

Optical fall detection with Asynchronous Temporal-Contrast Vision Sensors for independently living older people

Ágoston Mihály Srp / Ferenc Vajda

Received 2013-02-21, revised 2013-07-19, accepted 2013-07-19

Abstract

Several studies have presented different issues of an ageing population including the need of enhancing care systems for older people using smart technologies. Falling accidents have a significant impact on healthy life expectancy and are a major problem among independently living older people. This paper presents a solution of the fall detection problem utilizing bio-inspired asynchronous temporal-contrast sensors and neural networks, realizing automated, robust, reliable and unobtrusive fall-detection. A noise reduction scheme suited to the unique nature of the sensor is presented, enabling their use in various applications in addition to fall detection. The process of transforming raw sensor output to a suitable neural network input is also described, along with the neural network creation process, including structure selection, training data assembly, and training algorithm selection for a truly large-scale network.

Keywords

Fall detection · asynchronous temporal contrast sensor · noise filtering · artificial neural network · older people · AAL · home-care

Acknowledgement

This work has been supported by the scientific program of the „Development of quality-oriented and harmonized R+D+I strategy and functional model at BME” (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002). The authors gratefully acknowledge the contributions of the Hungarian National Office for Research and Technology (NKTH) and the European Committee in the frames of the AAL Joint Programme (AAL-2008-1, Project “CARE - Smart Private Homes for Elderly Persons”).

Ágoston Mihály Srp

Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, Hungary
e-mail: srp.agoston@iit.bme.hu

Ferenc Vajda

Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, Magyar tudósok krt 2., H-1117, Hungary
e-mail: vajda@iit.bme.hu

1 Introduction

Human society in general and Europe in particular is experiencing major demographic changes since the turn of the 20th century. The unprecedented rise of the population’s average age will cause a massive increase in the population with age of 65 and over within our lifetime [22]. Our society will face two major challenges: elderly care services will require more investment due to the increasing and increasingly elderly population of seniors, while a simultaneous shortage of skilled care-givers will be caused by the decreasing population of working age. It has been shown that elderly people prefer living in their own home to living in nursing institutions, but they need some support to remain independent in their homes [23].

For the independently living elderly fall is one of the major health hazards [4]: approximately 30% of people of age 65 or above fall at least once each year. In principle only a fraction of these falls require immediate or any kind of medical attention. Even if no medical intervention would be required, help getting up again could be necessary. The cases where help (medical or otherwise) would be required are usually exacerbated by the time passing between the fall and its discovery: people who spent a short time “down” may not even require admittance to a hospital, while long down times may even result in death [5].

Section 2 gives a short description of the bio-inspired ATC vision sensor. Because of the nature of this type of sensor traditional image processing methods are not applicable, however this very nature makes these sensors well suited for the purpose of fall detection. A new algorithm to reduce sensor noise is introduced in Section 3; the performance and effectiveness of the algorithm are evaluated, a few “rules of thumb” for determining suitable parameters are listed and the algorithm is compared to other filters. Section 4 gives a short overview on the subsequent feature extraction. A form of machine learning has to be used in any fall detection system as we are currently not capable of sufficiently describing a fall algorithmically. As falls can be described as a spatial and temporal pattern using an Artificial Neural Network to recognize these patterns is a natural choice. Section 5 describes the design and training of the artificial neural network. Test results (including results of in-vivo

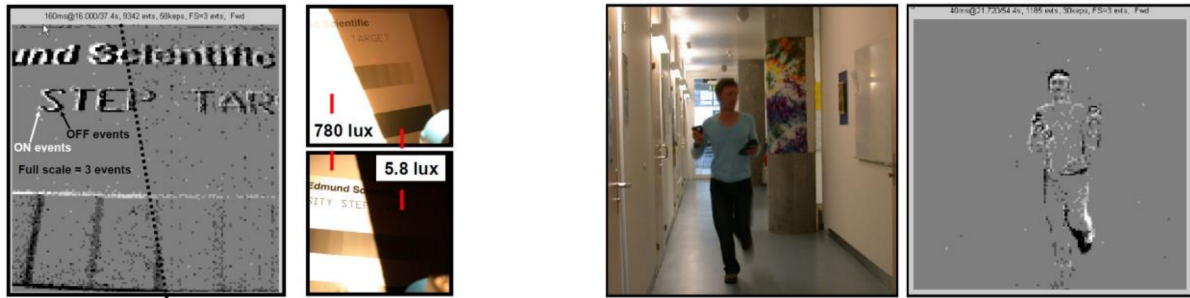


Fig. 1. Conventional image and ATC data

live testing) are shown in Section 6. Conclusions are presented in section 7.

2 Asynchronous Temporal-Contrast Vision Sensor

ATC sensors mimic the human retina; their working principles differ greatly from those of conventional cameras. As a result they are mostly independent of scene illumination, as they do not sense absolute illumination, but relative changes in intensity; ATC sensors directly encode object reflectance [10]. Another advantage of these sensors is their great dynamic range (Fig. 2), enabling them to reliably sense content in both over- and under-illuminated scenes. [9]

Each pixel quantizes local relative intensity changes, works independently and in continuous time and generates spike events. The output of the sensor is an asynchronous stream of time-stamped digital pixel addresses. These timed address-events (TAE) indicate scene reflectance [9]. The most important thing to note is that ATC sensors realize background segmentation by the very nature of their sensing. As only the dynamic content is sensed, the semi-stationary background is discarded. Thus the need for difficult and computationally expensive background segmentation is eliminated, making optical fall detection possible in a real-life environment.

The use of ATC sensors makes realizing an effective, reliable and robust automated monitoring system for fall detection possible. An additional advantage of the ATC sensor in the context of care of older people is the lack of picture. The pictures shown in this article can only be extracted under laboratory conditions and even then only a silhouette of the observed person is visible; during normal operation no picture is generated at any point of the system. This is a huge advantage and helps to alleviate privacy concerns – equally for patients, relatives, caregivers and regulatory institutions.

3 Noise reduction

One of the most important and overarching advantages of ATC vision sensors over traditional cameras is the lack of background. The elimination of the need for background segmentation saves significant computational capacity, enabling use of previously computationally too expensive algorithms.

