

# Monitoring Abnormal Behavior of Hospital Patients Using RGB+D Sensors

Seung Chae and Kin Choong Yow

GIST College, Gwangju Institute of Science and Technology  
Gwangju, South Korea  
e-mail: chaeseung, kcyow@gist.ac.kr

**Abstract**— The ability to recognize abnormal actions or conditions of hospital patients is a very important problem as it may bring about timely medical response to a critical patient condition and may make the difference between life and death. In this paper, we propose a system that makes use of the Microsoft Kinect for Windows v2 to generate RGB+D (Red, Green, and Blue + Depth) image sequences of hospital patients. Dense 2D image features were extracted from the image sequences and then combined in a hierarchical manner to form compound features. These compound features were then mined to produce a class feature model to be used for action recognition. In the recognition phase, the RGB and the depth image data were processed separately and the responses merged to produce an overall response for action classification. Our experimental results show that this approach is able to generate good recognition rates and is comparable to other state of the art algorithms.

**Keywords**-Abnormal behavior detection; learning; Kinect; data mining; Ambient Assisted Living(AAL).

## I. INTRODUCTION

As many countries are rapidly becoming aging societies, significant challenges arise for the states' finances (to build more hospitals, nursing homes, etc.), as well as on the healthcare system (timely response by medical staff and post-operative care). To address these challenges, research in Ambient Assisted Living (AAL) explore modern AI (Artificial Intelligence) methods and techniques to create intelligent living environments or mobility assistants, to enable elderly or impaired people to live independently or to call for help only in an emergency.

One of the most important application environment of AAL is in the hospital. A patient is usually admitted to a hospital if he needs constant and intensive monitoring, or if he becomes too weak to move independently after a surgery. In these scenarios, a lapse in timely attention and/or response may cause the difference between life and death.

Due to the ever-increasing demand of hospital beds by the ever-increasing population of people who needs medical attention, the medical staff at a hospital always find themselves short-handed. To cope with the shortage of staff, a hospital may reduce its frequency of the medical ward rounds, leading to delays in detecting critical patient conditions. Patients are sometimes expected to call for help using the nurse call button in a non-emergency (e.g., if they need a drink of water), leading to the nurses' frustration, alarm fatigue, delays or even completely disregarding the patient's calls in real emergencies.

Due to the above reasons, AAL technologies can be very beneficial in a hospital environment. They can monitor

patients in a non-intrusive manner, therefore relieving the hospital staff to perform other important duties. It can also provide non-stop, 24/7 monitoring of the patient and can even detect the case where the patient is not able to call for help nor press the nurse call button, e.g., due to temporary paralysis or laboring breath.

Recent progress in using RGB + depth sensors to detect human's action makes it an efficient and non-intrusive technology for action monitoring. In this paper, we propose a system that makes use of the Microsoft Kinect for windows v2 to detect several critical actions/conditions of hospital patients, e.g., labored breathing, failed attempts to press the nurse call button, fall off the bed, etc. When such actions/conditions are detected, an alert can be sent immediately to the medical staff on duty, greatly increasing the chance of early medical treatment and hence the survival of the patient.

The rest of the paper is organized as follows: Section II discusses some related work, and Section III describes the proposed algorithm. Section IV discusses the implementation details and Section V provides the experimental results. Section VI concludes the paper.

## II. RELATED WORKS

Banerjee et al. [1] presented an approach for patient activity recognition in hospital rooms using depth data collected using a Kinect sensor. Their work detects the presence of a patient in the bed as a means to reduce false alarms from an existing fall detection algorithm. They, however, do not attempt to recognize the different actions of the patients in the hospital bed.

Saha et al. [2] studied the problem of emotion recognition from gestures using the Kinect sensor. Using the co-ordinates of joints from the upper body and the hands, a set of nine features were extracted. Using these features, they were able to uniquely identify gestures corresponding to five basic human emotional states, namely, 'Anger', 'Fear', 'Happiness', 'Sadness' and 'Relaxation'. However, in this work, the subjects were looking directly into the camera, hence it will be difficult to apply this technique to a hospital scenario.

Other techniques that address action recognition make use of depth information only or RGB information only. Ijjina et al. [3] propose an approach for facial expression recognition using deep convolution neural networks (CNN) based on features generated from depth information only. The ability of a CNN to learn local discriminative patterns from data is used to recognize facial expressions from the representation of unregistered facial images.

