User Independent Hand Gesture Recognition by Accelerated DTW

Shah Muhammed Abid Hussain and A. B. M. Harun-ur Rashid
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh

***Abstract*—Hand gesture recognition is becoming increasingly popular for applications in ubiquitous computing environment. Accelerometer based methods have proven themselves to be competitive in terms of both portability and recognition accuracy. But there are only a few algorithms which have achieved moderate accuracies in user independent gesture recognition. A novel accelerometer based user independent hand gesture recognition algorithm is proposed in this paper. For validation of accuracy the test was run over 3200 samples collected from 5 users over 5 days. The overall accuracy was found to be 96.4% for user independent gesture recognition. A high recognition accuracy and possibility of simpler implementation also gives this algorithm an upper hand among competitive methods published in recent literatures. A hardware-accelerated dynamic time warping (DTW) implementation is also proposed to pave the way towards continuous gesture recognition for DTW based methods.**

Keywords-gesture recognition; dynamic time warping; user-independent gesture; accelerometer; hardware acceleration

I. INTRODUCTION

Spatial hand movement based human computer interaction (HCI) has become a key component of context aware pervasive systems. The recent trend of integrating a large number of sensors into embedded systems is one of the main driving factors of its popularity. The spatial hand movement or in our case - 'gesture' is also a natural and convenient way of communication among humans. Thus sensing spontaneous hand gestures is of utmost importance in ubiquitous computing for enhancing the quality of life.

However, there are multiple technical challenges [1] involved in personalized gesture based interaction, including the need of expensive overheads. Hence, computer vision [2-3] and "glove" [4-5] based solutions are not practicable especially for portable and "out of line-of-sight" applications. The inclusion of accelerometers in contemporary smart phones and tablet computers has made accelerometer an appropriate candidate for sensing gestures.

Recently, accelerometer based gesture recognition has been discussed in many publications [1][6-13]. Most of these literatures involve statistical methods [6-11] and frequency domain approaches hence resulting in floating point calculations. Two recent publications on hand gesture recognition are uWave [1] and FDSVM [6]. uWave [1] requires only one gesture to start with and employs feedback dependent template adaptation. But its user independent gesture recognition rate is quite low to be practically useful. FDSVM [6] achieves a higher accuracy but like other statistical methods require a large number of training samples. Furthermore, the affinity propagation (AP) and compressive sensing (CS) algorithm combined with dynamic time warping [15] which is proposed in [14] outperforms FDSVM [6] and achieves a very high accuracy in both user dependent and independent gesture recognition. But this algorithm increases complexity which translates into a high computational time.

Dynamic time warping (DTW) [15] based methods [1, 12-14] can alleviate the need of floating point calculations while still providing a high level of accuracy. But for continuous gesture (continuous sensing for valid gestures) recognition applications DTW suffers; mainly because it requires high computational power due to its complexity. As the software based solutions are not providing the required acceleration, we have to move towards hardware based solution for DTW implementations [16].

In this work, we propose an accelerometer based gesture recognition system that uses a single 3-axis accelerometer to sense gestures. The core of our work is the DTW algorithm which finds the minimum cost path between two time sequences. We have developed our algorithm based on uWave's [1], but have applied few changes. These modifications improve accuracy more than 5% for user dependent case. Moreover, our algorithm can be implemented in user independent gesture recognition with similar high level of accuracy. Thus our algorithm achieves an overall accuracy of 99.2% for user independent case and 96.4% for user dependent gesture recognition. A vocabulary of 8 gestures identified by Nokia research [17] is adopted for testing our recognition algorithm. We collected gesture datasets from 5 individuals over 5 days and created a gesture database of 3200 gesture samples with 16 samples per gesture per individual per day. This work is an attempt towards development of a continuous gesture recognition system hence all of its components have been optimized (e.g. fixed number of samples for gestures) towards serving that purpose. We also propose hardware based solutions for DTW acceleration. While this acceleration is not essential for isolated gesture recognition, it is of utmost importance in the case of continuous gesture recognition. For continuous case, the system must perform a large number of DTW calculations greater than or equal to the data output rate of the accelerometer. Hence this

978-1-4673-1154-0/12/$31.00 ©2012 IEEE

work will pave the way towards the implementation of continuous gesture recognition system.