The advantages of the ATC sensor also present a challenge: conventional image processing techniques are not applicable as

there are no images in the traditional sense. There are many methods for noise reduction in images; unfortunately, ATC sensors do not produce images. Traditional signal processing methods do not fare well either, as they do not take into account, that the signal represents an “image-like” scene. The real solution for dealing with data from ATC sensors is to have algorithms operate in event-space. Development of these however requires a completely different mindset.

Analysis of the sensor output showed it to be rather noisy (Fig. 2). In conventional applications using a frame-based approach this kind of salt&pepper noise is easily eliminated with e.g. morphological operations and/or median filtering or component labeling, however these approaches do not work in event-space, therefore we developed a 2D filtering algorithm to reduce the noise of the sensors. An additional issue is that while some of this noise is easily identified as such, e.g. the noise events outside and some distance from the silhouette; others are not so easily separated. It is not possible to decide if event within the silhouette are real events generated by small movements from e.g. shifting fabric, or facial expressions, or simply sensor noise. However these noise events, as they are within the silhouette, do not significantly affect subsequent processing and may be ignored.

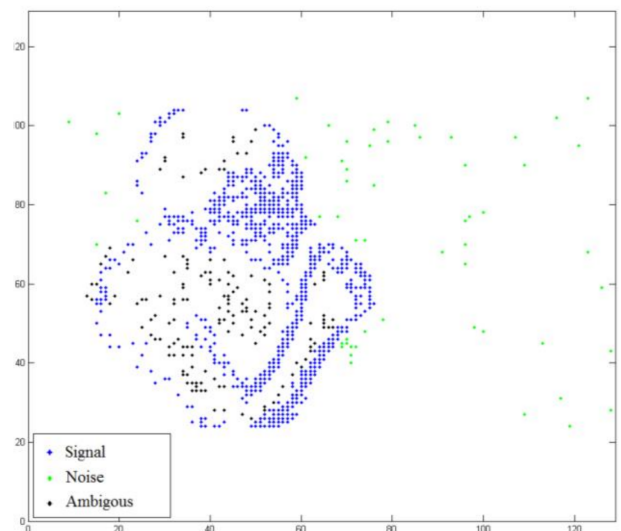


Fig. 2. Sensor noise

3.1 Filtering algorithm

The “event filter” algorithm consists of the following steps:

- 1 An observation window of length Δt is defined and only TAEs occurring within this observation window are considered. The appropriate value for Δt depends on the application and must be selected with care, as an inappropriate Δt may even render the filter useless. Too short and simultaneous events may be placed in different observation windows, leading to a worst case scenario where everything is disregarded as noise. Too long and unrelated events may be placed in the same observation window; if this happens to a sufficient number of spatially close, but temporally remote noise events the algorithm will be unable to remove them.
- 2 The sensing area of the sensor is divided into a number of equally sized buckets. Each TAE within the observation window is assigned to the appropriate bucket. Let us consider the sensing area as an event matrix \mathbf{S} of the size $h \times w$ (h rows and w columns), whose elements are lists containing the occurred events. Dividing the sensing area into $m \times n$ sized (m rows and n columns) buckets may be described mathematically as portioning the hypermatrix \mathbf{S} into sub-matrices $\mathbf{B}_{i,j}$ (1). (The sizes of the sensing area need to be divisible by the corresponding bucket sizes.)

$$\mathbf{S} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,w/n} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \cdots & \mathbf{B}_{2,w/n} \\ \vdots & \vdots & \mathbf{B}_{i,j} & \vdots \\ \mathbf{B}_{h/m,1} & \mathbf{B}_{h/m,2} & \cdots & \mathbf{B}_{h/m,w/n} \end{bmatrix} \quad (1)$$

The test results shown later in this paper were obtained with square buckets of size $b_{size} \times b_{size}$.

- 3 During the pre-filtering step the algorithm also takes the previous observation window into consideration. This ensures that even if cohesive events are separated by the observation window they are processed together. To differentiate the events of the separate observation windows the event matrix \mathbf{S} is amended with an index \mathbf{S}_k . For each bucket $\mathbf{B}_{i,j}$ of the current observation window (\mathbf{S}_k) the number of events in the bucket, its 8 neighbors and the corresponding 9 buckets in the previous observation window (\mathbf{S}_{k-1}) is determined. This is the event content ($\mathbf{C}(i, j)$) of bucket $\mathbf{B}_{i,j}$. If $\mathbf{C}(i, j)$ reaches a minimum threshold the bucket and its events are retained, otherwise they are discarded as noise. This filtering step eliminates a significant portion of the idle-noise of the sensor and thus reduces the number of calculations needed.
- 4 The actual filtering step is computationally more expensive: it checks all TAEs in a bucket against all TAEs in the bucket and its 8+9 neighbors. (If a bucket was previously discarded, then it is of course “empty”.) Outlying TAEs are thus removed and

only sufficiently large groups of TAEs are passed on. Outlying TAEs are events (\mathbf{e}_i) with less than $minpx$ neighbors (\mathbf{e}_j) in a set radius (\mathbf{maxD}) (2).

$$\sum_{i \neq j}^i num(Dist(\mathbf{e}_j, \mathbf{e}_i, \mathbf{maxD})) \geq minpx \quad (2)$$

The function “Dist” filters with a uniform kernel taking not only the spatial, but also the temporal distance of the examined events into consideration. Practical choices for the elements of \mathbf{maxD} are the corresponding bucket sizes and the length of the observation window, or a suitable fraction of these. Fig. 3 illustrates which TAEs are within the set distance from the currently examined TAE.

