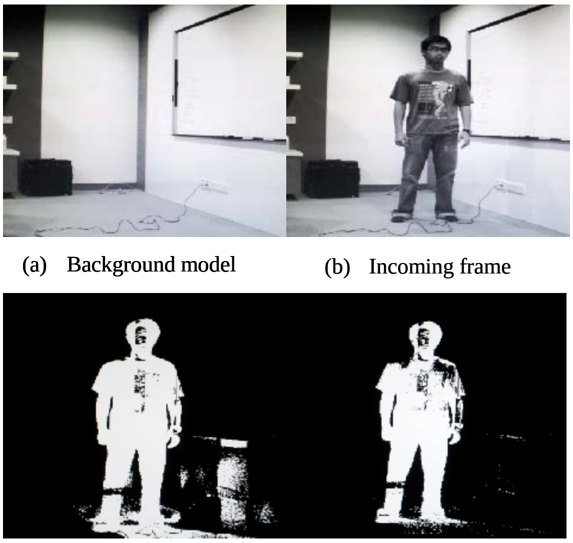
III. Fall Detection Algorithm

Visual based fall detection system requires a reliable object tracking system. In our work, we use background subtraction technique to detect and track the moving object efficiently. As shadows would affect the outcome of the object tracking, a shadow reduction technique is applied. In addition, median filtering and morphological processes are used to remove the noise. Lastly, the bounding box technique as proposed in [4] and [5] is incorporated to recognize and confirm falls.

A. Human Detection

Background subtraction has always been used for moving object detection due to its low complexity and producing accurate results. The crucial part in background subtraction is to generate the background model. According to S.-C. Cheung et al. [10], there are non-recursive and recursive background subtraction technique. The non-recursive technique generates the background model from a buffer of *N*number of previous input frames. On the other hand, the recursive method constantly updates a single background model based on each input frame. The recursive method does not required buffers; thus, it requires less memory storage. This leads to the reason of it being more suitable to be implemented in FPGA which has stringent memory constraint. In this work, frame differencing background subtraction is used. As we are targeting the home environment with static background, background training might not be necessary. The first image frame with no foreground object is captured as the background model. The subsequent incoming image frames will be compared with the background model and the difference will be calculated. Pixels will be classified as the foreground object if it satisfies the condition in (1).

$$\left| I_{New}[x,y] - I_{Bg}(x,y) \right| > T \tag{1}$$


(a) Background model (b) Incoming frame

(c) Before shadow reduction (d) Shadow reduction

Fig. 1 Illustration of shadow reduction technique in this paper

I_{New} and $I_{Bg}$are the intensity values for the incoming pixel and background pixel respectively at coordinate (*x*, *y*). *T* is a pre-determined value for the difference threshold. Pixels which do not fulfill the condition in (1) will be classified as the background object.

B. Shadow Reduction

Different types of shadow reduction techniques based on chromacity, physical, geometry and texture has been studied in [11]. In this paper, we find that chromacity-based method which uses YCrCb color space is the most suitable due to its linear transformation from RGB color space. It requires less hardware resources for the conversion and it is proven in [12] that this color space is able to differentiate shadows reasonably well. The color space transformation equations used are as shown in (2), (3) and (4).

$$Y=0.299R+0.587G+0.114B \tag{2}$$

$$C_r = 0.713(R - Y) \tag{3}$$

$$C_b = 0.565(B - Y) \tag{4}$$

Subsequently, shadows reduction can be done by complying with two assumptions: First, shadow region has a lower intensity (*Y*) value because it is less illuminated by the light source; second, the chrominance values (*Cr*, *Cb*) of the shadow region have less significant changes when compared with the background model. Thus, for each incoming pixel which has a lower intensity value as compared to the background model, we further compare its chrominance values with the background model. The pixels with absolute chrominance difference larger than a pre-determined threshold is classified as foreground. Using this classification, the shadows of the object can be significantly reduced, as shown in Fig. 1.

