International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 7, Issue 9 [September 2018] PP: 19-25

Fall Detection System Based OnActive Contours

Punit Khanolkar
Pioneer Research
Author Background: Punit Khanolkar grew up in India and currently attends PrabhavatiPadamshiSoni International Junior College in Mumbai, India. His Pioneer research topic was in the field of Computer Science and titled: "Computers That See: Exploring New Techniques of Computer Vision."

ABSTRACT :*Falls are the cardinal cause of injury and death in elderly individuals. Unfortunately, most fall detectors are based on wearable sensor devices, and the elderly often tend to forget to wear them. Furthermore, fall detectors based on artificial vision are not easily available in some parts of the world, and are typically very expensive. In this paper, we present an inexpensive fall detection system for home automation based on artificial vision algorithms such as active contours and Kalman filters. Test conducted on 30 videos have shown a detection rate of 95%.*

KEY WORDS:*Kalman filters, active contours, home automation.*

Date of Submission: 18-10-2018 Date of acceptance:04-11-2018

I. INTRODUCTION

The risk of falling is pervasive and entrenched amongst elderly individuals. Every year between 28% to 35% of people over 65 years old suffer at least one fall each year, and this figure increases to a staggering 42% for people over 70 years old (World Health Organization, 2007).Falls are accountable for 50% of elderly hospitalizations and approximately 40% of non-natural mortality for this sector of the population. Moreover, falls account for 87% of fractures among people age 65 years or older (Robert Bargar, 2017).

Falls are particularly perilous for people who live alone because a significant amount of time can pass before they receive any form of assistance. Roughly a third of people beyond 65 years of age live alone, and the population of the elderly is expected to escalate significantly in the next 20 years (Rodriguez R et al. 2012).

A multitude of technologies have been developed for detecting falls; however, they predominantly entail the elderly to wear sensor devices. Some elders, especially those who are diagnosed with dementia, tend to forget to wear such devices. Elderly individuals with dementia require extensive care to function independently. People suffering from dementia typically yearn to live in their own homes, but that is not always plausible. 13 percent of the world's population beyond 60 years of age have dependent living arrangements. (World Alzheimer Report, 2013). There are about 7 million people suffering from dementia in Europe alone, and this figure is expected to nearly double every 20 years. (World Health Organization, 2012).

A step towards home automation ameliorates their independence, safety and comfort as proposed by Brunete A. et al (2017) and lowers depression (Cotten SR. et al 2014). In a study presented by Brunete A. et al (2017), caregivers' conjecture that these technological advances can be extremely beneficial if used prudently, for example in avenues like security and leisure. Simply being aware of the fact that their patient is safe at home gives caregivers an imperative psychological interlude. Fall detection systems like the one delineated in this paper are an integral step towards creating smart homes.

The fall detection system suggested in this paper is based on a Windows computer with an 8GB ram and an integrated webcam. The device can monitor a room without any human intervention. Hence, the system is proficient at 24-hour monitoring. It is crucial to know that the system is inclined for people living alone at home, because if there is more than one person at home, and one of them falls down, the other can call for help. The system can also work with an overhead raspberry pi camera – In that scenario, the device can be installed into corners of walls and ceilings and can detect the subject from multiple angles.

In this paper, we present an active contour area detection algorithm that runs in reasonable time while retaining frame wise consistency. A "fall" is detected if the subject being tracked is 40% wider than the previous frame. This should detect falls lasting 0.4 -0.8 seconds. A rectangle is drawn around the periphery of thesubject.The system does not classify someone squatting or picking up something from the floor as a fall. As of now, we have to manipulate the 'detect parameters' to get the ideal recognition in varying light conditions – an aspect which can be automated in future versions.

The rest of the paper is organized as follows. We first review related work concerning the development of fall detection systems. Then, we elucidatethe proposed method in detail. Finally, we evaluate the results and conclude the paper with an analysis of its benefits and drawbacks and a discussion of future aspirations.