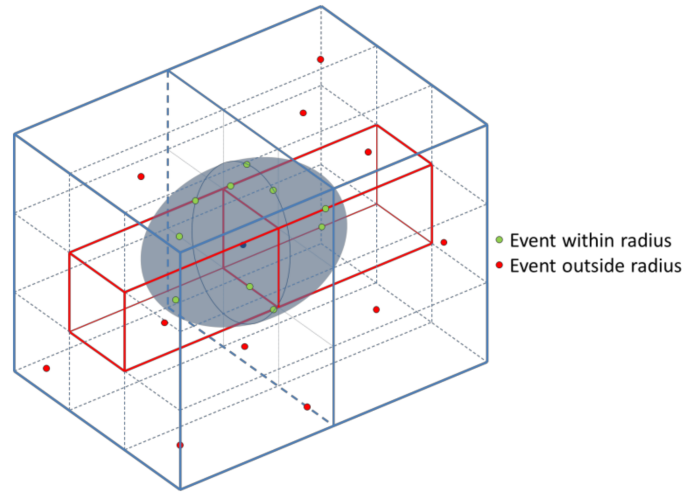


Fig. 3. Filtering of remaining TAEs

The event filter has some definite advantages. The first and most important is that the TAE stream does not have to be converted to an image, thus it is not necessary to reserve memory for a relatively sparse matrix. The algorithm places the incoming events into the appropriate bucket, stores them in lists and keeps track of the contents of the buckets. The 3 main parameters of the algorithm are: the length of the observation window (Δt), the size of the buckets (b_{size}), and the threshold for retaining the buckets ($minpx$). If suitable parameters are selected the pre-filtering removes a significant portion of the buckets, thereby reducing computational load. The main filtering step does not calculate the distance of all events from all other events, only those remaining after the pre-filtering are compared. These properties may make the filter an attractive choice for embedded applications.

3.2 Performance of the algorithm

For the investigation of the algorithm a test set containing 4 fall scenarios and 4 scenarios with activities of daily living (ADL) was assembled. A gold-standard was created for the test set. One option for a gold-standard is manual classification of each TAE, which is time consuming and prone to human error.

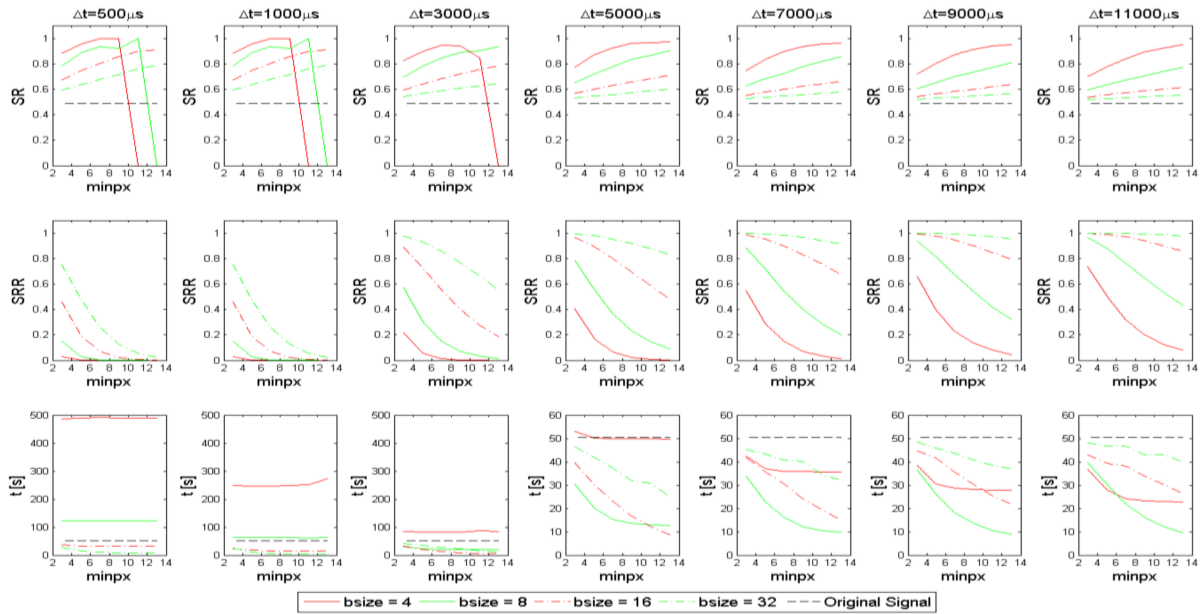


Fig. 4. Parameter effects on filter performance

To avoid these issues another algorithm was selected to serve as the gold-standard. The sensor developed by the Austrian Institute of Technology (AIT) utilizes two ATC sensors in a stereo arrangement to gain 3D data. The stereo algorithm removes all TAEs that it is unable to match to a pair on the other sensor. The classification of this algorithm was chosen to serve as the basis for comparison, as it is a proven solution which utilizes additional information in its decision. The event filter algorithm uses only 2D data to remain useable with a single ATC sensor.

We investigated the performance of our algorithm with various parameters. The Signal Ratio (SR) is the ratio of the number of signal TAEs versus all TAEs in the scene. The Signal Retention Ratio (SRR) is the ratio of the remaining signal TAEs after filtering versus the total number of signal TAEs. Fig. 4 shows SR, SRR, and runtime (t) of the algorithm as a function of these parameters. (The length of the test set is 50 seconds.)

Unsuitable parameters may render the filter unusable. Too short Δt coupled with an increasingly high $minpx$ causes a drastic decrease in SR and SRR as the pre-filtering steps removes more and more signal TAEs. A too short Δt has other consequences as well: the runtime may become exceedingly long and render on-the-fly filtering impossible. Well suitable parameters on the other hand enable a marked increase in SR while simultaneously maintaining a high SRR and enabling on-the-fly filtering.

The results prove that the algorithm is able to increase SR by removing noise. This increase may cause a decrease in SRR; the parameters have to be chosen depending on the requirements (noise reduction vs. signal retention) of the application. A hard to quantify advantage of the algorithm is that noise further from moving objects is removed with greater effectiveness; most of the remaining noise is very close to the moving silhouette and does not impact feature extraction significantly.

3.3 Pre-filter effectiveness

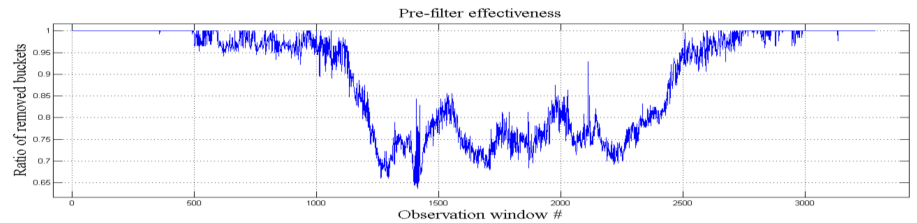
The effectiveness of the pre-filtering can indicate the suitability of the parameters. Given suitable parameters, (nearly) all buckets of an empty scene should be removed by the pre-filter; in a scene with dynamic content ideally none of the buckets containing TAEs related to scene changes should be removed.

