BOOSTING FRAME RATE PERFORMANCE OF FALL DETECTION SYSTEM ON HETEROGENEOUS PLATFORM

Nguyen Thi Khanh Hong[1], Le Huu Duy[1], Pham Van Tuan[2], Cecile Belleudy[3]

[1]*College of Technology, The University of Danang, Vietnam; ntkhong@dct.udn.vn; lhduy@dct.udn.vn*

[2]*University of Science and Technology,The University of Danang, Vietnam; pvtuan@dut.udn.vn*

[3]*University of Nice Sophia Antipolis, Nice, France; Cecile.Belleudy@unice.fr*

Abstract - Heterogeneous computing platform, Zynq- 7000 all programmable system-on-chip, not only accomplishes high efficiency solutions in accelerating the power consumption, execution time for implementing the Fall Detection application but also takes the advantage of Open source Computer Vision (OpenCV) libraries. The main goal of this work is to design and implement the Fall Detection System on Zynq platform. In addition, the execution time and calculated energy are extracted from the platform implementation. Besides, the Accuracy, Recall and Precision factors of Fall Detection System which are executed on the computer and platform implementation are compared. Finally, NEON optimization is used to boost the frame rate performance of Fall Detection System on Zynq Platform.

Key words - Fall Detection, energy consumption, execution time, boosting frame rate.

1. Introduction and Related works

It is necessary to have systems which can automatically monitor human activities in order to reduce the pressure on training and expanding force for health solutions. As a result, it is important to develop an automated Fall Detection application to prevent fall risk of elderly and rehabilitants and provide immediate help to them.

1.1. Fall Detection Approaches

Automatic fall detection in general can be performed by many different techniques: Indoor sensors [1], [2], [3]; Wearable sensors [4]; Video systems [5], [6], [7].

Among them, the wearable sensors help to capture the high velocities, which occur during the critical phase and the horizontal orientation during the post fall phase. However, in these methods the users have to wear the device all the time, and therefore, if it is inconvenient, it could bother them. Additionally, such systems require recharging the battery frequently, which could be a serious limitation for practical application.

On the other hand, video systems enable an operator to rapidly check if an alarm is linked to an actual fall or not. A block diagram of Fall Detection based on video processing is described in Figure 1.

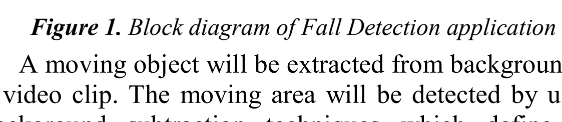

Figure 1. *Block diagram of Fall Detection application*

A moving object will be extracted from background of a video clip. The moving area will be detected by using background subtraction techniques which define the different of pixels in consecutive frames. After blobbing and smoothing the object, this result will be tracked by 2D modeling such as point tracking, kernel tracking (rectangle, ellipse, skeleton…), silhouette tracking. Then, calculating the feature extractions í done to understand what kind behaviors of object based on one of these modeling. The problems are that these features must encapsulate unique characteristics for the same action made by different people. In order to avoid misdetection and false alarms for this system not only depends on the techniques but also confronts some challenges such as dynamic background, brightness, occlusion, static object.

After tracking and extracting the object features, the problem of the system has to understand the meaning of the object actions through its features in the recognition block.

1.2. Implementation of Fall Detection Application

We now review some implementations for Fall Detection System which uses various methods. Besides, Michal Kepski and Bogdan Kwolek deploy the Kinect and accelerate-meter in fall detection system [8]. They implement this system on PandaBoard ES, which is a low-power and low-cost single board computer development platform based on Texas Instruments OMAP4 line of processors. In addition, a method for detecting falls at homes of elderly using a two-stage fall detection system is presented by Erik E. Stone et al. [9]. The first stage of the detection system characterizes a person's vertical state in individual depth image frames. The segmentation on ground events from the vertical state time series is then obtained by tracking the person according time. The second stage uses an ensemble of decision trees to compute a confidence that a fall precede on a ground event. Their database consists of 454 falls where 445 falls are performed by trained stunt actors and 9 are resident falls. The database is collected in nine years at the actual homes of older adults living in 13 apartments. This means that the data collection allows for characterization of system performance under real-world condition, which is not shown in other studies. Cross validation results are included for standing, sitting and lying down positions, within 4 m versus far fall locations and occluded versus not occluded fallers.