Gilbert et al. [4] proposed a technique that uses an overcomplete set of simple 2D corners in both space and

time in the RGB data only. These are grouped spatially and temporally using a hierarchical process, with an increasing search area. At each stage of the hierarchy, the most distinctive and descriptive features are learned efficiently through data mining. This method results in fast, accurate recognition with real-time performance on high-resolution video that outperforms all other methods reported thus far in the literature.

### III. PROPOSED APPROACH

#### A. Approach Overview

Our approach is based on the method proposed by Gilbert et al. [4] using mined hierarchical compound features. As we have additional depth data in addition to the RGB data captured by the Kinect sensor, we repeat the same process used for the RGB data on the depth data and generate two sets of association rules, one for the RGB data and one for the depth data. In the recognition process, the input RGB+D data is processed separately but is combined in the end to compute the overall support for a particular action.

Figure 1 shows an overview of our approach. In the learning process, we extract 2D Harris corners in 3 planes  $(x,y)$ ,  $(x,t)$ , and  $(y,t)$  from the training image sequence. Each corner is then encoded and grouped within the neighborhood of a  $3 \times 3 \times 3$  spatiotemporal cube. These encoded corners are used in an iterative grouping process, which forms descriptive compound features.

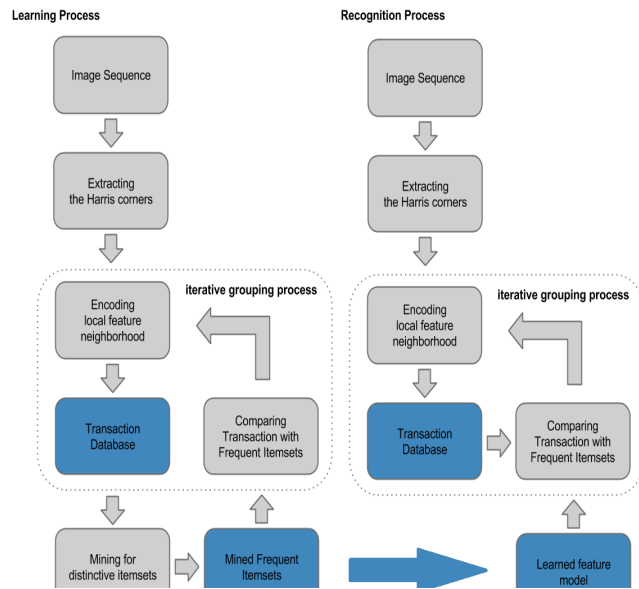


Figure 1. Overview of our approach.

Each set of grouped corners is called a Transaction, and these Transactions are collected to form a Transaction database, which is mined to find the most frequently occurring patterns, called the Frequent Item Sets. These mined item sets then become the basic feature of the next level of mining. In other words, the compound corners are re-grouped within an expanded spatiotemporal neighborhood to form a new Transaction database which data mining can be performed again. This process is repeated until we

achieve a small number of complex corners at the highest level. The Frequent Item Sets of the final stage then becomes the class feature model.

The process of recognition (classification of unseen data) is almost identical with the learning part. The Harris corners of the recognition image sequence is extracted and grouped in the same way. However, instead of mining these transactions, we check to see if they match those in the learned class feature model dataset. The recognition response is obtained by summing up all the confidences for each matched transaction.

The process is run twice, once for RGB data and once for depth data. The recognition response from both is then combined to give a final recognition response value. The image sequence is then assigned an action label according to the class that maximizes the response for that sequence. In the unlikely event that no matches occur and the model score is zero, the video would be classed as not containing any action.

#### B. Feature Extraction

The features that we use in this work are the 2D Harris corners in the three orthogonal planes of the video spatiotemporal domain  $(x,y)$ ,  $(x,t)$ , and  $(y,t)$ . The reason that 2D corners are preferred over 3D corners (proposed by Laptev and Lindeberg [5]) is because the 3D corners may be too sparse for this approach.

The 2D corners in each of the three orthogonal planes of  $(x,y)$ ,  $(x,t)$ , and  $(y,t)$  are detected independently using the OpenCV *goodFeaturesToTrack* function. Figure 2 shows an example of the 2D corners detected in a typical image.



Figure 2. Harris corner detection on a frame in three orthogonal planes,  $(x,y)$ ,  $(x,t)$ , and  $(y,t)$  [red =  $(x,y)$ , green =  $(x,t)$ , blue =  $(y,t)$ ].

