3D Measures Computed in Monocular Camera System for Fall Detection

Konstantinos Makantasis, Anastasios Doulamis, Nikolaos F. Matsatsinis
Computer Vision & Decision Support Laboratory,
Technical University of Crete,
73100, Chania, Greece
konst.makantasis@gmail.com, adoulam@cs.ntua.gr, nikos@ergasya.tuc.gr

***Abstract*—Traumas resulting from falls have been reported as the second most common cause of death. For this reason, computer vision tools can be exploited for detecting humans' fall incidents. In this paper, we propose a fast, real-time computer vision algorithm capable to detect humans' falls in complex dynamically changing conditions, by exploiting the motion information in the scene and 3D space's measures. This algorithm is using a single monocular low cost camera and it requires minimal computational cost and minimal memory requirements that make it suitable for large scale implementations in clinical institutes and home environments. The proposed scheme was tested in complex and dynamically changing visual conditions and as proved by the experiments it has the capability to detect over 92% of fall incidents.**

Keywords-machine vision; image motion analysis; features extraction; subtraction techniques

I. INTRODUCTION

Life expectancy in developed countries is increasing and population is aging. However, the quality of life, especially for elderly, is associated with their ability to live independently and with dignity, without having the need to be attached to any person in order to live a normal life and fulfill daily living. According to medical records, falls are the leading cause of injury-related visits to emergency departments and the primary etiology of accidental deaths in persons over the age of 65 years. The mortality rate for falls increases dramatically with age in both sexes and in all racial and ethnic groups, making this one of the most important problems that hinders these people's ability to have such an independent life, making necessary the presence and monitoring of their daily activities by care-givers.

For this reason, a major research effort has been conducted in the recent years for automatically detecting persons' falls. One common way is through the use of specialized sensors, such as accelerometers, floor vibration sensors, barometric pressure sensors, gyroscope sensors, or combination/fusion of them [1][2][3][4][5]. However, most of the previous techniques require the use of specialized wearable devices that should be attached to human body and thus their efficiency relies on the person's ability and willingness to wear them. On the other hand, a more research challenging alternative is the use of visual cameras, which is however, a prime research issue due to the complexity of visual content (illumination variations, background changes and clutter) and the fact that a fall should be discriminated than other ordinary humans' activities, like sitting and bending. Vision-based systems present several advantages as they are less intrusive, installed on building (not worn by users), they are able to detect multiple events simultaneously and the recorded video can be used for post verification analysis. Towards this direction, some works exploit 2D image data like for instance [6][7][8][9]. These works exploit foreground object's shape as well as its vertical motion velocity in order to detect a fall incident. H. Qian *et al.* [10] are based on human anatomy according which each part of the human body occupies an almost fixed percentage in length relative to body height, in order to train a classifier capable six indoor human activities, including fall incidents. However, none of these works exploit 3D information to increase system robustness. A 3D active vision system based on Time of Flight (ToF) cameras is proposed in [11]. Although, this work doesn't take into account the orientation of motion of the moving blob, and the measures that are provided by the camera could be affected by reflectivity objects properties and aliasing effects when the camera-target distance overcomes the non-ambiguity range. Multi-camera systems have been also proposed in [12], to exploit stereo vision. 3D processing, though more robust than a 2D image analysis in terms of fall detection and discrimination of a fall than other daily humans' activities; require high computational cost making these systems unsuitable for real-time large scale implementations.

In this paper, a new innovative approach is presented that exploits, on the one hand, monocular cameras to detect in real-time fall incidents in complex dynamically changing visual conditions and, on the other, it is capable to exploit actual 3D physical space's measures, through camera calibration and inverse perspective mapping, to increase system robustness. Due to its minimum computational cost and minimum memory requirements, it is suitable for large scale implementations, let alone its low financial cost since simple ordinary low-resolution cameras are used, making it

 ISBN: 978-1-61208-226-4

affordable for a large scale. In contrast to other 2D fall detection methods [6][7][8], our system is very robust for wider range of camera positions and mountings, as is proven by the experiments.