During the previously described testing the pre-filter removed ~88% of the buckets. Without context this number has no purpose. The ideal ratio varies significantly with scene content: for an empty scene it may reach 100%, for “full” scenes a markedly lower ratio may be ideal. The ratios also vary with the chosen application. In fall detection the pre-filter has to remove remote sensor noise to reduce computational load. If it is used to filter the sensor output of a mobile robot the requirements are different. If the robot is standing still, the requirements may be identical. If the robot is moving, the pre-filtering may even be discarded, as the whole scene is “moving” in relation to the sensor and generating events.

Fig. 5 demonstrates the function of the pre-filter. For empty scenes (#0-500) (nearly) every bucket is removed, the computationally more expensive main filtering need not be applied. Even with significant scene content (~#1300, scene shown in Fig. 7) ~65% of the buckets are removed during pre-filtering, thus the main filtering only has to remove relatively close noise events. The 65% are acceptable for a scene with this content, and together with the performance for empty scenes suggest that the chosen parameters are well suited.

Fig. 6 shows the average ratio of removed buckets for the test set. With ill-suited parameters (e.g. too small $bsize$) the ratio approaches and may even reach 100%, which is wholly unacceptable for real recordings that do not solely consist of empty scenes. The effectiveness of the pre-filter decreases when the bucket size is too big, the threshold too high, or the observation

Fig. 5. Influence of scene content on pre-filter effectiveness



window too long. In these cases enough noise may gather in a single bucket for it to be retained.

Testing proved that given suitable parameters the pre-filter was able to remove remote solitary noise events, thus reducing the computational requirements of the algorithm.

3.4 Determining suitable parameters

Based upon our examination results and our own experiences we have determined a few “rules of thumb” for determining the optimal parameters for the algorithm:

- A convex SRR curve usually indicates too small bucket size (*bsize*) and/or too short observation window (Δt). One or both should be increased.
- A disproportionately long runtime of the algorithm also indicates too small *bsize* and/or too short Δt .
- Suitable parameters usually result in a concave SRR curve; the better the parameters, the higher the SRR, and the lower its gradient.
- An unreasonably high ratio of removed buckets during pre-filtering of a scene with rich dynamic content may indicate unsuitable parameters; if *bsize* and Δt are otherwise appropriate the threshold (*minpix*) may be too high. If *bsize* and Δt are suitable (see previous rules) *minpix* should be decreased.
- An unreasonably low ratio of removed buckets during pre-filtering of a scene with sparse dynamic content may also indicate unsuitable parameters; if *bsize* and Δt are otherwise appropriate *minpix* may be too low. If *bsize* and Δt are suitable (see previous rules) *minpix* should be increased.
- Given suitable parameters the pre-filtering step should remove (nearly) 100% of the buckets in empty scenes while retaining (nearly) all buckets containing dynamic scene content in non-empty scenes.

3.5 Comparison to other filters

The event filter algorithm was compared to several popular filters which are used to remove salt&pepper noise. The images were generated by collecting all events within an observation window and plotting them in a single figure according to their x and y coordinates. The conventional filters were applied to this image. The images showing the result of the “event filter” and the “PCL ...” filters were attained by applying this technique to the TAEs remaining after filtering.

Of the traditional filtering algorithms only the performance of the 3-quantile filter is acceptable. This filtering inserts nonexistent events, which is not a huge problem, as it could be easily modified not to do so; however it also requires the conversion of the TAE stream into an image, which poses several issues, which are further elaborated on in [20]. The last two filters are part of the Point Cloud Library (PCL) [24]. The PCL algorithms [17], which were published after our first paper [20] describing the event filter algorithm, is very similar to the event filter. The statistical filter iterates twice through all points, first calculating the average distance to the nearest *k* points for each point; then calculating a threshold from their average and deviation. On the second pass all points are removed that have a higher average distance to their nearest *k* neighbors than the threshold. The performance of the PCL filter is nearly identical to our event filter, but it requires more computational resources. The other PCL filter checks the number of neighbors within a set distance from the examined point. The point is retained if the number reaches a set threshold. This filter also performs quite well, suppressing most noise and retaining an adequate number of signal points. The downside is, that it may retain groups of remote noise points, which may affect subsequent processing negatively, e.g. when a bounding box or cylinder are calculated.

The algorithm performed to our expectations, as it removed the majority of the idle-noise from the sensor, and our secondary goal was also met, as it only introduced a minimal latency to the processing stream. (The algorithm has to be applied on-the-fly to the received TAE stream before feature-extraction.) The performance of event filter is comparable to the performance of the other filters, or superior to those. Unlike traditional filters it does not require the conversion of the TAE stream to an image and has lower computational requirements than the algorithms of the Point Cloud Library. We found the algorithm to be suitable for our purposes, although care must be taken to maintain a balance between signal retention and noise suppression. We are confident that the algorithm can (with appropriate parameters) be suitable for other applications as well.

4 Feature Extraction

The input of the fall detection is a cloud of “simultaneously” occurred Timed 3D Events (T3DE). The T3DEs are generated by the AIT sensor system [6] from the filtered TAEs of two ATC sensors via stereo matching. AIT chose to employ their own algorithm to filter the two TAE streams in the sensor system. T3DEs are accepted as “simultaneous” if they fall within the

