Real-Time Monitoring System for Potentially Dangerous Activities Detection

Aleksandra Postawka  
Faculty of Electronics, Wroclaw University of Science and Technology, Poland  
Email: aleksandra.postawka@pwr.edu.pl

Abstract—Cognitive impairments are an unavoidable community problem. People suffering from such diseases need all day long attention with varying care difficulty depending on the type of disorder. What makes care harder in the case of autism is the frequent occurrence of self aggressive behaviors. The monitoring system is supposed to detect such situations and differentiate them from similar normal activities.

In this paper the Averaged Hidden Markov Models are used for potentially dangerous activities detection in the real-time monitoring system. The acceleration measure has been used in order to discriminate dangerous and normal situations. Additionally, algorithms for real-time activity recognition have been presented. The experiments have been conducted for a set of data containing hitting and touching sequences obtained from the depth sensor.

I. INTRODUCTION

Surveillance systems for dangerous situations detection are important in manifold fields of life and they differ depending on the intended purposes. Public places monitoring systems usually focus on detection of suspicious (abnormal) behavior [1] or aggressive behavior detection based on monitoring videos, e.g. Cassandra [2]. Healthcare dedicated applications often depict the problem of fall detection [3], [4] as that falls always indicate life threat. The monitoring of performed actions is also used for detecting and diagnosing serial diseases [5] and in order to save the history of movements in the life logging system [6].

Such healthcare systems are useful in the case of elderly persons who need some kind of continuous surveillance. However, people with cognitive impairments, e.g. Autism Spectrum Disorder (ASD), Alzheimer’s disease, need attention all the time. Therefore in such cases the assistance system is supposed to support the caregiver instead of replacing them. Several attempts have already been made in that direction. Rowe et al. created a sensor-based system for people with cognitive impairments [7] which included i.a. bed occupancy sensor, door opening sensors and alerts for caregiver. The survey of AI methods used for elders with cognitive impairments is presented in [8].

In the case of low functioning autism the caregivers have to face even more complicated problems as that the people who suffer from this disease additionally have the tendency toward self aggressive and self injurious behavior [9]. These potentially dangerous actions are individual for each ASD. In many cases the speed of motions is the only difference between potentially dangerous actions and similar normal activities, e.g. hitting and touching. The monitoring system should differentiate such situations and alert the caregiver if necessary. The problem has been depicted in this paper.

The research is a continuation of author’s work on action recognition with the use of Hidden Markov Models (HMMs). The whole system is supposed to recognize different actions including dangerous situations. The latter should cause an alert for a caregiver. The research on activity recognition with the use of Averaged HMMs, which use multiple learning sequences, has been done in author’s latest work [10]. This paper concerns only the problem of potentially dangerous actions real-time recognition as a part of more complex system. For the discrimination of normal and potentially dangerous activities the acceleration measure has been used.

As mentioned, potentially dangerous actions are different for each person with low functioning autism and have to be modeled individually. Moreover, it is common that the trajectory of motion differs depending on the person performing the activity. Therefore, the models describing one’s behavior have to be created based on individual long term observation. While designing such models for a specified autistic person the self aggressive activities are well-known by the caregiver.

The notation used in this paper can be found in Section II. The idea of the monitoring system and methods used for action recognition are presented in Section III. The results of conducted experiments are included in Section V. The conclusions and future work directions are presented in Section VI.

II. NOTATION

Averaged HMM (AHMM) is a HMM model obtained from multiple left-to-right HMMs [10] which had been generated based on different learning sequences describing the same action. The structure of these component models and algorithms for AHMM creation have been described in [10]. Since the resultant AHMM model is a HMM model, the common HMM notation will be used in this paper:

\[
\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}
\]

\(\lambda\) - the complete parameter set of the AHMM model,  
\(O = \{O_0, O_1, \ldots, O_{T-1}\}\) - the sequence of observation symbols,  
\(O_{j:k}\) - the part of observation sequence including symbols from \(j\)-th to \(k\)-th, inclusive,  
\(T\) - the length of observation sequence \(O\),  
\(P(O|\lambda)\) - the probability that the observation sequence \(O\) is generated by model \(\lambda\),  
\(I_\hat{O} = \{i_j, \ldots, i_k\}\) - the sequence of decoded states using the Viterbi algorithm [11] for the observation sequence \(O\).

III. MONITORING SYSTEM

The monitoring system is composed of a mobile workstation (computer) and a Microsoft Kinect 2.0 sensor which is
connected to the device. The application running on the computer consists of several modules (Fig. 1): feature extraction, data collection, learning, test, real-time recognition and can be launched in different modes. Each of the modules is described in dedicated subsection.