II. RELATED WORK

Previous research has already proposed fall detection systems based on active contours. Salimi Kim (2016) first refined the original contour detection process to extend its functionality to fall detection. To achieve this, he introduced a new active contour detection method based on difference between the areas of two subsequent frames. In this method, the first frame is extracted is slightly blurred to minimize noise errors. Then, the next frame is extracted and the difference in terms of width and length is calculated between the two frames. Every object that is brighter than the thresholdLimit is white and everything else is black. The areas that are white are than dilated to dilatePixels. Finally, a rectangle is drawn over the white areas that are larger than the minArea. There are, however, limitations on this algorithm. Although it gives reasonably consistent results, it does not eliminate noise effectively. Furthermore, its mechanism is limited to detecting a single object in each frame – an aspect which we have improved in this paper.

As for incorporating noise reduction algorithms into the system, Koldo de Miguel et al. (2017) proposed a Kalman filter based algorithm to create a fall detection system which can be tracked via a live feed on android smartphones. The system combines the use of Kalman filters with optical flow, and state classification algorithms such as KNN filters for subject state recognition.The approach is generally limited to camera movements, but the background subtraction technique provides inspiration for our approach.

The method proposed by Miao Yu et al. (2007) uses a head tracking algorithm distantly similar to the one used in this paper. The paper uses a motion history image and code-book background subtraction to determine whether large scale movement occurs within the scene being tracked. Based on the magnitude of the movement information, particle filters with different state models are used to track the head. The head tracking procedure is performed in two video streams taken by two separate cameras and three-dimensional head position is calculated based on the tracking results. Lastly, the authors use three dimensional horizontal and vertical velocities of the head are used to detect the occurrence of a fall.

Other approaches usually resort to use a two-dimensional convolutional neural network and are intended for higher parallelism. Adrian Nunez-Marcos et al. (2017) proposed a solution which uses convolutional neural networks to decide if a sequence of frames contains a person falling. To model the video motion and make the system scenario independent, the authors have used optical flow images as input to the networks followed by a novel three-step training phase. This method has received state of art results in all three public datasets used in the paper: the system achieves a sensitivity of 99% and a specificity of 96%. We can take inspiration from this paperandtry to implement a neural network in our paper to achieve superior outcomes.

III. BACKGROUND

3.1 Background subtraction

If the background of a scene remains unchanged the detection of foreground objects would be uncomplicated.Let us assume that each frame is converted to a grayscale image before it is processed. Basically, a frame (I), at the time (t), when there are no foreground objects in the scene (empty room, road without cars) is declared as the background model and then each pixel value (P) is compared to the pixel value at the same coordinate (x, y), in the frame, at a specific time. Each pixel that is different from the background model would be declared as foreground (F). (Tamersoy 2009)

P[F(x,y,t)] = P[I(x,y,t)] - P[I(x,y,0)] - Static Frame Difference

```
import cv2
import sys
camera = cv2.VideoCapture(0)
backgroundFrame = camera.read()
backgroundFrame = cv2.cvtColor(backgroundFrame, cv2.COLOR_BGR2GRAY)
while 1: currentFrame = camera.read()
currentFrame = cv2.cvtColor(currentFrame, cv2.COLOR_BGR2GRAY)
foreground = cv2.absdiff(backgroundFrame, currentFrame)
cv2.imshow("backgroundFrame", backgroundFrame)
 cv2.imshow("foreground", foreground)

key = cv2.waitKey(1) & 0xFF
 if key == ord("q"):
      cv2.destroyAllWindows()
camera.release()
sys.exit()
```

3.2 Adaptive Background subtraction

Adaptive backgrounding is a method where the background model is created using averaging images over time (1...n). This algorithm maintains a background model and the parameters of the model evolve over time. In this paper, we used a gaussian based model to write code for this method. The code which we have written for this method has been included below.