The rest of this paper is organized as follows: in Section 2 problem formulation is presented. Section 3 presents 2D and 3D measures for features extraction. In Section 4 experimental results along with the fall detection algorithm are presented, and, finally, Section 5 concludes this work.

II. APPROACH OVERVIEW

Humans' fall incidents can be characterized by motion features that are very discriminative in the fall detection context and in humans' posture. Information about humans' posture can be derived by the actual width-height ratio, and it is valid that in a 3D space this ratio is bigger in value when a fall event occurs than the same ratio with humans in standing position. The most commonly used feature to detect a fall is that of vertical motion velocity, which, besides fall incidents discrimination, is also able to provide useful information about fall intensity and thus possible injuries. Vertical motion velocity V, during a sequence of frames, can be expressed by (1).

$$V = \sum_{i=k-m}^{k} h_a(i) - h_a(i-1) \quad (1)$$

where $h_a(k)$ stands for the actual height of a human in 3D space at the k^{th} image frame (time). Vertical motion velocity is calculated over a time window of length m to estimate the speed of the motion which is also an evident of how severe would be a fall. Index k denotes the current frame for processing. We choose to use actual humans' height, measured in physical world units (e.g., cm, inches), and not their projected height being measured in pixel units, since this yields a more robust performance not be affected by cases where the human is far away or very close to the camera. To measure the actual height, however, we need to exploit 3D information. In addition, actual height can provide information about the moving object, in a way that the system becomes capable to discriminate if the moving object might be a human or something else, like a pet.

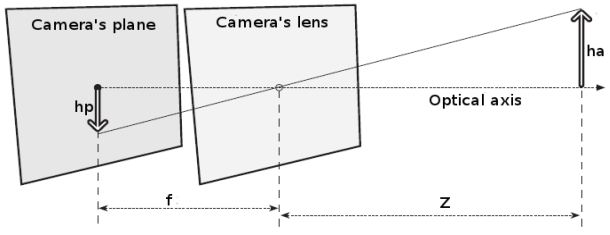


Figure 1: Object in camera's plane and 3D space

Width-height ratio computation requires, firstly foreground extraction (Section III.A), to extract the foreground object, which initially is unknown and secondly information about its left-most, right-most, top-most and bottom-most points, to calculate its projected height and projected width, as explained in Section III.B.

Vertical motion velocity, V, computation requires knowledge of the actual height of foreground object in 3D space. Representation of an object in camera's plane is presented in Figure 1. From Figure 1, it appears that the actual height of foreground object can be given through (2), if camera's focal length f, distance Z between the camera and foreground object and foreground object's projected height h_p are known.

$$h_a = Z\frac{h_p}{f} \quad (2)$$

The projected height can be obtained by the use of a foreground detection algorithm (Section III.B), the focal length can be obtained through camera calibration, as this process provides information about camera's geometry, and the distance between the camera and the foreground object can be obtained through the construction of a reference plane that is the orthographic view of the floor, as explained in Section III.C.

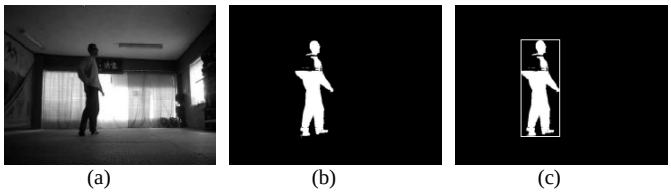
III. 3D MEASURES FOR FALL DETECTION

This Section presents 2D and 3D measures used for features extraction, as well as, the fall detection algorithm.

A. Foreground Extraction

For foreground extraction we use the iterative scene learning algorithm described in [8]. This algorithm, unlike the classic background subtraction techniques, which fail in large scale implementations because of their computational cost and memory requirements, is computationally efficient and has the ability to operate properly in real-time and in complex, dynamic in terms of background visual content, and unexpected environments.

