Action Recognition Through Device Sensors

Jevons Wang
Department of Computer Science
New York University
New York, NY, USA 10003
Email: jw4106@nyu.edu

Jia C. Shang
Department of Computer
& Information Sciences
Temple University
Philadelphia, PA, USA 19121
Email: jiacheng.shang@temple.edu

Jie Wu
Department of Computer
& Information Sciences
Temple University
Philadelphia, PA, USA 19121
Email: jiewu@temple.edu

***Abstract*—We have proposed a new form of action recognition using motion sensors built in mobile and wearable devices. Due to the miniaturization of hardware sensors, action classification through mobile sensors has become a much more attainable task. Using Android and Tizens integrated development environment, we have devised applications for each of these devices to document raw sensor data for analysis. Utilizing dynamic time warping, we attempt to recognize and classify actions based on differences in euclidean distances to build a strong database for further development.**

***Index Terms*—Wearable device, action recognition, action classification, dynamic time warping, time series**

1. Introduction

With a large increase of health-based technological advancements, new applications are being brought to everyday life. Step counters, heartbeat sensors are just some examples of allowing users to keep track of their health. In recent years, sensor usage has increased greatly. The size of sensors has diminished drastically whilst the capabilities are still growing. Sensors have a wide array of applications in many field of studies. Originally these sensors were used mostly in large industries for automated factory machines due to their sheer size and impracticality, but in recent years there has been a miniaturization which caused a leak into the world of mobile devices. With the increase of sensor usage, human action classification through sensor data becomes a much closer goal. With the ability to classify human actions accurately, we will be able to monitor users through their everyday actions and motions to help calculate and notify caloric intake and usage for a healthier lifestyle. Due to the popularity of wearable devices such as Apple Watches and Android Smartwatches, action classification would play integral part in this new wave of health applications. With the use of accelerometer, gyroscope and magnetometer callback that are built into our smart devices, we are able to display sensor data in the form of a three-dimensional vector for data analysis and action differentiation.

2. References

Previous work in action classification was performed by Allen Y. Yang. The Wearable Action Recognition Database[1] created to classify daily actions such as sitting, walking, running and drinking water. The database is available to the public and is an open source project. Using multiple body sensors and a centralized communication system, the Wearable Action Recognition Database(WARD) system design is able to send sensor data over a network. Allen Y. Yang used a distributed sparsity classifier to not only conserve energy usage in sensors, but also to preserve accuracy for global classification.

2.1. WARD

The wearable action database includes 20 human subjects. There are 13 males and 7 females along with 13 recognizable actions that cover daily human activities. Motion sensors are located on each arm, ankle and the waist which then connects to a centralized communication system. The database serves as a benchmark for future action recognition algorithms which follows the four principles:

- The database contains sufficient number of human subjects with large age differences.
- Action classes are general enough to cover habitual activities in daily life.
- The locations of the sensors are practical for full-fledged commercial systems
- The action data contain sufficient variation, measurement noise and outliers in air discovery and configuration of a heterogeneous network of sensor equipped motes, as well as a library of signal processing functions available on each node. It has also been designed to allow the convenient introduction of support for additional sensors and processing functionality as needed.

2.2. Wearable Sensor Network Design

The system design for creating the WARD database uses DexterNet[2]. DexterNet is a hierarchical sensor platform which has a three layer architecture: Body Network Layer(BNL), Personal Network Layer(PNL) and a Global Network Layer(GNL). Each body sensor ,which falls under the BNL, has both a triaxial accelerometer and a biaxial gyroscope. The PNL system communicates with the wireless sensors via a Tmote Sky base station connection which allows for local configurations to minimize resource usage between the PNL and GNL. Tmote Sky has a TinyOS on an 8MHz microcontroller with 10kb RAM. The sensors only trigger during events rather than a constant callback which allows for a low-resource system design. The body and personal network layer are built on the framework called SPINE(Signal Processing in Node Environment)[3]. SPINE allows for the two network layers(BNL/PNL) to interact with one another through an access point and feeds information to the Global Network Layer. The GNL is what allows for constant monitoring throughout a given location by connecting multiple PNLs to the network.