Fig. 6. Parameter influence on pre-filter effectiveness

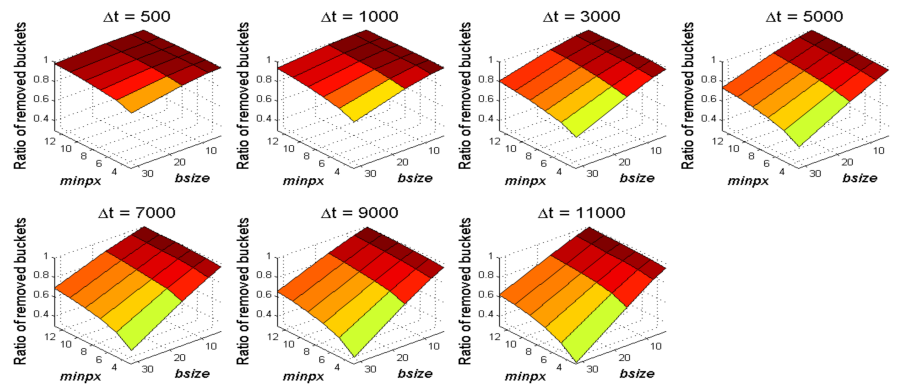
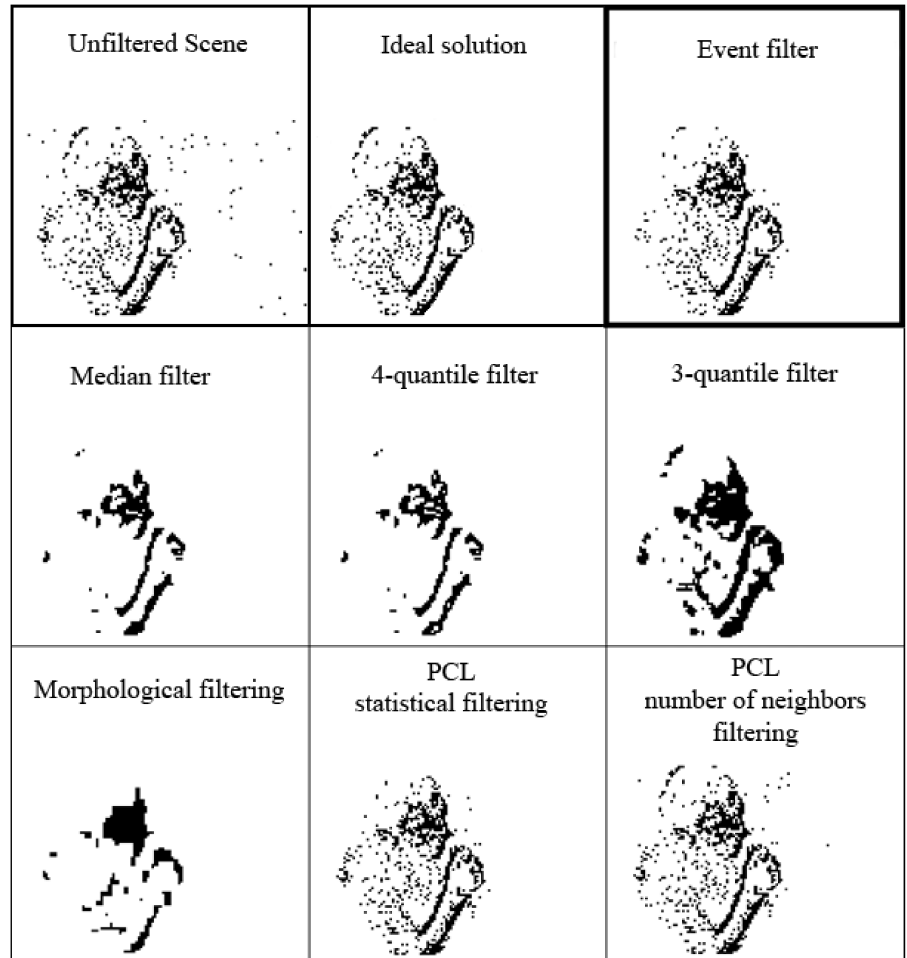


Fig. 7. Filter comparison



same observation window.

After analyzing the possibilities and considering their advantages and the problems associated with them the position, speed and acceleration of the CoG of the body and of the highest point, the Bounding Cylinder of the silhouette, its height to diameter ratio and the total number of simultaneously occurred Events were used as features in our fall detection system (Fig. 8). Cooperatively employed these features enable realizing a reliable and robust automated fall detection system.

The size of the feature vector and the required observation length for effective fall detection resulted in an input vector for the neural network that was too large to use effectively. After

some optimization and introduction of a sampling scheme suited to fall detection we were able to reduce it to a manageable size. The features were selected based mainly on the works of Anderson et al. [2], Juang et al. [7], Li et al. [8], Nait-Charif et al. [11], Noury et al. [12], Nyan et al. [13], Planine et al. [14], and Rougier et al. [15, 16]. The selection process for the features and the optimization and sampling scheme are explained in detail in [6, 20, 21].

5 Machine Learning

In the final analysis falls are certain spatial and temporal patterns; hence it is a logical step to employ a learning system to distinguish falls and non-falls. There are a number of possible

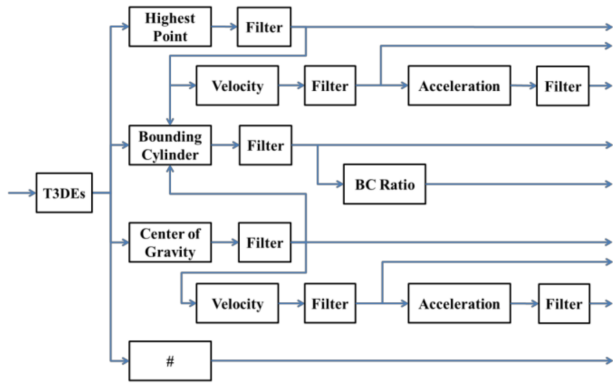


Fig. 8. Features used for fall detection

approaches to machine learning that may be used; after careful consideration we chose to use an artificial neural network (ANN) as the learning system in our fall detection scheme.

ANNs are universal approximators, allowing us to relatively quickly and easily model phenomena which are only vaguely understood or cannot be adequately explained. These qualities make ANNs a very attractive choice for fall detection. We chose to employ a fully connected multilayer feedforward network as they can easily be adapted to recognize patterns of temporal nature by using a tapped delay line.

5.1 Training and testing data

We have analyzed the most common types of falls and have created a list of 47 fall scenarios and a separate list of activities of daily living [21]. These lists were used as the basis for recording the training and test dataset. The recordings were done by stuntmen with the aid of nursing personnel so that the recorded falls match the way older people fall as closely as possible. The recorded dataset contains 467 fall scenarios and 50 non-fall scenarios. Recordings of the same scenario were done from different positions with different lighting conditions. The scenarios (where applicable) begin with the subject entering a scene and doing some normal activity (e.g. walking around) and end with the end of the fall.

