A method for real-time detection of human fall from video

M. Kreković, P. Čerić, T. Dominko, M. Ilijaš, K. Ivančić, V. Skolan i J. Šarlija
Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva
Zavod za elektroničke sustave i obradbu informacija
Zagreb, Hrvatska
E-mail: miranda.krekovic@fer.hr

Abstract - In this paper we present a method for real-time detection of human fall from video for support of elderly people living alone in their homes. The detection algorithm has four steps: background estimation, extraction of moving objects, motion feature extraction, and fall detection. The detection is based on features that quantify dynamics of human motion and body orientation. The algorithms are implemented in C++ using the OpenCV library. The method is tested using a single camera and 20 test video recordings showing typical fall scenarios and regular household behaviour. The experimental results show 90% of human fall detection accuracy.

I. Introduction

Thanks to the increase of the general standard of living and improving health care in developed countries in the past 50 years, life expectancy lengthened by 30 years, and thus increased the percentage of elderly in the general population. According to the census of 2001. year, people over 65 make up 14.35% of the population. Different studies have shown that it is in social and financial interests to enable older people to remain living at their own homes as long as possible.

Fear of a fall is one of the most prominent reasons why elderly people tend not to live alone. There are around 60000 reported falls per year, causing the cost of treatment of 400 million dollars. The risk for serious injuries is increased if the person remains unconscious or immobilized after the fall because of their inability to call for help. Because of this, devices that automatically detect the fall are in the focus of interest. Their goal is to shorten the time between the fall and the arrival of aid, increasing the likelihood of successful treatment, as well as reducing the cost of treatment. There are devices that detect the decrease in the acceleration and its direction, while some use the gyroscope to determine the position of the person's body. But the biggest drawback of such technology is that these devices are awkward and uncomfortable to wear and require frequent battery changes.

Intelligent video surveillance is the simplest way of detecting the fall and timely acquiring aid. Cameras can be installed in each room in the house and, beside the detection of falls, can be used for other purposes, such as ensuring homes of unwanted guests by serving as an alarm system. The goal of this project is to build a computer vision system which uses surveillance cameras and a video analysis method to detect a fall in the room. The project aims to detect the fall of a person in a room using one or more surveillance cameras. The cameras are connected to the computer which performs real-time processing and analysis of acquired video and shows the actual video footage. The fall detection event will be used to alarm the family and/or appropriate emergency services.

II. Problem Description

There are many problems that occur in the automatic detection of falls using video surveillance cameras (selecting an optimum resolution of a video, changing of room lighting conditions cause difficulties in image processing, activities that people do every day with certain movements that resemble the fall). Therefore, it is almost impossible to build a video system with a 100% detection accuracy. Our goal is to build a system that will accurately detect falls in at least 85% of cases.

One of the first problems is the selection of surveillance cameras. Video recordings taken with expensive, high quality cameras (e.g. Axis network video cameras) contain the high level of video compression (e.g. MPEG4) that can generate a non-existent objects in the picture. If the room is illuminated by natural light, change of night and day, weather conditions and artificial lighting will significantly affect the changes in light. These changes should be taken into account while separating moving objects from a static background. In addition to these changes in light, changes in the background may occur due to short-term movements of objects that comprise for example shifting chairs or relocation of a book. Such facilities must be integrated into the background. Therefore it is best to dynamically build a background during certain time intervals. Another problem with the lighting is the occurrence of reflection (the colors are usually lighter than usual) and the shadow of moving objects (the colors are darker than usual). They can cause a problem if they are detected as moving objects. Generally, the biggest problem in the segmentation will be insufficient contrast and diversity between the person and the background, for example, a person in a white dressing gown walking in front of a white wall or a colorful object moving in front of a crowded and textured background.

In these cases, the person will blend with the background and his/her fall will not be detected. Another significant problem is camera position. Large items that are in the room, such as furniture (wardrobe, bar, sofa, table, etc.) may block the view of some parts of the rooms.

In case a person falls behind something large, the fall will not be detected. Also, in the irregularly shaped or very large rooms, one camera is not enough to cover the desired area and those outside the camera's sight will not be recorded.

In the fall detection we use the method in which the silhouette of a person is described using an ellipse whose parameters are measured and calculated over time. In the event of a fall, the ellipse will change its shape, parameters will take on values characteristic for a fall, and the fall will be detected. However, some activities such as squat, lying, suction, stretching, exercise, etc. can have similar parameter values. In such situations, a totally different activity can be mistaken for a fall.