A. Feature extraction

Feature extraction module analyzes data obtained from the depth sensor. A basic data unit is a frame collected at a given time. The frame consists of 25 3D points obtained from Microsoft Skeleton Tracking which describe the position of significant skeleton joints. The skeleton data are normalized and the feature vector has been calculated according to methods developed in [12]. In effect all the chosen features are independent of the sensor tilt angle and the distance from the depth camera. The output of this module is an HMM observation symbol calculated due to formula included in [12]. This symbol (label) describes both hands position. Based on the number of all the possible observations it is easy to calculate the observation symbols for the right and left hand.

B. Data collection

The data collection module consists of two stages: the long term observation file creation and sequences selection. In the first stage the application is run in the recording mode. At each time unit the feature extraction module returns the HMM observation symbol for the current frame. The frame and the label are saved together in a singular file row. The multiple repetitions of each of modeled actions are recorded. In the second stage the recorded sequences of movements for activities are manually cut out from the long term observation file and placed in proper file directories. Choosing the proper activities are manually cut out from the long term observation file creation and sequences selection.

In the second stage the recorded sequences of movements for activities are manually cut out from the long term observation file creation and sequences selection. For each manually chosen sequence the beginning and the end of the sequence is trimmed so as to not repeat the same symbols on the edges of sequences. Therefore the sequences are not prolonged by the idle fragments.

The output of this module are multiple sorted sequences for all considered activities. The sequences are already divided into the training and test sets.

C. Learning

The input for the learning module is the training set of sequences for each activity to be modeled. For each activity the AHMM model is created based on multiple learning sequences (component models) [10]. Beyond the usual HMM model parameters also the real number of states $N_R$ value is calculated. $N_R$ is the id of the last state that should always be accessed in order to recognize the sequence as generated by this model. The id is set based on all the component models. The reason for this additional parameter is that for each first $k < T$ symbols in the recognized activity’s sequence $O_{0:T-1}$ the condition $P(O_{0:k-1}|λ) > 0$ is fulfilled. Sequences that are too short are being rejected based on the knowledge of the minimum id in decoded hidden state sequence. The output models are saved in files in proper directory - for the left, right or both hands.

D. Test

The test module uses the set of AHMM models obtained from the learning module and the training set with sequences generated in data collection module. This module is used only for experiments and developed methods validation. The performed tests have been described in Section V.

E. Real-time recognition

The input of the real-time recognition module is a set of AHMM models and a HMM symbol obtained from the current frame. The activity models are read from the files generated in data collection module. This module is used in dedicated subsection.

The Same As Before - the same activity is recognized as before:

\[
\begin{align*}
c_1 &= P(O_{k:t-1}|λ_j) = \max \{P(O_{k:t-1}|λ_j), λ_j ∈ H \} > 0 \\
c_2 &= I_{O_{k-1}}^∗[t-2] ≥ N_R \\
c_3 &= P(O_{k:t-1}|λ_j) = \max \{P(O_{k:t-1}|λ_j), λ_j ∈ H \} > 0 \\
c_4 &= I_{O_{k-1}}^∗[t-2] ≥ N_R \\
\end{align*}
\]

\[
\begin{align*}
c_1 &\land c_2 \land (c_3 \lor c_4) \\
\end{align*}
\]

New Recognized - the activity is recognized for the first time:

\[
\begin{align*}
c_1 &= 3λ_j ∈ λ_H P(O_{k:t-1}|λ_j) > 0 \\
c_2 &= I_{O_{k-1}}^∗[t-1] ≥ N_R \\
c_3 &= P(O_{k:t-1}|λ_j) ≠ \max \{P(O_{k:t-1}|λ_j), λ_j ∈ H \} > 0 \\
c_4 &= I_{O_{k-1}}^∗[t-2] < N_R \\
\end{align*}
\]

\[
\begin{align*}
c_1 &\land c_2 \land (c_3 \lor c_4) \\
\end{align*}
\]
Too Short - there is at least one model that recognizes the partial sequence, however the sequence is too short:
\[ \forall_{\lambda \in \Lambda_H} \left( P(O | \lambda_t) > 0 \Rightarrow I^*_H[T-1] < N_R \right) \]

Not Recognized - no one activity is recognized:
\[ \forall_{\lambda \in \Lambda_H} \left( P(O | \lambda_t) < 0 \vee I^*_H[T-1] < N_R \right) \]

The state machine for possible transitions in the recognizer has been presented in Fig. 2.

For each new observation symbol \( O_t \neq O_{t-1} \) a new partial sequence of observations is created and the symbol \( O_t \) is nearly always added to previously created sequences (excluding one element sequences). Each sequence is kept until its state is changed into Not Recognized. For each sequence in the list and for each model \( \lambda \in \Lambda_H \) the posterior probability \( P(O | \lambda) \) calculated (excluding the situation when \( \lambda = O_{t-1} \)). Depending on the number \( n \) of models for which \( P(O | \lambda_t) > 0 \) different decisions are taken. If \( n = 0 \) then the sequence’s state is set into Not Recognized and it is deleted. If \( n = 1 \) then the hidden state sequence is decoded using the Viterbi algorithm and the last decoded state id is checked. The number greater or equal to \( N_R \) indicates the new recognized activity, otherwise the sequence is Too short yet (it will be prolonged in the future). If \( n > 1 \) then the model with greatest posterior probability is chosen and the same procedure is applied as in previous case.

