Aleksandra Postawka [ID]
Jarosław Rudy [ID]

LIFELOGGING SYSTEM BASED ON AVERAGED HIDDEN MARKOV MODELS: DANGEROUS ACTIVITIES RECOGNITION FOR CAREGIVER SUPPORT

Abstract

In this paper, a prototype lifelogging system for monitoring people with cognitive disabilities and elderly people as well as a method for the automatic detection of dangerous activities are presented. The system allows for the remote monitoring of observed people via an Internet website and respects the privacy of the people by displaying their silhouettes instead of their actual images. The application allows for the viewing of both real-time and historical data. The lifelogging data (skeleton coordinates) needed for posture and activity recognition are acquired using Microsoft Kinect 2.0. Several activities are marked as potentially dangerous and generate alarms sent to caregivers upon detection. Recognition models are developed using Averaged Hidden Markov Models with multiple learning sequences. Action recognition includes methods for differentiating between normal and potentially dangerous activities (e.g., self-aggressive autistic behavior) using the same motion trajectory. Some activity recognition examples and results are presented.

1. Introduction

In its original meaning, *Lifelogging* is a practice sharing one's life data with other people or even the general public with the use of various wearable devices. This practice can be traced back to the 1980s when Steve Mann started his work on his line of Wearable Wireless Webcam devices [6]. Since then, the popularity of lifelogging has greatly increased. This was partially facilitated by the development of lighter and more sophisticated wearable devices.

While lifelogging started out as a social phenomenon, its applications have since spread. Nowadays, it can be used for the remote or automatic monitoring of people in places like hospitals or for elderly care. Lifelogging can mean the acquisition of any kind of real-time personal or life data, but it most commonly means the acquisition of video data. In this case, however, remote care lifelogging systems have to ensure the privacy of the people being observed.

Human Activity Recognition (HAR) is a research field that aims at classifying or predicting the actions of an individual based on observations. Such a process is very important for lifelogging, as it allows us to classify what the observed person is doing (activity, body position, gestures, *etc.*) without assistance or with limited help from the supervising individuals.

While HAR is used for recognizing a variety of activities in many applications, there is a subfield of HAR called *Abnormal Human Activity Recognition* (or AHAR). This subfield is concerned in recognizing abnormal, dangerous, or alarming situations that frequently arise in healthcare, elderly care, or care for cognitively impaired persons (*e.g.*, autistic children). Such situations include fainting, seizures, falling, self-aggression, unattended home exits, and others. It is important to recognize such abnormal activities automatically with high accuracy and a low false-positive rate.

In this paper, we present an approach to dangerous activity recognition (which falls under AHAR) by the use of Hidden Markov Models (HMMs). In particular, we employ Averaged Hidden Markov Models (AHMMs) as the basis for developing a classifier that is able to recognize activities (including posture changes and hand activities) and classify an activity as potentially dangerous. We also present a prototype of a lifelogging application that uses the developed classifier together with a Microsoft Kinect 2.0 sensor. The application allows for the real-time monitoring a person without invading the person's privacy while gathering data for future consultation and providing automated alarm messages via e-mail or SMS.

The remainder of this paper is structured as follows. Section 2 presents an overview of literature on lifelogging as well as activity and gesture recognition. Section 3 contains the notation used throughout the remainder of the paper. Section 4 contains the primary findings of the paper, including mathematic basics, methods, and the process leading to the development of activity recognition models based on HMMs as well as some examples, testing results, and definitions of the positions and activities being recognized. Section 5 describes the prototype lifelogging application utilizing the developed models. Finally, Section 6 presents the conclusions.

2. Literature overview

In this section we present a review of the papers related to lifelogging as well as the detection of abnormal and dangerous activities and gesture recognition, focusing on those works most relevant to this paper. We also focus on the applications of the presented approaches, including healthcare, support, and the monitoring of children and the elderly.

2.1. Lifelogging

As explained in Section 1, lifelogging started out as a practice of sharing one's life data with other people, effectively becoming a social phenomenon with little connection with science. However, new applications for lifelogging have been found recently (healthcare in particular). This has led to a number of papers studying various aspects of lifelogging or proposing it as a solution to some problems. In this overview, we focused on lifelogging aspects relevant to the topic of this paper. For further reading on other aspects of lifelogging, please consider [2].

Various limits and controversial aspects of lifelogging were the topic of a paper by O'Hara *et al.* [9]. One such aspect discussed in the paper is privacy. The authors argued that the privacy concerns associated with lifelogging are overblown, as the activity data collected by lifeloggers would often be voluntarily released into the public domain. The authors also claimed that the information gathered through lifelogging can be empowering for the lifelogger and help him construct an online identity. However, while true for the original life-sharing application of lifelogging, privacy is more of a concern for other applications (like healthcare).

One example of such an application was presented in a paper by Wu *et al.*, where the SensCare system for semi-automatic activity logging for elderly care was introduced [19]. In their work, mobile sensors were employed with the ability to log a user's activity for 24 hours without charging. The authors concluded that their unsupervised method of activity logging had a reasonable accuracy (an average F1 score of 0.65) and that the system is directly beneficial to the users, as it allows them to recall their activities at particular points in the past. Moreover, the authors argue that the lifelogging systems existing at the time generated low-quality activity logs, severely limiting the uses of lifelogging for elderly care.

Yoshihara *et al.* proposed a lifelogging system for the elderly through a combination of passive and active perception [21]. Passive perception was realized by a network of sensors (including smartphones), while active perception was realized by a conversation system for direct human-to-robot partner communication. Finally, the lifelog data was visualized to the elderly peoples' families and caretakers through a cloud database.

Similarly, Van Kasteren *et al.* introduced an unintrusive activity monitoring system for elderly care [4]. The system employed a wireless sensor network for collecting observations. Two approaches were used: an HMM as an example of generative mo-

dels and Conditional Random Fields (or CRFs) as an example of a discriminative model. Finally, four real-world datasets were used to test the effectiveness of the approaches. The results indicated that CRFs are more prone to overfitting. The authors also concluded that the layout of a house greatly affects the performance of the activity recognition.

As a final paper concerned with lifelogging, we discuss a work by Jalal *et al.* [3] where a depth video sensor was used together with HMMs for human activity recognition for elderly care in smart indoor environments. In the first step, information from the depth sensor was used to train Markov models to recognize the activities. In the second step, the trained system was used to produce lifelogs. The complete system was tested against smart indoor activity and MSRDailyActivity3D datasets. The results were promising, making the approach proposed by the authors suitable for home, office, or hospital environments.

2.2. Abnormal activity recognition

In this subsection, we overview papers related to the recognition of abnormal activities, including fall detection and child tantrum analysis. For a more exhaustive review on video-based abnormal human activity recognition, please consider [11].