To distinguish the characteristics of each corner, we encode each one with a 3-digit number. The first digit of code represents the scale. The points of interest were detected in multiple scales, relevant to the size of the search window used. The size of the search windows we used was  $\sigma_i = 3 \times 2^{i-1}$  with  $i = 1, \dots, 5$ , viz.,  $3 \times 3$ ,  $6 \times 6$ ,  $12 \times 12$ ,  $24 \times 24$  and  $48 \times 48$ . This is sufficient range for the video in our experiments, where in order to achieve real-time or near real-time processing, the size of the image has to be quite small

(we use 192 x108 images for our RGB data and 128x106 for our depth data).

The second digit of code represents the channel. The channel indicates where the interest pointed is detected. In this study, the channel has the value from 1 to 3 with 1 = (x,y), 2 = (x,t) and 3 = (y,t). The last digit of code represents the dominant orientation of the corner. In our system, we quantize the dominant orientation (from  $-180^\circ$  to  $180^\circ$ ) into 8 orientations, i.e., the orientation is divided into eight equal-sized bins of  $45^\circ$  and assigned with a value from 1 to 8).

For example, a corner with code of 125 means that this corner was detected at scale 1 (i.e., 3x3 search window), from channel 2 (i.e., the (x,t) plane) and at an orientation 5 (i.e., between  $0^\circ$  to  $45^\circ$ ).

### C. Data Mining

In our system, Association rule mining [6] is used to figure out the recurring patterns within the data. For example, an association rule has a form  $\{A, B\} \rightarrow C$ , where A, B and C are Item sets. A, B are called the antecedents and C the consequence. This association rule example implies that if there is a customer who bought items A and B, (s)he is likely to buy item C simultaneously. To analyze each rule quantitatively, we measured a 'support' and a 'confidence' value for each rule.

To process the Transaction association rules, Agrawal and Srikant [7] developed the 'Apriori algorithm' (Figure 3). This algorithm is the best known algorithm for frequent item set mining and association rule learning over transactional databases. There are two steps in Apriori algorithm. The first step is to find all item sets that have minimum support, i.e.,  $C_k$ . And the next step is to use frequent item sets to generate association 'rules' (Figure 4).

#### Algorithm Apriori(T)

```

 $C_1 \leftarrow \text{init-pass}(T);$ 
 $F_1 \leftarrow \{f \mid f \in C_1, f.\text{count}/n \geq \text{minsup}\};$ 
// n: no. of transactions in T
for ( $k = 2; F_{k-1} \neq \emptyset; k++$ ) do
     $C_k \leftarrow \text{candidate-gen}(F_{k-1});$ 
    for each transaction  $t \in T$  do
        for each candidate  $c \in C_k$  do
            if  $c$  is contained in  $t$  then
                 $c.\text{count}++;$ 
            end
        end
     $F_k \leftarrow \{c \in C_k \mid c.\text{count}/n \geq \text{minsup}\}$ 
end
return  $F \leftarrow \cup_k F_k;$ 
    
```

Figure 3. The Apriori algorithm

#### Algorithm RuleGeneration(X)

```

For each frequent itemset  $X$ ,
For each proper nonempty subset  $A$  of  $X$ ,
    Let  $B = X - A$ 
     $A \rightarrow B$  is an association rule if
         $\text{Confidence}(A \rightarrow B) \geq \text{minconf},$ 
         $\text{support}(A \rightarrow B) = \text{support}(A \cup B) =$ 
             $\text{support}(X)$ 
         $\text{confidence}(A \rightarrow B) = \text{support}(A \cup B) / \text{sup-}$ 
             $\text{port}(A)$ 
    
```

Figure 4. The Rule Generation algorithm

To clarify the concept of Association rules and Apriori algorithm, let's take an example from a real market basket analysis. Let  $I = \{i_1, i_2, \dots, i_m\}$  be a set of items (an 'item' can be a real item in a basket and  $I$  is the set of all items sold in the store). Transaction  $t$  is a set of items, where  $t \subseteq I$  (a transaction can represent items purchased in a basket), and the Transaction database  $T$  means a set of transactions, which can be written as  $T = \{t_1, t_2, \dots, t_n\}$ . An association rule is an implication of the form:  $X \rightarrow Y$ , where  $X, Y$  are item sets.

