E-Journal of Advanced Maintenance Vol.9-2 (2017) 66-71
Japan Society of Maintenology

Insider Malicious Behaviors Detection and Prediction Technology for Nuclear Security

Shi CHEN [1,*], Kazuyuki DEMACHI [1], Tomoyuki FUJITA [1], Yutaro NAKASHIMA [1], and Yusuke KAWASAKI [1]

[1] *The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan*

ABSTRACT

After Fukushima Daiichi nuclear power plant accident, the importance of nuclear security is increased, especially as a threat to nuclear power plants, sabotage by insider is significant. In response to the increasing threats to Nuclear Power Plant, human malicious behavior detection is necessary for nuclear security. Hand motion is an important part of human activity and has a high contribution for high-accuracy detection of insider malicious behaviors. Hand motions can be distinguished by the position of each fingertip, both stretched and bend fingers of both left and right hands can be classified as different parts by using depth data and body index frame of Microsoft Kinect v2. Fingers were identified by using K-means clustering algorithm. Finally, it was built a hand motion time series data by using the developed real-time hand motion detection system. However, as malicious behaviors detection isn't enough for nuclear security, future malicious behaviors prediction should be taken into consideration.
In this research, the real-time hand motion detection system was developed ~~by~~ using Kinect v2. In addition, we explored the possibility of malicious behavior detection and prediction by using Stacked Auto-Encoder.

KEYWORDS

Malicious Behavior Detection, Hand Motion Tracking, Kinect, Deep Neural Network, Stacked Auto-Encoder

ARTICLE INFORMATION

Article history:
Received 7 November 2016
Accepted 27 April 2017

1. Introduction

The importance of nuclear security ~~is~~ increased after Fukushima Daiichi nuclear power plant accident. Especially as a threat to nuclear power plants, sabotage by insider is worthy of attention. Considering current countermeasures to insider sabotage, the most challenging is to distinguish the ordinary maintenance behavior and the sabotage behavior since some sabotage behaviors may be hidden in ordinary maintenance behavior. In this case, human malicious behavior detection is necessary for nuclear security. Moreover, hand motion has a high contribution for human activity and a significant portion of maintenance behaviors and malicious behaviors are realized through hand motion. Thus, hand motion analysis should be taken into consideration for nuclear security.

Microsoft Kinect v2 is proposed to be used as the monitoring video camera in detection of insider malicious behavior. Kinect v2 is the new version of game controller technology introduced by Microsoft. It is composed of two cameras, namely a RGB and an infrared (IR) camera (as shown in Fig.1), The RGB camera captures color information with a resolution of 1920x1080 pixels, while the IR camera is used for the real-time acquisition of depth frame with a 512x424 pixels resolution. Also, Kinect v2 provide a new function called body index frame which can be used to segment human body from background.

Time series data analysis is a useful method in abnormal behavior detection. Thus, we captured hand motion time series and distinguish normal and abnormal behavior. However, abnormal behavior detection isn't enough for nuclear security, prediction technology of future malicious behavior should also be developed and implemented.

*Shi CHEN, E-mail: chenshigo@yahoo.com

ISSN-1883-9894/10

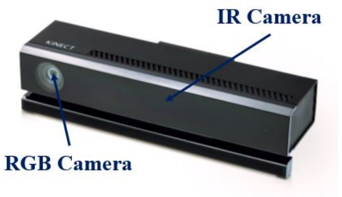

Fig.1 Microsoft Kinect v2

In this research, we tracked hand motion by using depth data and body index frame of Kinect v2. ~~Moreover,~~ Stacked Auto-Encoder was used to extract features of hand motion time series data and to recognize malicious behaviors.

2. Hand Motion Capturing

Hand motions are detected by capturing the fingertips positions. In this research, stretched fingers and bend fingers of both left and right hands would be classified into different parts. ~~and~~ Not only the fingertips of stretched fingers but also bend fingers can be detected (at most 6 persons).