It exploits the intensity of motion vectors along with their directions to identify humans' movements. For motion vectors estimation, the "pyramidal" Lucas-Kanade algorithm [13] was used, which has the ability to catch large motions by using an image pyramid. Motion vectors estimation is followed by the creation of a binary mask in order to indicate areas of high motion information.



(a) (b) (c)

Figure 2: (a) original frame, (b) extracted foreground, (c) minimum bounding box

 ISBN: 978-1-61208-226-4

This information is used as a computationally efficient background/foreground updating mechanism that updates the background at every frame instance by using the intensity of motion vectors within an area. If motion vectors' intensity is greater than a threshold then this area is denoted as foreground, otherwise it is denoted as background.

B. 2D Foreground Object Width-Height Ratio

Width-height ratio estimation requires information about the projected width and projected height of foreground object. In order to estimate the projected width and projected height of foreground object a minimum bounding box that includes the foreground object was created. Figure 2 shows foreground extraction and minimum bounding box for a captured frame. By using the four corners of the bounding box the left-most, right-most, top-most and bottom-most points of foreground object can be estimated and width-height ratio can be expressed by (3).

$$R = \frac{w_p}{h_p} = \frac{p_{rm} - p_{lm}}{p_{tm} - p_{bm}} \tag{3}$$

where w_p and h_p are projected width and projected height and p_{rm}, p_{lm}, p_{tm}, p_{bm} are the left-most, right-most, top-most and bottom-most points of foreground object respectively.

C. Estimation of 3D Measures for Detecting Falls

As mentioned before, vertical motion velocity computation requires camera calibration as this process relates camera measurements with measurements in the real, three dimensional, world according to (4). This relation is a critical component in any attempt to find the dimensions of an object in a three dimensional scene.

$$q = MQ, \quad q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} X \\ Y \\ W \end{bmatrix} \tag{4}$$

where q is a point on camera's plane, Q is the same point in three dimensional world and M is camera's intrinsic matrix. The two parameters, c_x and c_y, have to be introduced to model a possible displacement between the principle point and the center of the imager, while two different focal lengths are used because the individual pixels on a typical low-cost imager are rectangular rather than square. Our approach to camera calibration is derived from [14], which tries to determine optimal values for intrinsic parameters based on image observations of a known target.

Besides camera calibration, vertical motion velocity computation requires the construction of a reference plane that is the orthographic view of the floor. This construction is a perspective transformation, which can be though as a specific case of projective homographies. As described in [15], an affine space R^n is transformed to a projective space P^n by the following mapping:

$$(x_1, x_2, \dots, x_n)^T \rightarrow (x'_1, x'_2, \dots, x'_n, x'_{n+1})^T = (x_1, x_2, \dots, x_n, 1)^T$$

and the inverse mapping, from the projective space P^n to the affine space R^n, is given as:

$$(x'_1, x'_2, \dots, x'_n, x'_{n+1})^T \rightarrow (x_1, x_2, \dots, x_n)^T =$$

$$= \left(\frac{x'_1}{x'_{n+1}}, \frac{x'_2}{x'_{n+1}}, \dots, \frac{x'_n}{x'_{n+1}}\right)^T$$

where $x'_{n+1} \neq 0$.

For a projective space P^n, a projective homography is defined as a nonsingular matrix $H_{(n+1)x(n+1)}$. A point $\boldsymbol{x}$ is projectively transformed to $\boldsymbol{x'}$ as follows:

$$\boldsymbol{x'} = H\boldsymbol{x}, \quad \boldsymbol{x}, \boldsymbol{x'} \in P^n \tag{5}$$

where $\boldsymbol{x}$ denotes pixel coordinates in the homogeneous coordinates and $\boldsymbol{x'}$ is a new position of a pixel in the wrapped output image.

