ISSN 1870-4069

Training NAO using Kinect

Michalis Chartomatsidis, Emmanouil Androulakis, Ergina Kavallieratou

University of the Aegean Samos,
Dept of Information & Communications Systems, Greece

kavallieratou@aegean.gr

Abstract. This paper describes how the motions of the humanoid NAO robot can be controlled using the Microsoft sensor, Kinect. An application is implemented by which the robot can be controlled using real time tracking. An option to capture and save some motions is also included in order to test if it is possible to train the robot to execute automated motions. The basic question to answer by this work is whether NAO is able to help the user by lifting up an object. To answer that, a series of experiments were performed to validate if the robot could mimic both the captured and the real time motions successfully.

Keywords. Kinect, NAO, skeleton tracking, verification.

1 Introduction

The Robotics is a modern technology, which includes the study, design and operation of robots, as well as researches for their further development. The definition given to the robot by the Robot Institute of America is: "A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks". Nowadays, there are various types of robots:

- Industrial robots: used in industrial manufacturing environment.
- Household robots: This category includes a variety of devices such as vacuum cleaners robot, robot pool cleaners etc.
- Medical Robots: used in medicine, e.g. surgical robots.
- Service Robots: these have a specific and unique use as demonstration of specific technologies or robots used for research.
- Military robots: used to neutralize bombs or in other fields such as search and rescue.
- Entertainment Robot: this consists of robotic games to motion simulators.
- Space robots: robots used in space stations.

Among the above categories, the anthropoid robots are extremely popular since they are the human dreams since many decades. However, the ability to imitate human is not natural. It requires a lot of work and research.

pp. 27–37; rec. 2016-09-20; acc. 2016-10-07

In this paper, the Microsoft Kinect Sensor is used to give natural human motion to the Aldebaran NAO anthropoid.

Next, in section 2, previous research is mentioned, while in section 3 Kinect and NAO are shortly described. Our methodology is presented in section 4, while in section 5, several experiments are described. Finally, our conclusion and future work are drawn in section 6.

2 Related Work

There are many works that combine Kinect and NAO and explain how these can help in many different situations.

In [1], a study on learning sign language to children with vision and hearing problems was presented. Both NAO and Kinect were used. Initially the Kinect device was used in order to record the movements of a man and then these movements were transferred to the memory of the robot. Also existing techniques for learning as well as a new web technique were presented. The results showed that learning through the internet is accurate up to 96%, comparatively much larger than the other techniques under study.

In [2], a system that allowed NAO to perform movements made by man in real time was presented. The movements were captured by the Xsens MVN system. The robot was called to perform a series of complex movements, predominantly balance so that they could come to some conclusions about the ability of the robot to adjust the center of its gravity without falling.

Guo, Melissa et al. [4], presented a platform by which children with autism can interact with a robot. This study was based on studies that showed that most children with autism communicate better with robots than with people. By the help of Kinect some movements were performed and then incorporated in the robot’s memory. The children were able to play with the robot as they could make movements and the robot in turn to repeat these movements in real time. The results showed that this could be a better therapy for children with autism.

Lopez Recio et al. [5] studied whether the NAO could help to elderly physiotherapy. Patients instead of following the guidance of a therapist look at the robot that moves and repeat. Moreover, beyond the physical robot a virtual one was used and the performance of the patients was compared. The results initially showed that by the natural Robot patients had better performance compared to the virtual one. Furthermore, in some cases the robot spent much time to complete a movement and the patients were not focusing on the robot. On the other hand, when the robot executed the motions in normal speed, patients were more successful.

3 Hardware

3.1 Kinect Sensor

The Kinect sensor (Fig.1) is a device created by Microsoft for the Xbox 360, Xbox one and PCs. Initially, it was used for video games in Xbox but later it became a strong tool for programmers. It can track human skeleton, recognize voice and provide depth or color data that can be used in many applications. The hardware consists of an RGB camera, an IR emitter, IR depth sensor, a tilt motor and a microphone array. Since Kinect was released, Microsoft SDK has been the official tool for developing applications. Besides that, there are some other tools, like OpeNI & Nite, CL NUI and libfreenect. Here, the Microsoft's SDK is used, which allows programming in many program languages such as C++ and C#, it does not require a calibration pose to track the user and it is able to track more joints than any other developing tool.