The hand motion detection system in this research consists of three main parts: hand region classification, fingers segmentation and fingers identification. We developed our system using Visual Studio 2015 with C# as programming language while body index data, depth data and skeleton data of Kinect v2 were used to calculate fingertip positions.

2.1. Hand Region Classification

For hand region classification, body regions should be segmented from the background at first. For this purpose, Kinect body index frame is used. Body index frame is the new function provided by Kinect v2, which has 512×424 resolution frame and at most six bodies can be detected. For each pixel in body index frame, Kinect system sets an index:

1) If the pixel belongs to the region of human body, the index is 0~5 (index of each body is different);
2) If the pixel belongs to the background, the index is 255.

Thus, human body can be segmented from background by using body index frame.

After body segmentation, ~~we have to do~~ hand region classification is performed. For each hand, stretched hand region and bend hand region should be divided. As shown in Fig.2, the distance from each hand region to Kinect camera are different, in this case, depth data can be used for hand region classification. Let D(n) be the set of pixels' depths in Kinect depth frame, I(n) be the set of pixels' body index in Kinect body index frame, d_{max} be the wrist point which is in the farther distance to the Kinect Camera, d_{min} be the wrist point which is in the closer distance to the Kinect Camera and d is the threshold to calculate bend hand pixels. For each pixel's depth D(i) in D(n):

1) If $D(i) \le d_{max}$ and $D(i) > d_{max} - d$, meanwhile, $I(i) \neq 255$, then I(i) can be classified as the farther distance stretched hand pixels;
2) If $D(i) \le d_{max} - d$ and $D(i) > d_{min}$, meanwhile, $I(i) \neq 255$, then I(i) can be classified as the farther distance bend hand pixels;
3) If $D(i) \le d_{min}$ and $D(i) > d_{min} - d$, meanwhile, $I(i) \neq 255$, then I(i) can be classified as the closer distance stretched hand pixels;
4) If $D(i) \le d_{min} - d$, and $D(i) \neq 0$, meanwhile, $I(i) \neq 255$, then I(i) can be classified as the closer distance bend hand pixels;

Finally, pixels can be divided into four parts:

(i) Left stretched hand pixels;
(ii) Right stretched hand pixels;
(iii) Left bend hand pixels;
(iv) Right bend hand pixels.

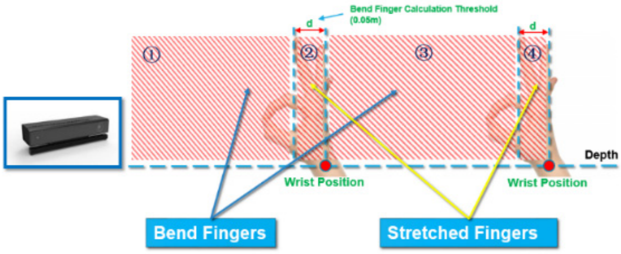

Fig. 2 Hand Region Classification by Depth

To reduce computing complexity, hand pixels should be limited to some specific region. For this purpose, we proposed a simple algorithm called "rectangle limitation algorithm" to captured these specific regions. The algorithm can be summarized as follow:

1) Get center points of both left and right hand;
2) Build four rectangles from center point with width W and height H for each hand; (The values of W and H depends on the distance from body to the Kinect camera)
3) Only the pixels within this region can be classified as hand pixels; (As shown in Fig.3)

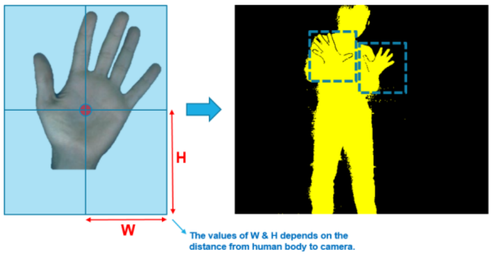

Fig.3 Rectangle limitation algorithm used for reducing computing complexity

The results of hand region classification are shown in Fig.4.

