Fusion of Color and Depth Camera Data for Robust Fall Detection

Wouter Josemans[1], Gwenn Englebienne[1] and Ben Kröse[1,2]
[1]*Informatics Institute, University of Amsterdam, P.O. Box 94323, 1090 GH Amsterdam, The Netherlands*
[2]*Digital Life Centre, Amsterdam University of Applied Science, P.O. Box 1025, 1000 BA Amsterdam, The Netherlands*

Keywords: RGB-D Camera, Data Fusion, Fall Detection.

Abstract: The availability of cheap imaging sensors makes it possible to increase the robustness of vision-based alarm systems. This paper explores the benefit of data fusion in the application of fall detection. Falls are a common source of injury for elderly people and automatic fall detection is, therefore, an important development in automated home care. We first evaluate a skeleton-based classification method that uses the Microsoft Kinect as a sensor. Next, we evaluate an overhead camera-based method that looks at bounding ellipse features. Then, we fuse the data from these two methods by validating the skeleton tracked by the Kinect. Data fusion proves beneficial, since the data fusion approach outperforms the other methods.

1 INTRODUCTION

The emergence of many new types of sensors on the consumer market gives rise to many different applications. Since sensors often provide incomplete or incorrect data, data fusion methods are being used to combine data from different sensors to increase robustness. One application where robustness is a prerequisite is the automatic detection of falls of elderly people. Falls are a major source of injury for elderly people (Gallagher et al., 2001). In fact, about 30% of people aged 65 or above fall every year (Gillespie, 2004), and for people above 65 years of age, 80% of all injuries can be attributed to falls (Kannus et al., 1999). Therefore, a lot of research has gone into automated solutions; i.e. fall detection through cameras, microphones, accelerometers, pressure sensors or motion sensors. These types of sensors do not require the fallen person to explicitly call for help, but can automatically detect catastrophic events and sound an alarm. Currently, research has been shifting in the direction of ambient methods, especially camera-based methods. In this research, we focus on camera-based methods for fall detection.

1.1 Problem Statement

In real applications, the use of computer vision systems still has its limitations, for example because of the limited field of view, or errors cause by lighting changes. In our research we focus on data fusion and we find a way to combine data from the Kinect and an RGB overhead camera that performs better than either method separately. In order for a fall detection system to be useful in practice, several issues must be addressed. First of all, the accuracy of a system must be high. Falls do not occur often, but when they occur, they must not be missed. The system should also avoid having too many false alarms, though some false alarms may be acceptable depending on whether or not it is possible to remotely verify if a fall took place. In this research, we focus on improving fall detection accuracy.

1.2 Related Work

Camera-based fall detection methods have been presented earlier. (Foroughi et al., 2009) uses a Support Vector Machine classifier for detecting falls. From the camera image, several features are extracted: approximated ellipses of the body shape, projection histograms representing the vertical and horizontal distribution of the foreground pixels, and temporal changes of the head's position. (Yu et al., 2009) uses a single RGB camera. First the ellipse that best matches the body of the subject is found, and the parameters of this ellipse are used to determine which of the three states the body is in: standing, bending or lying. A fall is detected if the subject transitions from standing to bending to falling in a short period of time. A similar approach is described in (Liu et al., 2010), but instead of looking at an ellipse, the silhouette of the

Josemans W., Englebienne G. and Kröse B..
Fusion of Color and Depth Camera Data for Robust Fall Detection.
DOI: 10.5220/0004213406080613
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 608-613
ISBN: 978-989-8565-47-1

target is classified into a pose. Again, quick transitions between certain poses indicate a fall. In (Tao et al., 2005), a similar method is presented, but this one looks at the change of a bounding volume over time. Falls are then detected as abrupt changes in the feature space. A volumetric approach is also used by (Anderson et al., 2009), in combination with fuzzy logic rules to determine if a fall took place.