C. Fall Detection

The system consists of two main steps, namely fall recognition and fall confirmation. A sudden change in human body shape's aspect ratio is a widely used method in visual based fall detection to recognize a fall due to its simplicity yet with relatively high accuracy. The required measurements are as shown in Fig. 2. A normal standing person should always have a height (*H*) greater than width (*W*) which calculates to an aspect ratio > 1. In contrast, a person who falls should have an aspect ratio < 1. However, this is not always true under the circumstances like occlusion and falling parallel to the field of view of the camera. Nevertheless, this technique is able to detect other fall activities efficiently.

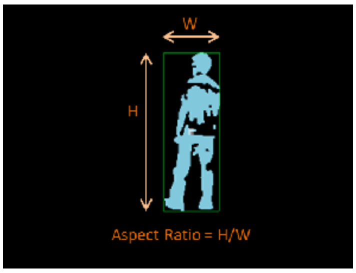


Fig. 2 The aspect ratio of the bounding box

After recognizing a fall, the inactivity period is calculated. The inactivity period is the time in which the fallen object does not have significant movements possibly because of unconsciousness or immobilization. The timer will start counting when the following rules are fulfilled.

1) Aspect ratio is smaller than 1.
2) The centroid of the object does not move more than 5 pixels in any direction.

In our work, we set the inactivity period to 6 seconds to confirm a fall. If one of the above mentioned two rules is not fulfilled during the inactivity period calculation, the timer will stop and the activity will be determined as non-fall.

IV. Hardware Implementation

The proposed system was implemented on Terasic'st Pad development board, which is an integration of Terasic's DE2-115 development board with Altera Cyclone IV (EP4CE115) FPGA device, a 5 megapixels CMOS camera sensor and a LCD touch panel. The system level abstraction of the proposed visual based fall detection system is as shown in Fig.3. The system was carefully designed like an embedded system with highly exploitation of the parallel and pipeline architecture of the FPGA.

A. Image Acquisition

The interfacing between the CMOS sensor and FPGA consists of a configuration module, a capturing module and a raw to RGB conversion module. The CMOS sensor outputs the raw data in Bayer format which consists of four color components, namely Green1, Green2, Red, and Blue. Internally, the CMOS sensor is able to capture images up to a resolution 2560x1920. In this work, the row and the column pixels are configured to twice skipping and binning mode. This reduces the resolution by combining 2 adjacent same-color imager pixels at horizontal and vertical direction to produce