Through multiple tests, it was deemed that the sensors placed on the ankles were difficult to use for classifying certain actions. All actions performed on the ankles have a similar pattern throughout therefore the ankle sensor data is somewhat too broad for usage. The current system design requires recalibration of the accelerometers to form linear corrections. This flaw generates a certain, yet constant, amount of measurement error. Noted that some sensor output under 1g may be shifted up 15 percent in certain sensors. The gyroscopes also tend to rotate under linear motions consistently across experiments for certain sensor boards.

2.3. Related Works

Previous projects using sensors for health monitoring includes using kinematic sensors to measure elderly people living independently at home. The walking detection allowed for researchers to determine the mobility of a subject based on their everyday activity. Using custom accelerometers known as *ACTIMOMETERS*[5], P. Barralon was able to help determine the mobility of independent senior citizens. These customized accelerometers were used to measure both activity and mobility in patients.

Similarly another project used accelerometers to monitor mobility in elders[6]. Using accelerometers and a centralized connection system located on the waist, they were able to detect any sudden increase in acceleration which could indicate falling. This system design allows for the elderly to live independently with a safe system of monitoring.

Previous projects classified actions through specific algorithms. Previous techniques include uses of k-nearest-neighbors[8,9] and cluster analysis[7]. More advanced time-series algorithms for action classification include techniques such as the hidden Markov Model[10]. For our project, we will be using the time-series technique known as *dynamic time warping*.

3. System Design

For our proposed research, we used a mobile and wearable device to help us record sensor data. In our case, we used a Galaxy S6 Edge and Samsung Gear S3 Frontier as our hardware. For our wearable, we used Tizen Wearable v2.3.2 and for our Samsung Galaxy S6 Edge we used Android 7.0 Nougat. Using the latest devices allows us to accurately record data as well as have access to a wider array of sensor types.

3.1. Sensors

For our android device, we are using the accelerometer, gyroscope and magnetometer to classify human actions. For the wearable device, we are using the same three sensors with the implementation of a 4-channel audio recorder as well. Future use of the 4-channel audio recorder will be to synergize environment sounds to specific actions similar to that of body worn microphones[10]. The accelerometer helps us determine the movement of our devices whilst the gyroscope helps us determine the orientation and rotation during the action. The implementation of a magnetometer allows us to determine the orientation relative to earths magnetic field. In specific cases the magnetometer will be able to help us differentiate our subjects based on specific attributes and location.

3.2. Devices/Operating Systems

For the two devices, we had to develop separate applications using different IDEs. For the Galaxy S6 Edge, we used Android Studio to log sensor data directly onto the device while for the Galaxy S3 Frontier, we used Samsungs new IDE Tizen Studio to create a sensor log on the Tizen operating system. Unlike the previous WARD project, the data collection protocol for our devices will be offline rather than online.

3.2.1. Android Studio. Firstly, we had to create an application for our android device that could not only gather sensor data, but also create accessible logs for evaluation. For the application design, we have a simple start and stop button which initiates the sensor callback for the mobile device. The sensor refresh rate is set to the android game setting which allows us the maximum refresh rate which ranges between 37-39(ms). Each axis for the three sensors are tab separated values with proper line break after each sensor log. To minimize resource expenditure, each sensor session is saved as an appended string which is written to a text file at the very end. As the sensor is constantly changing values, each value is appended to the resulting value which is written to the file upon termination. This allows for one single action of writing to a text file rather than multiple file writes for each sensor refresh. This technique allows for the application to run in the background without accessing too much of the devices resources. The maximum string size is 2^{31} - 1. Each file row is, at max, 54 characters long:

3 tabs and each float being approximately 10 characters at max. The maximum length of each session comes out to approximately 4418 hours minimum. Since we will be conducting short sessions for analysis, saving sensor data as a singular string not only minimizes battery usage but also ensures that length will not be a long-term issue. After each session, a folder with 3 text files is created and stored inside internal memory of the device.