The first category of abnormal situations often considered in the literature is a human fall. As such, a number of approaches exist for detecting such situations (most commonly based on visions). For example, Miaou *et al.* proposed a system employing a MapCam omni-camera for elderly fall recognition [7]. The MapCam is used to map a 360-degree field and eliminate blind zones. Personal information of the observed people are used to fine-tune the system in order to reduce the number of false alarms. Tests indicated that the accuracy of the system was 68% and 79.8% without and with personal information, respectively. The authors claimed that the system was reliable based on a Kappa value of 0.798.

The next paper on fall recognition for elderly care by Nana *et al.* employed Kinect combined with a three-axis accelerometer for the generation of data that was then classified with Support Vector Machines (SVMs) [8]. The experiments indicated that the accuracy of the proposed system was above 98%. Acyclic graph SVMs for fall recognition were also used by Yu *et al.* [22]. The authors employed background subtraction, ellipse fitting, and a projection histogram for distinguishing different human postures. Experiments based on a dataset of 15 people in a simulated home environment indicated a high fall detection rate of over 97% while maintaining a low false detection rate of below 1%.

Yin *et al.* proposed an abnormal activity recognition system based on wireless sensors attached to a human body combined with SVM and kernel nonlinear regression methods [20]. The authors emphasized the need for reducing the false positive rate. The experiments on the real data indicated the effectiveness of the approach even without explicitly labeling the abnormal activity data. This is beneficial, as abnormal data is rarely compared to normal data.

Khan and Sohn proposed an abnormal human activity system based on R-transform, Kernel Discriminant Analysis, and HMMs [5]. The authors considered a set of abnormal activities other than human fall, including chest pain, fainting, vomiting, and headaches. The authors also ensured the privacy of the elderly people by extracting binary silhouettes from video sensors. The tested average recognition rate was over 95%.

Pal *et al.* considered gesture recognition for the purpose of avoiding occupational hazards from prolonged sitting in a particular posture [10]. The authors employed a similarity matching algorithm with 7 similarity measures for recognizing 14 different gestures. The test results were statistically validated using the Friedman test. The observed accuracy recognition was above 94%.

The last group of approaches concerns the care for people with cognitive impairments, neurodevelopmental disorders (such as autism), or diseases. One example of such an approach is a work by Rowe *et al.* [17], who proposed their CareWatch system for the care for people with Alzheimer's disease. The monitoring system was designed to automatically recognize unattended home exits and improve caretaker sleep. The effectiveness of the CareWatch system was tested on two clinical trials, with one of them being completed and one still ongoing. Results showed the system decreased the number of unattended exits and improved caretakers' sleep. The system operated for more than 200 months of combined operation times with no major failure.

A system for monitoring and recognizing potentially dangerous activities (including self-aggression) of cognitively impaired people was proposed by Postawka [14]. Averaged HMMs with an acceleration measure combined with a depth sensor were used to distinguish dangerous and normal behavior. Tests using a set of data containing hitting and touching sequences were used to ascertain the effectiveness of the proposed approach. Finally, Yu *et al.* proposed an HMM and Kinect sensor approach for observing the behavior of children [23]. Combined with medical knowledge, this method was used to help recognize abnormal activities in children. The developed prototype system was used to provide doctors and parents with valid and reliable data concerning autism symptoms for early autism detection.

2.3. Final notes and summary

As for the final part of the literature overview, we present a few more papers and reviews related to human activity and gesture recognition in general. We also summarize the overview and draw several conclusions about the state of research on lifelogging and abnormal activity recognition.

Postawka considered the problem of recognizing warm-up exercises with Kinect 2.0 [13]. The approach used an averaged HMM, constructed using a proposed Sequences Concatenation Algorithm. The resulting model was tested against 13 exercises. The results indicated that the final averaged model combined the features of the component models, recognized more sequences, and did not confuse the recognized activities with others. Similarly, Siddiqui *et al.* considered the problem of recognizing

human gestures, which is important from the viewpoint of Human-Computer Interaction (HCI) [18]. Once again, Kinect was used to train an HMM-based algorithm on three types of gestures. The algorithm was then evaluated against 228 test videos. The final accuracy of the approach exceeded 80%. For a more exhaustive study of the aspects of and approaches to human activity recognition with RGB cameras, depth sensors, and other devices, please consider [1].

From the above literature overview, we can conclude that the recognition of abnormal human activities has garnered considerable attention among researchers, and it only continues to develop. Its applications are mostly associated with healthcare and elderly care, but they can also be used in detecting the early symptoms of various cognitive impairments. Furthermore, AHAR is often combined with lifelogging, which has long since stopped being merely the sharing one's life data with others; nowadays, it is being used to monitor the activities of elderly, children, patients, or even office workers.

However, while privacy might not be a big concern for regular lifelogging, it is a concern for its healthcare applications, as privacy helps ensure that those being monitored feel comfortable. Finally, we note that video and depth sensors (Microsoft Kinect in particular) are often used as a means of gathering lifelog data. Moreover, utilizing Hidden Markov Models is one of the most commonly used approaches in achieving Abnormal Human Activity Recognition.

3. Notation

In this section, we present the mathematical notation that will be used with the Hidden Markov Models:

- $\lambda = \{A, B, \pi\}$ – 3-tuple defining complete parameter set for HMM,

- N – number of states,

- M – number of observation symbols,

- T – length of observation sequence,

- $A = \{a_{ij} : i, j \in \{1, \ldots, N\}\}$ – state transition matrix,

- $B = \{b_{ij} : i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\}\}$ – probability distribution matrix for observed symbols,

- $\pi = \{\pi_i : i \in \{1, \ldots, N\}\}$ – initial state distribution vector,

- $O = O_1, O_2, \ldots, O_T$ – observation sequence,

- $I = i_1, i_2, \ldots, i_T$ – decoded state sequence,

- $O_{j:k} = (O_j, O_{j+1}, \ldots, O_{k-1}, O_k)$ – subsequence of O, including symbols from j-th to k-th, inclusive.

Additionally, for Averaged Hidden Markov Models (AHMMs), the following notation has been applied:

- D – number of component models,
- $x^{(d)} : d \in \{1, \ldots, D\}$ – value x from d-th component model,
- $\bar{\bar{X}}$ – cardinality of X.

4. Methods

All of the methods used in the research are based on basic Hidden Markov Models and the algorithms described in Subsection 4.1. In particular, the solutions used for action recognition are based on the Averaged Hidden Markov Models [13] obtained by averaging multiple component models with a *similar* structure. The process of parameter calculation for such similar models is briefly described in Subsection 4.2. The equations for averaging the component models are presented in Subsection 4.3.

The purpose of a lifelogging system requires the developed methods to be applicable in real-time action recognition (as described in Subsection 4.4). The developed system is also capable of potentially dangerous action detection – these methods are described in Subsection 4.5. The modeled positions and activities are presented in Subsection 4.6.