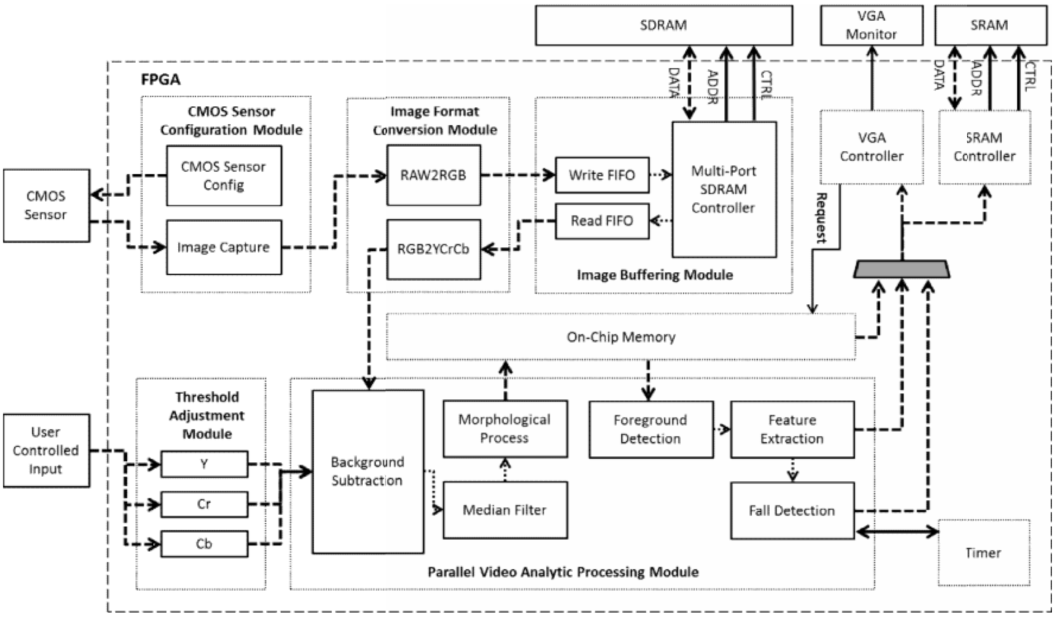
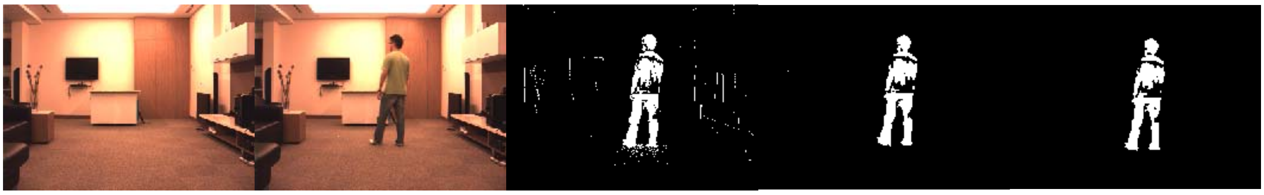
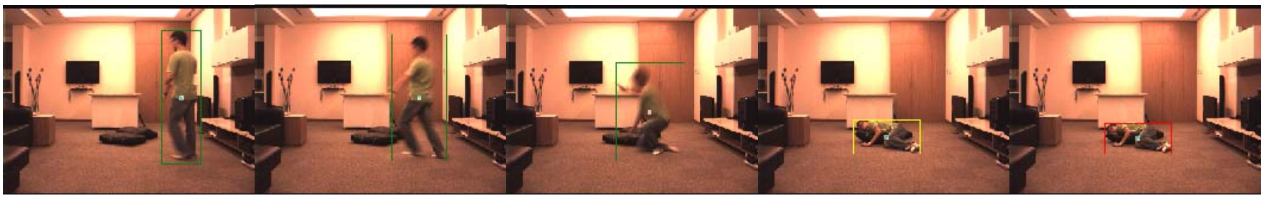


Fig. 3 Block diagram of the proposed FPGA implemented of visual based fall detection



(a) Background frame (b) Incoming frame (c) Background subtraction (d) Median Filtering (e) Morphological process

Fig. 4 (a) The background image frame with no object. (b) The incoming new image frame with a moving object. (c) The output from the background subtraction with noise. (d) The output from median filter which significantly removes the noise. (e) The output from morphological process which refines the object boundaries



(a)Object standing, aspect ratio > 1 (b) Object starts falling, aspect ratio > 1 (c) Object falling, aspect ratio > 1 (d) Object lie on the floor, aspect ratio < 1 (e) Inactivity period over, fall confirmed

Fig. 5 (a) An object is detected (b) The object is experiencing a fall (c) The object is falling (d) The object finally lies down on the floor, fall recognized and the bounding box turns yellow as the signal (e) The object did not move after an inactivity period, fall confirmed and bounding box turns red as the signal

one output pixel. As a result, the actual resolution of the image at the raw data is reduced to 1280x960. Finally, at the conversion from raw to RGB, the image is further downscaled into 640x480 by skipping the odd row and odd column of the image.

B. Image Buffering

After the raw to RGB conversion, the image will be stored in an external SDRAM. The DE2-115 development board has a SDRAM with memory capacity of 128MB from two 64MB SDRAM devices. Each of them has a separate 16-bit data lines connected to the FPGA and share the same control and address lines. An example of image capturing application from Terasic uses four 16-bit ports at the SDRAM controller with two for writing and the other two for reading. As pixel has a total 30-bit of data (10-bit for each component from RGB), the SDRAM controller can only write a single image at one time as it requires both write-ports and read-ports to write and read images respectively. This is not feasible in our application because two frames (background frame and incoming frame) are required at anyone time to conduct the background subtraction. Hence, the width of each SDRAM controller port is expanded from 16-bit to 32-bit. As a result, only a read-port and a write-port are required for an image frame.