Martin Humenberger et al. in [10] present a bio-inspired and embedded fall detection system by the combination of FPGA and DSP. Bio-inspired means that the use of two optical detector chips with event-driven pixels is sensitive to relative light intensity changes only. The chips are used as stereo configuration which facilitates a 3D representation of the observed area with a stereo matching technique. Moreover, the stationary installed fall detection system has a better acceptance for independent living compared to permanently worn devices. The fall detection is performed by a trained neural network. First, a meaningful feature vector is calculated from the point clouds. Then the neural network classifies the actual event as fall or non-fall. All processing is done on an embedded device consisting of an FPGA for stereo matching and a DSP for neural network calculation achieving several fall evaluations per second. The results of evaluation indicate that a fall detection rate of more than 96% with false positives below 5% for the pre-recorded database consisting of 679 fall scenarios.

In the next section, the research objective is mentioned. Fall Detection application will be described with four steps: object segmentation, filter, feature extractions and recognition in Section 3. In Section 4 an insightful experiment of implementation and evaluation is described. Finally, Section 5 contains the conclusions of this paper.

2. Research objective

In this study, the Fall Detection System in High Level Languages specified in C/C++ integrated OpenCV, cross-compiled along with libraries which implement the communication Application Programming Interfaces (APIs) and runtime layer using gcc/g++ toolchains are designed. The toolchains generate an.elf file downloaded to the processor ARM Cortex A9 on Zynq platform supported by SDK tools. Our system is executed by the configuration of image resolutions, frequencies of processor cores. The recognition rate is then evaluated and compared with other system. For designing and implementing our Fall Detection System on Zynq platform, the case study is presented as follows:

- Input video is recorded by the Camera Web Cam-Philips SPC 900NC [1] that is mounted on the wall at the distance of 3m from the floor.
- Resolution of input video: 320x240 pixels, 680x480 pixels.
- Core frequency: 222 MHz, 333MHz and 667 MHz.
- Output: warning signal (FALL or NONFALL), execution time, energy consumption.

Moreover, the recognition parameters such as Accuracy, Recall and Precision are compared based on computer and Zynq platform. The configuration of computer is described as follows:

- CPU: Intel Core i3 2.6Ghz
- Ram: 2GByte
- Operating System: Windows 7

And the characters of Zynq platform are: The Zynq®-7000 XC7Z020 CLG484 -1 AP SoC is a product based on the Xilinx All Programmable SoC architecture. It integrates a feature-rich dual-core ARM® Cortex™-A9 based processing system (PS) and 28 nm Xilinx programmable logic (PL) in a single device. The ARM Cortex-A9 CPUs are the heart of the PS and also include on-chip memory, external memory interfaces, and a rich set of peripheral connectivity interfaces [11].

Finally, from the observed results which are extracted by implementation of Zynq platform, the NEON Optimized Libraries is applied. As Cortex-A9 on Zynq platform prevails in embedded designs, many software libraries are optimized for NEON and have performance improvements and cache efficiency. In our study, we extract the execution time, power consumption of whole Fall Detection System which is deployed on ARM processor of Zynq -7000 AP SoC platform. After that, NEON is used for boosting the frame rate performance of Fall Detection System.

3. Fall Detection Application

3.1. Object segmentation

Background subtraction is method used to detect moving object. This method detects and distinguishes object or foreground with the rest of the frame called background [12] by subtracting current frame to estimated background [13]. The estimated background is update as follows:

$$B_{i+1} = aI_i + (1-a)B_i \tag{1}$$

Where B_i is current background, B_{i+1} is a updated background, and a is update coefficient and is kept small to prevent artificial tails forming behind moving objects. In the study, an average of 3 consecutive frames is used instead of the current frame I_i

$$B_{i+1} = (1-a)B_i + a\frac{1}{3}\sum_{j=j-2}^{i} I_j \tag{2}$$

Where α is chosen 0.05 as in [12]. Figure 2a and Figure 2b show the input frame and the the result of background after estimating.