3.2.2. Tizen Studio. The second part of the development phase is to create a samsung-based application specifically for the Samsung Frontier. The issue arises where the Samsung Frontier uses a relatively unutilized operating system called Tizen. The IDE, Tizen Studio, is still relatively new which makes finding source code and working in a generally unused IDE very difficult. Furthermore, Android Studio and Tizen Studio follow different languages and structure. Tizen Studio runs native in C/C++ whilst Android Studio runs native in java making the Samsung Frontiers application development a much more laborious process. Following the same route as that of the android application, I had the refresh rate for the sensors set to 38 ms. The issue with the wearable device is that interaction between the android application and tizen application was unsuccessful. There were many errors and difficulties attempting to create a connection protocol between the two devices. We resorted to storing data locally and collected wearable sensor data separately from that of the mobile device. We ensured that the refresh rate of the sensor matched that of the mobile devices sensors so that the application of a moving average will be as accurate as possible.

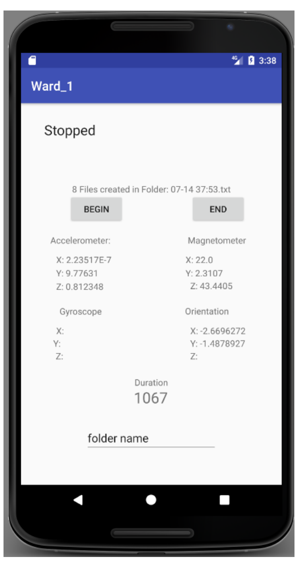


Figure 1. Application page

Figure 2. Before Execution

4. Evaluation

For the each sensor we have three variables each representing a specific dimension: X, Y and Z. Keeping the data in the form of a text file allows us to easily transition between MATLAB and Python due to each IDE having their own capabilities. Each dataset had to be cropped due to the initial gesture of executing and terminating the application. Best case scenario for graphing data would be strictly the action performed without the inclusion of starting and stopping the applications. Each subject was to hold the phone in their left hand and put the phone in their pocket before the start of each action. The orientation of the phone in the pocket was kept the same throughout each user as to maintain accuracy. To maximize the accuracy of dynamic time warping, removal of the initial and final gesture were necessary. Our current testing only utilizes the mobile device as our hardware. Our calculations and evaluations are required for only a singular device for the sake of accuracy before we implement a second variable.

4.1. Testing

For evaluation, we used a total of three subjects with different body frames to help us determine subtle differences in each action performed. Using three subjects of different sizes allows us to gather a larger range for action classification so that if we were to add new subjects, the range should hypothetically fall within the initial range of three subjects. We used a total of five exercises which were squatting, sitting, jumping jacks, push-ups and sit-ups. For each action was performed a total of 5 times for each session. After each session termination, the application returns the time each action took in milliseconds which we use as our timestamp for graphing.

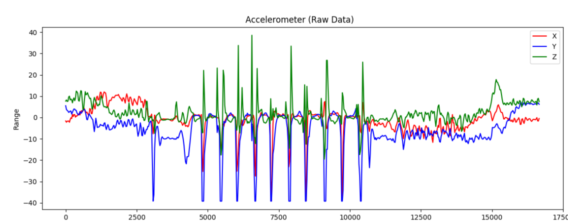


Figure 3. Accelerometer Graphed in Python

4.2. Graphing/Data Smoothing

Using pythons matplotlib API, we were able to graph the sensor data and recognize specific features for each action. Upon data analysis, the graphs for the accelerometer and magnetometer contained a large amount of spiking values. We determined that the sensor data contained too much noise and required a filter to smooth the graphs for further analysis. Different actions can cause bouncing inside the pocket between each session which will cause sensor values to oscillate depending on the sensor and action type. To compensate for the noise, we applied a moving average filter to smooth out the data.