The rest of the paper is organized as follows: Section II describes the data collection. Section III presents the technical details of the algorithm and our gesture recognition system. Section IV presents the prototype implementations. Section V discusses the results. Finally, section VI concludes the paper.

II. Data Collection

We next present our used equipment, gesture vocabulary and data collection process.

A. Input Device and Settings

We have collected our gesture datasets by using the Altera DE-0 Nano FPGA board which incorporates Analog Devices ADXL345 [18], a small, thin, ultralow power, 3-axis accelerometer with high resolution (13-bit) measurement up to ± 16 gravitational acceleration units (g). Fig. 1 shows the board form factor and the accelerometer along with its axes.

The accelerometer was operated in 3-wire SPI mode interfaced with Cyclone IV FPGA, programmed in Verilog HDL. As our ultimate aim in accelerometer based hand gesture recognition is to develop a continuous gesture recognition system, we decided to take fixed time lengths of samples as exhibited by [16]. We set the output data rate of accelerometer at 25 Hz. We found this rate to be optimal for gesture recognition as it results in filtering out most of the unwanted hand shake acceleration and noise. It also alleviates the necessity of smoothing which is a component of uWave [1]. But it is still high enough a rate to capture the important features of gestures. We took 85 samples per gesture on each of the axes which gives us a maximum gesture input time of 3.40 seconds. From few trials, we found out that it is almost impossible to make a hand movement involving accelerations beyond 3g. So we set the accelerometer resolution at ± 4g with 10-bit fixed resolution. Thus the output values received from the accelerometer ranged from -512 to +511 with the value +128 representing 1g on the positive axis.

B. Nokia Gesture Vocabulary

We adopted the Nokia gesture vocabulary which has also been used in demonstrating the accuracy of accelerometer based gesture recognition literatures [1] [6-14]. It consists of 8 gestures, 4 of which are 2D gestures and 4 others are 1D gestures. Since our input device, DE-0 Nano Board is of cuboid shape, the user gets an overall perception of its axes which are also aligned with the axes of accelerometer embedded in it. Fig. 2 shows these gesture paths and their implementation axes.

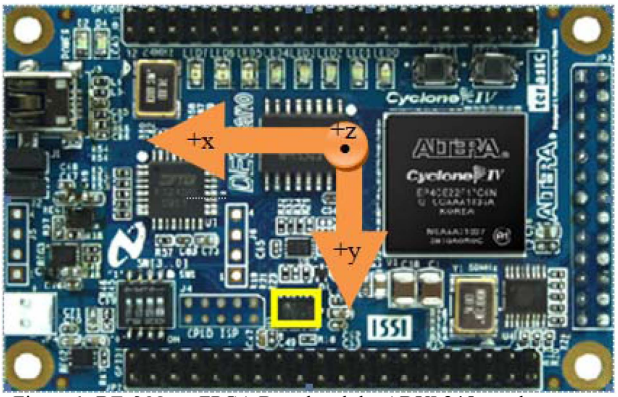

Figure 1. DE-0 Nano FPGA Board and the ADXL345 accelerometer (highlighted) along with its axes

C. Data Collection Process

The users had to perform a gesture from vocabulary right after pressing a start switch. The accelerometer samples the gestures for 3.40 seconds. But the gestures of the vocabulary involve gestures of various temporal lengths and practically, the same gesture takes different periods on several performances. Thus, the users had to hold the device motionless after performing the gesture until the 3.40 seconds window was closed. The users were allowed to perform gestures at different tilts given that they maintain a constant tilt throughout the gesture length i.e. 3.40 seconds. They were also advised to perform gestures at the designated axis/axes with minimal movement in the remaining axes/axis respectively.

III. Gesture Recognition System

In this section we present the technical components of our algorithm: tilt independence and dynamic time warping. We also discuss the necessity of decoupling the axes and the test procedure that we followed.

A. Tilt Independence

The time series data received from the accelerometer consists of the acceleration data of the user as well as the tilt data. Since we have assumed that tilt is arbitrary yet constant in a gesture, we can discard the tilt information in our received data just by averaging all the axis readings separately and negate the average value from the corresponding accelerometer data. This gives us approximately pure acceleration data of user.