Moving object is estimated by subtracting the current frame from background and comparing with threshold value τ. Pixels are considered if

$$|I_i - B_i| > \tau \tag{3}$$

Where is τ is predefined threshold. The result after being compared to τ is described in Figure 2c

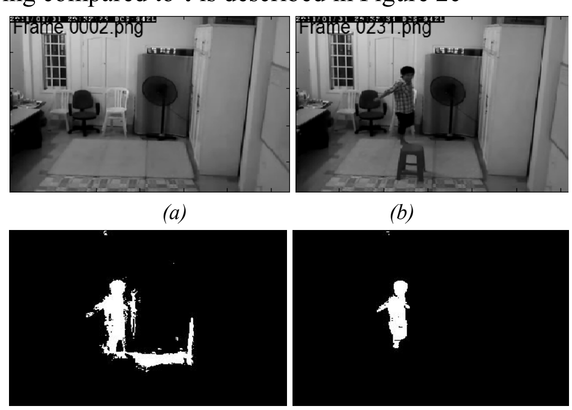

Figure 2. *(a) Estimate Background; (b) Input Frame; (c) Background Subtraction method; (d) GMM method.*

However, the result of background subtraction process is greatly affected by shadow of the object. In order to distinguish object from background, another method of estimating background/foreground is applied. An adaptive Gaussians mixture model (GMM) that was proposed by

[1] http://www.p4c.philips.com/cgi-bin/dcbint/cpindex.pl?ctn=SPC900NC/00&scy=gb&slg=en

Stauffer and Grimson at [16] is applied here. In this work, the values of a particular pixel over time is considered as a "pixel process", and each pixel is modeled by a mixture of K Gaussian distributions, which is used to estimate that pixel belongs to foreground or background. Thanks to probability distribution, GMM method could produces a better result than Background subtraction, even in the case of shadow caused by object (Figure 2d).

3.2. Morphology Filter

Morphology Mathematic (MM) methods are used to improve the quality of image from the object binary image. Some of MMs are dilation, erosion, opening, closing or the combination of these.

3.3. Body modelling and features extraction

3.3.1. Ellipse model

Ellipse model is a simple model describing the motion or other factors of object like velocity, location, or the shape of the human body. In this model, a single object is surrounded by an ellipse that represents human body. Three main and important parameters are considered in an ellipse model as follows [5]:

a. Centroid of ellipse

It is the location $O(O_x, O_y)$ or the centroid coordinates of ellipse each frame, and it is calculated as an average of all x coordinates and all y coordinates of white pixels in binary image.

b. Vertical angle of the object

It presents the angle of object. It is also the angle between the major axis of ellipse and horizontal line. Figure 3.

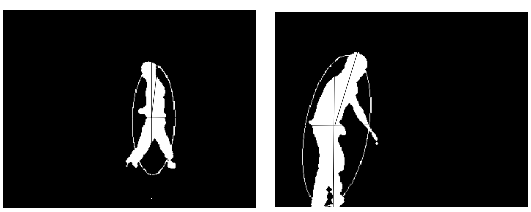

Figure 3. *Current angle of an object*

c. Major and minor axis of the object:

Major and minor axis are twice as much as the distance from centroid of ellipse O to O_1 and O_2 respectively, in which:

O_1 is the average of all x coordinates and the all y coordinates of white pixels which have a limited angle so that $|W\hat{O}h-\theta|<60^0$ and O_2 is the average of all x coordinates and the all y coordinates of white pixels which have a limited angle so that $|W\hat{O}h-(\theta+\pi/2)|<60^0$.

3.3.2. Feature extraction

5 major features are extracted from ellipse model of moving object and binary image:

a. Current angle

Current angle is vertical angle of the ellipse which presents the object [5].

b. Coefficient of motion

This is also known as an image of motion history and is considered as the velocity of the moving object. The equation of Cmotion is described as follows. And Figure 4 shows Motion History Image in case of moving and falling.