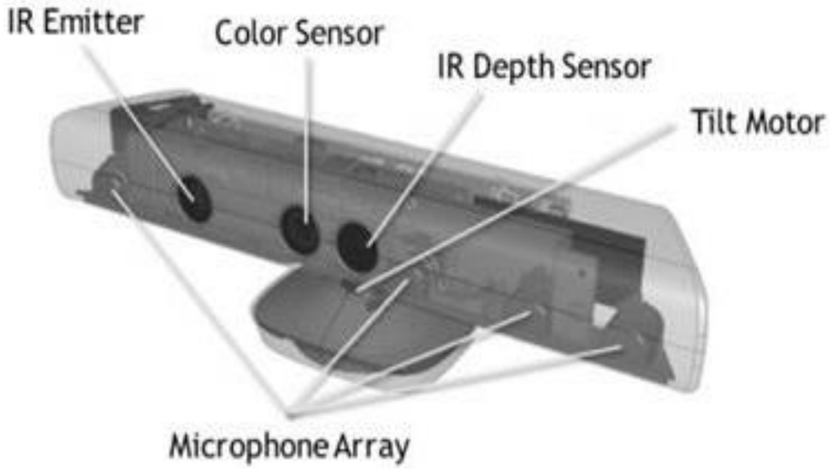

Fig. 1. Kinect sensor's hardware.

3.2 NAO Robot

Manufactured by the French robotic company, Aldebaran Robotics, NAO (Fig.2) is a widely used robot in many research institutes. In this thesis the academic version was used that has 25 joints, resulting 25 degrees of freedom (DOF), two cameras that allow it to interact with the environment, four microphones and loudspeakers for listening to voice commands and numerous sensors to enable it to perceive the environment.

In order to move, NAO is using a dynamic model of square programming. In particular, NAO is receiving information from joint's sensors becoming more stable and resistant in walking. Random torso oscillations are being absorbed. The mechanism that controls its movement is based on reverse kinematics, which manages Cartesian coordinates, joint control, and balance. For example, if during a motion, NAO realizes that is about to lose balance then it stops every motion.

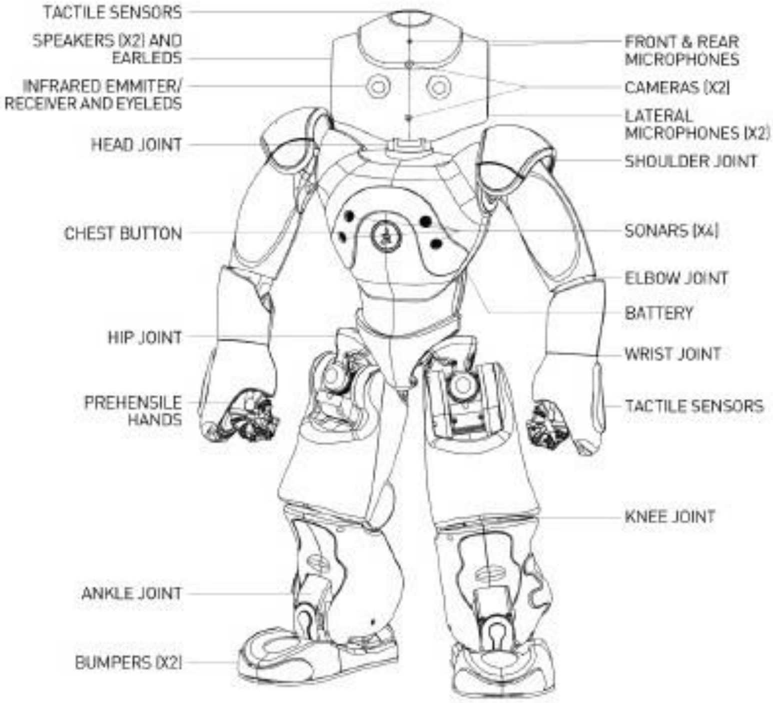

Fig. 2. NAO robot hardware.

In case it loses balance, it features a fall manager mechanism which is responsible to protect it in case it falls. The main purpose of this mechanism is to detect any change in the center of mass which is determined by the position of the feet. When NAO is about to fall down every other motion is being terminated and the hands are positioned depending on the fall direction. Also the center of mass is reduced and robot's inflexibility decreases.