4.1. Hidden Markov Models

Modeling with the use of HMMs is a probabilistic method for the study of discrete time series [12]. The complete model λ consists of three parameters: transition matrix A, symbol distribution matrix B, and initial state distribution π. In the issue of HMMs, there are three key problems that must be solved in order to use the models in practical applications [16]:

Evaluation. Compute the probability of observation sequence $P(O|\lambda)$ given observation sequence O and model $\lambda = \{A, B, \pi\}$. The mathematical solution to this problem is presented in the equation below:

$$P(O|\lambda) = \sum_{\text{all } I} P(O|I, \lambda) P(I|\lambda). \tag{1}$$

The problem could be equivalently stated as "score/evaluate the model based on the model parameters and observation sequence." The problem is usually solved by a dynamic programming method called a *forward-backward procedure* with a complexity of $O(N^2 T)$ instead of complexity $O(N^T T)$ obtained by the direct implementation of Equation (1) [16]. The method is very useful when we have multiple models and we want to choose the one that describes the given observation sequence in the best possible way.

Decoding. Estimate state sequence I given observation sequence O and model $\lambda = \{A, B, \pi\}$. The problem could be stated differently depending on the chosen optimality criterion [16]. If the aim is to choose states i_t that are individually most likely, then the impossible state sequence could be obtained in the model with disallowed transitions. Therefore, the optimality criterion used in this problem is usually the single best path; i.e., the issue is to maximize $P(O, I|\lambda)$. For the solution of the latter criterion, the *Viterbi algorithm* is used (along with dynamic programming).

Learning. Adjust model parameters $\lambda = \{A, B, \pi\}$ in order to maximize probability $P(O|\lambda)$ given observation sequence O. So far, there is no analytical method to solve this problem. Solutions are usually obtained with the use of an iterative procedure (such as the *Baum-Welch algorithm*) or gradient methods [16].

4.2. Component models

The AHMM model is based on a single *base model* and on multiple *child models*, which are structured in a similar fashion to the base model. In the learning phase, all of the component models use the temporarily created *Unique Lists (UL)*, including only the HMM symbol changes in the learning sequence (e.g., learning sequence $L = (1\ 1\ 1\ 5\ 5\ 19\ 19\ 5)$ would have generated such an UL: $L_{UL} = (1\ 5\ 19\ 5)$). The process of UL generation is described by Equation (2).

$$L = (a_i)_{i=1}^{\bar{\bar{L}}} \Longrightarrow L_{UL} = (a_i),\ a_i \neq a_{i+1}. \tag{2}$$

The described procedure is only used for model structure designation; i.e., to establish the beginning conditions for the Baum-Welch algorithm [16]. The component models structure is defined by filling the zero and non-zero values in Matrix B, as the Baum-Welch algorithm does not change zero to a positive number. Matrix A is initially an upper triangular matrix defined as follows:

$$A = \{a_{ij}\} : a_{ij} = \begin{cases} 0 & \text{for} \quad i < j \\ \frac{1}{N-i+1} & \text{for} \quad i \geq j \end{cases} \tag{3}$$

where $i, j \in \{1 \cdots N\}$.

Base model. The base model is a simple left-to-right HMM model [16] where subsequent states correspond to the subsequent HMM symbols in the UL. Basically, such a structure could be interpreted as a non-ergodic Markov chain. The learning sequence chosen for generating the base model (later called the *pattern* and denoted as P) is the longest one in terms of the number of elements in the UL.

$$P = L_{UL_i} : \forall_{j \in D} \bar{\bar{L}}_{UL_i} \geq \bar{\bar{L}}_{UL_j} \tag{4}$$

The model parameters are calculated using the Baum-Welch algorithm. The number of states in the base model is equal to $N = \bar{\bar{P}} + 1$. The last state is introduced for the eventual noise added in the later phases and $B_N = B_{N-1}$.

Child models. Child models are generated based on the structure of the base model and a new learning sequence. This process is realized by matching the symbols in a pattern and a new sequence (denoted as N) using ULs (Fig. 1). The purpose of the algorithm is to maximize the number of overlapping symbols while keeping the symbol order. The problem could be stated as follows:

$$\text{maximize} \quad \bar{\bar{X}}, \tag{5}$$

$$\text{subject to} \quad P = (p_i)_{i=1}^{\bar{\bar{P}}},$$

$$N = (n_i)_{i=1}^{\bar{\bar{N}}},$$

$$X = \{(p_i, n_j)\} : p_i = n_j,$$

$$\forall_{k,l,m,r}[(p_k, n_l) \in X \wedge (p_m, n_r) \in X \wedge k < m \implies l < r].$$

The child model has the same number of states as the base model, and the overlapping symbols are assigned to the same state. The groups of symbols between the overlapping pairs are undesirable both in the pattern and in the new sequences. Such a group of mismatched symbols in the pattern means a long skip transition between states, while a group of mismatched symbols in the new sequence means noise added to the next used state (Fig. 1, case 2).

In contrast, the alternately mismatched symbols from the pattern and new sequence (Fig. 1, case 1) stand for using the existing state from the base model with the HMM symbol replaced with the one from the new sequence. In such a case, we do not lose the information about the symbol order.

Pattern sequence	1	6	7	3	2	5	7	8
New sequence	1	3	4	9	6	8		

Case 1:	1	6	7	3		2		5		7	8
	1			3	4		9		6		8

Case 2:	1	6	7	3	2	5	7				8
	1			3				4	9	6	8

Figure 1. Case 1: example of child algorithm; Case 2: unwanted situation

The problem might be resolved by an optimal algorithm; however, it is only suitable for small instances. The method has been implemented as a tree-based algorithm where the tree leaves correspond to the P and N index changes. For each leaf in the new iteration, we get 3 new paths: (p_i, n_{j+1}), (p_{i+1}, n_j), and (p_{i+1}, n_{j+1}). The algorithm is accelerated by cutting off the branches that cannot give a better complete solution than the best partial solution already found (here, the worst future case is applied). Number of leaves Y depends on the problem instance and is located within the range of $2^{\bar{\bar{P}}+\bar{\bar{N}}} \leq Y < 3^{\bar{\bar{P}}+\bar{\bar{N}}}$.

Because of the memory complexity, an approximate algorithm has been created [13] that finds the optimal solution in most of the investigated cases. The method cuts off each of the branches with a worse partial solution. The number of leaves is estimated to be $Y < 2^{\bar{P}+\bar{N}}$.