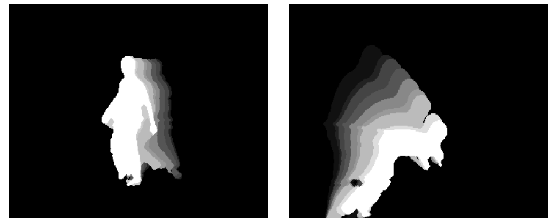

Figure 4. *Motion History Image*

c. Deviation of the angle (Ctheta)

Ctheta is standard deviation of vertical angle calculated from 15 successive frames. Ctheta is usually higher when a fall occurs [5].

d. Eccentricity

Eccentricity e at current frame is computed:

$$e=\sqrt{1-\frac{b^2}{a^2}} \tag{9}$$

Where a, b is semi-major and semi-minor axis of ellipse perspective. e is smaller when direct fall happens

e. Deviation of the centroid.

Ccentroid is standard deviation of centroid coordinates calculated from 15 continuously frames. Ccentroid falls rapidly when a fall occurs.

3.3.3. Recognition based on Template Matching

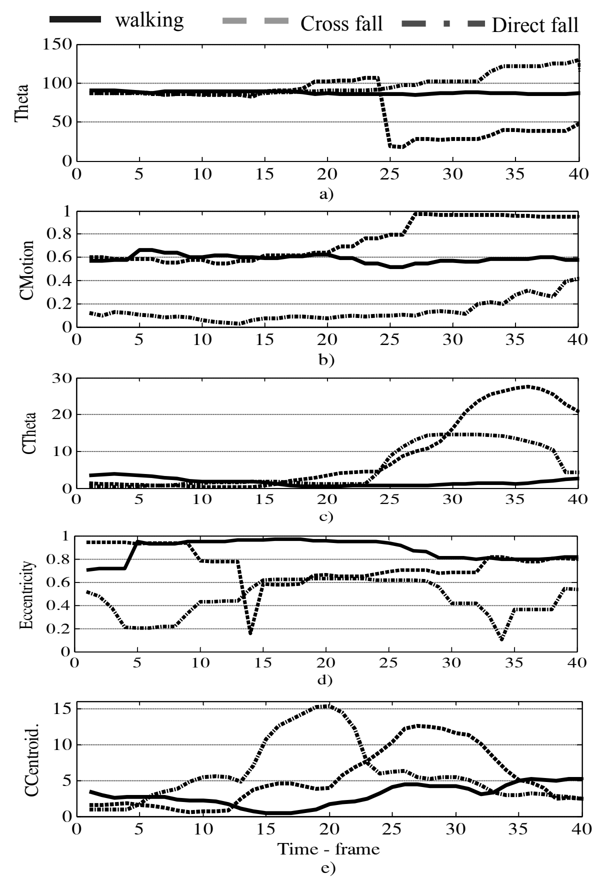

Figure 5. *Feature evolution for walking, fall-down in two direct*

Five extracted features from extraction block will be reasonably combined to recognize fall action. First, suitable thresholds are indicated through training process. A decision is estimated based on combination of some features with appropriate rules. This will be applied in test process to recognize action. From training, some rule are applied as follows: Theta and Cmotion are necessary in all case of data because there is a major change in two features when fall-behaviors occurs. So the combination of two features could effectively distinguish fall from non-fall behavious of old people such as walking, bending, sitting or lying in the bed; Eccentricity plays a key role in direct fall detection because other features are difficult to recognize in this case. More specific information of each case is shown in Figure 5.

4. Implementation & Evaluation

4.1. Platform implementation

4.1.1. Classification Performance

The DUT-HBU database [5] is used in this system. All video data are compressed in.avi format and captured by a single camera in a small room with the changeable conditions such as brightness, objects, direction of camera, etc. Database: the fall direction is subdivided into three basic directions: *Direct fall*: object falls face to the camera; *Cross fall*: occurs when the object falls cross to the camera; *Side fall*: the object perpendicularly falls to the both sides of the camera. In terms of non-fall videos, usual activities which can be misrecognized with fall action such as lying, sitting, creeping, bending are also classified into three directions above.

4.1.2. Classifying Evaluation

ROC (Receiver Operating Characteristics) is one of the methods to evaluate the efficient and accuracy of a system by calculating the Precision (PR), Recall (RC) and Accuracy (Acc). See in the Equation 10.

$$PR = \frac{TP}{TP+FP}; RC = \frac{TP}{TP+FN}; \quad Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{10}$$

Where TP, TN, FN, FP are defined as follows:

True positives (TP): amount of fall actions which are correctly classified as fall.