B. Dynamic Time Warping

Dynamic time warping (DTW) is a well-known algorithm in many areas including handwriting and online signature matching, sign language recognition, gesture recognition, data mining and time series clustering, computer vision and computer animation [15]. DTW algorithm has earned its popularity by being extremely efficient at the time-series similarity measure which minimizes the effects of shifting and distortion in time by allowing “elastic” transformation of time series in order to detect similar shapes with different phases [15]. DTW uses dynamic programming to find the minimum

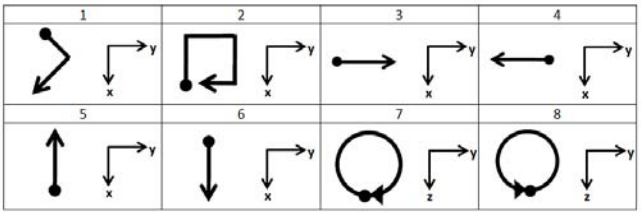

Figure 2. Gesture vocabulary adopted from [17] along with implementation axes. The dot denotes the start and the arrow the end

cost path between two time sequences. The original DTW algorithm can be summarized as follows:

Assume we have two time sequences A and B, of length n and m respectively, where $A = \{a_1, a_2, ..., a_n\}$ and $B = \{b_1, b_2, ..., b_m\}$. We first define a distance $d(i, j)$ as the Euclidian distance between the samples a_i and b_j. Then the accumulated cost $p(i, j)$ can be expressed as $p(i, j) = d(i, j) + min\{p(i-1, j), p(i-1, j), p(i-1, j-1)\}$ where $i, j \geq 1$ and also $p(1, 1) = 0$, $p(i, 1) = d(i, 1) + p(i-1, 1)$ & $p(1, j) = d(1, j) + p(1, j-1)$. Consequently, the overall minimum cost of aligning A and B is $p(n, m)$.

This algorithm satisfies the boundary conditions: $p_{initial\ (1)} = p(1, 1)$ and $p_{final\ (k)} = p(n, m)$, monotonicity conditions: $i_{s-1} \leq i_s$, $j_{s-1} \leq j_s$ and step size condition: $i_s - i_{s-1} \geq 1$, $j_s - j_{s-1} \geq 1$ where $p_s = (i_s, j_s)$. There have been proposed few modified or constrained versions of DTW to speed it up but most of them do so only at the expense of accuracy. Others are not suitable for development of a continuous gesture recognition system. DTW algorithm implementation can be thought of a matrix construction and is illustrated in Fig. 3.

C. Decoupling the axes

We performed DTW calculations on all the 3 axes of a particular gesture sample independently. We took the total cost to be the aggregate of the costs of all 3 axes. The advantage of decoupling of the axes can be realized from the Fig. 4. In this figure we have shown two gesture samples performed by two different users which are A and B respectively. Both users have performed the 2nd gesture from the vocabulary which involves x and y axes of the accelerometer. DTW will match x-axis and y-axis individually of two gesture samples by expanding (hence shifting) the sample B or by compressing the sample A if it is run on each of the axis separately. But if the DTW is run only once by taking the distance function involving all the axes as in uWave [1], then it yields a significant matching cost (mismatch). This is because it cannot take into account the variation of temporal speeds of between the two axes. Clearly, x-axis has to be shifted less to be matched, whereas y-axis has to be shifted more to be matched. This is only possible if the axes are decoupled. Thus decoupling results in a better matching.