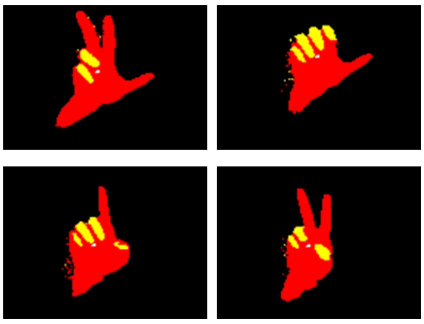

Fig.4 Results of hand region classification (red pixels: stretched hand region; yellow pixels: bend hand region; black pixels: background)

2.2. Fingers Segmentation

The objective of fingers segmentation is to segment stretched fingers from the palm. Firstly, one circle is draw whose center is hand, and the pixels within this area will be considered as palm (Fig.5-A). Some noise may also exist as shown in Fig.5-B, but these can be filtered by analyzing body index data and depth data of Kinect. Thus, stretched fingers can be segmented as demonstrated in Fig.5-C.

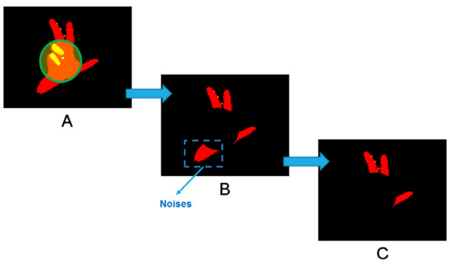

Fig. 5 Algorithm of Fingers Segmentation

The results of fingers segmentation are shown in Fig.6.

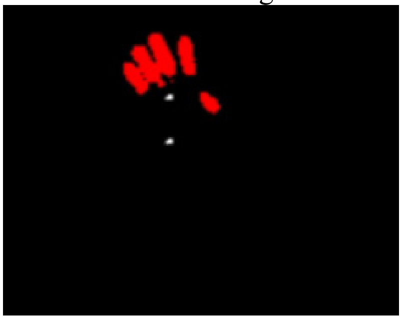

Fig.6 Results of Fingers Segmentation

2.3. Fingers Identification

To identify fingers, at first each finger (both stretched fingers and bend fingers) should be classified into different clusters. For this purpose, K-means clustering algorithm was used in this research; the implementing steps can be summarized as follows:

1) K initial "means" (in this case K=5) are randomly generated within the hand region pixels;
2) K clusters are created by associating every observation with the nearest mean;
3) The centroid of each of the K clusters becomes the new mean;
4) Steps 2 and 3 are repeated until convergence has been reached.

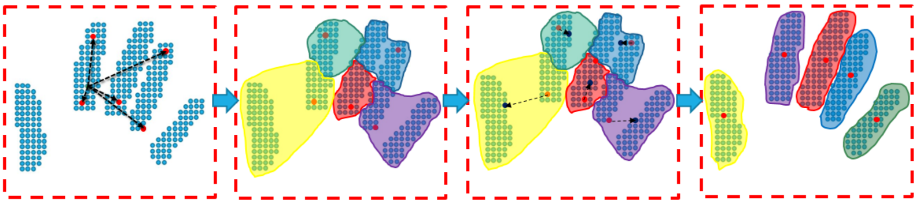

Fig.7 Steps of K-means clustering algorithm

However, one significant disadvantage of K-means clustering algorithm is that different initial means can result in different final clusters. Inappropriate initial means may lead to inappropriate results. To improve K-means clustering algorithm, the most important thing is to find an appropriate initial means for calculation rather than using random means. In this case, centroids of previous frame will be used as initial means of current frame, and new centroids can be calculated. These results will be more accurate after several frames because of accumulation effect.