Furthermore, NAO is equipped with sensors. These sensors allow NAO to have access to information through touching objects. Furthermore two sonar channels provide information which is used to calculate the distance between the robot and an obstacle. The detection range varies from 1cm up to 3 meters but for distance less than 15cm the distance from the obstacle cannot be calculated.

NAO is using an operating system, called NAOqi and it is based on natural interaction with the environment. Here, the NAOqi was used on computer, in order to test its behavior through simulations. It allows homogeneous communication between modules such as movement, sound or video. Also it can be used on many different platforms such as Windows, Linux or Mac. The software development is possible in different programming languages such as C++ and Python. For the purpose of this thesis NAOqi was used in Windows by using Python. Moreover, the Choregraphe was used, a software that allows to observe NAO's behavior.

4 The Proposed Methodology

In this section, it is described how the Kinect sensor can send the tracked data to NAO and how NAO is able to receive this data and repeat the motions of the user. Moreover, the proposed methodology to train NAO will be described.

The basic idea can be described in four steps:

1. the Kinect tracks the user and saves his skeleton joints,
2. the angle that is formed between three of the joints is calculated,
3. an offset is being calculated for this angle and
4. the value of the angle is being sent to robot.

Before presenting how Kinect and NAO are communicating, it should be indicated that in order to have a better understanding about robot's behavior, is is separated in seven body parts: head, left arm, left hand, right arm, right hand, left leg and right leg. For each of the body parts, it was further studied the motion of the joints that were considered important for this thesis as shown in the table below (Table 1).

Table 1. Joints and body parts that were studied.

Body Part	Joint (code name)
Left Arm	LShoulderPitch
	LShoulderRoll
	LElbowRoll
	LHand
Right Arm	RShoulderPitch
	RShoulderRoll
	RElbowRoll
	RHand
Head	HeadRoll
Left Leg	LHipPitch
Right Leg	RHipPitch

4.1 Communication

In order for NAO to communicate with the Kinect sensor, the best way proved the use of the socket technology. Each of the body part mentioned above was acting as a server and the Kinect sensor was acting as a client. Thus, the Kinect sensor was connected to these seven body parts.

Furthermore, in order to have all the servers online and ready to receive the values from Kinect, threads were used. Thus, every server was able to run regardless the other.

4.2 Angle Calculation

In order to calculate the angle formed between three joints, simple mathematics are used. Basically, four parameters are used, the detected human skeleton and three skeleton joints. Two vectors per two joints are created and converted to units, and then the dot and cross product are calculated. Here's an example in order to understand deeply the calculation (Fig.3). Let’s suppose that the elbow angle is to be calculated. From the detected joints the x, y, z coordinates of the shoulder (j1), elbow (j2) and wrist joint (j3), are used. The j2j1 and j2j3 vectors are created normalized to units and the cross and dot products are calculated. Using the last two, the atan2 of the angle is calculated and converted in radians.

After the angle calculation, an offset is applied to the angle and the new value is sent to the robot. The methodology of how this offset is calculated is described in the next paragraph.

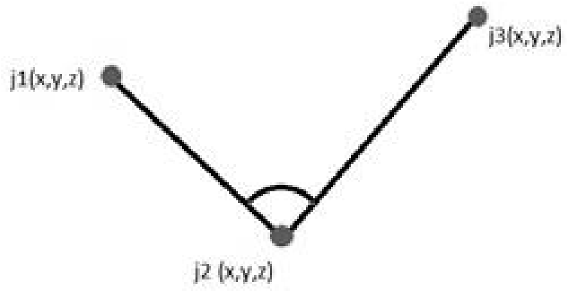

Fig. 3. Calculation of angle.

4.3 Offset Calculation

The determination of the appropriate offset for every angle was a quite challenging procedure. Many trials were performed before end up with the correct values. The main difference that makes this procedure difficult is that the value range that NAO can accept, is different than the value range the Kinect returns. The basic procedure to find the correct values can be described through the stages:

- Every joint of the robot is studied for the selected body parts through Choregraphe and considered a starting pose, a mid and a final. Then the angle value for each pose is kept.

- Then the same steps are used to find the angle values for the user using Kinect's skeleton tracking.
- Finally, by the corresponding values an offset value is calculated for each joint.