4.3. Averaging models

In the last stage, all of the D similar component models are averaged. The vector of weights w has to fulfill condition (6). Particularly, the arithmetic average (i.e., $w = \{w_d = \frac{1}{D}\}$) meets the requirements. The elements of matrices A and B and vector π are calculated due to Equations (7), (8), and (9), respectively.

$$w = \{w_d : d \in \{1, \ldots, D\} \wedge w_d \geq 0 \wedge \sum_{d=1}^{D} w_d = 1\} \tag{6}$$

$$\bar{a}_{ij} = \sum_{d=1}^{D} w_d \cdot a_{ij}^{(d)} \tag{7}$$

$$\bar{b}_{ij} = \sum_{d=1}^{D} w_d \cdot b_{ij}^{(d)} \tag{8}$$

$$\bar{\pi}_i = \sum_{d=1}^{D} w_d \cdot \pi_i^{(d)} \tag{9}$$

Example 1 – Model generation

In an imaginary recognition system, the space is divided into $M = 9$ 3D fragments mapped into HMM symbols; i.e., $b_{ij} \in \{1, 2, \ldots, 9\}$. For an activity, the two learning sequences were recorded. For each discrete sample, the HMM symbol was assigned. As a consequence, the model is created based on two learning sequences: $L_1 = (5\ 5\ 5\ 5\ 5\ 7\ 7\ 7\ 6\ 6\ 6\ 6\ 3\ 3\ 3\ 2\ 2\ 2\ 2)$ and $L_2 = (7\ 7\ 7\ 7\ 7\ 4\ 4\ 4\ 2\ 8\ 8\ 8\ 8\ 1\ 1)$. For each symbol list, the UL is created: $L_{1_{UL}} = (5\ 7\ 6\ 3\ 2)$ and $L_{2_{UL}} = (7\ 4\ 2\ 8\ 1)$.

The base model is created based on sequence L_1, while both ULs have the same length (otherwise, the longer one would be chosen). Number of states $N = 6$ is the list length increased by 1. Initially, matrix B is filled with 1 for respective symbols from $L_{1_{UL}}$ and with 0 otherwise. This means that only the following elements are positive: b_{15}, b_{27}, b_{36}, b_{43}, b_{52}, and b_{62}. Matrix A is an upper triangular matrix according to Equation (3), and vector $\pi = (1\ 0\ 0\ 0\ 0\ 0)$. Next, the initially filled-in matrices are forwarded to the Baum-Welch algorithm. The calculated parameters of the base model are presented in Figures 2 and 3.

The child model is generated based on the base model and new sequence L_2. The solution for the child structure problem for $P = L_{1_{UL}}$ and $N = L_{2_{UL}}$ resolved by the optimal or approximate algorithm is presented in Figure 4. The initial conditions for Matrix B in the Baum-Welch procedure include the positive values for b_{15}, b_{27},

b_{36}, b_{43}, b_{44}, b_{52}, b_{61}, b_{62}, and b_{68}. Matrix A is an upper triangular matrix again, and vector $\pi = \{\pi_i = \frac{1}{6}\}$. The parameters calculated by the Baum-Welch algorithm are presented in Figures 5 and 6. The initial distribution vector equals $\pi = (0\ 1\ 0\ 0\ 0\ 0)$.

The averaged model is created based on the base model, child model, and Equations (7–9). The final graph and HMM symbol distribution for the states is presented in Figures 7 and 8. The initial distribution vector equals $\pi = (\frac{1}{2}\ \frac{1}{2}\ 0\ 0\ 0\ 0)$.
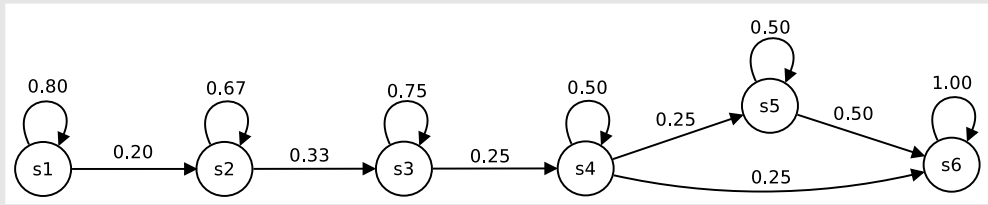
Figure 2. Base model – graph reflecting transition matrix
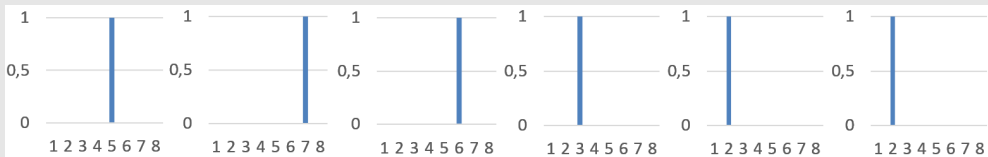
Figure 3. Base model – HMM symbol probability distribution for states

| Pattern sequence | 5 | 7 | 6 | 3 | 2 |
| New sequence | 7 | 4 | 2 | 8 | 1 |

| | 5 | 7 | 6 | | 3 | 2 |
| Output | | 7 | | 4 | | 2 |

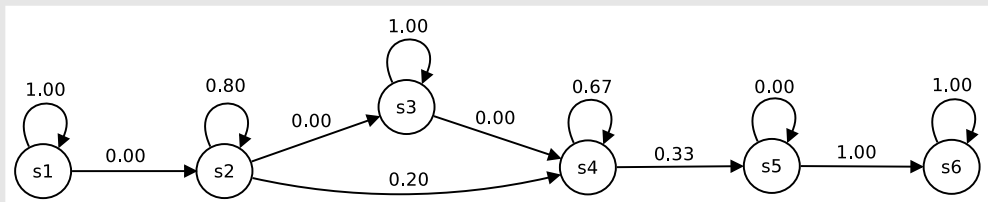**Figure 4.** Solution for child structure problem

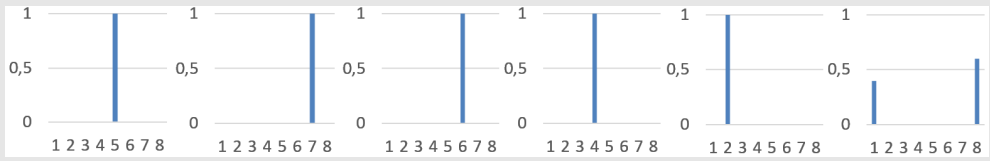**Figure 5.** Child model – graph reflecting transition matrix

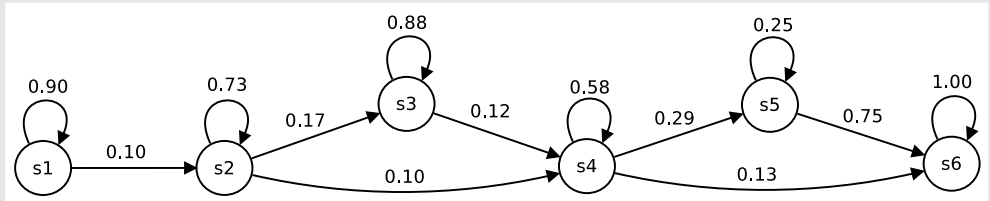**Figure 6.** Child model – HMM symbol probability distribution for states

Figure 7. Averaged model – graph reflecting transition matrix
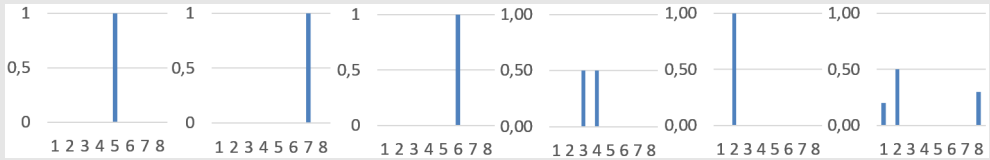

