Abnormal Activity Detection for Bank ATM Surveillance

Rajvi Nar
Computer Science Dept.
rajvinar.333@gmail.com

Alisha Singal
Electronics & Communication Engg. Dept.
alishasingal236@gmail.com

Praveen Kumar
Computer Science Dept.
praveen.kverma@gmail.com

The LNM Institute of Information Technology
Jaipur, India

August 18, 2016

Abstract

Posture recognition is one of the most interesting fields in computer vision because of its numerous applications in various fields. The problem we face with simple camera can be solved by the usage of a 3D camera. In this project, we explore a technique of using skeleton information provided by Kinect 3D camera for posture recognition for effective real-time ATM intelligent monitoring. To achieve posture recognition we can use kinect to track bone joints and their positions. By analyzing the position information, the system detects abnormal behaviors.

Keywords- Abnormal behavior, Kinect, Posture recognition

1 Introduction

ATM came into popularity in terms of research and usage in the early to mid 1990s. In the last few decades, ATM has become one of the most important facilities used by consumers worldwide for withdrawal of cash or to carry out other transactions. With the popularity of ATMs, banking has become much convenient. But at the same time ATMs are one of the most vulnerable sites. For this reason, ATMs have a video surveillance system installed. Unfortunately, the current systems are not so efficient to detect abnormal behaviour due to many reasons. If a criminal suspect is wearing a hat, a mask, or sunglasses, it is not possible to get a clear picture of the suspect with recognizable facial features [**1**]. What we need are intelligent systems that can automatically give proper warning feedback in real time. In this project, we design a real-time system for ATM monitoring which checks for abnormal behaviors in the ATM. Using Kinect, the system tracks the people present in the ATM room and calculates their position relations, and derives features which can be used to effectively analyze the person's behavior. When the system detects an abnormal behavior, it alarms the people present in the ATM as well as the ATM monitoring employees.

2 Related works

There is a lot of work that has been proposed for posture recognition. However, most of this work utilizes color information captured from simple RGB cameras. In [**2**], Chella et al. proposed a system for simultaneous people tracking in real life also detecting posture abnormality in any kind of environment in the context of human-computer interaction. As soon as the tracking algorithm tracks a person, the system also estimates his/her posture. In [**3**] Bernard, B. proposed a real-time, generic, and operational approach for recognition of human posture with a static camera. The 2D techniques represent the silhouettes of the observed person to provide a real-time processing. The proposed approach is composed of two main parts: the posture detection that recognises the posture of the person in observation by using information computed on the studied frame, and the posture temporal filtering that filters the posture by using information about the posture of the person on the previous frames.

The above mentioned human posture recognition works are dedicated to color cameras. In these works, in order to recognize human posture, region of interest in image has to be determined. The main disadvantage of these works is that their approach is sensitive to the change in clothing and lighting conditions. Unlike these works, the work proposed in this paper aims at detecting human posture from the skeleton (using depth camera). It is therefore going to react the same in day or night,whether its executing outside or inside irrespective of lighting intensity which makes it suitable for application for ATM surveillance.

3 Overview

An overview of our proposed model is presented in the figure 1. Our model consists of three modules: data acquisition, data processing and feature extraction,and posture recognition.

The technique we have used to label a posture as normal or abnormal is Logistic Regression. As we are trying

978-1-5090-2029-4/16/$31.00 @2016 IEEE

to estimate the probability of a posture being abnormal depending on the skeleton information, regression analysis is a good approach to this problem. Moreover, as our dependent variable is dichotomous (normal behavior and abnormal behavior), Logistic Regression will be an efficient regression model in this case. The whole process from data acquisition to posture recognition is described below.