By analyzing positions of the centroids of each cluster, fingers can be classified. For each cluster of finger pixels, the fingertip is the pixel which has closest distance to camera. We have already developed a real-time hand motion detection system and positions of each fingertip can be captured with the frame rate of 22fps. The result of hand motion detection can be seen in Fig.8.

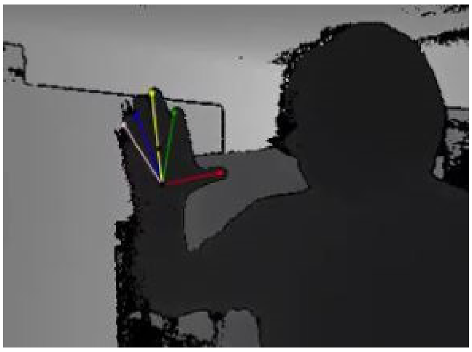

Fig. 8 Results of Hand Motion Detection

3. Behavior Recognition

To distinguish malicious behavior and ordinary maintenance behavior, different malicious motion should be classified into different patterns. For this purpose, Deep Neural Network is considered as a useful method. In this research, Stacked Auto-Encoder was implemented. As shown in Fig.9, Stacked Auto-Encoder is a neural network consisting of multiple layers of sparse auto-encoders in which the outputs of each layer are wired to the inputs of the successive layer. For each auto-encoder, when the number of neurons in the hidden layer is less than the size of the input, the auto-encoder learns a compressed representation of the input. Finally, the probability of each pattern can be output by a SoftMax classifier.

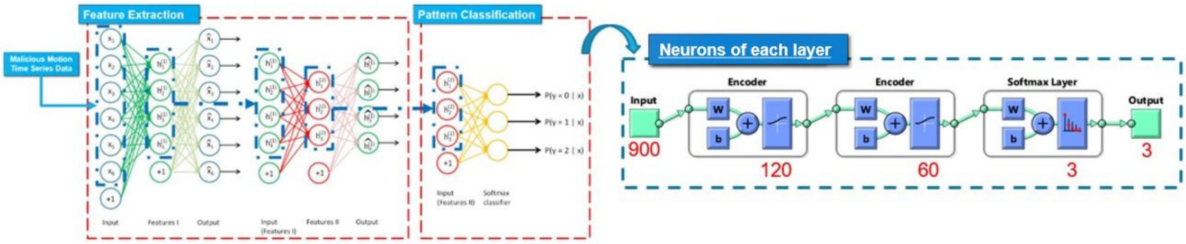

Fig. 9 Structure of Stacked Auto-Encoder

The structure of Stacked Auto-Encoder in this research can be demonstrated in Fig.9. First, a sparse auto-encoder was trained on the raw inputs (time series data of all malicious motions, 900 neurons in each sample) to learn primary features (120 neurons) on the raw input. Next, these primary features will be feed into the second sparse auto-encoder to obtain the secondary feature activations (60 neurons) for each of the primary features. Finally, these secondary features will be treated as "raw input" to a SoftMax classifier, and it will be trained to map secondary features to digit labels.

4. Experiment and Result

Recently, we assumed some malicious motions for behavior detection as shown in Fig.10:

1) Cutting motion (by using scissor, etc.);
2) Patting motion (control panel, etc.);
3) Turning motion (switch, etc.).

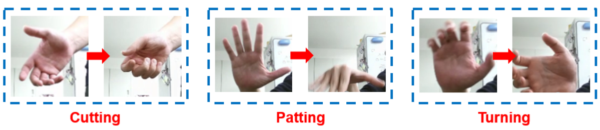

Fig. 10 Assumed Malicious Motions

For practical purposes, the following factors of neural network trainset should be taken into consideration:

1) Varying characteristics of individual hands;
2) Relative distance and angle to camera;
3) Motion speed.

In current experiment, we considered first two factors of trainset and captured three malicious motions with 5 persons in different distance and angle to camera. The factor of motion speed will be considered in future experiments. Conversion from captured fingertip positions to neural network trainset can be demonstrated in Fig.11. Each training sample of the trainset includes fingertip positions from 60 successive frames with 15 variables (3D positions of five fingers) for each frame.