Figure 8. Averaged model – HMM symbol probability distribution for states

4.4. Real-time recognition

Real-time recognition in an HMM is a much more complicated task than the typical evaluation. The requested information is not just the question of whether sequence O could be generated by model λ (by calculating posterior probability $P(O|\lambda)$) but also whether the recognized activity is complete or not [14]. The posterior probability for observation sequence $O = O_{1:T}$ and model λ fulfills the following condition:

$$P(O_{1:T}|\lambda) > 0 \implies P(O_{1:t}|\lambda) > 0, \text{ where } t \in \{1, \ldots, T\}, \tag{10}$$

Thus, all of the partial beginning sequences of the complete activity sequence would be recognized. Therefore, additional parameter N_R (the ID of the last state that must be accessed) has been added to the AHMM model. In the component models, the value is set based on the learning sequence and obtained model. In the resultant model, the value is chosen as $N_R = min\{N_R^{(d)}\}$.

Moreover, to check whether the recognized activity is complete or not, the last state of the AHMM model is searched for the last symbol from the sequence. Vector B_N contains the probability distribution of the symbols that occurred at the end of all of the learning sequences. Value $\bar{b}_{Nj} = 0$ means that the complete activity sequence cannot end with symbol j.

Summarizing, for each new symbol sequence O, the posterior probability is calculated. If it is greater than zero, then condition $\bar{b}_{NO_t} > 0$ is checked. If the condition is fulfilled, then the state sequence is decoded using the Viterbi algorithm [16]; finally, the last decoded state is compared with the N_R value. If it is equal or greater, then the sequence is marked as recognized and complete.

As the activity sequence could start at each time moment, a list of partial sequences is generated (Fig. 9). For each new observation symbol $O_t \neq O_{t-1}$, a new partial sequence (single-element) is created and all other sequences already existing in the list are prolonged by this new symbol. Each of these sequences are tested by the algorithm from the previous paragraph. Partial sequence $O_{t_1:t_2}$ is deleted from the list only when $P(O_{t_1:t_2}|\lambda) = 0$ for all λ models.

$$\{\} \stackrel{1}{\Rightarrow} \{\{1\}\} \stackrel{4}{\Rightarrow} \left\{ \begin{array}{c} \{4\} \\ \{1,4\} \end{array} \right. \stackrel{3}{\Rightarrow} \left\{ \begin{array}{c} \{3\} \\ \{4,3\} \\ \{1,4,3\} \end{array} \right. \stackrel{7}{\Rightarrow} \left\{ \begin{array}{c} \{7\} \\ \{3,7\} \\ \{4,3,7\} \end{array} \right.$$
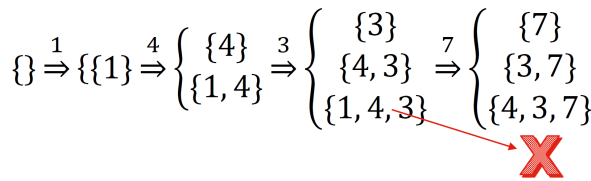
Figure 9. List of partial sequences in real-time recognition

4.5. Potentially dangerous action detection

The potentially dangerous activities that are much different from all normal activities are quite easy to detect using the previously described methods. The problem arises when the motion uses the same trajectory as another modeled activity and the only difference is the speed of motion [14].

Posterior probability $P(O|\lambda)$ is not a sufficient threshold, as can be seen in the exemplary diagram in Figure 10a. In the chart, the histograms of the $log(P(O|\lambda))$ value for the similar motions (hitting the head and touching the head with right hand) have been presented. The diagrams are not separable.
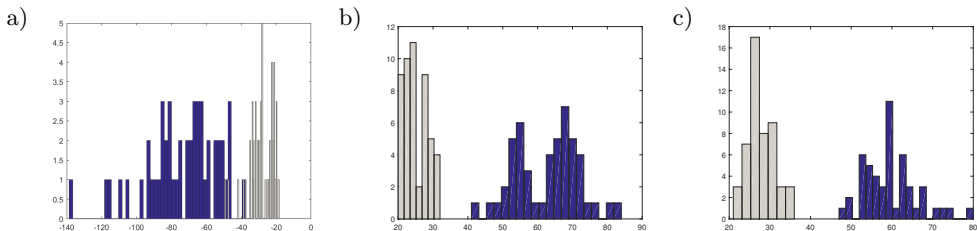

Figure 10. Examination of feature f quality as discriminant. Histograms for hitting and touching head with one/both hands: a) $f = log(P(O|\lambda))$, right hand; b) $f = \bar{\bar{O}}_{t_1:t_2}$, right hand; c) $f = \bar{\bar{O}}_{t_1:t_2}$, both hands

In reality, with equally quantized time, number of samples $\bar{\bar{O}}_{t_1:t_2}$ is a kind of a measure for the speed of motion. In order to get rid of the idle processes at the beginning and end of the action, the repeating symbols have been removed; e.g., $L = (2\ 2\ 2\ 5\ 5\ 19\ 25\ 25) \implies L = (2\ 5\ 5\ 19\ 25)$. As can be seen in Figures 10b and 10c, the number of observations in the sequence discriminates the classes; thus, this feature has been calculated for each pair of normal and potentially dangerous movements that use the same motion trajectory.

4.6. Experiments and results

The activities modeled in the system are presented in Table 1. The left column presents the list of 7 positions recognized in the system. The position recognition is realized based on tree-based classifier [15], with a recognition accuracy above 99% for the test data.

Table 1

List of modeled exercises

Position	Position change	Activity
Standing	(A) Standing → Sitting (chair)	Right arm twisting forward
Sitting (chair)	(B) Sitting (chair) → Standing	Right arm twisting backward
Kneeling	(C) Standing → Kneeling	Raising and lowering right hand
Bending	(D) Kneeling → Standing	Left arm twisting forward
Lying	(E) Standing → Bending → Standing	Left arm twisting backward
On all fours	(F) Sitting (chair) → Bending → Sitting (chair)	Raising and lowering left hand
Sitting (ground)	(G) Sitting (ground) → Lying	Both hands twisting forward
	(H) Lying → Sitting (ground)	Both hands twisting backward
	(I) Sitting (ground) → Standing	Raising and lowering both hands
	(J) Standing → Sitting (ground)	Clapping hands
		Clapping hands over the head
		Crawling forward
		Crawling backward
		Touching head with right hand
		Touching head with left hand
		Touching head with both hands
		Hitting head with right hand
		Hitting head with left hand
		Hitting head with both hands

The middle column contains a list of the 10 activities based on the position changes. These activities are modeled with AHMMs. The average recognition rate for these models is above 95%. The experiment results for these activities are discussed later in this subsection. The activities are labeled with letters for easier reference.