When all the temporary sequences are evaluated then the models labeled as New Recognized are searched for the one with longest observation sequence. The activities may be comprised of shorter activities and the goal is to find the longer one. Therefore, the recognized sequences are not deleted but left to be prolonged in the next iterations. If the new recognized sequence is labeled as potentially dangerous then the sequence is no more prolonged and the information about the danger is the priority.

IV. POTENTIALLY DANGEROUS ACTIONS DETECTION

The speed of motions is usually the indicator of potential dangerousness of activities for autistic children. If the activity is different from other actions then the modeling is quite easy. The problem arises when the same sequence is recognized both by the model of a normal activity and the model of activity which is supposed to be dangerous and the threshold cannot be designated by the posterior probability. The experiments revealed that the posterior probability could improve the discrimination between normal and potentially dangerous activities, however some errors occurred even for the training set. The number of observations in the recognized sequence turned out to be a much better feature. The obtained results have been presented in Section V-A.

The chosen solution is reasonable only in the case when the sequences are selected according to the method described in Section III-B. If the symbols on the edges of sequences are multiplicated then a different method should be chosen. Our real-time recognition system is designed in order to create such a trimmed partial sequences.

V. EXPERIMENTS

The experiments have been conducted for the data set of 450 activity sequences (75 sequences for each of 6 activities). The data have been divided into a training set (50 sequences for each action) and a test set (25 sequences). In the Experiment 1 the threshold values for normal and potentially dangerous actions have been calculated. The Experiment 2 compares the action recognition with and without the usage of the thresholds from Experiment 1.

A. Experiment 1

In this experiment the number of observations has been examined as a feature discriminating normal and potentially dangerous actions. The histograms have been calculated for a pair of hitting and touching actions for the left-, right- and both-handed activities (Fig. 3). Due to the results the number of observations separates the classes completely. In each case the threshold for the fast and low motion movements discrimination has been designated based on density function estimation for each of the histograms. The threshold has been calculated in Matlab as a curves intersection.

B. Experiment 2

The aim of this experiment was to examine the trained models’ effectiveness for the test set and to compare the results obtained with and without the use of additional threshold (the number of observations). The each of 6 models has been examined by the each of 150 test sequences (25 for an activity). The confusion matrices have been presented in Table I (without the threshold) and in Table II (with the threshold). The models for the left, right and both hands have been denoted as \( L, R \) and \( B \), respectively. Similarly, the models for hitting and touching have been labeled with letters \( H \) and \( T \), respectively.

In Table I there are many false positives values as that the hitting model recognizes a few sequences from the touching
model for the same hand type and inversely. It is quite obvious that the one-handed models recognize also a few sequences from the both-handed models. In Table II the results are much more desirable and (for each cell) not worse than in the first case. There is only one false positive as the model for hitting the head by the right hand has recognized the sequence supposed to be recognized by the model for touching the head by both hands. However, it is always better to recognize the touching sequence as a hitting sequence than conversely (we want to detect each self aggressive behavior). The results confirm that the recognition method based on the number of observations as a threshold is well chosen for this kind of problem.

The one-handed activities are recognized with the rate of at least 76% (up to 100%). The recognition rate for the both-handed actions is much lower (at least 32%). The reason is that in both-handed models the number of possible HMM symbols is squared and there are additionally the dependencies between hands movements. It can be noticed that much better effects could be gained by using the single hand models for left and right hands and comparing the time differences between the obtained activities labels.

The models for fast motion movements recognize more sequences than those for slow motion and similar movement trajectory. The greatest difference occurs for both hands movements where hitting the head is recognized with the rate of 32% and touching the head is recognized in 52% of cases. One-handed potentially dangerous activities are detected with the recognition rate of at least 84%.

VI. Conclusion

The methods used for normal and potentially dangerous action discrimination significantly improved the action recognition. The fast motion movements were detected in 52% of the test sequences for both-handed activity and in at least 84% cases of one-handed actions. The models for the potentially dangerous actions have a greater recognition rate than the models for similar normal activities. However, the self aggressive behaviors have a higher detection priority than the calm ones.

The future work will be targeted mainly at autistic children behavior modeling. Also the alerts for the potentially dangerous actions will be created. The models will be used also for emotion recognition.

The recognizer modules for left-, right- and both-handed activities need to be parallelized because of the algorithms complexity.

ACKNOWLEDGEMENT

This work was supported by the statutory funds of the Faculty of Electronics 0401/0218/16, Wroclaw University of Science and Technology, Wroclaw, Poland.

REFERENCES