False positives (FP): amount of non-fall actions which are wrongly considered to be fall.

False negatives (FN): amount of fall actions which are wrongly rejected and classified as non-fall actions.

True negative (TN): amount of non-fall actions which are correctly classified as non-fall.

4.1.3. Recognition performance

In this study, Template Matching algorithm is used in recognition block. We combine five features: θ, Ctheta, Cmotion, Ccentroid, e and four models of the fall to detect a fall event. In some case, the models are not enough to describe all cases when falls may occur.

The recognition parameters such as Recall, Precision and Accuracy are calculated by using the clear data set in DUT-HBU database [5]. Figure 6 presents the comparison of these parameters which are executed on computer and implemented on Zynq platform. The result of computer is higher about 8% than Zynq platform. However, the Recall, Precision and Accuracy are achieved by 90.5%, 86.2% and 87.1%. These parameters are considerably improved compared with the same of study in [14].

4.1.4. Experiment results for the Fall Detection System on platform

In our study, the measurement of power is taken by the Fusion Digital Power Designer GUI. The TI USB Adapter includes Power Management Bus (PMBus). PMBus is an open standard power-management protocol. This flexible and highly versatile standard allows for communication between Zynq platform and PC based on both analog and digital technologies and provides true interoperability, which will reduce design complexity and shorten time to market for power system designers. The Table 1 show**s** the power and energy consumption at various image resolutions and frequencies.

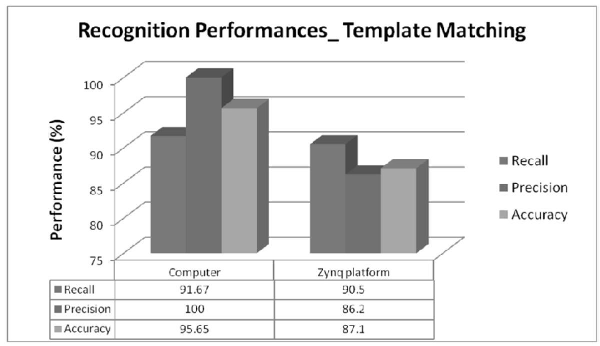

Figure 6. *The results of Template Matching Algorithm.*

Table 1. *The Power/Energy of Fall Detection System on platform*

Image resolution	Frequency (MHz)	Power mW	Energy mJ
320x240	667	420	45.11
	333	304.55	65.26
	222	254.55	79.83
640x480	667	437.27	188.73
	333	323.64	268.68
	222	269.09	335.39

4.1.5. NEON for boosting performance

Xilinx Zynq®-7000 AP SoC platform is an architecture that integrates a dual-core ARM®Cortex™-A9 processor, which is widely used in embedded products. Both ARM Cortex-A9 cores have an advanced single instruction, multiple data (SIMD) engine, also known as NEON. It is specialized for parallel data computation on large data sets. Parallel computation is the next strategy typically employed to improve CPU data processing capability. The SIMD technique allows multiple data to be processed in one or just a few CPU cycles. NEON is the SIMD implementation in ARM v7A processors. Effective use of NEON can produce significant software performance improvements [15].

SIMD is particularly useful for digital signal processing or multimedia algorithms, such as: Block-based data processing; Audio, video, and image processing codecs; 2D graphics based on rectangular blocks of pixels; 3D graphics; Color-space conversion; Physics simulations; Error correction.

The NEON to optimize Open Source Libraries such as ffmpeg and OpenCV is applied in this study. The Table 2 describes the improvement of average execution time and frame rate of two implementation stages on Zynq Platform.

5. Conclusion and Future works

In this paper, a Fall Detection Application is implemented on Zynq-7000 AP Soc platform with two video input resolutions and various frequencies. Its recognition performance has been evaluated and compared with the other system in terms of recall, precision and accuracy. The platform implementation of the application shows an average accuracy of almost over 85%. We also measure on-line power consumption and execution time of this system. Besides, the NEON optimizes Open source libraries to improve the frame rate performance with maximum number of 3fps In this system, we can use the other method such as accelerating on hardware, hardware/software co-design.