Finally, the right column presents 19 activities based on hand movements. The activities labeled as potentially dangerous are shown in italics. The average recognition rates for the normal activity models are above 92% and 69% for one- and two-hand activities, respectively [13]. The least recognized activity is crawling, as such a motion is hardly captured by the device. The both-hand motions without

crawling are recognized with an accuracy above 86%. The potentially dangerous activities are detected with a recognition rate of 84% and 52% for one- and both-hand motions, respectively [14]. The system is fully scalable – the activities can be easily modified by adding or removing models from the appropriate directory.

In this work, the emphasis has been placed on the activities based on posture changes. Such models are versatile and can be applied to many problems. For example, the classifier could be adjusted to detect the number of position changes within a time period, which can then be used to estimate the emotional state of the observed person (nervousness detection).

AHMMs have been adopted to this problem for the first time. The HMM observation symbols (position numbers) are obtained from the position classifier. For each of the 10 activities, 41 sequences have been recorded. The recording contains the different variants of each activity; e.g., two examples of *Standing* ⟶ *Bending* ⟶ *Standing* activities are presented in Figure 11.
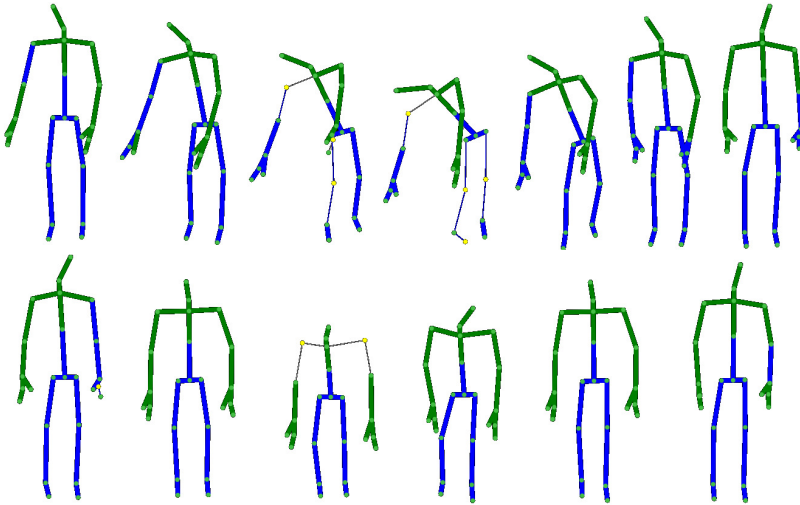

Figure 11. Two different variants of same activity (a few chosen frames)

The bone thickness in the picture only corresponds to the joint accuracy obtained from the Kinect library (the thinner bones are inferred). The sequences composed of the human silhouette coordinates and the HMM symbols were manually cut out from the recording and sorted by the activity ID. Next, the sequences were divided into a training set (16 sequences per action) and test set (25 sequences per action). The learning phase proceeded in accordance with the description in Section 4.1. The learned models were first examined by the sequences from the training set so as to make sure that there are no false positives and each of the learning sequences is recognized. Finally, the model's confusion matrix (visible in Table 2) was calculated for the test set.

Table 2

Confusion matrix for activities based on position changes (labels taken from middle column of Table 1)

	A	B	C	D	E	F	G	H	I	J
A	24	0	0	0	0	0	0	0	0	0
B	0	25	0	0	0	0	0	0	0	0
C	0	0	24	0	0	0	0	0	0	0
D	0	0	0	22	0	0	0	0	0	0
E	0	0	0	0	24	0	0	0	0	0
F	0	0	0	0	0	24	0	0	0	0
G	0	0	0	0	0	0	25	0	0	0
H	0	0	0	0	0	0	0	23	0	0
I	0	0	0	0	0	0	0	0	24	0
J	0	0	0	0	0	0	0	0	0	23

The average recognition rate for the activities based on position changes is 95.2%. The good results are obtained due to the high classifier accuracy. However, it should be noted that the real accuracy mainly depends on the observed person's position relative to the Kinect device. If the person is fully visible to the device, then the recognition rate should be similar to the one measured in this experiment. Otherwise, the limb coordinates may be incorrectly estimated by the Kinect library; therefore, the classifier may return misleading information. On the other hand, such noisy recordings (based only on a person's silhouette) are not recognized by humans either.

Another important achievement is the lack of false positives. In the case of short sequences (often containing 2 or 3 symbol changes), there is a risk of adding too many transitions and too much noise to the model. In effect, such a model would recognize not only its own activities but also many others that are seemingly not similar. Such a problem occurred while not enforcing the alternate symbols in the child models.

5. Lifelogging application prototype

In this section, we present a prototype logging application created using the developed activity recognition models along with a posture classifier. This application can be divided into two major components: (1) a vision platform; and (2) a workspace PC with a server. The architecture of the system is shown in Figure 12.

The vision platform consists of a Microsoft Kinect 2.0 sensor and a step motor. Kinect is connected to the workspace server and sends the sensor data of the observed person to the server using JSON. Such a system works, but the angle of observation is limited. In order to alleviate this, a step motor is added to allow the Kinect sensor to be rotated in order to change the observation angle. The step motor is controlled by the input from the workstation server.
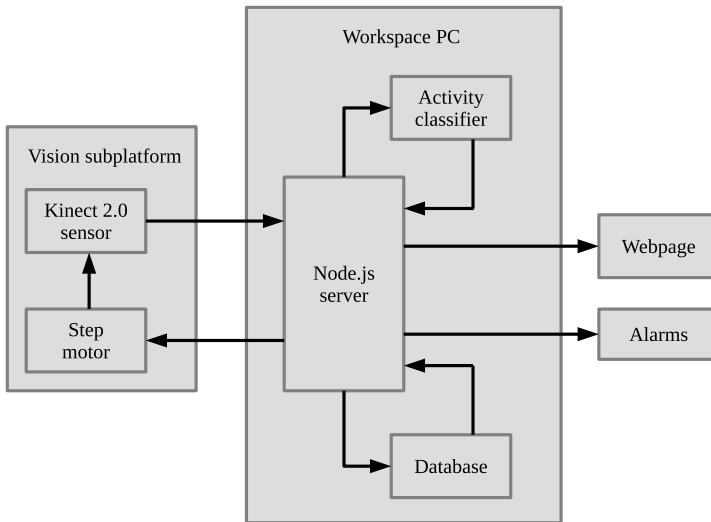
Figure 12. Lifelogging application architecture