### D. Learning

To learn the frequent mined corner configurations (i.e., the class feature model), we apply the same method of neighborhood encoding as Gilbert et al. [4] for all the features that were detected in the feature extraction stage. In the encoding scheme, a regular  $3 \times 3 \times 3$  grid is used to establish a neighborhood for encoding the relative position of corners.

Figure 5 shows four corners that have been detected in the region around a central corner that is marked with a red cross. For the neighborhood encoding, we are interested in the relative positions of each corner relative to the center corner. In a  $3 \times 3 \times 3$  grid, there are a total of 27 cells. Each cell is numbered from 0 to 26, starting from the smallest value of  $t, y$  and  $x$  an increasing first in  $x$ , then in  $y$  and finally  $t$ . For example, in Figure. 5, the top-left corner of the front  $3 \times 3$  grid (i.e., at  $t-\omega$ ) will have a position code of 00 and the bottom-right hand corner of the back  $3 \times 3$  grid (i.e., at  $t+\omega$ ) will be 26. The center cell will therefore be 13.

Each corner has its individual three-digit code, and in the neighborhood encoding, each corner is then prefixed with an integer that denotes the grid cell where it occurs. For the central corner in Figure 5, the cell number is 13, and hence, the center feature is represented by the five-digit number 13125.

This five-digit number is known in data mining as an item and encoding all of four corners in the grid will yield four items, e.g., 00321, 08237, 13125 and 20112. The items are then concatenated into a large 1D vector, known within the mining community as a Transaction vector. Here  $T = \{00321, 08237, 13125, 20112\}$ . For the purposes of the training stage, each Transaction vector is appended with the label of the associated action class,  $\alpha$ . Hence, the Transaction vector becomes  $\{00321, 08237, 13125, 20112, \alpha\}$ . This encoding process is then repeated for all 2D corners detected in the video sequence to produce  $D_1$ , the transaction database for the first stage of mining.

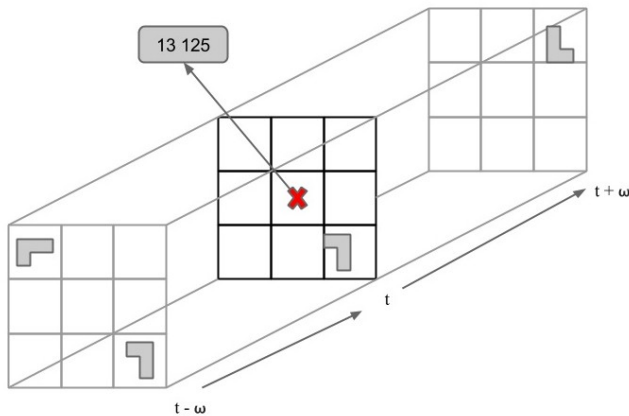


Figure 5. The grid centered on a corner shown by a cross. Four other corners are found within the neighborhood defined by the grid.

The process is then repeated at a second level  $l = 2$ , where the compound feature detected in level  $l = 1$  becomes the input to the next higher level of hierarchical grouping. Using the same encoding scheme of level  $l = 1$ , we prefix each of these compound feature with the cell position that it was found. At these higher levels, we still use a  $3 \times 3 \times 3$  grid with 27 cells, but each cell now has a larger size, where  $\omega'_l = 2 \times \omega^{l-1}$ . In our experiments, we use  $l$  from 1 to 3.

#### E. Recognition

After the training has taken place, the frequently recurring distinctive and descriptive compound features for each class  $\alpha$ ,  $M(\alpha)$ , are produced. To classify an unseen RGB video sequence, we use the same procedure as Gilbert et al. [4]. The video is analyzed in the same way as in the learning process, but instead of mining patterns from  $D$ , only patterns that exist in  $M(\alpha)$  are passed to the next level. The confidence of each transaction in  $M(\alpha)$  is used to weight the matches, as a high confidence would indicate that the Transaction  $T$  is distinctive compared to other classes. The use of the confidence ensures that if the transaction is matched with several classes, the confidence will provide a measure of the discrimination between those classes. The response  $R$  of the classifier is given by

$$R_\alpha = \frac{1}{|D \cap M(\alpha)| |M(\alpha)|} \sum_{\forall T_i \in D} m(T_i, M(\alpha)), \quad (1)$$

where

$$m(T_i, M(\alpha)) = \begin{cases} \text{conf}(T_i \Rightarrow \alpha), & T_i \in M(\alpha) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The entire process is then repeated for the depth image sequence. The two responses from the RGB and the depth data are then combined to give an overall response. The image sequence is then assigned an action label according to the class that maximizes the response for that sequence. In the unlikely event that no matches occur and the model score is zero, the video would be classified as not containing any action.