C. Parallel Video Analytic Processing

Every pixel read from the image buffering module will go through a series of processing in a highly pipeline and parallel structure. As shown in Fig. 6, the first pixel processed will go to the next module immediately. This limits the maximum delay time at pixel level instead of the frame level.

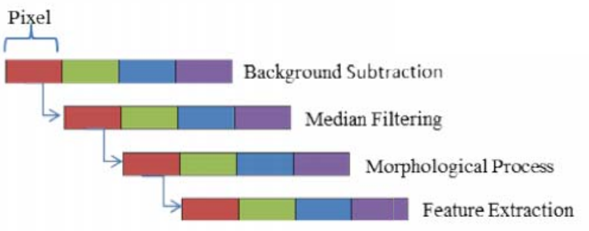


Fig. 6 The pipeline flow of the processed pixels

D. Background Subtraction

To conduct the background subtraction, the incoming frame and background model will be read together from the SDRAM in pixels at each positive-edge of the triggering clock. The difference between the pixels will be calculated and grouped into foreground or background objects based on the algorithm discussed in Section III. To reduce the delay of the computations, most calculations are done using continuous assignment statement in hardware description language which by default has a zero timing delay (maximum delay = signal propagation delay).

E. Median Filtering and Morphological Process

Due to the motions at the background and the vibration of the camera itself, background subtraction tends to observe noise that the morphological process alone cannot remove. Thus, we apply a 5x5 median filtering before the morphological process to reduce the noise and fill up the tiny holes in the silhouette of the object. At morphological stage, opening-closing method is applied to refine the boundary. The results of the filtering and morphological process can be seen in Fig. 4.

F. Feature Extraction

A bounding box is created for each foreground object for the feature extraction process. A pixel will be considered **to** belong to one object if its distance with the object's bounding box is shorter than a pre-determined value in either horizontal or vertical direction. To address the multi-object problems, we have incorporated a multi-object tracker which is able to detect up to 4 objects. In the case of overlapping of bounding boxes, two will be joined together if the center of the bounding box of either one is residing in the bounding box of the other one. Lastly, when bounding box is completed, the width, height, centroid and the center of the bounding box will be extracted.