An issue with the approaches described above is that the amount in which the body shape changes depends on the direction of the fall. This is not an issue in depth cameras, since they measure the absolute size of the body shape, and not the size of the projection of the body onto a 2D plane. Several methods have investigated the use of depth cameras, the Kinect in particular. For instance, (Rougier et al., 2011) uses the Kinect depth camera for fall detection. After foreground segmentation, the target's 3D centroid is found. The position and velocity of this centroid are monitored, once they reach certain threshold values they will trigger an alarm. A similar method that also uses the Kinect is presented in (Mastorakis and Makris, 2012), in which the change in the bounding box shape is examined, and the target is monitored for inactivity after a fall. Some successful attempts were made in combining data from different sensor types, such as (Töreyin et al., 2005), which combines video and audio data. Additionally, in (Alemdar et al., 2010) video data is combined with accelerometer data to do classification. However, combining the data from depth cameras and normal cameras is unexplored in the application of fall detection. Therefore, we want to focus on investigating data fusion between RGB and depth cameras in this research.

1.3 Research Question

The question we answer in this research is the following: Can we improve classification accuracy by fusing data from the Kinect and an overhead camera? An overhead camera is used because it gives a different angle of the room, which can help when the tracking target is occluded from certain angles. In our study we implemente a classification method that uses the Kinect, and a different classification method that uses the overhead camera. We then evaluate the performance of these methods. Next, we combine the data from these two methods, and compare the performance of this data fusion method to the performance of the separate methods.

1.4 Overview of Methods

We have designed a skeleton-based classification method that classifies a sequence of joint positions into a fall or a non-fall. We do this by compressing a high-dimensional sequence of joint positions using PCA, and then classifying on this low-dimensional data using an SVM classifier. The second method we implemented was the bounding ellipse method. For this method we look at the shape of the foreground of the overhead camera image. The parameters of this shape are then used in SVM classification. This is a method much like those described in (Tao et al., 2005), (Yu et al., 2009) and (Liu et al., 2010), though these methods use a heuristic rule to determine if a fall took place, whereas we use an SVM classifier. We chose to use an SVM classifier in order to compare this method fairly with our own methods, which also use SVM classifiers. Additionally, the SVM does not require adapting the heuristic classification rules to our particular dataset, thus avoiding biasing the results in favor of our method. The third and fourth approaches use data fusion to improve classification. Our third method takes the features from both classifiers and does classification on this larger feature vector. In our fourth method, we estimate the reliability of the skeleton tracked by the Kinect, and use this as extra features for our classifier. A more elaborate description of these methods is given in the next section. We evaluate these methods by classifying a dataset of 40 fall samples and 40 non-fall samples, as described in 3. These experiments are described in Section 4.

2 METHODS

2.1 Skeleton-based Classification Method

As seen in Figure 1, we take the skeleton joint data from the Kinect. An example of the skeleton tracked is shown in Figure 2. For each period of one second in the skeleton joint data, k skeleton tracking results are sampled and concatenated as one observation in our training data. For each skeleton pose, each of the 20 joints is described by X,Y and Z coordinates, which means we have $k \times 60$ dimensions per observation. The joints are represented in skeleton space coordinates, which are measured in meters with respect to the Kinect's position. Considering the high dimensionality of this vector, we would need a lot of training data to train our classifier. We can avoid this by using Principal Component Analysis. We chose to use 10 principal components in our feature vectors, because after experimentation we found that the first 10 principal components explain 96% of the variance in the

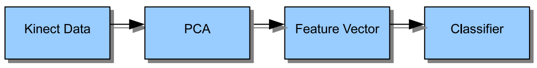

Figure 1: Overview of skeleton-based classification method.

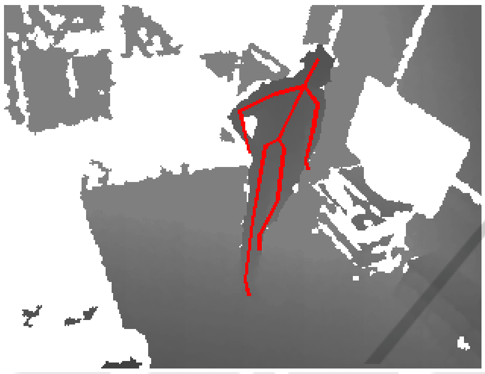

Figure 2: The tracked skeleton in the depth image.