## IV. IMPLEMENTATION

### A. Hardware/Software specifications

In our implementation, we make use of the Microsoft Kinect for Windows v2 that was released in the summer of 2014. The Kinect v2 comes with a 1080p color camera and a 512x424 depth sensing camera [8]. The Kinect was connected to an Intel Core i5-3210M 2.50GHz laptop running Windows 8.1, Microsoft Visual Studio 2013 and OpenCV 2.4.10.

### B. The six action classes

In our experiments, we have identified a set of six actions that we want to recognize. Two of the action class (*Still*. and *Rolling*.) represent normal action (or normal condition) of a patient on a hospital bed.

- 1) *Still*. This is the default normal action. The patient is either sleeping or lying motionless on the bed.
- 2) *Rolling*. This is also a normal action. The patient turns in the bed occasionally.
- 3) *Coughing*. This is the first abnormal action. The patient is coughing violently for a prolonged period of time, indicating a serious deterioration of the patient's condition.
- 4) *Bottle*. This represents a patient's repeated but failed attempts to reach for a water bottle on the bedside table. The patient requires water but is too weak to reach for it by himself.
- 5) *Button*. This represents a patient's repeated but failed attempt to reach for the nurse call button on the wall beside the bed. The patient could be in a life-threatening situation but is too weak to reach the button.
- 6) *Falling*. This represents the patient completely falling off the bed, which may be caused by excessive turning in bed, or when the patient reach out too far for an object (e.g., water bottle).

The default action (i.e., *Still*) is necessary because our algorithm will always assign an action to the test image sequence (the action class that have the largest response). If lying still is not considered as an action, then the algorithm will need some nontrivial modification to handle them so that it will not cause the unintended rising of the false positive rate.

Figure 6 shows examples of the RGB and depth image of each of the six action class.

### C. Generating the Image Database

For each action except the first (i.e., lying still), we capture about 5-8 seconds of the action (at 30 frames per second) in both RGB and Depth images at full resolution (RGB: 1920x1080, Depth: 512x414). Each action was preceded by about 1.5 to 2 seconds of no action (i.e., lying still) which actually constitutes Action 1 (*Still*).

Due to the computation complexity of the algorithm, we have rescaled the RGB images by a factor of 10 (to 192x108 pixels) and the Depth images by a factor of 4 (to 129x106 pixels) in our experiments.

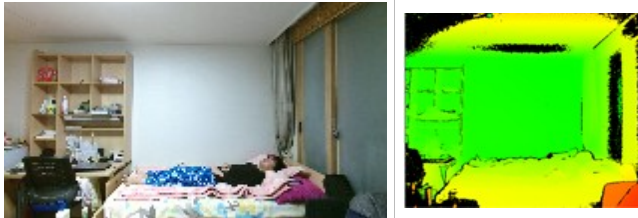


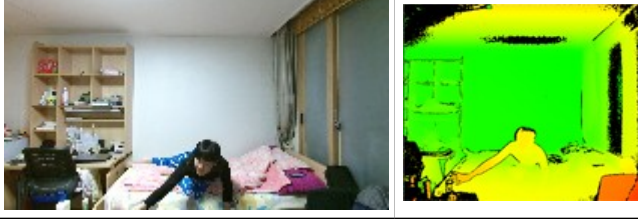
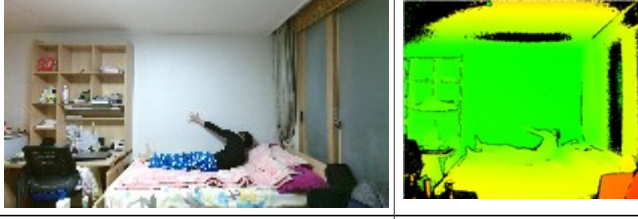
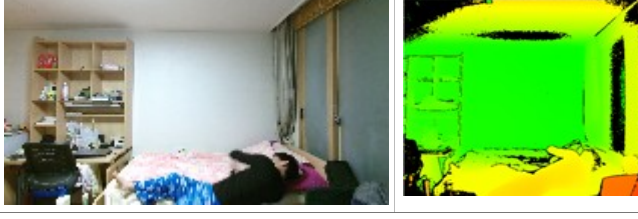
Action name	No. of frames
Still	220 frames
	
Rolling	168 frames
	
Coughing (Abnormal action)	210 frames
	
Bottle (Abnormal action)	219 frames
	
Button (Abnormal action)	215 frames
	
Falling (Abnormal action)	204 frames
	

Figure 6. Example RGB+D images of the six action classes.