Table 2. *Frame rate improvement by using NEON*

Image resolution	Frequency (MHz)	Average execution time(ms)		Frame rate (fps)	
		Standard	NEON	Standard	NEON
320x240	667	96.5	75.5	10.4	13.2
	333	182	163.2	5.5	6.1
	222	277.4	255.1	3.6	3.9
640x480	667	395.7	358.6	2.5	2.8
	333	761.5	690.3	1.3	1.4
	222	983.5	923.8	1	1.1

REFERENCES

[1] A. Sixsmith, N. Johnson, and R. Whatmore, "Pyroelectric IR sensor arrays for fall detection in the older population,"Journal of Physics IV (Proceeding), vol. 128, pp. 153–160, Sep. 2005.

[2] T. Pallejà, M. Teixidó, M. Tresanchez, and J. Palacín, "Measuring gait using a ground laser range sensor," Journal of Sensors (Basel), vol. 9, no. 11, pp. 9133–9146, Jan. 2009.

[3] Y. Zigel, D. Litvak, and I. Gannot, "A method for automatic fall detection of elderly people using floor vibrations and sound-proof of concept on human mimicking doll falls," IEEE Transaction on Biomedical Engineering, vol. 56, no. 12, pp. 2858–2867, Dec. 2009.

[4] M. Tolkiehn, L. Atallah, B. Lo, and G.-Z. Yang, "Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor," Conference Proceeding of IEEE Engineering in Medicine Biology Society, vol. 2011, pp. 369–372, Jan. 2011.

[5] Y. T. Ngo, H. V. Nguyen, and T. V. Pham, "Study on fall detection based on intelligent video analysis," International Conference on Advanced Technologies Communication, pp. 114–117, Oct. 2012.

[6] E. Auvinet, F. Multon, A. Saint-arnaud, J. Rousseau, and J. Meunier, "Fall Detection with Multiple Cameras : An Occlusion-Resistant Method Based on 3-D Silhouette Vertical Distribution," IEEE Transaction on Information Technology in Biomedicine, vol. 15, no. 2, pp. 290–300, 2011.

[7] S. Gasparrini, E. Cippitelli, S. Spinsante, E. Gambi, and U. Politecnica, "A Depth-Based Fall Detection System Using a Kinect® Sensor," Journal of Sensors, vol. 14, pp. 2756–2775, 2014.

[8] M. Kepski and B. Kwolek, "Human Fall Detection Using Kinect Sensor," in Proceedings of the 8th International Conference on Computer Recognition Systems CORES, 2013, pp. 743–752.

[9] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the Microsoft Kinect," IEEE J. Biomed. Heal. informatics, vol. 19, no. 1, pp. 290–301, Jan. 2015.

[10] M. Humenberger, S. Schraml, C. Sulzbachner, A. N. Belbachir, A. Srp, and F. Vajda, "Embedded fall detection with a neural network and bio-inspired stereo vision," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, pp. 60–67.

[11] M. Kepski and B. Kwolek, "Fall Detection on Embedded Platform Using Kinect and Wireless Accelerometer," Proceedings of the 13th International Conference on Computers Helping People with Special Needs, vol.2, pp. 407–414, 2012.

[12] A.M. McIvor, "Background subtraction techniques. In Proc. Of Image and Vision Computing:, Auckland, New Zealand,2000.

[13] A. Derek, James M.Keller, Marjorie Skubic, Xi Chen, and Zhihai He, "Recognizing Falls from Silhouettes" Proceedings of the 28th IEEE EMBS Annual International Conference New York City, USA, Aug 30-Sept, 2006.

[14] Hong Thi Khanh Nguyen,, Cecile Belleudy, Pham Van Tuan, "Fall Detection Application on an ARM and FPGA Heterogeneous Computing Platform", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Volume 3, pp 11349-11357, 20 August 2014.

[15] Haoliang Qin, "Boost Software Performance on Zynq-7000 AP SoC with NEON", Xilinx XAPP1206, 2014.

[16] C.Stauffer, W.E.L. Grimson, "Adaptive background mixture models for real-time tracking", IEEE Conf. Computer Vision and Pattern Recognition vol. 2 pp. 246-252, June 23-24 1999.

(The Board of Editors received the paper on 13/04/2016, its review was completed on 22/05/2016)