4.4 Training NAO

As already mentioned, it is implemented a program that trains the robot by doing some motions. In order to succeed that the user records a motion of his own and stores it into a file of skeleton objects. After that NAO is taking the command to execute the saved motion through a simple procedure: The saved file is read and for every three joints in the file the procedure that was described above is repeated.

This way NAO can be trained in doing some motions that may require to be executed many times. It's clear that using real time motion the user would have to execute the same motion many times and the robot would follow. This would be exhausting for the user and could not help in any way.

4.5 Avoiding Loss of Balance

The ability of NAO to be able to execute the motions without the danger of falling was one of the most important things to be implemented. Balance is a factor that cannot be ignored. Even if the robot holds an object there is the danger that it may fall down. To avoid this situation when the robot connects and is ready to receive the values from skeleton tracking it sets its center of mass to both legs. Therefore it is able to execute any kind of motions or hold any object without falling down.

4.6 Receiving Values

As mentioned, each server is responsible for some joints of each body part. In order to command NAO to move, some stages have to be followed:

- First, the message that is received through socket from the client is a sting type containing values for each joint of the body part.
- To use these values, they have to be converted into a list of float type numbers that can be accepted by the NAO, using the Python's *split()* method.
- After that, by the use of the *setAngles*() function, NAO takes the command to execute the move.

5 Experimental Results

In order to verify our technique, it was tested through a series of experiments. First, the real-time tracking and sending data to NAO was tested and then the robot was trained to execute a motion that was recorded earlier by the user.

