INTELLIGENT VISION-BASED FALL DETECTION SYSTEM: PRELIMINARY RESULTS FROM A REAL-WORLD DEPLOYMENT

Michael Belshaw[1,2], Babak Taati[1,2], David Giesbrecht[1], Alex Mihailidis[1,2]
Intelligent Assistive Technology and Systems Lab (IATSL), University of Toronto[1]
Toronto Rehabilitation Institute[2]

ABSTRACT

An automated, vision-based, ceiling-mounted Personal Emergency Response System has been developed to detect falls in real home environments. The system employs visual background modeling which separates a subject's shadowed silhouette and shadow-less silhouette regions. Analysis of these regions is performed to create velocity, area, and moment features. Machine learning then classifies these features to detect "fall" vs. "non-fall" events from the video input. In-home tests were conducted to verify the system's ability to operate in real world environments. During a seven day trial, all of the 11 simulated falls were successfully detected with 5.4 false alarms per day.

Keywords: Fall Detection, Emergency Response, Smart Home, Vision-based Activity Monitoring

I. INTRODUCTION

Falls are the most common cause of injury among older adults in the home and are the most expensive category of injury for the healthcare system in Canada [***1***]. The severity of injury due to a fall can significantly worsen if the fallen person is unable to acquire assistance. Personal Emergency Response Systems (PERS) tackle this issue and facilitate timely assistance to fall victims by placing emergency calls. The call for help is usually initiated by a wearable panic button or an accelerometer-based device. However, wearable devices are often ineffective as users may forget to wear the device or become unconscious following a fall [***2***] [***3***]. These limitations are exacerbated if the victim is cognitively impaired, reducing their awareness of the system's function.

Vision-based monitoring systems that use automatic real-time processing of images from a camera are a promising PERS alternative because users are not required to wear any devices and the system can self-activate in case of an emergency. As a result, several vision-based approaches have been explored over the past few years. These approaches can be grouped into two types: wall-mounted units in which the camera is able to see a horizontal view of the room and process parameters such as height [***4***] [***5***] [***6***], or ceiling-mounted units in which the camera has a downward facing view of the room and thus is less susceptible to occlusion [***7***] [***8***] [***9***].

Ceiling-mounted systems include systems such as Spehr et al., which detects falls by analyzing a person's body orientation [***7***]. Nait-Charif et al. and Lee et al. introduced the idea of inactivity zones to exclude locations in which falls should not be detected, such as on beds or chairs [***8***] [***9***]. Lee et al. employed background subtraction and applied thresholds to a limited set of features (speed, circumference, and feret diameter) extracted from silhouettes of foreground figures.

We expand on the single camera silhouette-based approach of Lee et al. [***9***]. We enhance the vision techniques and improve the detection of falls with a richer set of features, machine learning methods, voice prompting and speech

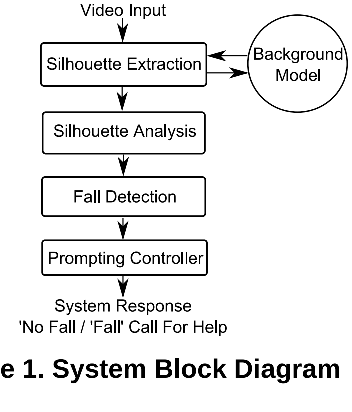

Figure 1. System Block Diagram

recognition in real-home environments outside the lab. This paper focuses mainly on the computer vision algorithms developed for the detection of falls.

II. FALL DETECTION SYSTEM

A. Overview

The fall detector is implemented in C++ and OpenCV on a 20 watt, x86 computer (1.0GHz Via Esther processor and 512 MB of RAM). An inexpensive Logitech QuickCam Pro 9000, with a field of view of approximately 48° by 61°, was used for 320 x 240 video capture at 5 frames per second in real-time, all in a ceiling unit.

All video processing is performed on-board the ceiling-mounted unit and no images are stored or broadcast. Falls are detected in real-time by performing: silhouette extraction to segment silhouettes (Sec. II-B), silhouette analysis to generate shape features (Sec. II-C), fall detection to classify the features as falls (Sec. II-D), and inactivity zones to decrease false positives (Sec. II-E).

Upon detecting a fall, the prompting controller activates speech recognition and custom prompting software to enable the user to overrule the decision and prevent false alarm emergency calls or to request a specific voice connection (e.g. a family member, a friend, a neighbor, a live operator, or a 911 emergency call). See Figure 1 for system block diagram of these main system modules.

B. Silhouette Extraction

The vision system is designed to monitor one person at a time. The background subtraction technique of Wren et al. [***10***] is used to model each background pixel as a single Gaussian distribution centered at the estimated intensity of the pixel. Background adaptation is inhibited in the vicinity of the tracked person (blob region) so that the human subject is not adapted into the background. Background subtraction is performed for each RGB color channel, at each pixel *i* according to:

$$D_i = I_i - B_i \qquad (1)$$

in which *I* is the current frame, *B* is the background image and *D* is the difference image.

