Definition And Performance Evaluation Of A Robust SVM Based Fall Detection Solution

Imen Charfi, Johel Miteran, Julien Dubois
Laboratory Le2i , UMR 6306
University of Burgundy
21000 Dijon, France
Email: johel.miteran@u-bourgogne.fr

Mohamed Atri, Rachid Tourki
Laboratory of E3E
University of Monastir
Faculty of Sciences of Monastir, Tunisia
Email: mohamed.atri@fsm.rnu.tn

Abstract—We propose an automatic approach to detect falls in a home environment. A Support Vector Machine based classifier is fed by a set of selected features extracted from human body silhouette tracking. The classifier is followed by filtering operations taking into account the temporal nature of a video. The features are based on height and width of human body bounding box, the user's trajectory with her/his orientation, Projection Histograms and moments of order 0, 1 and 2. We study several combinations of usual transformations of the features (Fourier Transform, Wavelet transform, first and second derivatives), and we show experimentally that it is possible to achieve high performance using a single camera. We evaluated the robustness of our method using a realistic dataset. Experiments show that the best tradeoff between classification performance and time processing result is obtained combining the original data with their first derivative. The global error rate is lower than 1%, and the recall, specificity and precision are high (respectively 0.98, 0.996 and 0.942). The resulting system can therefore be used in a real environment. Hence, we also evaluated the robustness of our system regarding location changes. We proposed a realistic and pragmatic protocol which enables performance to be improved by updating the training in the current location, with normal activities records.

I. INTRODUCTION

According to the Center for Research and Prevention of Injuries report [1], fall-caused injuries of elderly people in UE-27 are five times as frequent as other injuries causes which reduces considerably their mobility and independence. Among the diverse applications of computer vision systems, object detection and event recognition are of the most prominent related recognition and motion analysis, that is, researchers had the idea to spread it in fall detection. The fall event, extracted automatically from the video scene represents itself, crucial information that can be used to alert emergency. In this context, visual information on the corresponding scene is highly important in order to take the “right” decision. Therefore, video compression may be included into the acquisition system to reduce data-bandwidth. Meanwhile, detecting such particular situations allows the video compression to be controlled. For instance, the compression can be reduced after a fall to provide more details on the scene or the different compression rates can be applied on the background and the regions of interest. Detecting falls requires two main tasks. First, we have to use a robust method for human body tracking. Second, a robust feature extraction method should be proposed which describes the user’s behaviour to discriminate falls from other activities. In this paper, we aim a real time fall detection system that not only insures privacy protection but also overcomes occlusions problem and insures adaptive video compression. Our objective is to design a fall detector system that acquires realistic video data, detects falls and alerts emergency by sending the relevant part of the video. Thus, the main contributions of this paper are to propose to find the best combination between several features and transformation of features, and a robust human body tracking algorithm coupled with Support Vector Machine (SVM) based classification system. Moreover, we propose a pragmatic and novel study which measure the impact of the environment (i.e. location change) between the learning phase and the detection phase. The proposed dataset (250 videos), which will be publicly available, is also a contribution of this study.

Literature review is presented in section II, followed in section III by the description of the full detection and classification method. Classification features and their transformation are presented in section IV. In Section V, the dataset is presented and we define three protocols of experimental evaluation of our method. Section VI contains experiments and results while a conclusion is presented in section VII.

II. STATE OF THE ART

In [2], Noury and al. stated many fall detection techniques. Most existing fall detection systems are based on sensor devices such as accelerometers, microphones and cameras. Since our aim is to combine event detection and image compression, we will focus here only on video (and mono camera) based fall detection system.

