Fall Detection From Video Surveillance

JIN HE
HEXJ03058504
hejin.udem@gmail.com

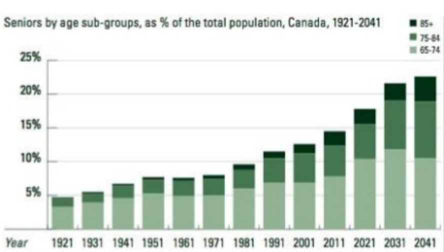
Abstract

This paper propose a approach to detect falls of a person who lives alone based on the analysis of images captured by multi-cameras set up around space. The system use cameras to capture the motion of a person in real time, extracting the features from each frame and decide whether there is a fall or not by voting from all the cameras simultaneously. The techniques I used is based on Baysian decision theory and unsupervised learning. This system can give a good result with sensibility 84.5% and specificity 85.1%.

Key words – fall detection, multi-cameras, minimum-error-rate classifiction, image processing

1. Introduction

Nowadays, Canada is facing the growing of senior population according to the statistics of Sante Canada in 2002, the figure following shows us that one Canadian out of eight was order than 65 years in 2001 and this ratio will increase constantly up to one out of five in 2026. We have to face such problem: more and more elderly will take the risk of falling in private house without care.



(Santé Canada, 2002)

Figure 1. The proportion of elderly

Usual way to detect falls is to use some wearable sensor like accelerometers and help button but the order people always forget to wear them or are not able to move after falls. The advantage of video surveillance can overcome these problems: Firstly, elderly don't have to wear any application, cameras could accord all the activities of targeted person automatically. Secondly, there is less modification of envoirnment. Last, video surveillance could provide us with high accuracy(87%) of fall detection.

2. Recognition Method Outline

To build a classifier, there are 4 processes in general.

1) Establish an adaptive background model
2) Obtain foreground segmentation
3) Image processing and extracting features
4) Training classifier

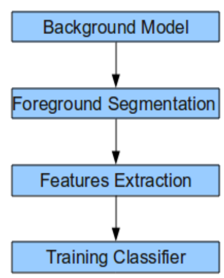


Figure 2. Building a classifier

To decide a classification, there are 8 steps in general.

1) Input new image sequences
2) Build background model
3) Input test images
4) Obtain foreground segmentation
5) Extracting features
6) Tested by classifier
7) Make votes
8) Final decision

At step 6 when we detect a possible fall, but we don't know whether it is a fall or just a lying down to take a rest, thus we should judge it by adding another condition: the changes of status . If it is unchanged or little changed. it must be a fall, otherwise it is not fall.

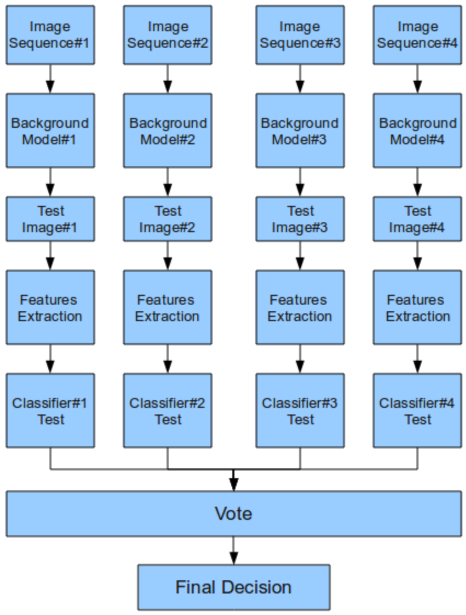


Figure 3. The process of detection

2.1 Feature Extraction

The key step is to establish initial background model, it affects directly the quality of data extracted. First of all, selecting a certain quantity of static background randomly (there is no people in the scene) and then calculating the average of them for the purpose of reducing noises generated from the cameras. The background model should be adaptive because of the changes of the scenes.

Next step is to obtain the foreground binary segmentation. In this step we subtract background model from input image and get some segmentation fragments, we select the biggest one as our target foreground. It could be a segmentation along with a lot of noise. In order to remove these noise I use function cvErode() and cvDilate() that provided by opencv (Open Computer Version Library) .

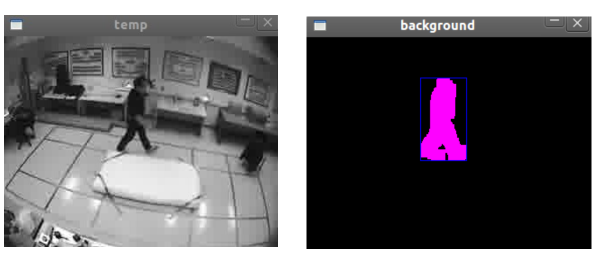


Figure 4. Input image and segmentation

After obtain segmentation that we need, the following step is to extract features of the segmentation. For choosing features we should take two factors into account: 1) the number of the features 2) the correspondence between the features. It is true that the more features we use , the higher precision system we could build , however we have to consider the complexity of computation: $n \propto m^d$ d is the number of the features. If d is big enough, the cost of computation is going to be larger. Another factor we should think is the correspondence between different features. We had better choose independent characteristics to describe data in order to maximize information contained in the features. In this case I choose 2 features: the scale of bounding box (width / height) and the velocity of scale change.

1. $a = width / height$

2. $b = a - a'$

the scale describes the state of person , if this person is walking, the scale is small, if the person begin to fall or just lies down regularly, this number is going to increase constantly until become stable.