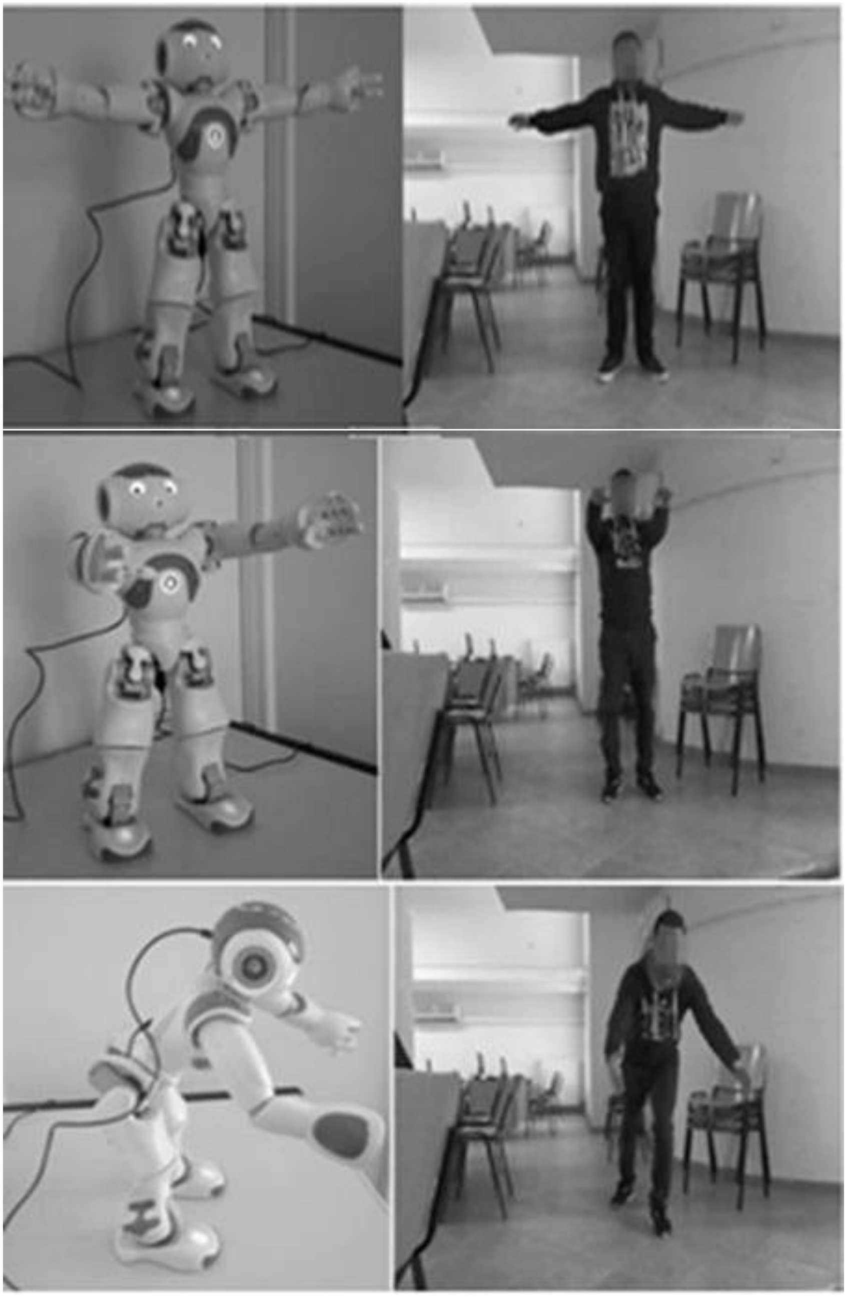

Fig. 4. Testing real time tracking.

5.1 Real Time Experiment

The first experiment is testing if NAO is able to follow the user in real-time. It's an experiment to test whether our angle and offset calculations were correct and the NAO can follow successfully the human motion in executing time and accuracy.

Our experiments (Fig.4) proved that NAO performs the motion with high success rate. However, it is difficult to control its palms because during some moves Kinect was unable to track whether the user's hand was open or closed. Any other move, which did not include opening or closing the palms, was performed accurately

5.2 Picking Up with One Hand

The second experiment (Fig.5) includes the recording of a motion by the sensor and then the imitation of this motion by the robot. Specifically, NAO picks up an object and leaves it a little further. It should be indicated that NAO does not recognize the object. As a result, in order this experiment to be successful; the object had to be at a specific location.

As mentioned in the previous experiment, the most important problem to face was to establish a way to show the robot it should close the palms throughout the movement. Using the angle between the joints left (right) elbow, left (right) wrist and left (right) hand, respectively, it was not possible to execute always the desired movement. The detection of the sensor proved significant errors in the calculation. The only way for the successful registration of the movement, was to keep a hand stable at the point where the sensor accurately understands the angle and control the opening-closing of the other hand. Specifically, in order for NAO to be able to close the right hand and grab the object, the user had to control it by using his left hand. In particular the angle that is formed between the joints Left Elbow - Left Wrist - Left Hand of the user is controlling whether the right hand of the robot is open or closed.

5.3 Helping the User

The main goal for our third and last experiment (Fig.6) was to find out if NAO is able to collaborate with the user, in order to help him lift an object.

Our first step was to train NAO into executing the proper move to lift the object. In order to perform that, a proper way to command NAO to close his palms had to be planned. As mentioned before closing NAO's palms was not something very easy. To succeed in that, a proper set of joints of the human skeleton, able to be tracked by Kinect, for the entire time that the motion was taking place had to be established. After a lot of trials, the conclusion was that the best way to make NAO's palms to close was by checking the movement of the user's head. In particular, if the user's head was looking down then NAO's hands were closing and if the user's head was looking up, NAO's hands were opening.

After that, the execution of the movement could be performed. As you can see in the fig.6 the motion consists of four stages:

1. Nao and user touch the object,
2. They both grab the object and start lifting it,
3. The object is at maximum distance from the ground and,
4. The object is put back to the ground.

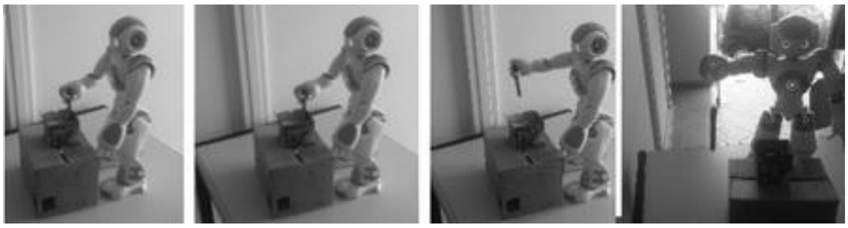

Fig. 5. NAO picks up an object.