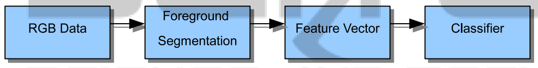

Figure 3: Overview of bounding ellipse classification method.

data. For classification, we use an Support Vector Machine (SVM) classifier with a Radial Basis Function (RBF) kernel. We chose this kernel because it allows non-linearities in our data.

2.2 Bounding Ellipse Classification Method

For our second method (Figure 3) we use an overhead RGB camera. The first step in this method is foreground segmentation. Our approach models the background by constructing a set of eigenbackgrounds(Oliver et al., 2000) using Principal Component Analysis (PCA). Applying PCA to a diverse set of background images results in the mean background image μ and a set E of k eigenbackgrounds. The background of a new sample $\mathbf{x}$ is reconstructed by projecting the sample onto the k eigenbackground and projecting back into image space. Then, we create a probabilistic model of the background. Given an image $\mathbf{x}$, this model will give us the probability for each pixel that it belongs to the background. We model each HSV component of each pixel i with a Gaussian distribution, taking the reconstructed background $\mathbf{b}$ as a mean:

$$p_{b,i}(\mathbf{x}) = \mathcal{N}(\mathbf{x_i}; \mathbf{b_i}, \sigma^2) \quad (1)$$

The variance σ was calculated from the training data. For the H and S components, we found a variance of 0.08 and for V a variance of 0.05. We decide a pixel

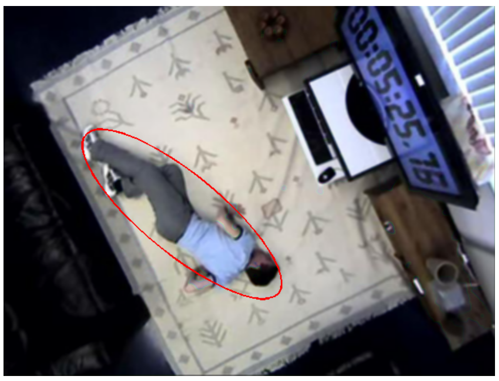

Figure 4: Example of an ellipse fit to the target.

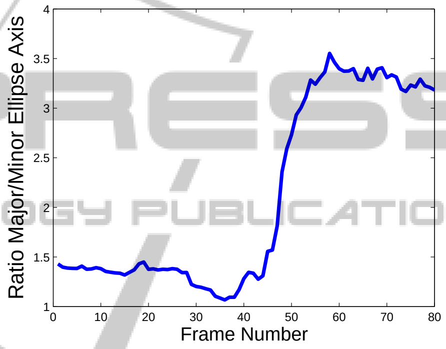

Figure 5: The ellipse major/minor axis ratio during a fall. Fall occurs between frames 40 and 60 (frame rate is 15 fps).

belongs to the foreground if $p_{b,i}(\mathbf{x}) < 0.5$. This gives us a foreground mask, on which we perform morphological operations to remove noise and find the largest blob. After obtaining a silhouette using foreground segmentation, we fit an ellipse to the silhouette and use its parameters as features. An example of an ellipse fitted to a target can be seen in Figure 4. For this ellipse, we are interested in the ratio r of major axis to minor axis. In Figure 5, we plot the value of this ratio over time, showing that it is a good indicator of a fall. To make the features location invariant, we also add the distance d from the bounding ellipse to the center of the image to our feature vector. We use the same classification method as in our skeleton-based fall detection method; an SVM classifier with an RBF kernel.

2.3 Data Fusion Method A

Figure 6 shows data fusion method A. For both data fusion methods, we used synchronized data so that we can relate a measurement taken with the Kinect directly to a measurement taken with the overhead camera. The process of synchronizing the cameras is described in section 3.1. Data fusion method A takes

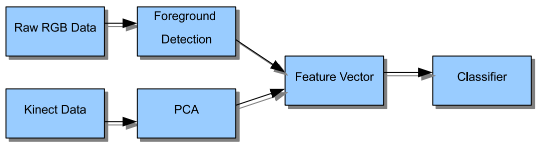

Figure 6: Overview of data fusion classification method A.