The algorithm for the fall detection is designed to measure the parameters and detect a fall of the largest object that moves. This is because smaller areas often do not represent people, but errors in the segmentation of the background or temporarily moved objects from the background. In case there is more than one person in the room at the same time, the system will detect a fall of only the largest of them. But this is not a problem because others present can provide assistance without signaling the system.

III. System Architecture

The main goal of the project is fall detection using surveillance video cameras. The camera should be placed in the corner of the room, so it's field of vision could capture the entire room. The fall detection consists of several basic activities: separating moving objects from the background, calculating the parameters for these areas and finally, fall detection itself.

A. Background Subtraction

Moving objects are extracted from the motionless background using an OpenCV algorithm BackgroundSubtractorMOG2 described in [1]. The algorithm uses Gaussian mixture model, as it proved to be very effective in previous work. A background model is created to monitor the difference between foreground and background pixels. Each pixel that does not fit into this model is considered a foreground pixel, part of a moving object. Additional parameters are adjusted to gain a better image for further analysis. The result is a series of binary images in which pixels with zero value represent the background (black areas), and pixels with the value equal to one are moving object (white areas).

B. Elliptical Approximation of an Object

Fall detection is based on the movement coefficient and human shape deformation. An ellipsoid contour is superimposed over the human shape to follow changes more easily. All bodies are outlined in the binary image using the OpenCV findContours function. Then, using the function contourArea, the largest contour is singled out. The largest contour in the figure represents a human. Finally, the function fitEllipse calculates the ellipse that fits best a set of contour points. It returns an ellipse defined by its center (x, y), orientation (angle θ) and lengths of large (a) and small (b) axis. The best approximation of an ellipse is obtained using the least squares method which minimizes the square error of approximation.

C. Movement Coefficient

Motion History Image (MHI) is used to calculate the movement coefficient. It shows recent motion in the image. Motion history is updated as following:

$$\mathrm{mhi}(x, y) = \begin{cases} \text{timestamp}, & \text{foreground}(x, y) \neq 0 \\ 0, & \text{foreground}(x, y) = 0 \text{ and} \\ & \quad \text{mhi} < (\text{timestamp} - \text{duration}) \\ \mathrm{mhi}(x, y), & \text{else.} \end{cases}$$

Foreground is the result of a background subtraction. Timestamp represents the current time in milliseconds or other units. Each pixel in the MHI where the movement occurred at the point of time takes the value of timestamp. If there is no action at that point of time the value of MHI remains intact. Duration represents the maximal interval of the motion track in the same units as timestamp. Every value that is lower than the difference between timestamp and duration is set to zero.

The more recent moving pixels are seen brighter in the MHI image. To quantity the motion of the person we have to calculate the movement coefficient, C_{motion}, using the formula:

$$C_{motion} = \frac{\sum_{pixel(x,y)} mhi(x,y)}{\sum_{pixel(x,y)} foreground(x,y)}$$

This coefficient is scaled to a percentage of motion. Value zero means that there is no movement and value equal to one corresponds to full movement.

D. The Algorithm

Fall detection algorithm consists of three steps, as shown in Figure 1. The first step is movement coefficient analysis. Value larger than 0.8 indicates possible fall. Sudden movements or walk perpendicular to the camera can also result with high movement coefficient, so it is necessary to check other parameters to distinguish fall from normal movement.

The second step of the algorithm is human shape analysis. When a person falls, there is a sudden change of orientation and ratio of major and minor axes of the ellipse which approximates that person. In that case the value of standard deviation of the angle σ_θ and standard deviation of the axes ratio $\sigma_{a/b}$ will be high. When person walks around the room, those values will be low. A large amount of movement is considered to be a fall if σ_θ is larger than 15° and $\sigma_{a/b}$ larger than 0.9.

The last step of the algorithm is used to check whether the person remains motionless on the floor for a few seconds after falling. The coefficient of movement has to be less than 0.2 between the twentieth and the thirtieth video frame from the suspected fall time. This corresponds to interval from 0.83 to 1.25 seconds after the fall and it means that observed object is no longer moving. Standard deviation of the y - coordinate of the center of the ellipse which approximates the object, σ_y, is always calculated for the last ten video frames. Over the next ten images since the fall, the standard deviation σ_y has to be larger than one eighth of the number of pixels per image width. This means that there has been a shift of the object in y - axis.