5.2 Training methods

A compilation of falls from standing were used as training data in this preliminary investigation; a separate compilation served as test data. This investigation was limited to this single category to simplify and shorten training, as the purpose of this initial phase was to obtain a proof of concept and determine suitable training algorithms and ANN structures.

According to Seiffert [19] large-scale datasets with a large number of training samples often lead to very large networks. Past a certain threshold conventional training algorithms perform very poorly and training time and memory limitations increasingly move into focus. We took this into account and began testing with various training methods to assess their usability. Based on the above mentioned paper and our own test results a

form of Conjugate Gradient Method was chosen for ANN training.

Various methods belonging to this group were tested, including the Fletcher-Reeves Update, the Polak-Ribière Update and the Scaled Conjugate Gradient. Neither the Fletcher-Reeves Update, nor the Polak-Ribière Update did scale up well; during training with the full dataset these “faster” methods failed to converge. The Scaled Conjugate Gradient method retained a reasonable performance despite the increased amount of training data.

5.3 Artificial Neural Network Structure

Defining a neural networks structure is an essential part of the design process. Although there are some formalized methods for defining the structure this step remains a mixture of experience, gut-instinct and luck.

Several possible structures for the neural network were investigated in the initial phase mentioned in the previous subsection. Based on the complexity of the task we expected to be able to solve the problem with a 3-layer network, and initially limited our investigations to these. The layer activation function arrangement $\text{tansig} \text{ tansig} \text{ logsig}$ proved clearly superior with all structures.

The next phase consisted of initial tests with a compilation of falls not limited to a single category and two output neurons corresponding to fall and not-fall probability. This allows a measure of confidence to be calculated from the two outputs. These tests showed a marked drop in performance, even after considerable training. Careful and detailed examination of the network weights showed the weights of the final layer to be saturated. This led to the addition of a 4th layer with a linear activation function (purelin) to avoid saturation. A modified mean square error was used to measure the performance of the different structures (3). (The operator “.” denotes element wise squaring; the length of the samples is h .)

$$\begin{aligned}
 \mathbf{se}_f &= (\mathbf{output}_f - \mathbf{target}_f)^2 \\
 \mathbf{se}_{nf} &= (\mathbf{output}_{nf} - \mathbf{target}_{nf})^2 \\
 \mathbf{se}_{conf} &= (\mathbf{output}_{nf} + \mathbf{output}_f - 1)^2 \\
 \mathbf{error} &= \frac{\sum_{i=1}^h [(\mathbf{se}_f(i) + \mathbf{se}_{nf}(i) + \mathbf{se}_{conf}(i)) \cdot (1 + 99 \cdot \mathbf{target}_f(i))]}{h + 99 \cdot \sum_{i=1}^h \mathbf{target}_f(i)}
 \end{aligned} \tag{3}$$

This measure basically calculates the mean error of the fall (\mathbf{se}_f) and not-fall outputs (\mathbf{se}_{nf}), then calculates the square error of their sum (\mathbf{se}_{conf}). Ideally the sum of the outputs should be 1, so the square error of their sum serves as a confidence measure. The final performance is gained by computing the weighted average of the sum of these components. In fall detection false negatives, i.e. missed falls are not acceptable; false positives on the other hand are acceptable, if there are not too many of them. Accordingly weights were chosen to be 99 for fall examples and

1 for non-fall examples to severely penalize missed falls. The method itself is not especially sensitive to the weights as long as they remain in a reasonable range. Both outputs were subjected to cropping at 0 and 1 prior to calculation of the performance measure.

The training data set was divided into 9 sets, each containing an example from all fall scenarios. (The 10th set was reserved for independent testing.) To reduce the time required (as due to the size of the networks and the training data set training times were significant) the size of the 3rd layer was fixed to 2 during the first phase of testing; cross evaluation combined with a 15x15 grid search with a step size of 20 was used to determine the optimal number of neurons in the first two layers.

After careful evaluation of the performance and the actual network outputs, a few structures with the “best” performance were selected for further study. The next phase of testing used these candidate networks, changing the number of neurons in the 3rd layer. Layer sizes of 2, 4, 8, 16, 32, and 64 were tested; sizes 2 and 4 proved clearly superior. After consideration of the performances of these candidates and of the available computational capacity 5 structures with the “best” performances were selected to investigate possible decision strategies.

5.4 Decision strategies

Five possible decision strategies (Table 1) were investigated with the 5 ANN candidates.

Tab. 1. Decision strategies

#	Decision strategy
1	$out_f > thresh$
2	$1 - out_{nf} > thresh$
3	$(out_f > thresh) \& (1 - out_{nf} > thresh)$
4	$out \cdot (1 - kim_{nf}) > thresh$
5	$out_f \cdot (1 - out_{nf}) \cdot (1 - out_f + out_{nf} - 1) > thresh$

At this point a slightly different protocol was used to determine performance. Successful detection of a fall in a fall scenario was counted as a true positive. A fall signal in a non-fall scenario or outside the fall section of a fall scenario was counted as a false positive. Receiver Operating Characteristics (ROC) was used to compare the decision strategies.

The ROC curves of Fig. 9 were gained by averaging the classification results of the five different neural networks. While the difference between the decision strategies is not major, strategy 4 had the best performance. Having determined the best decision strategy we compared the performance of the 5 chosen ANNs while using the best strategy (Fig. 10). ANN 3 had the best fall detection performance, but also the highest false positive rate; ANNs 2, 4 and 5 had the lowest false positive rate, but their true positive rate was also the lowest. ANN 1 proved to be the best compromise between sensitivity and specificity.