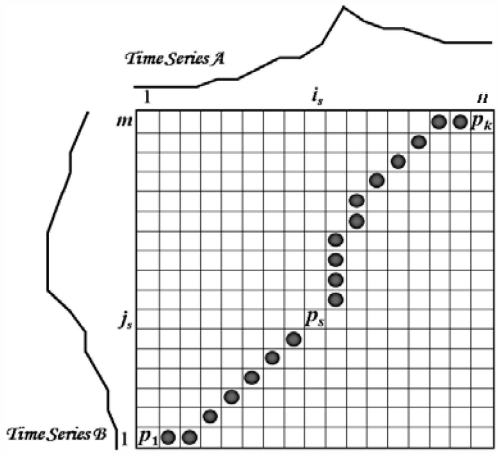

Figure 3. DTW of two time sequences

D. Test Procedure

Although DTW works very well in matching time sequences of different durations, we have limited its use in matching sequences of same duration of length $n = 85$, for our own interest stated in section II A. Thus time and space complexity of the DTW algorithm are both $O(n^2 = 7225)$. In training mode, 16 samples per gesture type are taken. The average values of all 3 axes were negated from their corresponding actual values. After that an internal DTW calculation was performed between the same gesture types. The lowest cost yielding gesture sample for each type was selected to represent that gesture type in the template. This way, the gestures in the template was "idealized". In test mode, the input gesture was negated from the average values on 3 axes in a similar manner and 8 DTW calculations were performed taking the test sample and 1 gesture sample from template each time. From experience, we found out that the confusion between gesture 7 and gesture 8 is the major factor for decreased accuracy in gesture recognitions that's why we advised our users to make these gesture paths slightly parabolic with the major axis being the y-axis. Its reason was to weight a higher cost at the y-axis which is the primary differentiator between these two gestures.

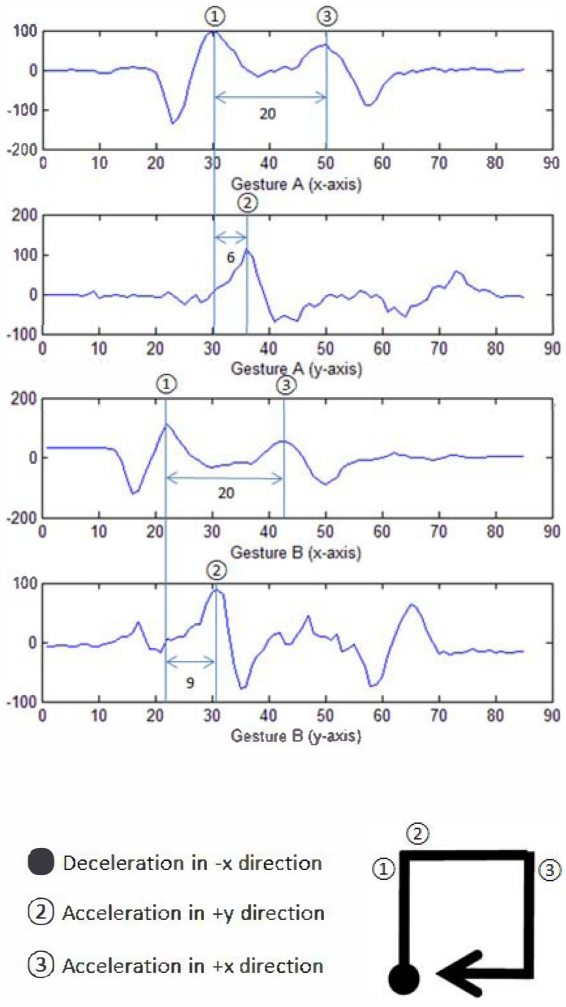

Figure 4. Analysis of a two-axis gesture given by two different users (A and B)

IV. PROTOTYPE IMPLEMENTATION

We evaluated our algorithm in MATLAB on a Core i5 based PC running at 2.26 GHz. Our aim was to compare DTW based gesture recognition at different systems. So we implemented this same algorithm in the Altera DE0-Nano [19] Board by instantiating Nios II/e [20] soft processor and hence by building an SOPC. As there was no floating point calculation involved, we achieved exactly the same result. To get the flavor of hardware accelerated acceleration in the same DE0-Nano [19] Board, we also implemented a hardware DTW module. We collected the time required to perform just the DTW calculations in the hardware as this is the only part of the system to be bottlenecking the system's speed. The timing analysis was done by the built in SignalTap II [21] Logic Analyzer.

V. RESULTS

Our gesture recognition algorithm has been tested on 3200 samples which were collected from 5 individuals over 5 days. Each user has his/her 640 gesture samples in our database. 1 gesture from every type was taken arbitrarily from database for a fixed user in training mode and the test mode was run on the other 632 samples for the same user. Thus the overall accuracy of our gesture recognition system over 3200 samples is achieved to be 99.2% for user dependent case. For, user independent case, the template was constructed from a particular user using the training method described in III C. Then the test procedure was run on all the gestures for the all other users leaving that particular user out. In this case the overall accuracy was achieved to be 96.4%. The confusion matrices for user dependent and user independent gesture recognition are shown in Fig. 5 and Fig. 6 respectively.