Figure 1: Project Components

In **data acquisition**, using kinect, we capture different types of information i.e. color, depth and skeleton information. **data processing** aims at finding the 3-D locations of body-joints using SimpleOpenNI toolbox. PrimeSense has provided two separate pieces of software that are useful to us. First is the OpenNI framework which includes the drivers for accessing the basic depth data from Kinect. Another feature is user tracking in which the algorithm use and process the depth image to determine the joints positions of any person within the camera's range. We then define relevant features based on angles made by lines joining some of the joints. Also, we normalize the data as needed. Kinect can also be used to extract a color stream as in a traditional camera apart from the depth and skeleton information. Besides the depth and skeleton information, Kinect also provides a color information similar to a traditional camera. However, in this project, we focus on exploring the possibility to use only skeleton information for human posture recognition. **Human posture recognition** aims at learning the postures provided as a training data set and classifying the test posture using one of the predefined classes: alarming condition and not alarming condition.

3.1 Data acquisition

The Kinect sensor is a motion sensing device. Its name is a combination of the terms 'kinetic' and 'connect'. It was originally designed as a natural user interface (NUI) for the Microsoft Xbox 360 video game console to create a new control-free experience for the user. It enables the user to interact and control software on the Xbox 360 with gesture recognition.

We used kinect as it's highly effective in providing depth information of the subjects as compared to other similar devices. Kinect consists of multiple sensors. Kinect offers an RGB camera that captures twelve 1280x960 resolution images per second. It also provides an IR sensor which helps us capture a depth image. It projects multiple dots which allows the final camera on the right side, the CMOS depth camera, to compute a 3D environment. The device is mounted with a motorized tilt to adjust the vertical angle. Kinect can detect up to 2 users at the same time, and compute their skeletons in 3-D with 2-D joints representing body junctions like the feet, knees, hips, shoulders, elbows, wrists, head, etc.

Figure 2: Kinect

The software on the Xbox processes the depth image in order to locate people and find the positions of their body parts. This process is known as **skeletonization** because the software infers the position of a user's skeleton (specifically, his joints and the bones that connect them) from the data in the depth image [**4**].

By using the appropriate **Processing library**, the user position data can be accessed which provides information about various joints in the skeleton. The joint information is collected in frames. For each frame, the positions of 2-D points are estimated and collected. Each joint provides two main pieces of information. Firstly, we get the position of joints in x, y, and z coordinates. Lastly, status of joints can be accessed. If Kinect is able to track the joint, it sets the status of the joint to 'tracked'. If the joint couldn't be tracked, the algorithm tries to get the position of joints from other joints. If successful, the status of this joint is set to 'inferred'. Otherwise, the status of the joint is set to 'non-tracked'.

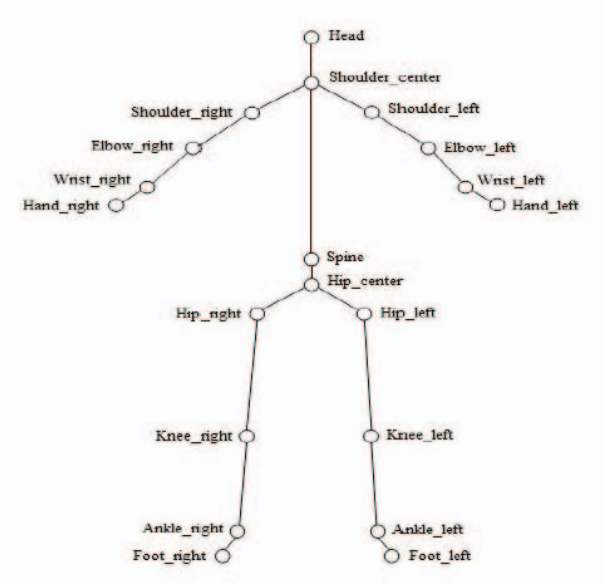

Figure 3: Joints obtained from skeleton data

3.2 Data processing and Feature extraction

With the skeleton tracked by Kinect, joint positions are obtained. Since, joint vector has 3 coordinates and a skeleton consist of 20 joints, the feature vector has 60 dimensions.