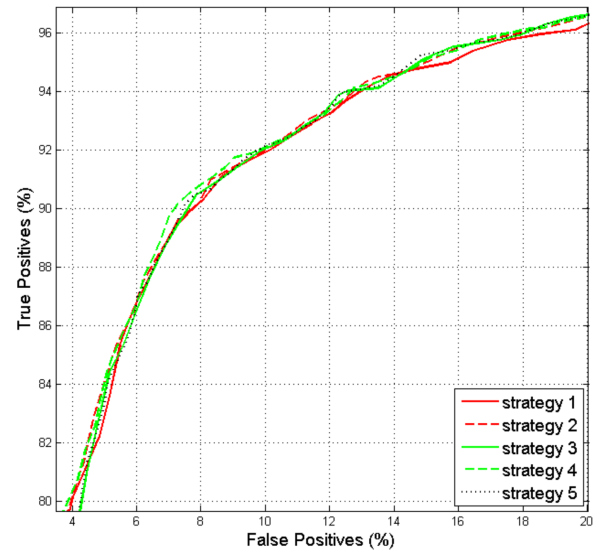


Fig. 9. ROC of classification with different decision strategies

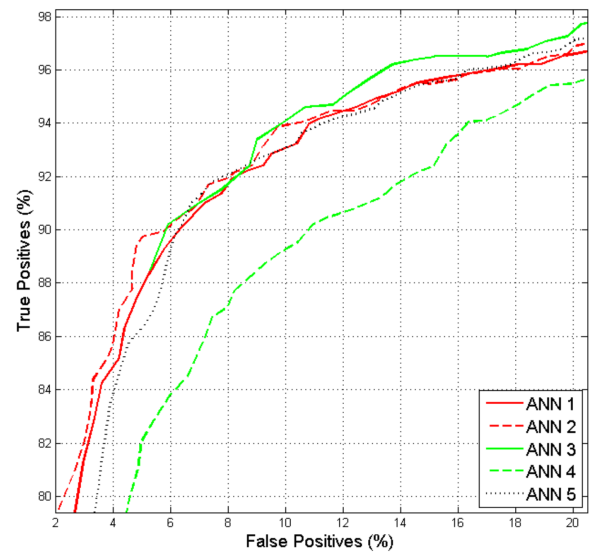


Fig. 10. ROC of classification with different neural networks

6 Test results

Actual testing was done with the dedicated testing subset of the data consisting of 46 fall scenarios and 5 non-fall scenarios. Results for all five candidate structures were promising. All networks achieved good generalization and exceeded expectations by a wide margin. Fall detection rate (falls detected versus falls observed) of ANN 1 with decision strategy 4 for the whole dataset was on the order of 97-98% with an alarm threshold of 0.8 and over 99% with an alarm threshold of 0.7. False alarm rate can be defined multiple ways. One possibility is number of false alarms versus number of observed non-fall scenarios. This disregards the length of the scenarios. Nursing institutions prefer another method of calculating false-alarm rate: number of false alarms during a defined time period. False alarm rate for the first case is ~8.64/day. While this may seem a little high at first glance, it must be considered that all non-fall scenarios were selected for their similarity with falls. Another mitigating factor is that a significant portion of false alarms coincide with

very low numbers of TAEs and are therefore easily handled in post-processing.

Li et al. [8] and Anderson et al. [2] did not include data on performance in their papers. The solution proposed by Nyan et al. [13] detected all falls without false alarms, however their system covers a very limited area with 6 cameras and is only useable with worn optical markers. The system proposed by Juang et al. [7] classified 97.8% of 400 postures correctly, with 100% sensitivity for “lying down” samples. This result is promising regarding fall detection. The paper’s focus was posture classification not fall detection, so no such data was published making a direct comparison impossible. Nait-Charif et al. [11] analyzed activity and detected falls by the lack of activity. The system detected all 9 test falls, but no information about the nature of those falls is revealed. Rougier et al. [15, 16] proposed two different solutions. The first [16] tracks the 3D trajectory of the head to detect falls. This approach was tested with 10 ADL and 9 fall scenarios. The system recognized 6 out of 9 falls (66.6% detection rate) and one of the ADL scenarios was misclassified as a fall (10% false alarm rate). The other system [15] utilizes human shape and motion history. It was tested with 17 fall and 24 ADL scenarios. Detection rate was ~88% with 15 out of 17 falls recognized while false positive rate was ~12.5% with 3 false alarms out of 24 non-falls. The detector developed by Alwan et al. [1] was tested with 70 dummy falls and 35 object drops and had “100% true positives and 0% false alarms” in controlled experiments. While the results are promising, no description of fall scenarios was given, and there were no ADL scenarios used, only object drops, limiting the significance of the obtained test results. The ultrasound detector proposed by Dobashi et al. [3] for fall detection in the bathroom also shows an impressive 100% detection rate with no false alarms. The test set consisted scenarios of 10s duration; one empty bathroom, and 3 each of standing, tumbling to left, right and forward scenarios were incorporated. The proposed system elegantly solves fall detection in the bathroom; it would be interesting to evaluate its performance in a more general environment with a more diverse (regarding both falls and ADL) set of scenarios.

After incorporating modifications prompted by laboratory testing and gaining the necessary approvals and permissions live in-vivo testing was conducted with the system described in this paper from 1st of June to 31st of August, 2012 in 3 apartments in one of the care institutions of Senioren Wohnpark Weser GmbH. Given suitable installation, the system produced 2.1 false alarms per week per room on average, which is acceptable according to resident caregivers. Fortunately for the inhabitants there were no falls during the test period in the monitored rooms; unfortunately this prevented validation of the detection rate. However AIT did several live demonstrations of the system at various Ambient Assisted Living workshops and conferences, where it was set up and interested parties could test the fall detection capability; the system recognized all falls during these demonstrations and reception was very positive. Additionally the data used

in the independent test set was recorded with the aid of stuntmen and experienced caregivers to ensure the obtained data are as realistic as possible. The proposed fall detector recognized 98% of the fall scenarios, including the very difficult scenarios of syncope (vertical slipping against a wall ending in sitting position) and rolling out of bed, while producing few false alarms even in difficult to distinguish situations like e.g. exercising, crouching, sitting down, lying down, etc.

7 Conclusion

In this paper we presented an optical fall-detection scheme involving the new bio-inspired ATC vision sensors and artificial neural networks. We discussed the difficulties stemming from using this new technology and how to solve them. We introduced an algorithm for filtering the rather noisy output of the ATC sensor, analyzed the performance of the algorithm and the effectiveness of the pre-filter. We described a few “rules of thumb” as a guideline for determining suitable parameters for the algorithm and compared it to other filters. We presented the features we found most suitable for fall detection with ATC sensors. We discussed the suitability of the different training algorithms for truly large-scale networks and large amount of training data. We discussed the importance of selecting an appropriate structure for the neural network and described the process for selecting our candidate networks for the fall-detection scheme.