The workspace machine hosts a Node.js server (written in JavaScript), which employs the developed models to recognize and classify the activities and positions. An SQL database is used to store the observed data for future viewing. The server analyzes the obtained image data and sends control signals to the step motor in order to keep the observed person in the center of the observation area. The server also hosts a webpage that can be used to access the application from anywhere in the world via the Internet. The application also includes an alarm service for automatically messaging the caretakers via e-mail or an SMS when a dangerous activity is detected. Screenshots of the working application presenting some of its features are shown in Figures 13, 14, and 15.
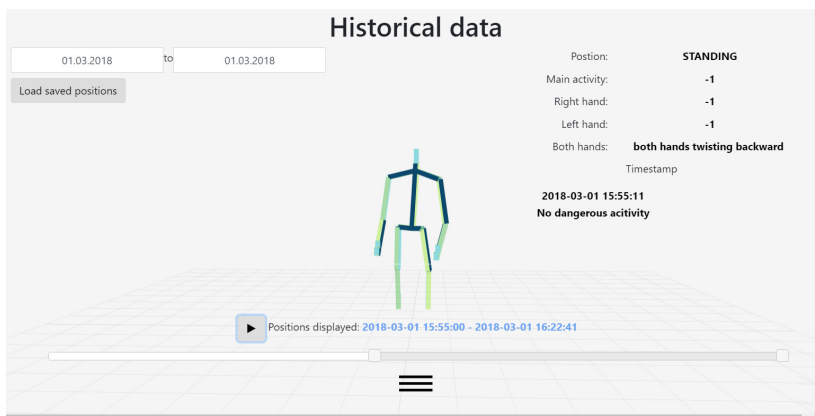

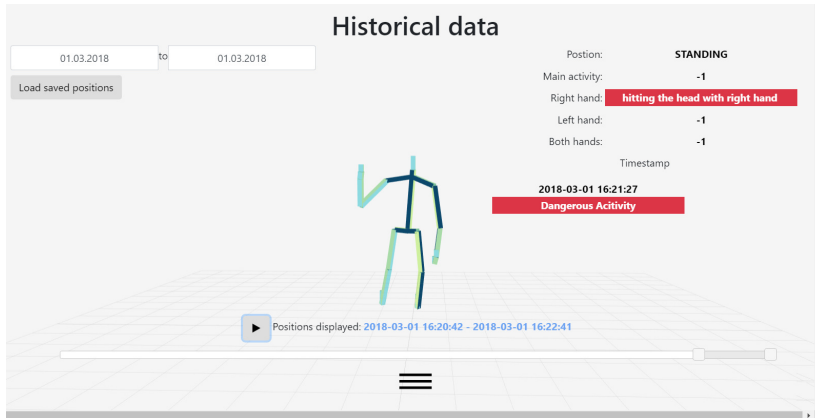**Figure 13.** Example detection of non-dangerous activity

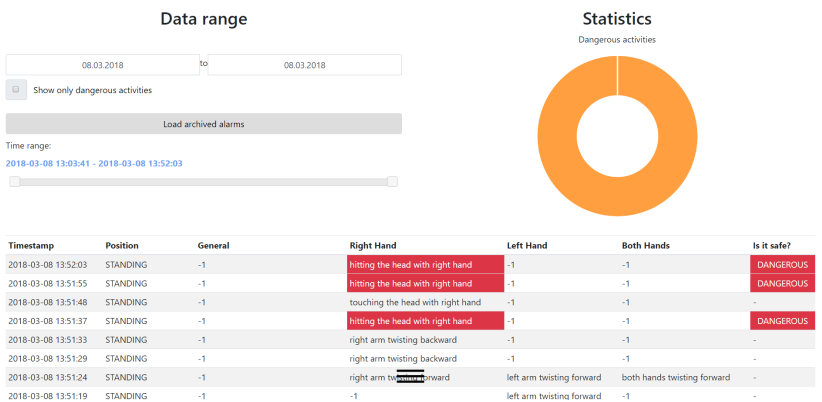**Figure 14.** Example detection of potentially dangerous activity

Figure 15. Example recognition history and statistics

The application has the following functionalities and features:

- the possibility of real-time monitoring of the activities of observed person by multiple caretakers from anywhere in the world,
- the possibility of viewing historical data for a chosen time period, complete with silhouette visualization,
- ensuring privacy by visualizing only the silhouette of the monitored person (in both the real-time and historical modes),
- support for various web browsers with no need for dedicated software,
- automated recognition and classification of positions and activities with recognition of potentially dangerous activities,

- easily extendable – more recognition models for new activities can be added simply by replacing a few files,
- automated warning messages through e-mail or SMS in the case of recognizing dangerous activity,
- the possibility of automatically keeping the monitored person in the center of the viewing area with the use of a step motor,
- accounts for caretakers.

It should be noted that the presented application is just a prototype serving to test the developed recognition method in practical applications. The classifier performs well on its own; however, the addition of the database, server, and other functionalities decreased the rate at which the data from the Kinect sensor is processed, visualized, and classified. This in turn negatively affects the accuracy of the classifier. In order to fix this, the application code or architecture must be optimized. Alternatively, the classifier could be improved to work faster (*e.g.*, by the use of parallel computing). Moreover, the number of currently recognized activities is still limited, and more work is necessary to define models for a wider range of possible activities.

The next issue concerns the mechanism of controlling the observed area by rotating the Kinect sensor with the step motor. The drawback of this approach is the fact that rotating about the X axis (up and down) forces the sensor to recalibrate itself with regards to the floor level, which takes several seconds. Fortunately, rotation about the X axis is not necessary in practice, because the observed person is unlikely to move any part of his/her body higher than 2.5 meters above ground level. Thus, the sensor can remain fixed with regards to the X axis as long as it covers the area from the floor to 2.5 meters above it. However, rotation about the Y axis (left and right) might be required in practice. Fortunately, slow movement about the Y axis is possible. Alternatively, the application could be modified to employ more than one sensor observing the same person and connected to the same server. Such a setup would have two advantages. First, we can place a restriction that all sensors cannot rotate at the same time, meaning at least one sensor remains operational when the others are recalibrating. Second, multiple sensors observing the same person could improve the accuracy of the position and activity recognition.

The issues above must be fixed before the application can be moved to the next development stage; however, even at the current stage, the prototype highlights how the developed activity recognition models could be used in practice without invading the privacy of the observed people.

6. Conclusion and future work

In this paper, an Abnormal Human Activity Recognition model for the purpose of lifelogging and the automated or semi-automated monitoring of elderly or other care-requiring people were presented. The models are based on the concept of Averaged

Hidden Markov Models, which were used to recognize actions as well as detect potentially dangerous activities; *e.g.*, the self-aggressive behavior in autism. The AHMM activity models can be created for individuals and easily added to the system without changing the source code. This feature is extremely important in the case of people with autism, but it may be also useful in the case of different afflictions.

The methods for AHMM calculation based on HMM observation sequences were presented in this paper. The description also included methods for potentially dangerous activity detection and an algorithm for real-time action recognition with the use of AHMMs. The recognition data included 7 human positions, 10 position changes, and 19 activities. The data was captured by a Microsoft Kinect 2.0 depth vision sensor, which allows us to monitor the observed people without invading their privacy.

Moreover, a prototype lifelogging system was presented in this paper. The application shows how the developed recognition models could be used in practice and allows for the monitoring of observed people based on both real-time measurements and historical data.