P[F(x,y,t)] = |P[I(x,y,t)] - P[mode{I(x,y,t-1), ..., I(x,y,t-n)}]| > Threshold

```
import sys import cv2
 threshold = 10
camera = cv2.VideoCapture(0)
 _, backgroundFrame = camera.read()
backgroundFrame = cv2.cvtColor(backgroundFrame, cv2.COLOR_BGR2GRAY)
i = 1
 while 1:
 _, currentFrame = camera.read()
currentFrame = cv2.cvtColor(currentFrame, cv2.COLOR_BGR2GRAY)
 foreground = cv2.absdiff(backgroundFrame, currentFrame)
foreground = cv2.threshold(foreground, threshold, 255, cv2.THRESH_BINARY)
cv2.imshow("foreground", foreground)
alpha = (1.0/i)
backgroundFrame = cv2.addWeighted(currentFrame, alpha, backgroundFrame, 1.0-alp ha, 0)
cv2.imshow("backgroundFrame", backgroundFrame)
i += 1
key = cv2.waitKey(1) & 0xFF
 if key == ord("q"):
cv2.destroyAllWindows()
camera.release()
sys.exit()
```

3.3 Gaussian Probability Distribution

Gaussian distribution, also called Normal distribution, is a continuous probability distribution. It is determined N (μ, σ^2) where μ is the mean or the expected value and σ^2 is the variance. The mathematical value of μ determines the location of peak distribution and σ^2 controls the width of the curve. Both, adaptive and non-adaptive background subtraction are copiously based on this system of continuous probability distribution.

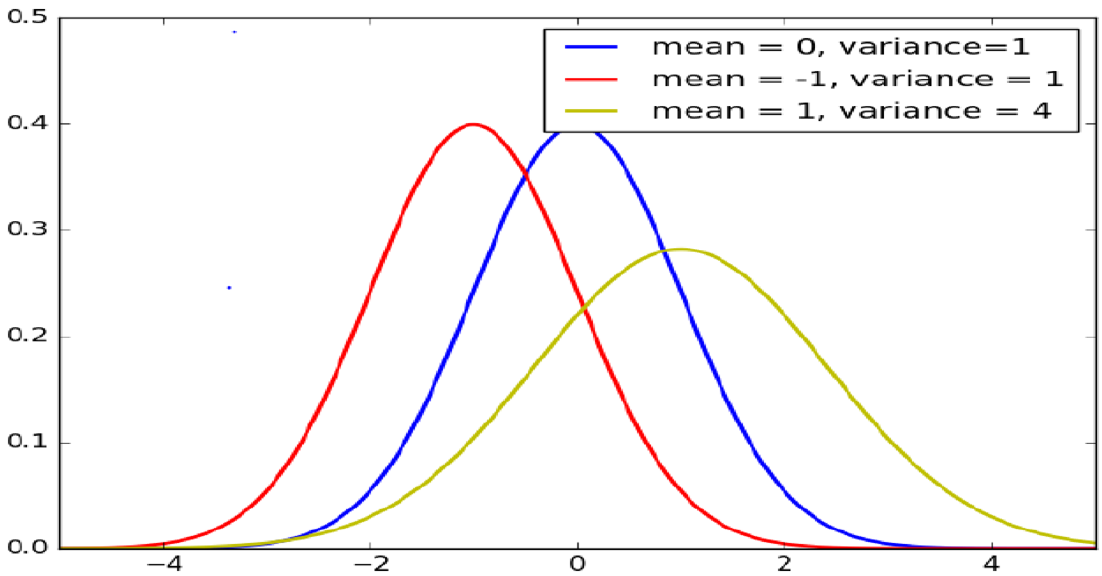


Figure 1: Gaussian distribution

3.2 Kalman Filters

A Kalman filter was chosen to keep track of the subjects in the scene. The Kalman filter was chosen instead of other algorithms like (exponential) moving average filters because its prediction properties, noise reduction properties and fast response.

Kalman filters address noisy and imprecise data and generate predictions of new states based on past data. A lineal version of the filter was implemented utilizing data extracted from the subjects in a scene. The chosen parameters are differentially processed by the filter depending on whether they are used for fall detection or future state prediction. The following equations are applied:

Centre of mass x and y position:

$x(k) = x(k\text{-}1) + x(k\text{-}1) \bullet dT,$ (1)

$y(k) = y(k\text{-}1) + y(k\text{-}1) \bullet dT,$ (2)

$x(k) = x(k\text{-}1),$ (3)

$y(k) = y(k\text{-}1),$ (4)

Height and width:

$h(k) = h(k\text{-}1)$ (5)

$w(k) = w(k\text{-}1)$ (6)

Ratio change speed:

$V_{Ratio\ (k)} = V_{Ratio\ (k\text{-}1)} + V_{Ratio\ (k\text{-}1)} \bullet dT,$ (7)

$V_{Ratio\ (k)} = V_{Ratio} = V_{Ratio\ (k\text{-}1)}$ (8)

Angle of the eclipse adjusting to the individual:

$\Phi\ (k) = \Phi\ (k\text{-}1) + \Phi\ (k\text{-}1) \bullet dT,$ (9)

$\Phi\ (k) = \Phi\ (k\text{-}1)$ (10)

The parameters used for fall state detection are ratio, ratio change speed and angle. The main functions of the filter are to reduce measurement noise and absorb periodic changes characteristic of specific movements such as walking.

The parameters for future state prediction are the center of mass and ratio. The center of mass is filtered to reduce noise and acquire a quicker response to changes. Noise was found to be negligible compared with subject size and was independent of a subject’s proximity to the camera.

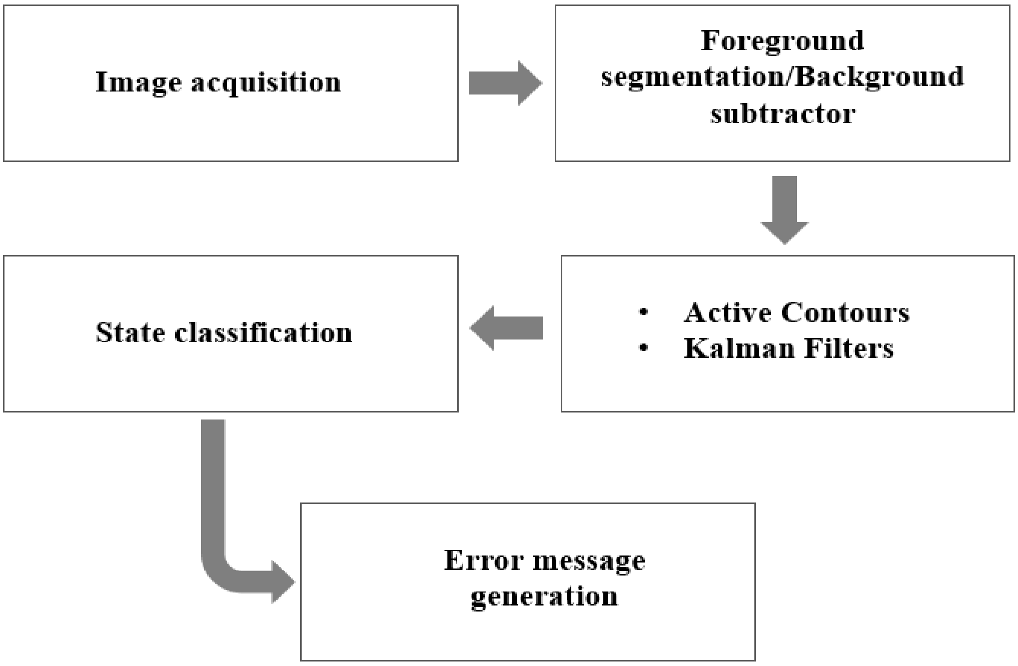
IV. MY PROJECT

We use a scheme based on background subtraction, feature extraction and intelligent fall detection using an 8GB windows computer with an Intel® Core™ i5 processor to conduct all tests.

The paramount objective of our algorithm is to distinguish between subjects in a fall state. To achieve this goal, the algorithm extracts data from the subject in a scene to identify his/her incumbent state.

Data acquisition requires multiple preliminary steps: subtracting the subject from the background, progressively learning the subject’s changing environment and identifying uninteresting objects (to facilitate their rapid recognition as background), following the subject through the scene and identifying subjects that are partially occluded by furniture. A Kalman filter is used to reduce noisy data and absorb the repetitive periodic changes common to various human actions.

We have given a step-by-step outline of how our system functions in the flowchart below.



To measure the system's performance, a total of 30 videos were recorded using a 12-megapixel Apple iPhone 7's primary mobile camera. The videos were recorded in three different locations: a house, a building compound and a classroom.