4.2.1. Moving Average Filter. A moving average filter is also known as a low pass filter. The moving average filter accepts a certain number of input points and then returns a single output point. This technique allows us to remove large changes in nearby values to give us a smoothening effect. With the application of a moving average filter, we tend to lose valuable data along with noise. As we are able to cut down large values from the raw data, keeping track of our window size is also important. For the project, we used a default window size of 32 for all three sensors. For further testing purposes, it would help to change the window sizes based on the number of sensor values we have for each action as well the sensor type we will be using. Gyroscope values tend to change less dramatically than magnetometer values, which would require us to use a larger window frame for a gyroscope than that of a magnetometer.

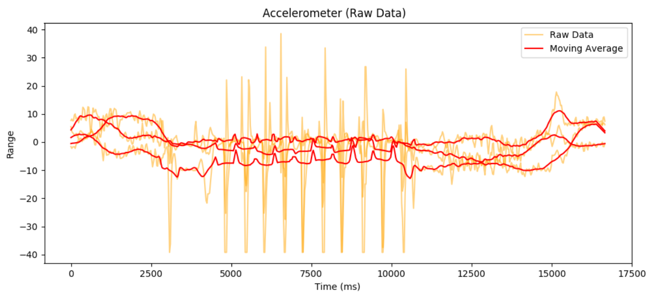


Figure 4. Accelerometer Graphed in Python (Moving Average Filter Applied)

4.3. Dynamic Time Warping

One of the key elements for our research is the time series analysis algorithm known as dynamic time warping. Considering that we are calculating sensor changes over different amounts of time, the application of dynamic time warping allows us to generate a matrix for classification. One common application is in speech recognition where different speaking speeds can be warped so that the features of the graph is matched accordingly. In our case, each subject performs an action at different speeds, but the issue is gone when dynamic time warping is applied. Using MATLABs function dtw(x,y), we are able to get a returning float which is the euclidean distance between the two graphs as shown in figure 8.

4.4. Results

Each sensors values have to be analyzed differently. For an accelerometer, each axis represents the change in movement based on that specific direction while a gyroscopes three values each represent either roll, pitch or yaw. The three axis for the accelerometer and magnetometer represents a vector value, so we apply the three values to the three dimensional euclidean distance formula(1) to acquire a single value. We classify the euclidean distance between

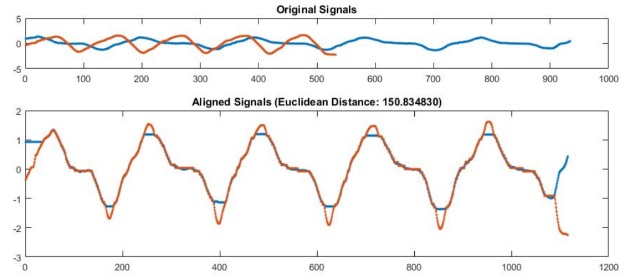
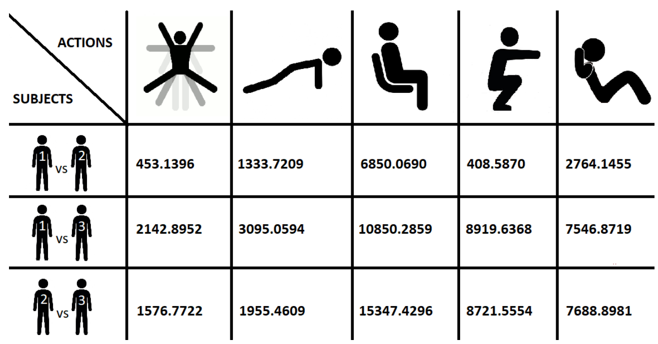


Figure 5. Dynamic Time Warping applied to gyroscope pitch graph for jumping jacks.

each sensor using a matrix. The column would represent each possible combination between subjects whilst the rows represent each action as displayed in figure 6. With this combination, we are then able to gather values for each possible combination of subjects and actions. In Table 1, we follow the same format as the table in figure 6. The values given are the gyroscopic(G) components of Yaw, Pitch and Roll as well as magnetometer (M). Each excercise is compared to one another with our subjects having the initials J, N and D. For example, the first row and first column's value in Table 1 would represent the difference in graphs of the Jumping-Jack action between a combination of all three subjects.