To obtain the silhouette images of the person with a shadow (initial silhouette S_I) and without a shadow (final silhouette S_F), a shadow isolation technique is employed using two of the principles outlined in [***11***] :

1. *Most shadow regions will have a pixel intensity that is darker than the background scene* (See Figure 2b top)
2. *Most shadow regions in the difference image do not contain strong edges* (See Figure 2d top)

To extract the information relating to the first principle, the difference image is processed into two separate difference images D_D and D_L (e.g. in Fig 2b, 2c respectively) depending on whether the pixels are darker or lighter than the background model. This is expressed by:

$$\begin{aligned} D_{D,i} &= \left|D_i\right| && \text{if} \quad D_i < 0 \\ D_{L,i} &= D_i && \text{if} \quad D_i \geq 0 \end{aligned} \qquad (2)$$

To extract edge information related to the second principle, edge subtraction is performed on a grayscale Sobel filtered version of *I* and *B* which are subtracted to generate the edge difference image D_E (examples are depicted in Figure 2d) [***12***]. The edge information enables silhouette extraction even if the subject is the same color as the background. Also, the edge information is less sensitive to shadows.

The difference images D_L, D_D and edge difference image D_E are then thresholded and combined to create a foreground mask called the initial silhouette S_I. S_I will tend to include the subject and their shadow. The final

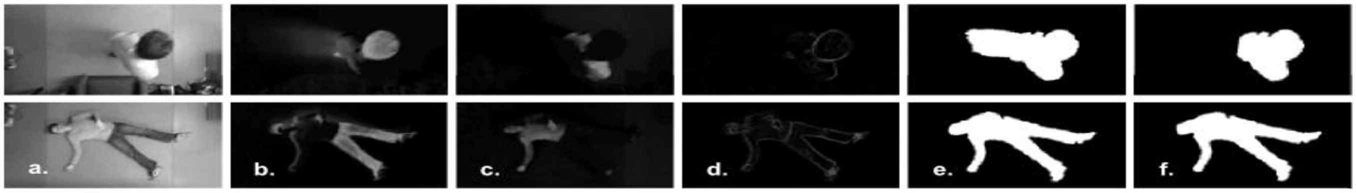

Figure 2. Silhouette Extraction With Cropped Input:
a. Input *I*, b. Darker D_D, c. Lighter D_L, d. Edge D_E, e. Initial S_I, f. Final S_F

silhouette S_F is extracted using the same thresholds for D_L and D_E but a higher more conservative threshold for D_D to exclude shadow regions [***13***]. Fig. 2e, 2f depict the initial and final silhouettes S_I and S_F.

The two silhouettes (S_I and S_F), both convey useful shadow and silhouette information and comparing them is beneficial for the detection of falls. For instance, if the human subject has a significant shadow extending far from the silhouette's center, the subject is more likely to be standing up and unlikely to have fallen. Machine learning can detect these feature patterns for such falls.

C. Silhouette Analysis

For each video frame I, the following set of geometric and temporal features

$$\mathbf{f} = \{v_I, v_F, a\} \ U \ \mathbf{h_F} \ U \ \mathbf{h_I} \qquad (3)$$

are extracted from the initial silhouette S_I and the final silhouette S_F. The v_I, v_F represents the velocity of the silhouettes. The a represents the ratio of the silhouette areas (S_I area) / (S_F area). The vectors $\mathbf{h_I}$ and $\mathbf{h_F}$ represent the seven Hu moments for S_I and S_F. If the subject has no significant cast shadow a approaches 1. If the subject has a significant cast shadow $a > 1$. The vectors $\mathbf{h_I}$ and $\mathbf{h_F}$ Hu moments [***14***] are translation, scale, and rotation invariant and thus are ideal for describing a silhouette's shape [***15***] [***16***]. These extracted silhouette features (**f**) are time-smoothed over 16 frames to reduce the effects of noise.

D. Fall Detection

For each time interval t, the time-smoothed features $\mathbf{f}(t)$ of a subject's silhouette form the set of observations which are used to classify the subject's status. Following a set of experiments using the Weka machine learning tool [***17***] to classify the "fall" vs. "non-fall" events, neural networks (NN) were empirically chosen as the classifier that performed this task well, with a low computational overhead.

Twenty video segments of approximately one minute length were collected in a mock-up living room environment and were used for training. All video frames were annotated to indicate if the frame did or did not contain a fall. A Multi-Layer Perceptron NN with one hidden layer was trained using observations extracted from manually annotated video data. The resulting classifier had a true positive rate of 97% and false positive rate of 5%. The "true positive rate" is the number of frames correctly estimated as fall events divided by the total number of frames in which actual fall events occurred. Similarly, the "false positive rate" is the number of frames incorrectly estimated as fall events divided by the total number of frames in which non-fall events occurred.

E. Inactivity Zones

As in [***8***] and [***9***], it is also possible to define inactivity zones. Defining these zones is useful around objects (such as beds or chairs) where a fall is unlikely, but it is common for the subject to remain inactive. When a subject enters an inactivity zone, the system is restricted from classifying the subject's behavior as a potential fall.