Next, we compare the DTW calculation timings obtained from implementations in MATLAB, prototypes in DE0-Nano [19] SOPC [22] without hardware acceleration and with hardware acceleration in Table I. We observe that a 12 fold improvement in DTW calculation time for hardware accelerated case. These recorded times are the time required to perform 24 DTW calculations (3 calculations per axis for 8 gestures).

	100	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	100	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	100	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	100	0.0	0.0	1.03	0.0
	0.0	0.0	0.0	0.0	100	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	100	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	95.87	2.29
	0.0	0.0	0.0	0.0	0.0	0.0	3.10	97.71

Figure 5. Confusion Matrix for user dependent case for 3200 samples. Columns are recognized gestures and rows are the actual identities of input gestures. (Average accuracy of 99.2%)

	98.55	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	92.26	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	2.25	100	0.0	0.0	0.0	0.0	0.41
	0.04	4.87	0.0	100	0.0	0.0	2.71	0.0
	0.0	0.33	0.0	0.0	100	0.0	0.0	0.0
	1.41	0.29	0.0	0.0	0.0	100	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	87.38	6.58
	0.0	0.0	0.0	0.0	0.0	0.0	9.91	93.01

Figure 6. Confusion Matrix for user independent case for 3200 samples. Columns are recognized gestures and rows are the actual identities of input gestures. (Average accuracy of 96.4%)

The achieved accuracies for both user-dependent and user-independent gesture recognition by different popular algorithms are presented in tabular form in Table II. The gesture recognition algorithm we presented yields a higher accuracy but there are still scopes for further improvement. From the confusion matrix we observe that the two gestures – 7th & 8th are the hardest to recognize correctly. This was due to the fact that they have a common axis (z-axis) of similar gesture movement. Their y-axis components may not also yield a high enough cost to differentiate between them, if the acceleration values are small compared to the z-axis. These two gestures can be modified to have a parabolic shape in which the major axis will be the y-axis. This has been proven to improve the accuracy, and our algorithm achieves a near perfect recognition rate. While our gesture recognition algorithm could have had benefits due to the DE0-Nano [19] board ergonomics, fixed gesture length or the use of a high accuracy accelerometer with low noise, a detailed study is required in order to quantify them.

TABLE I. COMPARISON OF SPEED FOR DIFFERENT IMPLEMENTATIONS IN SECONDS

MATLAB	SOPC without hardware acceleration	SOPC with hardware acceleration
0.3312	0.7056	0.0583

TABLE II. RECOGNITION ACCURACY FOR FEW ALGORITHMS

Algorithm	User Dependent	User Independent
uWave[a] [1]	93.5%	75.4%
FDSVM[b] [6]	95.21%	89.29%
DTW with AP & CS [14]	99.79%	96%
Proposed	99.2%	96.4%

a. Without template adaptation
b. For 12 gesture vocabulary

VI. CONCLUSION

We have proposed a novel accelerometer based gesture recognition system that achieves an accuracy of 99.2% for user dependent case although by using more training sets than used by [1] & [6]. But for the user independent case we have achieved an accuracy of 96.4% in which case large number of training set is highly practicable. We modified the uWave [1] algorithm to make it useful in independent gesture recognition and also applied an intelligent template construction from 16 sample gestures. We also issued some guidelines to the users for data collection which helped to improve the accuracy. Nevertheless, this accuracy is achieved at the cost of increased DTW calculations, which is about 3 times compared to that of uWave [1].

Although thousand fold improvements in DTW calculation speed is shown in [16], those were only achieved when we have larger time sequence or "query length". Hence, the 12 fold improvement by our hardware acceleration is reasonable, but not sufficient for implementing a continuous gesture recognition system consisting of an 8 gesture vocabulary at our desired data date. Further improvement in DTW acceleration can be achieved with increased overheads (e.g. memory, logic etc.) or with optimization of implementation architecture.

The vocabulary we used did not include any 3D gesture. However, as our proposed algorithm is a generalized one, it can be applied for recognizing the 3D gestures also.