## V. EXPERIMENTAL RESULTS

### A. Recognition Rate

We test our system with 52 test sequences (of 2 second duration each, i.e., 60 frames). The results are shown in Table I.

TABLE I. RECOGNITION RATES

Action	No. of Test Sequences	No. of Successful Recognitions	Recognition rate(%)
Still	4	4	100.0
Roll	6	5	83.3
Cough	11	9	81.8
Bottle	8	6	75.0
Button	13	11	84.6
Fall	10	10	100.0

### B. Confusion Matrix

Figure 7 shows the confusion matrix of our experiments.

	Still	Roll	Cough	Bottle	Button	Fall
Still						
Roll	17					
Cough				9	9	
Bottle			13		12	
Button			8	7		
Fall						

Figure 7. Confusion matrix

### C. Comparison with other methods

We compare our method with traditional classification algorithm such as binary decision tree, ensemble tree, k-NN (k-Nearest Neighbors algorithm), SVM (Support Vector Machine) with radial basis function kernel and neural network classifier with back-propagation learning.

TABLE II. COMPARISON WITH OTHER METHODS

Method	Average classification accuracy (%)
Binary decision tree	76.63
Ensemble tree	90.83
k-NN	86.77
SVM with radial basis function kernel	87.74
Neural network classifier with back-propagation learning	89.26
Our method	86.54

The results are shown in Table II. We observe that our system is comparable to other state-of-the-art methods.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a system that makes use of the RGB+D sensors available on the Microsoft Kinect for Windows v2 to determine abnormal behaviors of hospital patients. Our proposed system processes the RGB information and Depth information separately, and then combines the responses to make the action classification.

In the learning phase, for each of the RGB and Depth image sequence, 2D Harris corners are detected in each of the three orthogonal planes of the video spatiotemporal domain  $(x,y)$ ,  $(x,t)$ , and  $(y,t)$  and then encoded based on their relationship in an expanded spatiotemporal neighborhood. These encoded corners formed Transactions in a Transaction Database which is subsequently mined to obtain the class feature models. Two separate transaction databases were kept for the RGB data and the depth data.

During the recognition phase, a similar process of extracting 2D Harris corners from the RGB test sequence and the Depth test sequence is used. These corners were also encoded into Transactions but instead of mining them, they are matched with the class feature model. The recognition response is obtained by summing up all the confidences for each matched transaction.

Our experiments showed that this approach is able to generate good recognitions rates and is comparable to other state-of the art algorithms. Our future work will include

collaborating with an actual hospital to perform live trials on their patients.

## REFERENCES

- [1] T. Banerjee, M. Enayati, J. M. Keller, M. Skubic, M. Popescu, and M. Rantz, "Monitoring patients in hospital beds using unobtrusive depth sensors" in Conf Proc IEEE Eng Med Biol Soc. 2014, 2014, pp. 5904-5907.
- [2] S. Saha, S. Datta, A. Konar, and R. Janarthanan, "A Study on Emotion Recognition from Body Gestures Using Kinect Sensor", International Conference on Communication and Signal Processing, 2014, pp. 56-60.
- [3] E. P. Ijjina and C. K. Mohan, "Facial Expression Recognition Using Kinect Depth Sensor and Convolutional Neural Networks", 13th International Conference on Machine Learning and Applications (ICMLA), 2014, pp. 392 – 396.
- [4] A. Gilbert, J. Illingworth, and R. Bowden, , "Action Recognition Using Mined Hierarchical Compound Features", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 33, Issue 5, 2010, pp. 883- 897.
- [5] I. Laptev and T. Lindeberg, "Space-Time Interest Points," Proc. IEEE Int'l Conf. Computer Vision, 2003, pp. 432-439.
- [6] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. 1993 ACM SIGMOD, 1993, pp. 207-216.
- [7] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Proc. 20th Int'l Conf. Very Large Data Bases, 1994, pp. 487-499.
- [8] Microsoft Kinect for Windows v2 features [Online]. Available at <http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx>. Retrieved 23 Mar 2015.