ACTIONS / SUBJECTS					
1 vs 2	453.1396	1333.7209	6850.0690	408.5870	2764.1455
1 vs 3	2142.8952	3095.0594	10850.2859	8919.6368	7546.8719
2 vs 3	1576.7722	1955.4609	15347.4296	8721.5554	7688.8981

Figure 6. Accelerometer value for three dimensions. Values are Euclidean distances between each subject.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \qquad (1)$$

4.5. Inclusion of New Action

After generating a reliable classification matrix with a set range between three subjects, we incorporated a new action. Jumping high knees, which involve the action of alternating knee lifts up to the subjects stomach, was performed 5 times in the session. Jumping high knees have similar characteristics to that of jumping jacks, squats and our sitting action. In terms of phone orientation, jumping high knees have the same positioning as that of squats due to the phones placement in the users pockets.

The reason we chose jumping high knees is due to the

TABLE 1. CLASSIFICATION MATRIX

(G) Pitch	JJ	PU	SS	SQ	SU
JxD	28.5152	120.4667	73.1589	65.2134	75.2715
DxN	37.0191	75.8154	304.6321	314.3645	168.0158
NxJ	47.8124	146.1982	287.8842	65.2134	147.1741
(G) Roll	JJ	PU	SS	SQ	SU
JxD	45.5764	132.1708	133.0344	204.7261	159.7833
DxN	36.0883	43.6229	304.6321	98.4889	106.9023
NxJ	61.8827	135.1469	175.6078	204.7261	119.8945
(G) Yaw	JJ	PU	SS	SQ	SU
JxD	53.5918	40.8197	194.9447	30.7168	107.6607
DxN	67.8164	65.4857	304.6321	63.7893	114.0996
NxJ	46.1869	49.1422	92.7353	30.7168	88.9827
(M) Vector	JJ	PU	SS	SQ	SU
JxD	5311.0764	12930.0587	35633.2158	4855.2263	8325.1804
DxN	8408.7165	23327.4120	56890.9173	36268.9604	24798.5105
NxJ	10798.6265	19387.0682	74616.9410	40411.0566	39255.2597

fact that the action shares many similarities between the actions we currently have whilst at the same time containing certain features that can make it distinguishable. This allows for future actions to be more easily recognized.

TABLE 2. JUMPING KNEES

(A) Vector	JJ	PU	SS	SQ	SU
JNxD	448.2305	4247.0406	5325.2449	1894.4237	8969.9414
JNxN	1687.7892	3333.4114	16961.1449	8733.7938	15963.8875
JNxJ	38.9451	2848.6439	4652.6753	1610.5577	4029.9876
(G) Pitch	JJ	PU	SS	SQ	SU
JNxD	36.9305	101.1439	215.6683	321.4366	184.8711
JNxN	51.5676	73.0009	64.8086	61.7337	76.0791
JNxJ	12.6811	172.0071	217.9381	321.4366	179.4237
(G) Roll	JJ	PU	SS	SQ	SU
JNxD	48.2583	75.5987	218.336	66.8399	70.1634
JNxN	62.7086	63.7738	78.8307	73.0001	68.8761
JNxJ	10.8158	91.9781	135.5114	66.8399	93.9976
(G) Yaw	JJ	PU	SS	SQ	SU
JNxD	74.2453	113.7198	205.3022	87.5820	112.7518
JNxN	82.1154	74.2333	102.5962	95.7560	81.5021
JNxJ	35.7603	100.1559	97.4966	87.5820	92.4320
(M) Vector	JJ	PU	SS	SQ	SU
JNxD	5651.5856	34626.2761	43302.0076	11220.0451	65434.7465
JNxN	11290.5332	25854.8222	10724.1603	47154.8415	97068.5102
JNxJ	132.3367	21145.1436	33767.3311	11099.2611	27477.8490