By using perspective transformations, any parallelogram can be transformed to any trapezoid, and vice versa. In our case, we want to transform the camera's plane to a reference plane that represents the orthographic view from above of the camera's plane. Then according to the inverse perspective mapping algorithm described in [16], $\boldsymbol{x'}$ and $\boldsymbol{x}$ can be expressed by the following relations:

$$\boldsymbol{x'} = [x' \quad y' \quad 1] \; and \; \boldsymbol{x} = [x \quad y \quad 1] \tag{6}$$

where x, y, x', y' represent Cartesian coordinates on image plane and reference plane respectively, homography matrix $H = [h_{ij}]$can be normalized so to have $h_{33} = 1$ and through (6) equation (5) is expressed as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{7}$$

This equation that represents a perspective transform requires at least four non-collinear points in order to be solved. By using observations of a known target, a larger set of points can be found and this equation can be solved in a least square sense. The quality of the transformation is measured by the Back Projection Error [16], associated with H(8).

$$E = \sum_{i=1}^{n} \left(x'_i - \frac{h_{11}\, x_i + h_{12} y_i + h_{13}}{h_{31} x_i + h_{32} y_i + h_{33}}\right)^2 + \left(y'_i - \frac{h_{21}\, x_i + h_{22} y_i + h_{23}}{h_{31} x_i + h_{32} y_i + h_{33}}\right)^2 \tag{8}$$

 ISBN: 978-1-61208-226-4

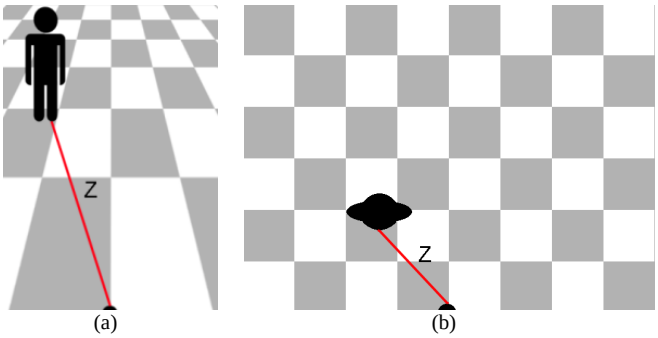


Figure 3: (a) camera's plane, (b) reference plane

Figure 3 shows both camera's and reference planes, while Figure 4 shows input images along with their inverse perspective transformation output images. To approximate the distance Z between foreground object and camera, we use the bottom-most point of foreground object, p_{bm}. As shown in Figure 3(b), on the reference plane the relation between camera's natural units (pixels) and the units of the physical world (cm) is linear and thus distance Z is straightforwardly calculated.

This results in a simple model and a single solution in which a point in the physical world (X, Y, Z) with actual height h_a is projected on the image plane with projected height h_p in accordance with (9). However, the appearance of errors during perspective transformations affects the actual height estimation, as it depends on distance estimation on created reference plane.