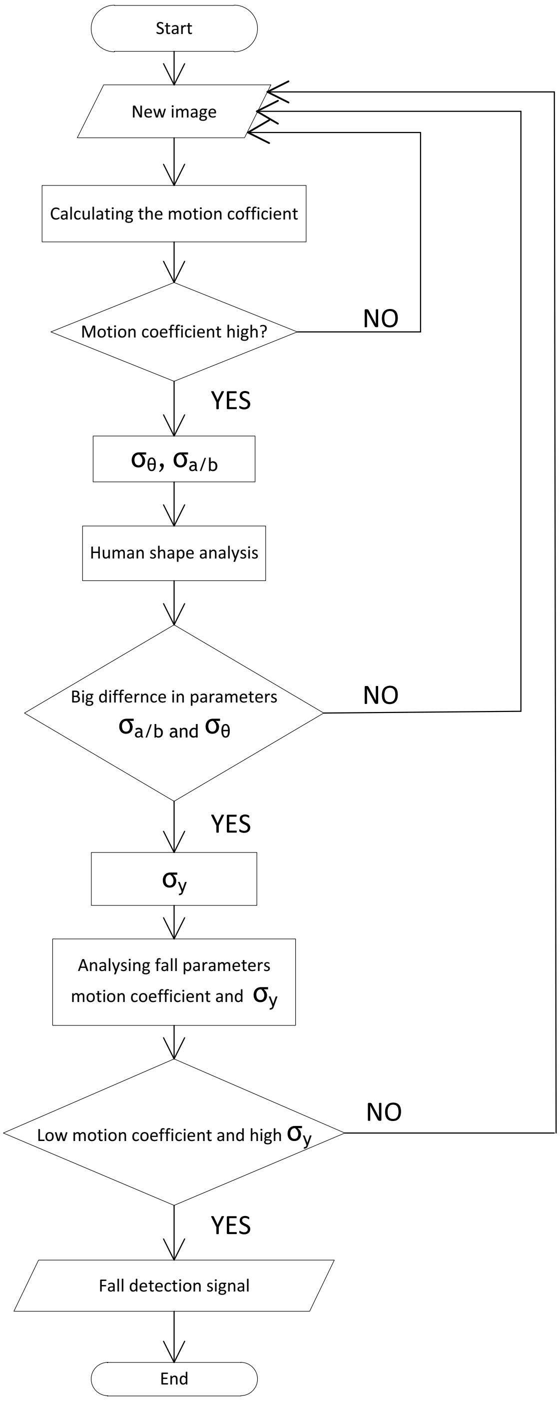


Figure 1. Flowchart

Figure 2. The result of background subtraction

Figure 3. The result of elliptical approximation of an object

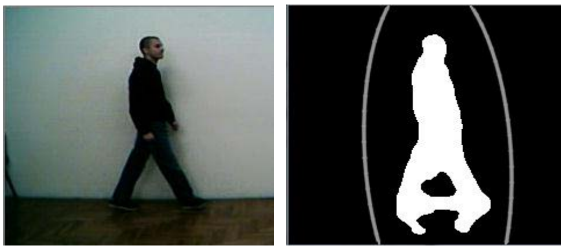
Figure 4. Motion history image

IV. RESULTS

All algorithms implemented in the project were written in C + + using the OpenCV library. It allows the system to work in real time using a webcam. The system supports the monitoring of only one camera.

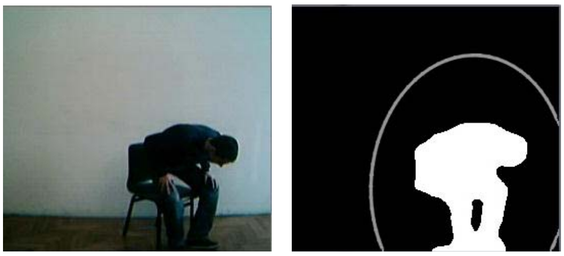
Twenty different shots were used for testing the fall detection algorithm. The most images show everyday situations when people stay in room. Since the algorithm works by calculating the movement coefficient and ellipse approximation of the object, the key test specimens are those in which a person is doing sudden movements, jumping, sitting or lying down. The algorithm has successfully detected a fall in eighteen test samples which shows an efficiency of 90%. Two sections that are used to detect the fall (movement coefficient and ellipse parameters) successfully collaborate and complement their individual shortcomings. One of the examples that illustrate this is the situation where a person is lying down.