5. Conclusion

Inspired by the increase in sensor application to technology as well as the sudden attention to user health, using mobile and wearable device for action recognition becomes an important research. These techniques of data analysis as well as using dynamic time warping will lead us to new health applications where we may be able to monitor human caloric usage based on specific exercises with higher accuracy rather than relying only on what we currently are limited to.

The creation of a sensor logging application is the first step in classifying human actions. With the use of dynamic time warping, we are able to take two different time series and generate a euclidean distance which allows us to create a numerical range for future classification of actions as well as differentiation between current actions within our database. With the incorporation of a new foreign exercise which was previously not one of our recognized actions, we are able to show that there are recognizable differences in new actions.

Acknowledgments

We would like to thank Daniel Bautista from University of California, Davis and Nicholas Boyd from Saint. Joseph's University for help testing the application. We would also like to thank Temple University for providing us with the necessary technology to conduct this research and the National Science Foundation for giving us this opportunity. This research was supported in part by NSF grant CNS 1460971.

References

[1] Yang, Allen Y., Roozbeh Jafari, S. Shankar Sastry, and Ruzena Bajcsy. "Distributed recognition of human actions using wearable motion sensor networks." *Journal of Ambient Intelligence and Smart Environments* 1, no. 2 (2009): 103-115.

[2] Kuryloski, Philip, Annarita Giani, Roberta Giannantonio, Katherine Gilani, Raffaele Gravina, Ville-Pekka Seppa, Edmund Seto et al. "DexterNet: An open platform for heterogeneous body sensor networks and its applications." In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on* , pp. 92-97. IEEE, 2009.

[3] Aiello, Francesco, Fabio Luigi Bellifemine, Giancarlo Fortino, Stefano Galzarano, and Raffaele Gravina. "An agent-based signal processing in-node environment for real-time human activity monitoring based on wireless body sensor networks." *Engineering Applications of Artificial Intelligence* 24, no. 7 (2011): 1147-1161.

[4] P. Barralon, N. Vuillerme, and N. Noury. "Walk detection with a kinematic sensor: Frequency and wavelet comparison," in *Proceedings of the 28th IEEE EMBS Annual International Conference*, pp. 1711-1714, 2006.

[5] P. Barralon. "Accelerometric data classification and fusion in the framework of telemonitoring," Ph.D. dissertation, Joseph Fourier University, October 2005.

[6] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy. "Wearable sensors for reliable fall detection," in *Proceedings of the IEEE Engineering in Medicine and Biology Conference*, pp. 35513554, 2005.

[7] T. Huynh and B. Schiele. "Analyzing features for activity recognition," in *Proceedings of the Joint Conference on Smart Objects and Ambient Intelligence*, pp. 159163, 2005.

[8] B. Najafi, K. Aminian, A. Parschiv-Ionescu, F. Loew, C. Bla, and P. Robert. "Ambulatory system for human motion analysis using a kinematic sensor: Monitoring of daily physical activity in the elderly," *IEEE Trans. Biomedical Engineering*, vol. 50, no. 6, pp. 711-723, 2003.

[9] S. Pirttikangas, K. Fujinami, and T. Nakajima. "Feature selection and activity recognition from wearable sensors," in *Proceedings of the International Symposium on Ubiquitous Computing Systems*, 2006.

[10] P. Lukowicz, J. Ward, H. Junker, M. Stger, G. Trster, A. Atrash, and T. Starner. "Recognizing workshop activity using body worn microphones and accelerometers," in *Proceedings of the International Conference on Pervasive Computing*, pp. 1832, 2004.