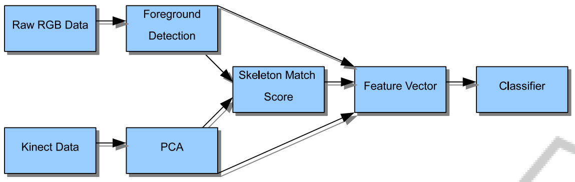

Figure 7: Overview of data fusion classification method B.

the features from the bounding ellipse and skeleton-based methods, and appends these features into one feature vector. This vector is used for classification.

2.4 Data Fusion Method B

In data fusion method B (Figure 7), we use the overhead camera to validate the skeleton tracked by the Kinect. The idea is that the skeleton not being tracked properly will result in false alarms in fall detection. Incorrectly tracked skeletons are unstable, and may change erratically between measurements, causing motions that are interpreted as falls. If we can measure how well the skeleton is being tracked, we can quantify how much weight we can give to the skeleton data. For validating the skeleton, we see how well its reprojection into the overhead camera image matches the foreground. Figure 8 shows the skeleton reprojected into the overhead camera image. In order to better model the shape of the human body, we expand each joint using a cylinder, the result of which can be seen in Figure 8 on the right. After foreground segmentation and skeleton reprojection, we have two binary images F and S. We define our measure of how well the skeleton matches the foreground as follows:

$$m = \frac{|F \cap S|}{|F \cup S|} \tag{2}$$

The higher the value of m, the better the match between the two image masks. For each pair of measurements (skeleton and foreground segmentation) we calculate a match score m. These match scores, along with features from the bounding ellipse and skeleton classification methods, are appended into a feature vector, on which SVM classification is done.

3 LAB & DATA SET DESCRIPTION

3.1 Lab Setup & Synchronization

The data for our experiments was recorded in a simulated living room using two cameras: a Microsoft Kinect and a Vivotek FD7131 overhead camera. The Kinect is connected to the recording laptop by a USB cable, sending data to the Microsoft Kinect SDK (version 1.0), which analyzed the depth image to track the user's skeleton at a frame rate of 10 to 30 fps. The overhead camera is connected through the network, sending data over the network compressed using the MPEG-4 video codec. Data is sent at a rate of 15 fps at a resolution of 640x480. The overhead camera has tangential distortion, which we corrected using the model by (Brown, 1971). Data was recorded during different periods of the day, with different lighting conditions. In some samples, the scene was illuminated only by sunlight, in other samples, artificial light was used. To synchronize the data from the two data sources with sub-second precision, we displayed a timer on the television screen in the room. Then, we can get a time stamp from network camera image with sufficient accuracy and correlate it to the Kinect data. While this is not a viable solution in a real-life scenario, there are many ways to deal with this issue depending on the available hardware.

3.2 Data Set & Preprocessing

The recorded falls all take place in an area of roughly 3x3 meters, and the direction of the fall is varied. In total, 40 fragments of about 5 minutes were recorded. Each fragment contains mostly general activity with a fall at the end. These falls were manually annotated. For SVM classification, we need all samples to have the same length, which means that they need to have the same amount of temporal data. Since the Kinect's frame rate is not consistent, we use linear interpolation to get the sample data at regular intervals. We apply interpolation to both the skeleton joints and the bounding ellipse features. From our data set we determined that we only need to look at the first second of data to capture the complete falling motion. The final dataset that we test each method on contains measurements taken during the first second of a fall, for each of the 40 falls. 8 measurements are used for every fall, with uniform spacing between measurements. Similarly, 40 samples were chosen from general activity footage. These contained samples of the target performing general activity.