The change in the ellipse condition is met but the amount of movement is insufficient, which avoids the false fall detection. Out of twenty different video sequences, the algorithm did not show a satisfactory result for two. The first video sequence shows the situation of a person who is completely still and after some time immediately falls. In the second shot, the person falls while entering the frame. Algorithm does not work properly in these situations because it fails to calculate the corresponding movement coefficient. This is the result of an insufficient number of frames in which a person is not blended in the background.



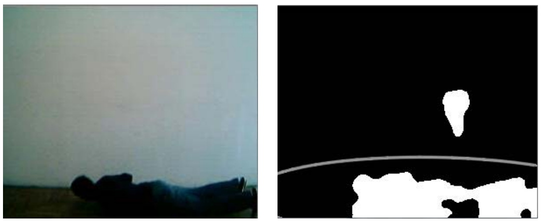
$C_{motion} = 0.71$, $\sigma_\theta = 2.49$, $\sigma_{a/b} = 0.38$

Figure 5. Walking – values of parameters are low



$C_{motion} = 0.29$, $\sigma_\theta = 4.43$, $\sigma_{a/b} = 0.23$

Figure 6. Sitting – angle deviation is high but rest parameters are low



$C_{motion} = 0.9$, $\sigma_\theta = 51.97$, $\sigma_{a/b} = 0.91$

Figure 7. Fall – all values are high

V. CONCLUSION

In this paper, we present a method for automatic fall detection using a system of camera for video surveillance. The developed method allows real time fall detection using the original algorithm. The algorithm has several stages that are performed sequentially. An output signal, which informs whether a person has fallen or not, is obtained by analysing the parameters. Applications other than the basic, can be, with some upgrades, multiple. One of the uses could be that the system is connected to a local health center in order to give the staff an insight into the situation and, if necessary, send assistance to people who have fallen. In order for the system to be used for commercial purposes, it is necessary to see how it behaves with a wider range of conditions and circumstances that can occur in households which would use it. This way the percentage of successful operation of the system remains high.

Although this algorithm gives satisfactory results and works relatively good in real time, there is still room for improvement. Given that all the test specimens are recorded under controlled conditions, which primarily refers to the constant illumination of the room and significant difference in the colors of the person's clothing and the background. The upgrade should primarily be performed on the background subtracting algorithm to make the whole algorithm become more robust.

Because of the complexity of the used functions, more time is required for calculating data, making the system run a little slower in real time. The first step in solving this problem is to find the section of code that is running the longest. If possible, this piece of code should be optimized and its execution time reduced.

Another way of solving this problem is to divide the algorithm into logical units that perform a specific job. Each unit gets its own thread, which deals with a specific task. All threads operate in parallel. Such a way of using thread parallelism is called a pipeline. Ideally, if all threads perform tasks that are approximately the same duration, the parallelism will be maximized.

Additionally, the system can be expanded in a way that it can monitor multiple cameras. Then cameras could be placed in every room of the house, and the system could be used for commercial purposes.

LITERATURE

[1] Rougier, C., St-Arnaud, A., Rousseau, J., Meunier, J., *Video Surveillance for Fall Detection*. Department of Computer Science and Operations Research, University of Montreal, Canada

[2] Spehr, J., Gövercin, M., Winkelbach, S., Steinhagen-Thiessen, E., Wahl, F.M. *Visual Fall Detection in Home Environments*.

[3] Vishwakarma, V., Mandal, C., Sural S. *Automatic Detection of Human Fall in Video*. Indian Institute of Technology, Kharagpur, India

[4] McIvor, A. M. *Background Subtraction Technique*, Reveal Ltd, Auckland, New Zealand

[5] Toyama, K., Krumm, J., Brumitt, B., Meyers, B. *Principles and Practice of Background Maintenance,*Microsof Research, WA 98052

[6] Davis, J. W., Bobick, A. F. *The Representation and Recognition of Action Using Temporal Templates*, CVPR97, 1997

[7] Davis, J. W., Bradski, G. R. *Motion Segmentation and Pose Recognition with Motion History Gradients*, WACV00, 2000

[8] Z. Živković, *Improved adaptive Gausian mixture model for background subtraction, international Conference Pattern Recognition*, UK, August, 2004