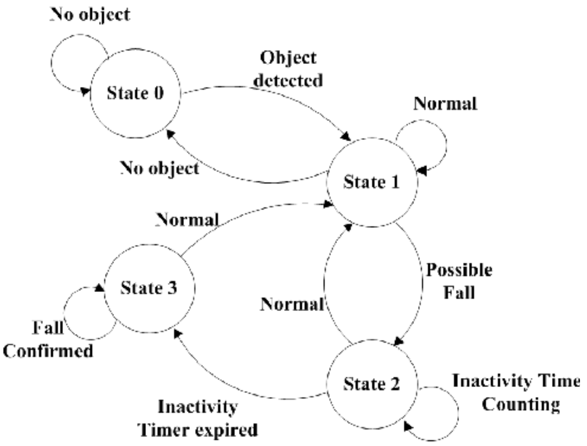


Fig. 7 State machine of the fall detection

TABLE I
HARDWARE RESOURCE UTILIZATION ON THE FPGA DEVICE

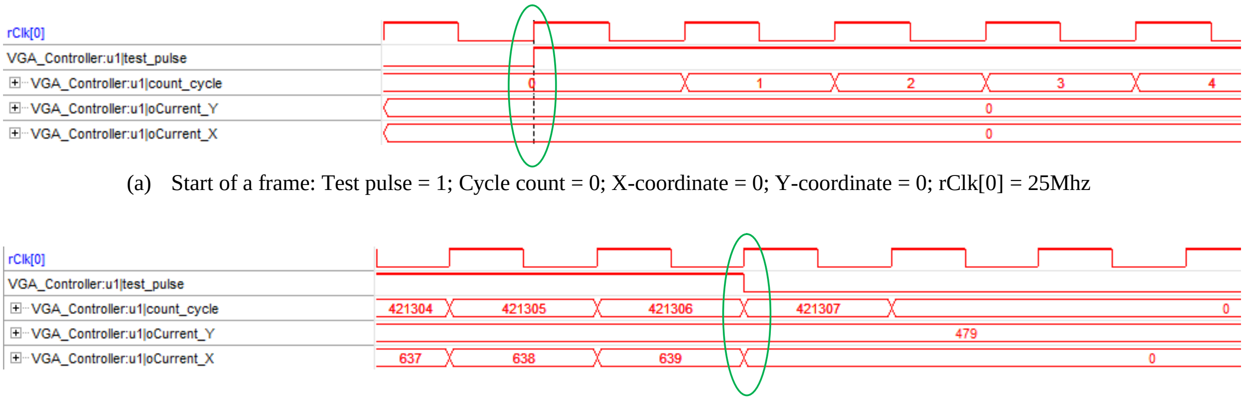
Resource	Total	Used	Percentage
Logic Element	114, 480	12, 894	11%
Block Memory (Kbits)	3,888	1,202	31%
Embedded Multiplier (9-bit)	532	16	3%

TABLE II
FALL DETECTION RESULTS OF THE PROPOSED FPGA IMPLEMENTATION OF INTELLIGENT VISUAL BASED FALL DETECTION

Incident	Fall	Non-Fall	Accuracy
Daily Activities	3	27	90.00%
Falls	25	5	83.33%

G. Fall Detection

The aspect ratio of the object is obtained by calculating the width to height ratio of the object using Altera Megafunction for floating point division. As the function works with single precision (32-bit) floating point format (IEEE 754), a converter is created to change the width and height of the object into single precision floating point format. Then, by giving the converted width and height as the input, we acquired the aspect ratio in floating point format from the division function. Whenever the aspect ratio changes from > 1 to <1, the time taken for the changes will be determined. In this case, we find the difference of the current aspect ratio with the aspect ratio 1 second ago. If the difference is larger than a pre-determined threshold (0.5), a possible fall will be indicated.


rClk[0]
VGA_Controller:u1|test_pulse
VGA_Controller:u1|count_cycle
VGA_Controller:u1|oCurrent_Y
VGA_Controller:u1|oCurrent_X
0
1
2
3
4
0
0
rClk[0]
VGA_Controller:u1|test_pulse
VGA_Controller:u1|count_cycle
VGA_Controller:u1|oCurrent_Y
VGA_Controller:u1|oCurrent_X
421304
421305
421306
421307
0
479
637
638
639
0

(a) Start of a frame: Test pulse = 1; Cycle count = 0; X-coordinate = 0; Y-coordinate = 0; rClk[0] = 25Mhz

(b) End of a frame: Test pulse = 0; Cycle count = 421307; X-coordinate = 639; Y-coordinate = 479; rClk[0] = 25Mhz

Fig. 8 (a) The test pulse is triggered at the first pixel of the image. (b) The test pulse is canceled at the last pixel of the image

The fall detection module is designed in state machine behavior as illustrated in Fig. 7. When an object is detected, the state machine will move from state 0 to state 1. When a possible fall event occurs, the state machine will move from State 1 to State 2. Then, the timer for inactivity period will start counting. As mentioned in Section III, if an inactive period of more than 6 seconds occurs, it will shift to State 3 and the event would be confirmed as fall. A signal for fall confirmation will be triggered as shown in Fig. 5 in which it is indicated by a red bounding box. At any circumstances that the aspect ratio becomes > 1, the situation will be considered as normal again and the state machine will shift to State 1. When the moving object leaves the field of view of the camera, the state machine will shift to State 0.