Apart from the feature described above, another feature can be extracted by calculating the joint angles. While working with postures, we observed that ten joints, namely Torso, Neck, Head, Left shoulder, Left elbow, Left wrist, Right shoulder, Right elbow, Right wrist, Left hip and Right hip, are the most important joints for representing postures. From these joints, we can calculate different sets of angles. Subsequently, we defined the following angle based features to recognize desired postures: angle between vector joining left hand to left elbow and vector joining left elbow to left shoulder, angle between vector joining left elbow to left shoulder and left shoulder to torso, angle between vector joining right hand to right elbow and right elbow to right shoulder, and angle between vector joining right elbow to right shoulder and right Shoulder to torso. The calculation of the subject's posture is based on the fundamental consideration that the orientation of the subject's torso is the most characteristic quantity of the subject during the execution of any action and for that reason it could be used as reference.

3.3 Posture Recognition

Posture recognition can be achieved using Logistic regression, a Machine Learning technique that for a given input, predicts a class and provides a probability associated with the prediction. These probabilities are extremely useful, as they provide a degree of confidence in the predictions.

3.3.1 Logistic regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is binary. The logistic regression is a kind of predictive analysis. Logistic regression is used to describe data, and to explain the relationship between one dependent binary variable and one or more independent variables.

Hypothesis function for Logistic regression is:

$$h_\theta(x) = \frac{1}{1+e^{-z}} \tag{1}$$

where z is $\theta^T X$ and $h_\theta(x)$ is called sigmoid function that looks like figure 4.

We can see that $h_\theta(x) \to 1$ as $z \to \infty$ and $h_\theta(x) \to 0$ as $z \to -\infty$, and that $h_\theta(x)$ is bounded between (0,1).
In Logistic Regression, the function below is used as the cost function:

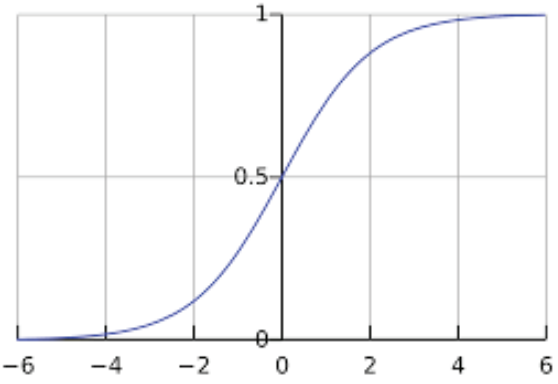

Figure 4: Logistic Regression

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases} \tag{2}$$

which basically means that if $h_\theta(x)$ is near 1, the cost will be small, and if $h_\theta(x)$ is near 0, the cost will be huge.

means that, if $h_\theta(x)$ is near 0, the cost will be small, and if $h_\theta(x)$ is near 1, the cost will be huge. Putting these two parts together, we can get the following cost function:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} log(h_\theta(x^{(i)}) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))] \tag{3}$$

The target is to minimize this cost function, so we can use the gradient descent method. For gradient descent, we just iteratively trim our θ vector:

$$\theta_j := \theta_j + \alpha(y^{(i)} - (h_\theta(x^{(i)}))x_j^{(i)} \tag{4}$$

where, α is the learning rate. Here we chose $\alpha = 0.1$. θ_j refers to the jth element in θ vector, i is the current training sample in use.

3.4 System implementation

The features we used in this project are the angles made by different bones in a given posture. We use these features to sense abnormal positions or intentions. We stored the training data (feature values) for the algorithm in a text file. The size of the sample data we took for training the algorithm was 100. Using this training data, we calculate the weights in MATLAB using the Gradient Descent Method. The calculated weights are then used in the processing code.
Using these weights we would try to guess the hyperplane which can distinguish our features in most optimal way. By multiplying newly made features (newly made features consist interdependency of features on each other as a feature) with weights calculated before we calculate the

probability for each $y = 1$ case. After some experimenting, we decided the threshold probability to be 0.55 to get the most accurate results. If probability is greater than the set threshold value then the system will fire the alarm as shown in Figures 5, 6, and 7.