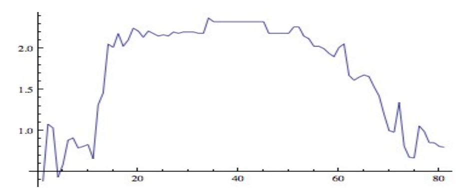


Figure 5. Scale changes

Figure 5 shows the scale changes from frames 220 to 300 in the 01 scene. We can see clearly that the scale grows sharply between frames 230 and frames 233 because of a fall. After that it reaches a platform at around 2.2 until frames 290 before this person stands up again.

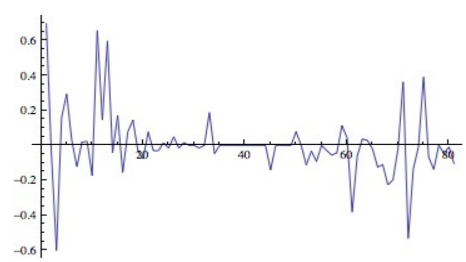


Figure 6. Velocity of scale changes

Graph above illustrates the velocity of scale change, the most dramatical changes happen in two section:

frames 220 to 240 and frames 280 to 300. There are both status changes in two section from walk to fall and from fall to stand respectively.

2.2 Parameters Estimation

Extracted features contain two classes in general : falls (lies down without move) or non falls, and the distribution of the data respect to two mixtured Gaussian, but we don't know labels, mean vectors and matrix covariances for each class. The parameters can be estimated by mean of EM algorithm with K-means initialization. And then we can get 4 parameters: $\mu 0$, $\mu 1$, Σ_0 ,Σ_1 The mixture Gaussian distribution as following picture shows in general.

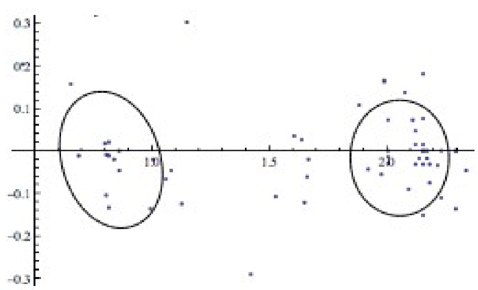


Figure 7. Two mixtured Gaussian

2.3 Bayesian Decision Theory

In this project, I use minimum-error-rate classification to decide possible falls or non falls. We define a decision function according to Bayesian theory :

$$p(w_i|x)=\frac{p(w_i)\times p(x|w_i)}{p(x)} \qquad \text{i= 0 or 1}$$

if $p(w_0|x)>p(w_1|x)$ x belongs to class 0 otherwise it is in class 1.

3. Experiments Results and Analysis

The whole system was built by C/C++ with OpenCV under Linux environment. First of all, I choose 4 cameras that installed on the sides of the room, then convert video into 320*240 image sequences. For training each classifier (4 classifiers in this case) , I use frames from 01 scenes to 10 scenes and test frames from 11 to 13 scenes.

(Frames)

	Training	Test
camera#1	842	253
camera#2	895	253
camera#3	816	253
camera#4	941	253

Table 1. Experiment data

I take vote method for final decision, for under some circumstance , single camera is not able to give right classification because of its angle or position however vote technique could reduce this kind of errors effectively. I test 4 frames that from different camera at each time (4 cameras working together) by using 4 classifiers individually, and then obtain the majority of votes for final decision. To evaluate our system, including sensibility, specificity, the definitions of these criteria as following :

$$sensibility=\frac{TP}{TP+FN}$$

$$specificity=\frac{TN}{TN+FP}$$

Experiment performance:

	Fall	Not Fall
Fall	TP: 49	FP: 29
Not Fall	FN: 9	TN: 166

Table 2 . Experiment results

My system gives a good result with sensibility 84.5% and specificity 85.1% , it can always get very good result if the scenes tested just contains fall, walk, sitting down the couch etc. but it is not easy to classifier sleeping or lying down from a real fall. Our system set a proper threshold after detecting a possible fall. The threshold is the velocity of scale change, if a fall happened the scale change must be small, even to zero. I test 116 frames where a person lying on the couch to have a rest, the 26 frames are misclassified as falls.

4. Conclusion and Discussion

In this work, I estimate mixtured Gaussian parameters by mean of EM algorithm, minimum-error-rate classification to classifer falls and vote method to get final decision. My detection system has proven its robustness and accuracy on the realistic image sequence of simulated falls and daily activities.

This system was built based on the hypothesis that there is no or little motion when a real fall happen, To realize this goal, I take three steps: the first is to classifier falls and not fall roughly, the second is to detect whether or not there has little or no motion after a possible fall, if answer is positive , the image is going to be recognized as fall. Last I put the results that come from different cameras together to make a final decision.

Such a system can be readily applied in homes to assist elderly, sick, needy or children. It can also be targeted to hospitals, welfare homes, etc, that take care of these groups of people who need help.

References

[1] Caroline Rougier, Jean Meunier, Alain St-Arnaud and Jacqueline Rousseau *Fall Detection from Human Shape and Motion History using Video Surveillance* . IEEE. 2007

[2] N. Noury , A. Fleury , P. Rumeau, A.K. Bourke. G. Ó Laighin , V. Rialle, J.E. Lundy . *Fall detection – Principles and Methods* . IEEE. 2007.

[3] Bin HUANG, Guohui TIAN and Hao WU . *A Method for Fast Fall Detection Based on Intelligen t -Space.* IEEE. 2008.