Unlike video monitoring the proposed system does not compromise privacy as no image is generated at any point of the system; it also does not require human resources for monitoring. It does not depend on the capability and willingness of patients’ to raise an alarm, unlike “panic button” solutions, which are rendered useless by e.g. falls resulting in unconsciousness. Installation of the proposed system does not require extensive remodeling of the living space (like e.g. pressure sensitive floors do). A direct comparison of the proposed system and the solutions published by Alwan [1] and Dobashi [3] by e.g. setting up all systems in a number of apartments for an extended period of live in-vivo testing would be interesting and would enable objective analysis and comparison of all systems’ capabilities in real-life environment.

References

- 1 **Alwan M, et al.**, *A Smart and Passive Floor-Vibration Based Fall Detector for Elderly*, Proceedings of Information and Communication Technologies, 2, (2006), 1003–1007, DOI 10.1109/ICTTA.2006.1684511.
- 2 **Anderson D, Keller J, Skubic M, et al.**, *Recognizing Falls from Silhouettes*, 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, (2006), 6388–6391, DOI 10.1109/IEMBS.2006.259594.
- 3 **Dobashi H, et al.**, *Fall Detection System for Bather Using Ultrasound Sensors*, Proceedings of the 9th Asia Pacific Industrial Engineering & Management Systems Conference, (2008), 1860–1865.
- 4 **Duthie E**, *Falls*, Medical Clinics of North America, 73(6), (1989), 131–1336.
- 5 **Gurley RJ, Lum N, Sande M, Lo B, Katz MH**, *Persons found in their*

- homes helpless or dead*, The New England Journal of Medicine, **334**, (1996), 1710–1716.
- 24 <http://pointclouds.org>.
- 6 **Humenberger M, Schraml A, Sulzbachner C, Belbachir A, Srp Á, Vajda F**, *Embedded Fall Detection with a Neural Network and Bio-Inspired Stereo Vision*, 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2012), 60–67, DOI 10.1109/CVPRW.2012.6238896.
 - 7 **Juang C, Chang C**, *Human Body Posture Classification by a Neural Fuzzy Network and Home Care System Application*, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, **37**(6), (2007), 984 – 994, DOI 10.1109/TSMCA.2007.897609.
 - 8 **Li Q, Zhou G, Stankovic J**, *Accurate, Fast Fall Detection Using Posture and Context Information*, Proceedings of the 6th ACM conference on Embedded network sensor systems. SenSys '08, (2008), 443 – 444, DOI 10.1145/1460412.1460494.
 - 9 **Lichtsteiner P, Posch C, Delbruck T**, *A 128x128 120 dB 15 μ s Latency Asynchronous Temporal Contrast Vision Sensor*, IEEE Journal of Solid-State Circuits, **43**(2), (2008), 566 – 576, DOI 10.1109/JSSC.2007.914337.
 - 10 **Lichtsteiner P, Posch C, Delbruck T**, *A 128x128 120 dB 30 mW Asynchronous Vision Sensor that Responds to relative Intensity Change*, 2006 IEEE International Solid-State Circuits Conference, (2006). Session 27, Image Sensors, 27.9.
 - 11 **Nait-Charif H, McKenna S**, *Activity Summarisation and Fall Detection in a Supportive Home Environment*, 17th International Conference on Pattern Recognition, **4**, (2004), 386 – 401, DOI 10.1109/ICPR.2004.127.
 - 12 **Noury N, Fleury A, Rumeau P, et al.**, *Fall detection – Principles and Methods.*, Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, (2007), 1663 – 1666, DOI 10.1109/IEMBS.2007.4352627.
 - 13 **Nyan M, Tay F, Mah M**, *Application of motion analysis system in pre-impact fall detection*, Journal of Biomechanics, **41**(10), (2008), 2297 - 2304, DOI 10.1016/j.jbiomech.2008.03.042.
 - 14 **Planine R, Kampel M**, *Emergency System for Elderly – A Computer Vision Based Approach*, 2011, pp. 79-83, DOI 10.1007/978 3 642-21303-8.
 - 15 **Rougier C, Meunier J, St-Arnaud A, et al.**, *Fall Detection from Human Shape and Motion History using Video Surveillance*, 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), **2**, (2007), 875 – 880.
 - 16 **Rougier C, Meunier J, St-Arnaud A, et al.**, *Monocular 3d head tracking to detect falls of elderly people*, Proceedings of the 28th IEEE EMBS Annual International Conference, (2006), 6384–6387.
 - 17 **Rusu R, Cousins S**, *3D is here: Point Cloud Library (PCL)*, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), (2011), 1 - 4, DOI 10.1109/ICRA.2011.5980567.
 - 18 **Scales L**, *Introduction to Non-Linear Optimization*, Springer-Verlag; New York, 1985.
 - 19 **Seiffert U**, *Training of Large-Scale Feed-Forward Neural Networks*, International Joint Conference on Neural Networks, IJCNN'06, (2006), 5324 - 5329, DOI 10.1109/IJCNN.2006.247289.
 - 20 **Srp Á, Vajda F**, *Possible techniques and issues in fall detection using asynchronous temporal-contrast sensors*, e& i Elektrotechnik und Informationstechnik, **127**(7-8), (2010), 223 – 229, DOI 10.1007/s00502-010-0751-0.
 - 21 **Srp Á, Vajda F**, *Fall Detection for Independently Living Older People Utilizing Machine Learning*, 8th IFAC Symposium on Biological and Medical Systems, (2012). paper A-0011.
 - 22 **Steg H, et. al.**, *Europe Is Facing a Demographic Challenge Ambient Assisted Living Offers Solutions*, European Overview Report, (March 2006).
 - 23 **Sun H, Florio V, Gui N, Blondia C**, *Promises and Challenges of Ambient Assisted Living Systems*, Sixth International Conference on Information Technology: New Generations, (2009), 1201 - 1207, DOI 10.1109/ITNG.2009.169.