It should be noted that the models used for activity recognition have to be chosen individually and adjusted to the specific problem. Further research will concern adapting the described methods to be used for the support and monitoring of children with low-functioning autism. Multiple long-term observations of a single child have to be recorded and analyzed in cooperation with their parents and psychologists in order to develop viable recognition models.

Acknowledgements

References

[1] Ann O.C., Theng L.B.: Human activity recognition: A review. In: *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, pp. 389–393, 2014. http://dx.doi.org/10.1109/ICCSCE.2014.7072750.

[2] Gurrin C., Smeaton A.F., Doherty A.R.: LifeLogging: Personal Big Data, *Foundations and Trends in Information Retrieval*, vol. 8(1), pp. 1–125, 2014. http://dx.doi.org/10.1561/1500000033.

[3] Jalal A., Kamal S., Kim D.: A Depth Video Sensor-Based Life-Logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments, *Sensors*, vol. 14(7), pp. 11735–11759, 2014. http://dx.doi.org/10.3390/s140711735.

[4] van Kasteren T.L.M., Englebienne G., Kröse B.J.A.: An activity monitoring system for elderly care using generative and discriminative models, *Personal and Ubiquitous Computing*, vol. 14(6), pp. 489–498, 2010. `http://dx.doi.org/10.1007/s00779-009-0277-9`.

[5] Khan Z.A., Sohn W.: Abnormal human activity recognition system based on R-transform and kernel discriminant technique for elderly home care, *IEEE Transactions on Consumer Electronics*, vol. 57(4), pp. 1843–1850, 2011. `http://dx.doi.org/10.1109/TCE.2011.6131162`.

[6] Mann S.: Wearable Wireless Webcam. `http://wearcam.org/netcam.html`. Accessed on 11th of December 2017.

[7] Miaou S.G., Sung P.H., Huang C.Y.: A Customized Human Fall Detection System Using Omni-Camera Images and Personal Information. In: *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, 2006. D2H2.*, pp. 39–42, 2006. `http://dx.doi.org/10.1109/DDHH.2006.1624792`.

[8] Nana P., Min D., Yue Z., Xin C., Sheng B.: *The Elderly's Falling Motion Recognition Based on Kinect and Wearable Sensors*, pp. 1129–1141. Springer International Publishing, Cham, 2017. `http://dx.doi.org/10.1007/978-3-319-48036-7_83`.

[9] O'Hara K., Tuffield M.M., Shadbolt N.: Lifelogging: Privacy and empowerment with memories for life, *Identity in the Information Society*, vol. 1(1), pp. 155–172, 2008. `http://dx.doi.org/10.1007/s12394-009-0008-4`.

[10] Pal M., Saha S., Konar A.: Distance matching based gesture recognition for healthcare using Microsoft's Kinect sensor. In: *2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*, pp. 1–6. 2016. `http://dx.doi.org/10.1109/MicroCom.2016.7522586`.

[11] Popoola O.P., Wang K.: Video-Based Abnormal Human Behavior Recognition – A Review, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42(6), pp. 865–878, 2012. `http://dx.doi.org/10.1109/TSMCC.2011.2178594`.

[12] Poritz A.B.: Hidden Markov models: a guided tour. In: *International Conference on Acoustics, Speech, and Signal Processing ICASSP-88*, pp. 7–13, 1988. `http://dx.doi.org/10.1109/ICASSP.1988.196495`.

[13] Postawka A.: *Exercise Recognition Using Averaged Hidden Markov Models*, pp. 137–147, Springer International Publishing, Cham, 2017. `http://dx.doi.org/10.1007/978-3-319-59060-8_14`.

[14] Postawka A.: Real-time monitoring system for potentially dangerous activities detection. In: *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 1005–1008. 2017. `http://dx.doi.org/10.1109/MMAR.2017.8046967`.

[15] Postawka A., Śliwiński P.: *A Kinect-Based Support System for Children with Autism Spectrum Disorder*, pp. 189–199. Springer International Publishing, Cham, 2016. `http://dx.doi.org/10.1007/978-3-319-39384-1_17`.

[16] Rabiner L.R., Juang B.H.: An introduction to hidden Markov models, *IEEE ASSP Magazine*, vol. 3, pp. 4–16, 1986. `http://dx.doi.org/10.1109/MASSP.1986.1165342`.

[17] Rowe M., Lane S., Phipps C.: CareWatch: A Home Monitoring System for Use in Homes of Persons With Cognitive Impairment, *Topics in Geriatric Rehabilitation*, vol. 23(1), pp. 3–8, 2007.

[18] Siddiqui S.A., Snober Y., Raza S., Khan F.M., Syed T.Q.: Arm gesture recognition on Microsoft Kinect using a Hidden Markov Model-based representations of poses. In: *2015 International Conference on Information and Communication Technologies (ICICT)*, pp. 1–6, 2015. `http://dx.doi.org/10.1109/ICICT.2015.7469478`.

[19] Wu P., Peng H.K., Zhu J., Zhang Y.: *SensCare: Semi-automatic Activity Summarization System for Elderly Care*, pp. 1–19, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. `http://dx.doi.org/10.1007/978-3-642-32320-1_1`.

[20] Yin J., Yang Q., Pan J.J.: Sensor-Based Abnormal Human-Activity Detection, *IEEE Transactions on Knowledge and Data Engineering*, vol. 20(8), pp. 1082–1090, 2008. `http://dx.doi.org/10.1109/TKDE.2007.1042`.

[21] Yoshihara Y., Tang D., Kubota N.: Life Log Visualization System Based on Informationally Structured Space for Supporting Elderly People. In: *2013 Second International Conference on Robot, Vision and Signal Processing*, pp. 78–83, 2013. `http://dx.doi.org/10.1109/RVSP.2013.25`.

[22] Yu M., Rhuma A., Naqvi S.M., Wang L., Chambers J.: A Posture Recognition-Based Fall Detection System for Monitoring an Elderly Person in a Smart Home Environment, *IEEE Transactions on Information Technology in Biomedicine*, vol. 16(6), pp. 1274–1286, 2012. `http://dx.doi.org/10.1109/TITB.2012.2214786`.

[23] Yu X., Wu L., Liu Q., Zhou H.: Children tantrum behaviour analysis based on Kinect sensor. In: *2011 Third Chinese Conference on Intelligent Visual Surveillance*, pp. 49–52. 2011. `http://dx.doi.org/10.1109/IVSurv.2011.6157022`.

Affiliations

Aleksandra Postawka
Wroclaw University of Science and Technology, Department of Computer Engineering, Wybrzeże Wyspiańskiego 27, 50-370 Wroclaw, Poland, aleksandra.postawka@pwr.edu.pl, ORCID ID: https://orcid.org/0000-0001-8439-6951

Jarosław Rudy
Wroclaw University of Science and Technology, Department of Computer Engineering, Wybrzeże Wyspiańskiego 27, 50-370 Wroclaw, Poland, jaroslaw.rudy@pwr.edu.pl, ORCID ID: https://orcid.org/0000-0003-1095-6041