Barring the house video set, all videos were shot in the same compound and classroom across multiple takes on different days and times. The classroom light conditions are principally artificial, with some natural light entering through the windows. The 10 videos recorded in the house were shot in a conventional home environment with bountiful amounts of natural light entering through the windows. Most videos are between 20 and 40 seconds long – the mean video duration is 27.3 seconds.

1.1 Parameters Used for measuring performance

The parameters used to assess the tests are sensitivity (the percentage of fall events detected), specificity (the percentage of events without falls detected correctly), precision (the percentage of fall alerts that represent actual falls) and accuracy (the percentage of correctly detected events).

Sensitivity = TruePositives/TotalPositives, (11)

Specificity = TrueNegatives/TotalNegatives, (12)

Precision = TotalPositives/(TruePositives + FalseNegatives), (13)

Accuracy = (TruePositives + TrueNegatives)/TotalEvents. (14)

Of all these parameters, sensitivity is the most crucial because the preponderant objective of a fall detector is to detect all fall events. Accuracy and precision are also fairly significant from a performance viewpoint.

V. EXPERIMENT RESULTS

We have summarized the results of our study in Table 1 and Table 2

Table 1: Fall Detection outcomes

Type of event	No. of Events	True Positives	False Positives	True Negatives
Falls	17	16		-
Occluded falls	3	3		-
Total Positive events	20			
Sitting	7	-		7
Occlusions	4	-		4
Walking between falls	17	-	1	16
Miscellaneous	2	-		2
Total negative events	29			
Total events	49	19	1	28

Table 2: Algorithm Performance

Parameter	Result (%)
Sensitivity	95.0%
Specificity	96.6%
Precision	95.2%
Accuracy	98.0 %

VI. CONCLUSIONS

Fast and high-quality fall detection using active contours is a topic which has become intriguing only recently, contrary to the well-known fall detection using neural networks. In this paper, we propose a novel approach that has two major contributions to this growing field of research.Firstly, we propose an improved object detection method, that can detect multiple objects at the same time. Secondly, we propose a Kalman Filter algorithm that keeps track of the subject in the screen using the subject's aspect ratio and linear velocity, to eliminate noise up to a considerable extent and make the system more seamless.

It is demonstrated that out system greatly outperforms other fall detection systems based on active contours. However, due to the fact that it does not use a convolutional neural network, its accuracy compared to systems which use one is less. Also, it is low in parallelism due to limited hardware – In the future, it would be promising to improve performance by using cameras which can run with code in python.

VII. FUTURE ASPIRATIONS

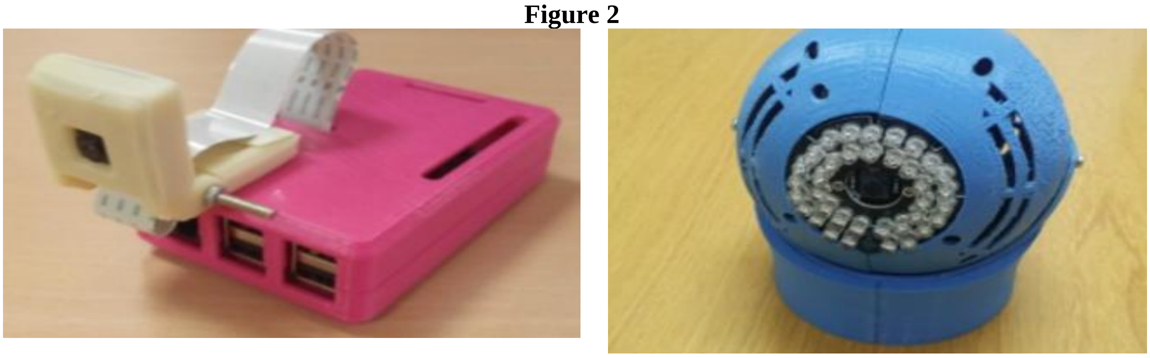
We could consider running our current code on a Raspberry Pi home camera. It would be judicious to use a Raspberry Pi 2 camera board due to its sound technical characteristics, widespread adoption and relatively low price. Implementing this would allow us to test our system from a vast array of locations as the camera can be installed at ceilings and corners of walls.