V. Experimental Results

Our hardware implementation was designed on the development board mentioned in Section IV using Verilog hardware description language. Table I shows the hardware resource utilization of the FPGA device. In order to display the results with the VGA resolution, our hardware implementation uses the VGA control clock to read and process the data from the memory. As the image processing follows the VGA control clock and the algorithm has no delay up to frame level, we can estimate the processing frame rate by calculating the number of cycles required to process from the first pixel until the last pixel. The approximation is as indicated in Fig. 8. Then, by using equation (5) below, the processing frame rate can be calculated ($N_{clock\ per\ frame}$is the number of clocks used to display one frame and $f_{clock}$is the operating frequency to read images). The result gives a value of 59.339 which is approximately 60 frames per second.

$$Frame\ rate = \left[N_{clock\ per\ frame} / f_{clock} \right]^{-1} \tag{5}$$

To evaluate the accuracy of the system in fall detection, we conducted a series of tests. Thirty different daily activities which include walking, sleeping, sitting, squatting, lying down and jumping were carried out to check the reliability of the system in differentiating between fall events and daily activities. Besides that, thirty trials of fall which include side-fall, forward-fall and backward-fall were tested. Table II shows the results of the test. Although there are false positives and false negatives for both daily activities and fall events, the overall performance presents satisfactory results. The accuracy of the system can be further improved by incorporating multi-camera and more sophisticated foreground detection method.

VI. Conclusion

In this paper, we have presented an FPGA implementation of intelligent visual based fall detection system which at the best understandings of the authors, has not been reported in the literature. By exploiting the parallel and pipeline architecture of the FPGA, the proposed system is able to process up to 60 fps at the resolution of 640x480. As the main processing frequency is only clocked at 25 Mhz, it shows the great potential to speed up more on the image processing by using a higher clock frequency and exploit further on parallel and pipeline architecture of the FPGA for the processing of large-scale vision system

Acknowledgment

This work, in collaboration with MIMOS Malaysia, was supported in part by a research grant from the Ministry of Science Technology and Innovation.

References

[1] U. Laessoe, H.C. Hoeck, O. Simonsen, Thomas Sinkjaer, Michael Voigt, “Fall risk in an active elderly population can it be assessed?”, Journal of Negative Results in BioMedicine, 2007, 6:2.

[2] Rabieyah Mat, Hajar Md. Taha, “Socio-Economic Characteristics of the Elderly in Malaysia”, 21st Population Census Conference, 2003.

[3] Report on seniors’s falls in Canada. Public Health Agency of Canada, Division of Aging and Seniors, 2005.

[4] Tao, M. Turjo, M. F. Wong, M. Wang, and Y. P. Tan, “Fall incidents detection for intelligent video surveillance,” in Proc. IEEE Int. Conf. Commu. and Signal Processing, 2005, pp. 1590-1594.

[5] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Robust video surveillance for fall detection based on human shape deformation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 5, May, 2011, pp. 611-622.

[6] Y. T. Chen, Y. C. Lin, and W. H. Fang, “A hybrid human fall detection scheme,” in Proc. of 2010 IEEE 17th International Conference on Image Processing, 2010, pp. 3485-3488.

[7] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Fall detection from human shape and motion history using video surveillance,” in Proc. 21st Int. Conf. AINAW, vol. 2. 2007, pp. 875–880.

[8] JiaLuen Chua, YoongChoon Chang, Wee Keong Lim, "Intelligent Visual Based Fall Detection Technique for Home Surveillance," International Symposium on Computer, Consumer and Control (IS3C), 2012, pp. 183-187.

[9] Asano, S., Maruyama, T., Yamaguchi, Y., "Performance comparison of FPGA, GPU and CPU in image processing," International Conference on Field Programmable Logic and Applications, 2009, pp. 126-131.

[10] S.-C. Cheung and C. Kamath, “Robust techniques for background subtraction in urban traffic video,” in Proc. of the SPIE, vol. 5308, 2004, pp. 881-892.

[11] A. Sanin, C. Sanderson, B.C. Lovell. “Shadow Detection: A Survey and Comparative Evaluation of Recent Methods,” Pattern Recognition, Vol. 45, No. 4, 2012, pp. 1684–1695.

[12] F. Kristensen, P. Nilsson, and V. Öwall, “Background segmentation beyond RGB,” in Proc. Asian Conf. Computer Vision, vol. 2, 2006, pp. 602–612.