3.5 Normal and Abnormal Postures

For demonstration purposes, we have selected the following three postures to be considered abnormal, and have used the same as our training set data -

- Aggressive posture (fig. 5)
- Fiddling with the camera posture (fig. 6)
- Peeping posture (fig. 7)

An alarm is raised if the current posture of the person matches with any of the postures mentioned above. We have considered any posture that does not fit the abnormal posture category as a normal posture. No alarm is raised in this case.

4 Result

After training the machine using the training data, the algorithm was be able to detect abnormal and normal behaviours as expected. For example, if the person is trying to destroy the ATM as shown in Figure 5, or is trying to peep as shown in Figure 7, or is trying to block the camera as shown in Figure 6, then the system fired the alarm. When there is no such activity as shown in Figure 8, no alarm was fired.

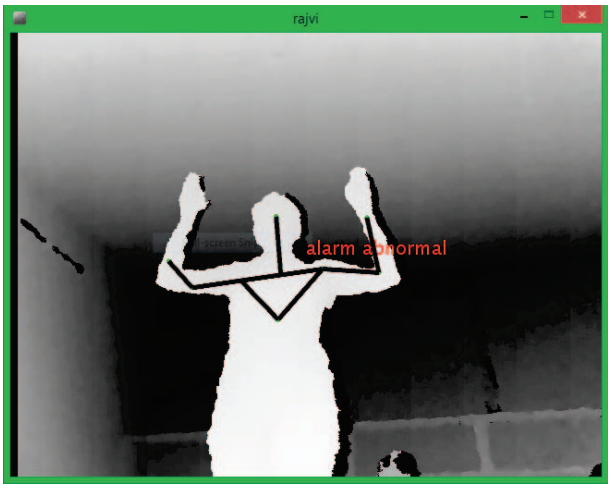

Figure 5: Aggressive posture

Figure 6: Fiddling with the camera posture

Figure 7: peeping posture

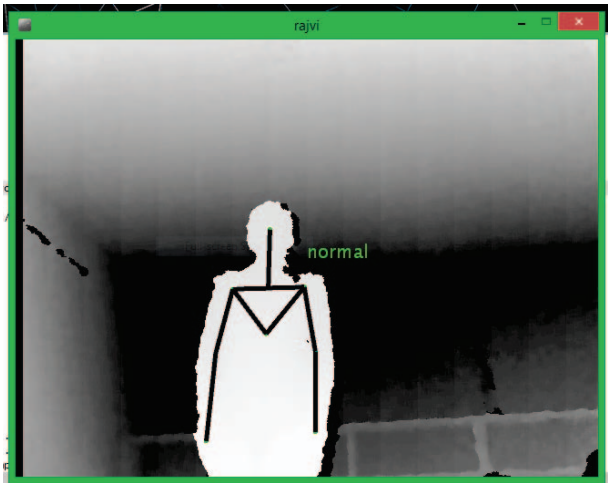

Figure 8: normal posture and no alarm

5 Conclusion

In this project we showed that posture recognition can be used to detect abnormal behavior of a person in an ATM. We achieved this by using skeleton data that can be extracted from the depth image provided by a 3D camera like Kinect. We used Processing language (built on Java) to write our code. A Machine Learning algorithm, Logistic Regression was used to calculate the probability of the current pose of the person under surveillance being abnormal. This was achieved by calculating the weights from a training data set. We used angles between different bones as features which we processed using a MATLAB program to calculate the weights for the algorithm. We used gradient descent method to calculate the optimum value of weights. And this way we were able to make the machine learn the abnormal behavior using algorithm and data.

References

[1] Min Yi, *Abnormal Event Detection Method for ATM Video and its Application*. Communications in Computer and Information Science, Springer 2011.

[2] Antonio Chella, Haris Dindo, Ignazio Infantino, *People Tracking and Posture Recognition for Human-Robot Interaction*. Vision Based Human-Robot Interaction, Palermo, Italy, March 2006.

[3] Bernard Boulay, *Human Posture Recognition for Behaviour*. [*cs.OH*]. Universite Nice Sophia Antipolis, 2007. English.

[4] Greg Borenstein, *Making things see*. Maker Media Inc., 2012.