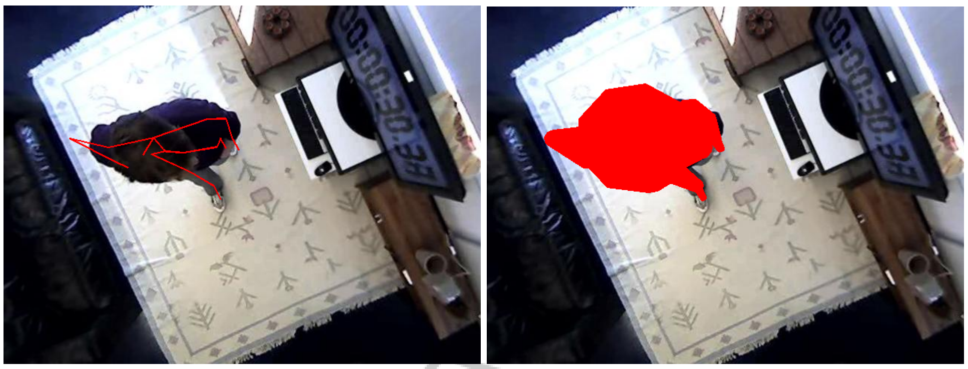

Figure 8: The skeleton reprojected into the overhead camera image. Next to it is the expanded skeleton reprojected into the same camera image.

Table 1: Classification results of classification. Numbers are mean/variance of the True Positive Rate and True Negative Rate of 5-fold cross-validation over 25 classification experiments.

	TPR mean	TPR variance	TNR mean	TNR variance
Skeleton-Based	0.969	0.007	0.926	0.003
Bounding ellipse	0.907	0.009	1	0.000
Data Fusion A	0.921	0.029	0.92	0.021
Data Fusion B	0.984	0.037	0.987	0.017

4 EXPERIMENTS

In our first two experiments, we look at classification performance for the skeleton-based classification method and the bounding ellipse classification method. We then evaluate our data fusion methods on the same dataset and show that classification results are improved for data fusion method B. For evaluating each of our classifiers, we use 5-fold crossvalidation for classification on our dataset. Each classification method is tested on the same 40 samples of falls and 40 samples of non-falls. Since there is some randomness in the selection of the folds, we repeat this cross validation experiment 25 times. The results are shown in Table 1. True Positives (TP) are falls classified as falls, while False Negatives (FN) are falls classified as non-falls. Similarly, False Positives and True Negatives represent false alarms and correctly classified non-falls, respectively. We see that the skeleton-based classifier performs quite well, with the scores indicating on average about 3 false positives and about 1 false negative. The false negatives were caused by the target falling near the edge of the screen, and a large part of the joints being (incorrectly) inferred. The false positives were fragments where the target was sitting or lying on the couch and the skeleton changed erratically between frames due to high uncertainty in the skeleton tracker. The bounding ellipse method has a relatively high number of false negatives compared to the skeleton-based classification method. These were falls where the shape of the bounding ellipse did not change enough due to the position of the person and the direction of the fall. Data Fusion Method A does not perform very well. It has a relatively high number of false negatives and false positives, so it seems this approach for data fusion is not very effective. There is a good explanation for this: the classifier has no information on which feature set is more reliable, which would be useful information for falls in which the skeleton is not tracked properly. On the other hand, data fusion method B shows a low number of false positives and a low number of false negatives, outperforming the other methods in terms of absolute numbers of misclassifications. This is due to our data fusion approach: the classifier has information on when the skeleton is not being tracked properly, and will attribute more weight to other features.

5 DISCUSSION

The question we wanted to answer in this research was: can we improve classification accuracy by fusing data from the Kinect and an overhead camera? When simply classifying on concatenated features, performance did not improve. However, using the skeleton match score resulted in high classification scores with only one false negative on average. Since this false negative is the same false negative that originally occurred in both the skeleton-based classification method and the bounding ellipse method, it seems that the fusion parameter worked exactly as intended; indicating when the Kinect skeleton is reliable. If the skeleton match score is low, then the bounding ellipse has more influence on the classification result, and fewer misclassification are made. Our conclusion is that classification results are indeed improved by data fusion.