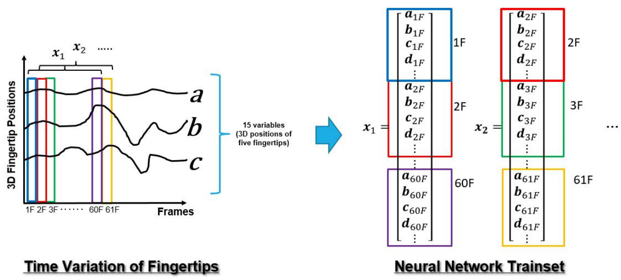

Fig. 11 Conversion from Captured Fingertip Positions to Neural Network Trainset

By training Stacked Auto-Encoder using this trainset, these three malicious motions can be classified into different pattern. Then this trained Stacked Auto-Encoder can be used to detect malicious motions. The testing time series data we captured are contained with different motion (both malicious motions and ordinary normal motions) in order (Cutting-Normal-Patting-Normal Turning). The results of behavior detection can be seen in Fig.12. This figure shows the probability of different motion for each frame.

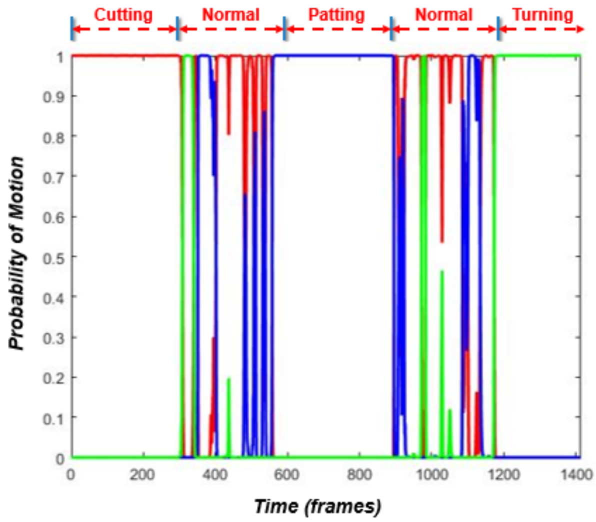

Fig. 12 Results of Behavior Detection (Red line: probability of cutting motion; Blue line: probability of patting motion; Green line: probability of turning motion)

Thus, all of these three motions can be detected from the testing time series data and distinguished from ordinary normal motions in this experiment. Detection accuracy is shown in Table 1:

Table 1 Detection Accuracy

Motions	Cutting	Patting	Turning	Normal
Accuracy (%)	99.33	98	71.33	61.17

5. Conclusion and Future Work

In this research, we proposed a hand motion detection algorithm for insiders' malicious behaviors detection and both stretched fingers and bend fingers can be detected and identified by using this algorithm. Meanwhile, a real-time fingertip tracking system was developed and time series data of each fingertip was successfully obtained with 22fps. Moreover, by using Stacked Auto-Encoder, assumed malicious motions can be classified into different patterns and detected.

For future work, hand motion detection technology will be improved and stacked Auto-Encoder will be implemented for the detection of sign of hand motion of malicious insiders' sabotage. Furthermore, the prediction of detected features for earlier response will be taken into consideration.

References

[1] Y. Li, "Hand gesture recognition using Kinect", Proc. Int. Conf. Software Eng. Service Sci., 2012, pp. 196–199.

[2] Ray S, Turi RH, "Determination of number of clusters in K-means clustering and application in colour image segmentation", Proceedings of the 4th international conference on advances in pattern recognition and digital techniques (ICAPRDT'99), Calcutta, India, pp 137–143.

[3] Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of Machine Learning Research 11.Dec (2010): 3371-3408.

[4] Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18.7 (2006): 1527-1554.