the magnitude and the orientation of the movement using the Gradient Image. They also analyzed the bounding box changes. Foroughi and al. [8] proposed a fall detection method combining the variations of the best-fit approximation of an ellipse around the human silhouette, the projection histograms and the changes of head pose as features for a Support Vector Machine based classifier (SVM). In the vision part of Toreyin and al. work [9], the aspect ratio of the moving region detected with a standard camera is analysed by the motion model with Hidden Markov Model (HMM). Tao and al. [10] used aspect ratio of the moving object’s bounding box, and Anderson and al. [11] adopted the width to height ratio of the silhouette bounding box and the off-diagonal term from the covariance matrix as the features to determine whether fall incident occurs. These features need to be extracted from the silhouette to train and perform classification with HMMs for temporal pattern recognition. In [12], Rougier and al. analysed with the GMM classification method the shape’s deformations through video sequences acquired from only one uncalibrated camera and they improve the performance by combining results of four uncalibrated cameras mounted in different points of view. We will focus in this paper on a mono camera system, showing that it is possible to combine efficiently some of the previously used features, with standard transformations and a powerful classifier in order to obtain a useful detector, regardless the direction of the fall.

III. THE FALL DETECTION OVERVIEW

The main steps of our detection system (figure 1) are as follows:

- Motion detection and tracking, using foreground/background segmentation. This step ends using morphological operators (erosion and dilatation) allowing to remove segmentation artifacts,
- Feature extraction, such as aspect ratio of the bounding box, ellipse orientation, etc,
- Feature transformation, capturing variations of previous features, since a fall is characterized by large movement and change of the human shape,
- Image level fall detection using SVM : we used a Support Vector Machine based system to classify each image. The SVM decision is followed by a filter (majority vote in a moving window of size $n_w = 5$) removing isolated decisions. The final decision is given at the frame rate.
- Slot image level fall detection using a novel metric to evaluate the performances introducing tolerance in the final decision.

SVM is a universal learning machine developed by Vladimir Vapnik [13] in 1979. The SVM performs a mapping of the input vectors from the input space (initial feature space) $R^d$ into a high dimensional feature space $Q$; the mapping is determined by a kernel function $K$. It finds a linear decision rule in the high dimensional feature space $Q$ in the form of an optimal separating boundary, which leaves the widest margin between the decision boundary and the input vector mapped into $Q$. A Radial Basis Function SVM (RBF) was used in this study:

$$K(x, y) = \exp \left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

(1)

Mapping the separating plane back into the input space $R^d$, gives a separating surface which forms the following nonlinear decision rules:

$$C(x) = Sgn \left( \sum_{i=1}^{N_v} y_i\alpha_i \cdot K(s_i, x) + b \right)$$

(2)

where $\alpha_i$ are the Lagrange coefficient obtained during the optimisation process. The separating plane is constructed from those input vectors, for which $\alpha_i \neq 0$. These vectors $s_i, i = 1, ..., N_v$, are called support vectors and reside on the boundary margin. All results with SVM were obtained with a home made software based on the LIBSVM library [14]. We used the default values of the SVM parameters defined in this implementation, excepted for the RBF parameter $\sigma$ which was automatically tuned in order to optimize the classification rate.

![Fig. 1. Overview of the fall detection method.](image)

Fig. 1. Overview of the fall detection method.

IV. FEATURE EXTRACTION

A. Low level features

For moving object detection, we applied the background subtraction method from L. Li [15] under the Bayesian decision that performed on many difficult videos of both indoor and outdoor scenes. The algorithm consists of four parts: change detection, change classification, foreground object segmentation and background learning and maintenance.

![Fig. 2. Low level features representation.](image)

Fig. 2. Low level features representation.

From this moving object detection, we defined an initial set $F$ of 14 features, containing, as depicted in figure 2,
height and width of the bounding box \((B_h, B_w)\), aspect ratio of the bounding box \((B_r)\), coordinates of the center of the bounding box \((C_x, C_y)\), coordinates of the center of the best fitting ellipse \((E_x, E_y)\), horizontal and vertical projection histograms \((H_{ph}, V_{ph})\) [16], moments of order 0, 1 and 2 \((m_{00}, m_{11}, m_{02}, m_{20})\) of the moving detection and orientation of the Ellipse \((E_o)\). The moving detection and feature extraction are implemented in C++ using the OpenCV Library [17].