Our achieved user independent gesture recognition accuracy is similar to that achieved by [14] but without the need of complex manipulations such as compressive sensing and affinity propagation hence reducing corresponding computational cost. Thus we believe our work is a major step towards continuous and user independent gesture recognition in improving the quality of life for disabled people and in general pervasive computing. Moreover, we recognize one of the major factors that limit mobile and embedded device scaling: the user–computer interface. While gesture based recognition can leverage this problem, there are various exciting applications awaiting this field of ubiquitous computing some of which are mentioned in [1].

Our future work will be focused on evaluating our proposed system against a larger gesture vocabulary, on improving architecture, on implementing a continuous gesture recognition platform and also on finding new opportunities to use accelerometer based gesture for human computer interaction.

ACKNOWLEDGMENT

The authors would like to thank the participants in the user studies and Terasic for their timely support for DE-0 Nano on numerous occasions.

REFERENCES

[1] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya and V. Vasuvedan, "uWave: Accelerometer-based personalized gesture recognition and its applications," in IEEE PerCom, 2009.

[2] Microsoft Kinect: en.wikipedia.org/wiki/kinect

[3] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, "A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory," in ICPR, 2008.

[4] M. Martin, J. Maycock, F. P. Schmidt and O. Kramer, "Recognition of manual actions using vector quantization and dynamic time warping," in Int. Conf. on HAIS, 2010.

[5] J. K. Perngm B. Fisher, S. Hollar, and K. S. J. Pister, "Acceleration sensing glove (ASG)," in Digest of Papers for Int. Symp. Wearable Computers, 1999, pp. 178-180.

[6] J. Wu, G. Pan, D. Zhang, Guande Qi and Shijian Li, "Gesture recognition with a 3-D Accelerometer," in Ubiquitous Intelligence and Computing, Springer-Verlag, 2009.

[7] T. Schlömer, B. Poppinga, N. Henze and S. Boll, "Gesture recognition with a Wii controller", in Int. Conf. on TEI, 2008.

[8] J. Mäntyjärvi, J. Kela, P. Korpipää, and S. Kallio, "Enabling fast and effortless customisation in accelerometer based gesture interaction," in Proc. Int. Conf. Mobile and Ubiquitous Multimedia. College Park, MA: ACM, 2004.

[9] V. M. Mäntylä, "Discrete Hidden Markov Models with application to isolated user-dependent hand gesture recognition, VTT publications, 2001.

[10] F. G. Hofmann, P. Heyer, and G. Hommel, "Velocity profile based recognition of dynamic gestures with discrete hidden markov models," in LNCS, vol. 1371, I. Wachsmuth and M. Fröhlich, Eds. Springer, 1997, pp. 81-95

[11] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca, "Accelerometer-based gesture control for a design environment," Personal Ubiquitous Computing, vol. 10, pp. 285-299, 2006.

[12] G. Niezen and G. P. Hancke, "Gesture recognition as ubiquitous input for mobile phones," in Int. Workshop on DAP, 2008 conjuction with Ubicomp, 2008.

[13] D. H. Wilson and A. Wilson, "Gesture recognition using the XWand," Technical Report CMU-RI-TR-04-57, CMU Robotics Institute, 2004.

[14] A. Akl and S. Valaee, "Accelerometer-based gesture Recognition via dynamic-time warping, affinity propagation, & compressive senesing," in ICASSP, 2010.

[15] P. Senin, "Dynamic time warping algorithm review", Information and Computer Science Department, University of Hawaii at Manoa, 2008.

[16] D. Sart, A. Mueen, and Walid Najjar, "Accelerating dynamic time warping subsequence search with GPUs and FPGAs," in ICDM, 2010.

[17] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," in IEEE ASSP Magazine, 1986, pp. 4-15.

[18] Analog Devices: ADXL345 Datasheet: http://www.analog.com/en/mems-sensors/inertial-sensors/adxl345/products/product.html

[19] Terasic Technologies: DE0-Nano Development Board: http://www.altera.com/b/de0-nano-dev-board.html

[20] Altera Corporation: Nios II Processor: http://www.altera.com/literature/lit-nio2.jsp

[21] Altera Corporation: Design debugging with Signal Tap II Logic Analyzer: www.altera.com/literature/hb/qts/qts_qii53009.pdf

[22] Altera Corporation: SOPC builder handbook: http://www.altera.com/literature/hb/qts/qts_qii54001.pdf