$$h_a = Z\frac{(h_p - c_y)}{f_y} \tag{9}$$

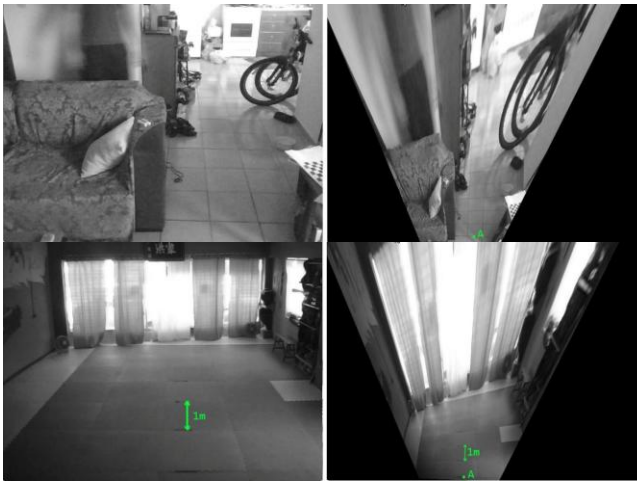


Figure 4: Input images and their inverse perspective transformation mappings

In order to use (1), actual height has to be approximated for every captured frame. Because of the motion of foreground object, errors in the calculation of its height may occur. Let us denote as $\hat{h}(i)$ this approximate height of the foreground object at the current frame of analysis i. In our approach, to reduce accumulation of the approximation errors to the following frames to process we use a heuristic iterative methodology, which updates the foreground height taking into account previous height information and the current one, yielding to a robust approximate solution, denoted as $h(i)$, which is computed by (10). This iterative procedure requires an initial value of $h(i)$ which in our case is set to average height for adult males, e.g., 175cm.

$$h(i) = \lambda h(i-1) + (1-\lambda)\hat{h}(i) \tag{10}$$

where λ is a parameter that regulates the importance of $\hat{h}(i)$ to the iterative procedure. For our experiments, λ is set to 0.8, since this value yields the more reliable performance.

By using this form for every captured frame, $h(i)$ converges to the actual height of foreground object. In order to reduce wrong estimations when a fall event occurs and height is significantly decreases, height $h(i)$ is being updated only if $\hat{h}(i)$ is bigger and smaller than a threshold (in our case ±20 cm). Figure 5 shows the approximation of foreground object's actual height. The gray line represents the approximation of foreground object's actual height for the first 1,000 frames of system operation, while the horizontal line represents its actual height.

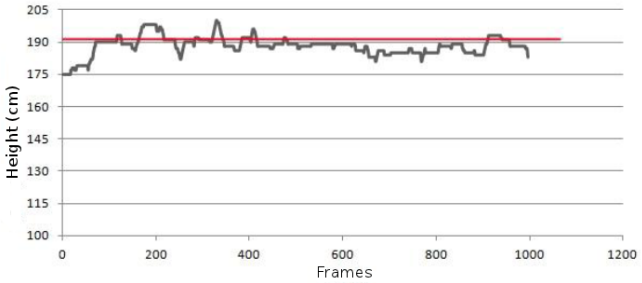


Figure 5: Actual height approximation

IV. EXPERIMENTAL RESULTS

The application was developed on a PC with 4GB RAM and a dual-core Intel processor at 2.1GHz. The camera that was used was a simple USB webcam with 640x480 pixels resolution. The code was written in C by using OpenCV library. By using this hardware, this algorithm operates in real time at 14fps. In quad-core computers, the time can be reached up to 17fps.