**B. Transformations and combinations of low level features**

The fall is often characterized by high variations of the previous features. In order to include these variations in the classification process, it is possible to use several well known transformations:

- The first derivative \((FD)\), representing the velocity.
- The second derivative \((SD)\), representing the acceleration.
- The Fourier transform, using the standard fast algorithm \((FFT)\).
- The Wavelet transform \((W)\).

We combined the original data \((DB)\) with coefficients of these several transformations that we calculated on a given size window. Indeed, the windowed Fourier Transform combined with SVM classifier has been used with success in several cases, such as 2D object recognition by Smach [18], cancer detection by S. Parfait [19]. It is also well known that the wavelets coefficients can be used as features for pattern recognition [20]. We used in the next experiments \(D_4\) Daubechies orthogonal wavelets [21].

**V. Dataset and evaluation protocols**

**A. The Dataset**

We acquired 250 video sequences in four different locations, 192 containing falls, and 57 containing several normal activities, motions, body transfers, for instance from a chair to a sofa. The frame rate is 25 frame/s and the resolution is 320x240 pixels. The video data illustrates the main difficulties of realistic video sequences that we can find at an elderly home environment. Our video sequences contain variable illumination as well as shadows and reflections that can be detected as moving objects, and typical difficulties like occlusions or cluttered and textured background. The actors wearing different clothes with different colors and texture performed various normal daily activities (walking in different directions, sitting down, standing up, crouching down, housekeeping, moving a chair) and falls (forward falls, falls when inappropriate sitting-down, loss of balance). All activities are taken in different directions without taking in account the camera point of view as shown in figure 3. Each video is manually annotated: the numbers of the frames defining the beginning and the end of the fall are recorded, and the localisation of the body is manually defined for 130 videos using a rectangle. This manually defined bounding box allows us to evaluate the classification features independently from the automatic body detection. The dataset was built from different locations (named “Home”, “Coffee room”, “Office” and “Lecture room”) in order to evaluate the robustness of our fall detection method against location change between training and testing. Some examples are depicted figure 3.

**B. Evaluation protocols**

We evaluate the fall detection process firstly at the SVM output level (measuring the error rates, computed from the well and misclassified frames) and secondly after the filtering, computing the sensitivity, the specificity, the accuracy, the recall and the final classification error rate. Since the goal of our work is to perform the fall detection in real time and in a continuous way, we defined an evaluation protocol based on a moving analysis window. The window size \(w\) has been fixed regarding the average fall duration, which is 14 images, with a standard deviation of 1.4. We fixed then \(w = 18\) images. All images of the whole video sequences are classified, regarding information contained in the last \(w\) images. The resulting latency is less than 1s, which is acceptable for a real time fall detection system. If a beginning of a fall is detected, the system waits \(w\) images before to continue the analysis. A good detection will generate only one true positive (TP). A TP occurs when the number \(N\) of successive decisions “Fall” of the classifier is higher than a threshold \(Z\), and if this detection occurs close to the ground truth fall. We define the distance \(D\) between the theoretical fall and the detected fall, measuring the number of images between the beginning of the ground truth fall and the detected fall. If \(D < D_{max}\) and \(N > Z\), where \(D_{max}\) is a parameter of the system, a true positive is considered (figure 4). If a fall is not detected, the classification process continues. A non detected fall will generate several false negative (FN) depending on the duration of the fall. In the next experiments, \(Z = 11\) and \(D_{max} = 18\).

In order to evaluate the robustness of our fall detection method against location change, we defined three protocols P1, P2 and P3:

- For P1, the training and the test set were built with videos from the “Home” and “Coffee room” subsets.
- For P2, the training set was built using the videos from “Coffee room” and the test set was built using “Office” and “Lecture room” subsets.
- For P3, the training set was built using the videos from “Coffee room” and some videos without any fall of “Office” and “Lecture room”. The test set was built from “Office” and “Lecture room”.