Figure 2a displays a fully independent fall detection system with an estimated cost of $120. Figure 2b shows a newer version of the unit which is based on the Raspberry Pi 3 board which includes a WIFI connection and consequently the price would be reduced.

V11. ACKNOWLEDGEMENTS

I would like to extend my sincerest gratitude to Professor Susan Fox for mentoring me throughout the program. Frankly, right before I joined the Pioneer Research Program, I had no idea of what Active Contours were about and I knew that the Pioneer Summer Program was very intensive, because it was compressed within a short span of time. However, Professor Fox made the experience easier, and more importantly enjoyable. I would like to thank the Pioneer staff for helping me schedule classes. Lastly, I would like to thank my Pioneer peers and friends – Advait, Matthew and Rohin – whose feedback and support helped me improve this paper, and helped me make my first formal paper one that I am really proud of.

Figure 2



(a) (b)

The addition of a convolutional neural network to our existing system could promulgate multifaceted possibilities for our existing system. We could implement multimodal processing to enhance supplement the system. Multimodal processing involves recognizing words such as "help" or any unusuallyhigh-pitched sounds from the video stream, and analyzing those sounds to interpret whether a fall might have occurred or not. This might prove to be integral in improving the accuracy of the incumbent system, as its addition would render a two-way fall detection system.

We could also try to automate the light parameters to get the ideal recognition as this would help save time during testing the system.

BIBLIOGRAPHY

[1]. Kalache, A., Fu, D., Yoshida, S. (2007). WHO Global Report on Falls Prevention in Older Age. World Health Organization. Geneva, Switzerland

[2]. Bargar, R. (2017). Falls and Fractures in Seniors. https://www.onhealth.com/content/1/seniors_-_falls_and_fractures

[3]. Rodrigues, R., Rodrigues, M., Laumura, G. (2012). Facts and Figures on Healthy Ageing and Long Term Care. European Centre for Social Welfare Policy and Research. Vienna, Austria.

[4]. Alzheimer's Disease International (ADI) World Alzheimer Report 2013. Alzheimer's Disease International (ADI). London, UK.

[5]. Brunete, A., Selmes, M., Selmes, J. Can smart homes extend people with Alzheimer's disease stay at home? J. Enabling Technol. 2017;11:6–12. doi: 10.1108/JET-12-2015-0039.

[6]. Miguel, K., Brunete, A., Hernando, M., Ernesto, G. (2017). Home Camera-Based Fall Detection System for the Elderly.doi: 10.3390/s17122864

[7]. Brunete A., Selmes M., Selmes J. Can smart homes extend people with Alzheimer's disease stay at home? J. Enabling Technol. 2017;11:6–12. doi: 10.1108/JET-12-2015-0039

[8]. Cotten S.R., Ford G., Ford S., Hale T.M. Internet use and depression among retired older adults in the United States: A longitudinal analysis. J. Gerontol. Ser. B Psychol. Sci. Soc. Sci. 2014;69:763–771. doi: 10.1093/geronb/gbu018.

[9]. Salimi, K. (2007). Improving safety for home care with a low cost computer vision embedded solution thus letting the aging generation live longer at home. Machine vision based fall detection for home care patients, from https://github.com/infr/falldetector-public/blob/master/Haaga-Helia%20documents/kim_salmi_fall_detection.pdf.

[10]. YU, M., NAQVI, S.M. and CHAMBERS, J.A., (2009). Fall detection in the elderly by head tracking. IN: IEEE Workshop on Statistical Signal Processing (SSP2009), Cardiff, Wales, 31 August-3 September, pp. 320-325

[11]. Nunez-Marcos, A., Azkune, G., Arganda-Carreras, I. (2017). Vision-Based Fall Detection with Convolutional Neural Networks. Wireless Communications and Mobile Computing Volume 2017, Article ID 9474806, 16 pages https://doi.org/10.1155/2017/9474806.

[12]. Tamersoy, (2009). Vehicle detection and tracking in highway surveillance videos. The University of Texas at Austin. Texas Scholar Works. University of Texas Libraries. http://hdl.handle.net/2152/ETD-UT-2009-08-316

Punit Khanolkar "Fall Detection System Based On Active Contours"International Journal Of Engineering Inventions, Vol. 07, No. 09, 2018, pp. 19-25