We were able to compare the skeleton-based classification method to an existing bounding ellipse method. Comparing to other state-of-the-art methods is more difficult, however, since a lot of these methods use a specialized sensor setup. Additionally, not all papers report the full confusion matrix in their results, or they have non-binary classifiers. This makes it hard to compare our results quantitatively to other state-of-the-art methods.

5.1 Future Work

Before a fall detection system is used in practice, it will have to be able to deal with the challenges of observing day-to-day life. For example, pets or TVs can make tracking a person more difficult. The Kinect proved quite useful for fall detection, but it needs to be better at tracking targets that sit down or lie on a couch for the Kinect to be truly reliable in a stand-alone setup. The limited range and field of view of the Kinect also make it difficult to apply in Kinect in some situations. Testing the method on real-life data instead of simulated falls should also be done; young people do not fall in the same way as elderly people. Unfortunately, not many real-life falls have been recorded.

5.2 Conclusions

In this research, we have evaluated four fall detection methods. The first two methods had a high number of either false positives or false negatives. To combine data from both methods, we implemented two data fusion methods. The results show that our data fusion method B outperforms the other methods, by using a match score that estimates the reliability of tracked skeleton. With this we have shown that data fusion between sensors of different modalities is beneficial for fall detection.

ACKNOWLEDGEMENTS

This work has been funded by the Foundation Innovation Alliance (SIA - Stichting Innovatie Alliantie), in the framework of the *Balance-IT* project.

REFERENCES

Alemdar, H., Yavuz, G., Özen, M., Kara, Y., Incel, Ö., Akarun, L., and Ersoy, C. (2010). Multi-modal fall detection within the WeCare framework. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 436–437. ACM.

Anderson, D., Luke, R., Keller, J., Skubic, M., Rantz, M., and Aud, M. (2009). Linguistic summarization of video for fall detection using voxel person and fuzzy logic. *Computer Vision and Image Understanding*, 113(1):80–89.

Brown, D. (1971). Lens distortion for close-range photogrammetry. *Photometric Engineering*, 37(8):855–866.

Foroughi, H., Rezvanian, A., and Paziraee, A. (2009). Robust Fall Detection Using Human Shape and Multi-class Support Vector Machine. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 413–420. IEEE.

Gallagher, B., Corbett, E., Freeman, L., Riddoch-Kennedy, A., Miller, S., Smith, C., Radensky, L., and Zarrow, A. (2001). A fall prevention program for the home environment. *Home care provider*, 6(5):157–163.

Gillespie, L. (2004). Preventing falls in elderly people. *Bmj*, 328(7441):653–654.

Kannus, P., Parkkari, J., Koskinen, S., Niemi, S., Palvanen, M., Järvinen, M., and Vuori, I. (1999). Fall-induced injuries and deaths among older adults. *JAMA: the journal of the American Medical Association*, 281(20):1895–1899.

Liu, C., Lee, C., and Lin, P. (2010). A fall detection system using k-nearest neighbor classifier. *Expert Systems with Applications*.

Mastorakis, G. and Makris, D. (2012). Fall detection system using kinects infrared sensor. *Journal of Real-Time Image Processing*, pages 1–12.

Oliver, N., Rosario, B., and Pentland, A. (2000). A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843.

Rougier, C., Auvinet, E., Rousseau, J., Mignotte, M., and Meunier, J. (2011). Fall detection from depth map video sequences. *Toward Useful Services for Elderly and People with Disabilities*, pages 121–128.

Tao, J., Turjo, M., Wong, M., Wang, M., and Tan, Y. (2005). Fall incidents detection for intelligent video surveillance. In *Information, Communications and Signal Processing, 2005 Fifth International Conference on*, pages 1590–1594. IEEE.

Töreyin, B., Dedeoğlu, Y., and Çetin, A. (2005). HMM based falling person detection using both audio and video. *Computer Vision in Human-Computer Interaction*, pages 211–220.

Yu, X., Wang, X., Kittipanya-Ngam, P., Eng, H., and Cheong, L. (2009). Fall Detection and Alert for Ageing-at-home of Elderly. *Ambient Assistive Health and Wellness Management in the Heart of the City*, pages 209–216.