**VI. Experimental results**

For all the feature selection experiments, the manually defined bounding box has been used, in order to consider only the intrinsic discriminative capacity of each feature, independently from the body detection. Performance of the system including the automatic body detection are presented at the end of this section. We studied the influence of the width \(w\) of the Fourier Transform and the \(N_w\), number of wavelet coefficients. The optimum was found for \(w = N_w = 32\). These parameters are fixed for all next experiments. For computational time reasons, it is not possible to use an
exhaustive feature selection method (the total number of low level features is $d=32\times14\times5=2240$). We decided to select firstly the best combinations of transformations and secondly the subset of low level features using selected combinations.

### A. Features selection

In order to optimise the final classification rate, we evaluate several combinations of transformations mentioned before (using Protocol P1). We evaluated the result at the SVM output level (so the error rate is computed at image level), and after the filtering, and the final decision using the previously described protocol. Since the number of combinations of transformations is relatively low (31 possible combinations), it was possible to use an exhaustive approach for this step, where the initial number of low level features is fixed to 14. The results are respectively presented in the table I and II.

Four combinations allow obtaining a global error equal to zero. In order to optimise the global computation time, we used in the next experiments these combinations. For a real-time implementation, the simplest combination can be retained, i.e. the original data combined with the first derivative transformation.

Then, we selected the features from the set $F$ using a standard SBFS approach, minimizing the error at the SVM output level. The final set $F_s$ of features used for next experiments is thus:

$$F_s = \{C_y,m_{00},E_x,B_r,E_o,m_{02},H_{ph}\}$$

Using the 7 low level features selected after applying the two transformations, the final dimension of the feature space is $d=32\times7\times2=448$.

### B. Performance of the automatic body detection

We applied the same method to the four best combinations found in the previous study to evaluate the full automatic detection and classification performance. Some examples of manual and automatic detection are presented figure 5.

The results, respectively presented in the table III and IV are very close to the results obtained using manual annotation, proving also the robustness of the automatic detection. In the best case, only one fall was not detected from a total of 54
C. Robustness to location change

As mentioned previously, we evaluated our fall detection method using the images acquired in several locations. The three protocols P1, P2 and P3 were evaluated using the combination of tranformations of the 7 low-level features selected in the section VI-A.

The results are presented in the table V. The global error rate obtained with P2 (0.46%) slightly increases compared to protocol P1 (0.38%) where all the videos where acquired in the same location. However, the results of protocol P3 (global error rate of 0.32%) show that it is benefic to add videos, regardless the direction of the fall or the position of the body in the scene, and there were three false positives.

In [12], Rougier obtained a global error rate of 4.6% whereas our error rate is 0.38%, however these rates are not really comparable since the test protocols are different: we evaluated the performance at the slot level, and Rougier evaluated the performance at the video level, which is not really applicable in the real world where video is acquired in a continous way. Moreover, the choice of the orientation of the camera of Rougier (very wide angle lens and camera fixed on the ceiling) is not suitable to our feature extraction. We evaluated our algorithm using their database, and obtained a global error rate of 4%, with a recall of 73% and a precision of 97.7%. The difference of performance comes mainly from the difference of camera setup: our camera was set at only 2m from the ground and used a narrower angle lens. Our setup better captures vertical motion which characterizes falls. This lead us to evaluate the robustness of our method to location change between training and testing, since in real application the final user it is not always able to train the system using fall simulation.

C. Robustness to location change

As mentioned previously, we evaluated our fall detection method using the images acquired in several locations. The three protocols P1, P2 and P3 were evaluated using the combination of tranformations of the 7 low-level features selected in the section VI-A.
specificity and precision are high (respectively 0.98, 0.996 and 0.942). The resulting system can therefore be used in a real environment. Hence, we also evaluated the robustness of our system regarding location changes. We proposed a realistic and pragmatic protocol which enables performance to be improved by updating the training in the current location, with normal activities records. The results are obtained using only one camera, and we plan in the near-future to implement the adaptive compression scheme described in the introduction.

**ACKNOWLEDGMENT**

The authors would like to thank EGIDE agency for funding, project Utique20801WB.

**REFERENCES**