III. EXPERIMENTAL RESULTS

Two in-home trials where conducted in two separate real living rooms with healthy adult subjects. For each trial the subjects simulated falls and performed daily living behaviors for a continuous period of seven days. The results from the initial study were used to debug the system while the second seven day study was used to gather performance data. Participants for the second study were instructed to simulate falls occasionally and log such events.

During the second trial, a total of 11 simulated falls were conducted during the seven days. All the detected falls, false alarms, and voice prompt outcomes were logged. The results in Table 1 are provided with and without the prompting controller intervention, which reduced false positives. The "true positive rate" and the "false positive rate" are calculated using the fall events detected by the system (true falls or false alarms) and divided by 3,024,000, the total number of frames occurring over the trial period. When a person has fallen, only one fall event is triggered for the fall sequence. The "false alarms per day" are

Table 1. Home Trial Results

Prompting Controller Intervention	True Positive Rate	False Positive Rate	False Alarms Per Day
No	100%	0.00126%	5.43
Yes	100%	0.00016%	0.71

calculated by taking the total number of false positive falls detected and dividing by the trial length—seven days.

The test indicates that all simulated falls were detected by the system. On average 5.4 false positives were triggered daily without prompting controller intervention. The number of false positives per day is reduced to only 0.7 with prompting controller intervention. Most of the false positives were contributed to lighting changes. Most notably, over 26% of the false alarms are from sunlight shadows cast by trees moving on windy days during the tests. Only 13% of the false positives occurred with humans in the scene.

Lessons Learned

The real world environment tests indicated that complex lighting situations and multiple occupants gathering closely cause the majority of false positives for the system. The system may be retrained with "active learning" to account for these incorrectly classified cases if silhouettes can be distinguished. To better cope with multiple subjects, the system could also be expanded to track multiple silhouettes. Improved background adaptation techniques are also needed to compensate for dynamic lighting situations. The prompting controller functioned well in decreasing false positives. It was able to work with background noise (e.g. a television), but speech recognition performance decreased if the subject's face was facing the floor. Audio processing to increase the speech signal to noise ratio (e.g. noise reduction, automatic gain control) could improve speech recognition results.

IV. CONCLUSIONS & FUTURE WORK

We have designed and constructed a vision-based personal emergency response system that uses a neural network to classify falls. In-home tests verify the features' robustness in detecting falls and also indicate that the speech prompting controller successfully decreases false positives.

For future work we are expanding the feature set, utilizing different machine learning methods such as support vector machines (SVM), tracking multiple people in a scene, and using a wider angle lens.

REFERENCES

[1] SMARTRISK, "The Economic Burden of Injury in Canada.," Toronto, ON., 2009.

[2] W. Mann, P. Belchior, M. Tomita, and B. Kemp, "Use of personal emergency response systems by older individuls with disabilities," *Assistive Technology*, vol. 17, pp. 82-88, 2005.

[3] E. Porter, "Wearing and using personal emergency response systems," *Journal of Gerontological Nursing*, pp. 26-33, October 2005.

[4] D. Anderson et al., "Evaluation of a video based fall recognition system for elders using voxel space," in *International Conference of the International Society for Gerontechnology*, Pisa, Italy, 2008.

[5] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3D head tracking to detect falls of elderly people," in *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, New York City, USA, 2006.

[6] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," in *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, New York City, USA, 2006.

[7] J. Spehr, M. Gövercin, S. Winkelbach, E. Steinhagen-Thiessen, and F. Wahl, "Visual fall detection in home environments," in *International Conference of the International Society for Gerontechnology*, Pisa, Italy, 2008.

[8] H. Nait-Charif and S. McKenna, "Activity summarization and fall detection in a supportive home environment," in *International Conference of Pattern Recognition*, Cambridge, UK, 2004.

[9] T. Lee and A. Mihailidis, "An intelligent emergency response system: preliminary development and testing of automated fall detection," *Journal of Telemedicine and Telecare*, vol. 11, pp. 194-198, 2005.

[10] C. Wren, A Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780-785, 1997.

[11] P. Rosin and T. Ellis, "Image difference threshold strategies and shadow detection," in *British Machine Vision Conference (BMVC)*, Birmingham, UK, 1995.

[12] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, "Detection and location of people in video images using adaptive fusion of color and edge information," in *International Conference on Pattern Recognition (ICPR)*, Barcelona, Spain, 2000.

[13] L. Ren, G. Shakhnarovich, J.K. Hodgins, H. Pfister, and P. Viola, "Learning silhouette features for control of human motion," *ACM Transactions on Graphics*, vol. 24, no. 4, pp. 1303-1331, 2005.

[14] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179-187, 1962.

[15] J. Davis and G. Bradski, "Real-time motion template gradients using Intel CVLib," in *IEEE ICCV Workshop on Framrate Vision*, 1999.

[16] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 491-500, 2002.

[17] G. Holmes, A. Donkin, and I. Witten, "WEKA: A Machine Learning Workbench," in *Australia and New Zealand Conf. Intelligent Information Systems*, 1994.