 ISBN: 978-1-61208-226-4

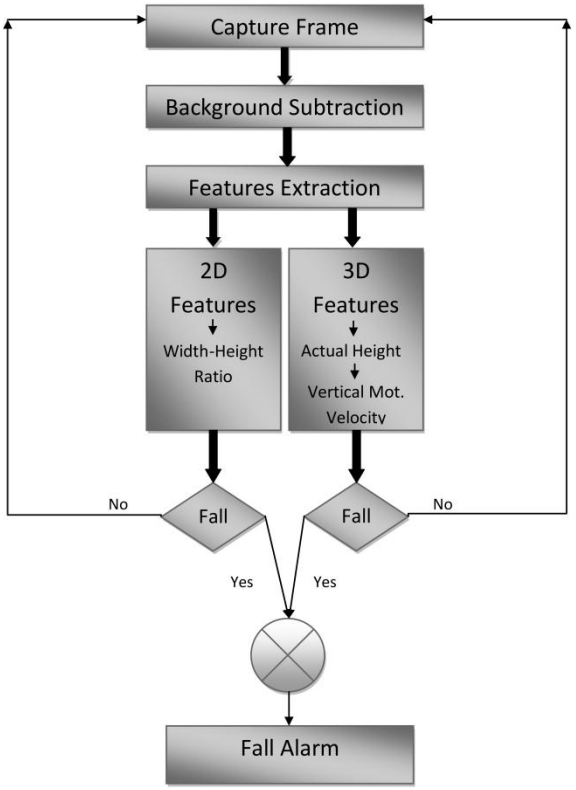


Figure 6: Workflow diagram of presented fall detection scheme

The workflow of the system is presented in Figure 6. For every captured frame, initially, the background subtraction algorithm takes place. The output of this algorithm leads to the extraction of the foreground, and thus, the features that are used by the fall detection algorithm (vertical motion velocity and width-height ratio). The fall detection algorithm, firstly checks if the width height ratio suggests a fall. If this feature suggests a fall, then the vertical motion velocity is calculated and compared with a threshold relative to the real height of the foreground object measured in cm. If this feature suggests a fall too, then a fall alarm occurs. By measuring vertical motion velocity in cm, the performance of the system is not affected by cases where the foreground object is far away or close to the camera.

(a) (b)

Figure 7: (a) Falls in every direction according to the camera position, (b) normal everyday activities that look like a fall, like bending and lying down

During the experimentation process one person simulated falls, in every direction according to the camera position and normal every day activities, that may look like falls; but, they are not real falls; see Figure 7. Because of the nature of the algorithm two different variables affect its performance; the width-height ratio of foreground object and its vertical shifting during a sequence of frames. Figure 8 and Figure 9 describe the performance of the system, concerning on successful fall detection, when the camera was placed at the height of 260cm. In the diagram in Figure 8, we keep constant the value of vertical motion velocity threshold, while in the Figure 9 we keep constant the value of width-height ratio threshold.

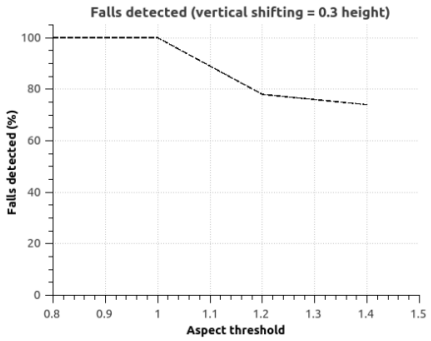


Figure 8: Performance in regard to width-height ratio when camera placed at 260cm

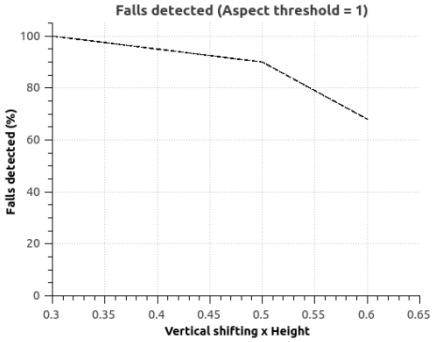


Figure 9: Performance in regard to vertical shifting when camera placed at 260cm

TABLE I: PERFORMANCE WHEN CAMERA PLACED AT DIFFERENT HEIGHTS

Camera's height		**Proposed system**	**System of** [8]
40cm	Falls detected	92.8%	92.8%
	Wrong detections	4	4
220cm	Falls detected	92%	72%
	Wrong detections	6	6
260cm	Falls detected	96%	73%
	Wrong detections	2	2

 ISBN: 978-1-61208-226-4

Table I summarizes its performance and compares it with system performance proposed in [8]. As this system is an expansion of [8], it was expected to perform better. Performance of system proposed in [8] was affected by the camera's position, in contrast, this system performs well for a much wider range of placements that permits a camera mounted in a higher position, favoring fall detection process by providing better coverage with less obstacles inserted into its field of view. This comparison was performed by using the same demo video as input into both fall detection systems.

V. CONCLUSION AND FUTURE WORK

This paper presented a fall detection scheme that exploits 3D measures by using a single monocular camera. The proposed scheme has the capability to operate in real-time and to detect over 92% of fall incidents in complex and dynamically changing visual conditions, while it presents low false positive rate. Its minimal computational cost and memory requirements, let alone its low financial cost since simple ordinary low resolution cameras are used, making it affordable for a large scale.