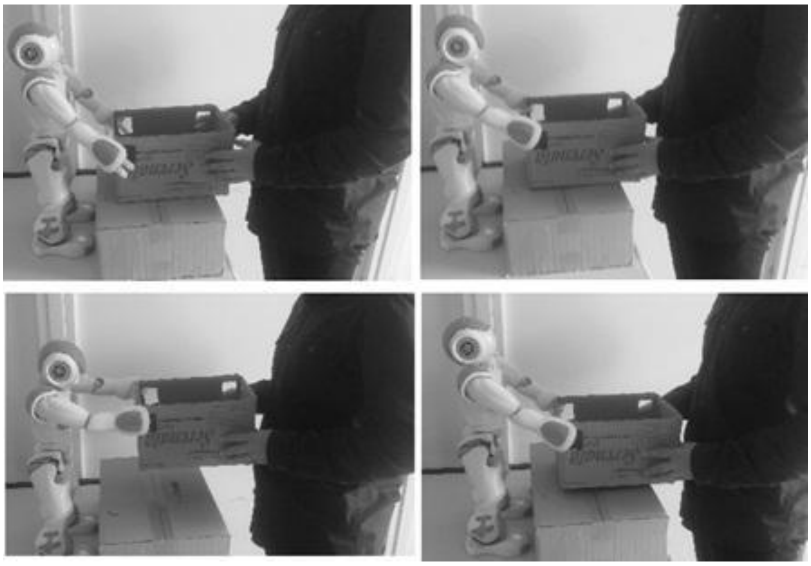

Fig. 6. NAO helps the user pick up an object.

It should be also mentioned here, that for the previous experiments NAO does not recognize its environment, the object needs to be placed at specific spot in order for the move to be successful.

It was experimented and proved that NAO is able to be programmed in order to help the user lift an object of certain size and weight in regular human execution time. This could mean that NAO robot is able to help people in their ordinary life.

6 Conclusions

In this paper, an easy way has been proposed to control the humanoid robot NAO by using the Microsoft Kinect sensor. Two methods, one for real time tracking movements and another for recording the motion and let the robot imitate it later, have been experimented. Furthermore, it has been tested successfully, the capability of NAO to help people in their ordinary life by lifting and moving light or collaborating with a person to extend or lift bigger and long objects.

As a future work, it is planned to introduce the vision of NAO or Kinect in the procedure for object recognition and introduce inversion of the human movement.

References

1. Isong, I., Kivrak, H., Kose, H.: Gesture imitation using machine learning techniques. In: Signal Processing and Communications Applications Conference (SIU) 2012 20th IEEE, 1–4 (2012)
2. Koenemann, J., Burget, F., Bennewitz, M.: Real-time imitation of human whole-body motions by humanoids. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on IEEE, 2806–2812 (2014)
3. Gouda, W. and Gomaa, W.: Nao humanoid robot motion planning based on its own kinematics. In: Methods and Models in Automation and Robotics (MMAR), 2014 19th International Conference On IEEE, 288–293 (2014)
4. Guo, M., Das, S., Bumpus, J., Bekele, E., Sarkar, N.: Interfacing of Kinect Motion Sensor and NAO Humanoid Robot for Imitation Learning (2013)
5. López, D., Márquez, E., Márquez, L., Waern, A.: The NAO models for the elderly. In: Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction, IEEE Press, 187–188 (2013)
6. Khoshelham, K.: Accuracy analysis of kinect depth data. In: ISPRS workshop laser scanning, 38 (5), 133–138 (2011)
7. Khoshelham, K., Elberink, S.: Accuracy and resolution of kinect depth data for indoor mapping applications. In: Sensors 12(2), 1437–1454 (2012)
8. Oikonomidis, I., Kyriazis, N., Argyros, A.: Efficient model-based 3D tracking of hand articulations using Kinect. In: BMVC, 1 (2), 3 (2011)
9. Tang, M.: Recognizing hand gestures with microsoft's kinect. In: Palo Alto: Department of Electrical Engineering of Stanford University (2011)
10. Weise, T., Bouaziz, S., Li, H., Pauly, M.: Realtime performance-based facial animation. ACM Transactions on Graphics (TOG), ACM, 30 (4), 77:1–10 (2011)
11. Kondori, F., Yousefi, S., Li, H., Sonning, S. and Sonning, S: 3D head pose estimation using the Kinect. In: Wireless Communications and Signal Processing (WCSP), 2011 International Conference on IEEE, 1–4 (2011)
12. Nirjon, S., Greenwood, C., Torres, C., Zhou, S., Stankovic, J. A., Yoon, H., Son, S.: Kintense: A robust, accurate, real-time and evolving system for detecting aggressive actions from streaming 3d skeleton data. In: Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on IEEE, 2–10 (2014)