This algorithm makes the assumption that only one person is present in the scene; so, primary priority in to-do list is its evolution in a way that it will operate properly when more than one person are present in the scene and even in crowded conditions.

Through our proposed scheme, besides the contribution to humans' fall problem, significant measures of a 3D scene can be calculated that can reveal much more information which might be useful in different kind of applications.

VI. ACKNOWLEDGMENTS

The research leading to these results has been supported by European Union funds and national funds from Greece and Cyprus under the project "POSEIDON: Development of an Intelligent System for Coast Monitoring using Camera arrays and Sensor Networks" in the context of the inter-regional programme INTERREG (Greece-Cyprus cooperation) – contract agreement K1 3 1017/6/2011.

REFERENCES

[1] S. Wang, J. Yang, N. Chen, X. Chen, and Q. Zhang, "Human activity recognition with user-free accelerometers in the sensor networks," in *Neural Networks and Brain, 2005. ICNN B '05. International Conference on*, 2005, vol. 2, pp. 1212 – 1217.

[2] M. N. Nyan, F. E. H. Tay, and E. Murugasu, "A wearable system for pre-impact fall detection," *Journal of Biomechanics*, vol. 41, no. 16, pp. 3475–3481, 2008.

[3] T. M. Le and R. Pan, "Accelerometer-based sensor network for fall detection," in *Biomedical Circuits and Systems Conference, 2009. BioCAS 2009. IEEE*, 2009, pp. 265 –268.

[4] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, "Barometric Pressure and Triaxial Accelerometry-Based Falls Event Detection," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 6, pp. 619 –627, Dec. 2010.

[5] 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," *Biomedical Engineering, IEEE Transactions on*, vol. 56, no. 12, pp. 2858 –2867, Dec. 2009.

[6] N. Doulamis, "Iterative motion estimation constrained by time and shape for detecting persons' falls," in *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, New York, NY, USA, 2010, pp. 62:1–62:8.

[7] Z. Fu, E. Culurciello, P. Lichtsteiner, and T. Delbruck, "Fall detection using an address-event temporal contrast vision sensor," in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, 2008, pp. 424 –427.

[8] A. Doulamis and K. Makantasis, "Iterative Scene Learning in Visually Guided Persons' Falls Detection," in *19thEUSIPCO*, Barcelona, 2011, pp. 779–783.

[9] H. Foroughi, A. Rezvanian, and A. Paziraee, "Robust Fall Detection Using Human Shape and Multi-class Support Vector Machine," in *Computer Vision, Graphics Image Processing, 2008. ICVGIP '08. Sixth Indian Conference on*, 2008, pp. 413 –420.

[10] H. Qian, Y. Mao, W. Xiang, and Z. Wang, "Home environment fall detection system based on a cascaded multi-SVM classifier," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, 2008, pp. 1567 –1572.

[11] G. Diraco, A. Leone, and P. Siciliano, "An active vision system for fall detection and posture recognition in elderly healthcare," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, 2010, pp. 1536 –1541.

[12] N. Thome, S. Miguet, and S. Ambellouis, "A Real-Time, Multiview Fall Detection System: A LHMM-Based Approach," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1522 –1532, Nov. 2008.

[13] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," presented at the IJCAI81, 1981, pp. 674–679.

[14] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 1106 – 1112.

[15] B. Cyganek and J. P. Siebert, "Front Matter," in *An Introduction to 3D Computer Vision Techniques and Algorithms*, John Wiley & Sons, Ltd, 2009, p. i–xx.

[16] A. Bevilacqua, A. Gherardi, and L. Carozza, "Automatic Perspective Camera Calibration Based on an Incomplete Set of Chessboard Markers," in *Computer Vision, Graphics Image Processing, 2008. ICVGIP '08. Sixth Indian Conference on*, 2008, pp. 126 –133.

 ISBN: 978-1-61208-226-4