Journal of ICT Research and Applications

Journal of ICT Research and Applications welcomes full research articles in the area of Information and Communication Technology from the following subject areas: Theory, Signal Processing, Electronics, Computer Network, Telecommunication, Wireless & Mobile Computing, Internet Technology, Multimedia, Software Engineering, Computer Science, Information System, and Knowledge Management. Authors are invited to submit articles that have not been published previously and are not under consideration elsewhere.

Abstracts and articles published on *Journal of ICT Research and Applications* are available online at http://journal.itb.ac.id and indexed by Scopus, Google Scholar, DOAJ, EBSCO Open Science Directory, Electronic Library University of Regensburg, Open J-Gate, New Jour, International Association for Media and Communication Research (IAMCR), Cabell's Directories, Zurich Open Repository and Archive Journal Database, and ISJD-Indonesian Institute of Sciences.

Publication History

Formerly known as:

- ITB Journal of Information and Communication Technology (2007 – 2012)

Editorial board of Journal of ICT Research and Applications

Publication assistant: Dini Sofiani Permatasari & Yuliah Qotimah. Address: *ITB Journal Publisher*, LPPM ITB, Ged. Rektorat ITB Lt. 5, Jalan Tamansari 64, Bandung 40116, Indonesia, Tel. (022) 2501759 ext. 115, Fax.: (022) 2504010; e-mail: journal@lppm.itb.ac.id.

Distribution: Distributed to other universities and research institutions for exchange publications and to regular subscribers.

Subscription per issue: Rp 100.000,- (domestic) and $50 (overseas). Order form for subscription should be sent to the editorial office.

Reg. No. 691-SIC-UPPGT-SIT-1963, Accreditation No. 56/DIKTI/Kep/2012 (Juli 2012- Juli 2017)

Published by *ITB Journal Publisher*, the Institute for Research and Community Services, Institut Teknologi Bandung, in collaboration with Indonesian Engineering Association (Persatuan Insinyur Indonesia - PII).

Cover Design by Chandra Tresnadi.

Table of Contents

CSLM: Levenberg Marquardt based Back Propagation Algorithm Optimized with Cuckoo Search

Nazri Mohd. Nawi, Abdullah Khan & M. Z. Rehman

Software and Multimedia Centre, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM).
P.O. Box 101, 86400 Parit Raja, BatuPahat, Johor DarulTakzim, Malaysia
Email: hi100010@siwa.uthm.edu.my

Abstract. Training an artificial neural network is an optimization task, since it is desired to find optimal weight sets for a neural network during training process. Traditional training algorithms such as back propagation have some drawbacks such as getting stuck in local minima and slow speed of convergence. This study combines the best features of two algorithms; i.e. Levenberg Marquardt back propagation (LMBP) and Cuckoo Search (CS) for improving the convergence speed of artificial neural networks (ANN) training. The proposed CSLM algorithm is trained on XOR and OR datasets. The experimental results show that the proposed CSLM algorithm has better performance than other similar hybrid variants used in this study.

Keywords: *artificial neural networks; back propagation; cuckoo search; levenberg marquardt; local minima.*

1 Introduction

An Artificial Neural Network (ANN) is a data processing model that is based on the biological nervous systems of a human brain [1]-[2]. The main constituent of this representation is the new foundation of the data processing system. It consists of a large number of tremendously interrelated processing elements known as neurons, all functioning together in order to solving many complex real world problems [3]. ANN have been effectively implemented in all engineering fields such as biological modeling, decision and control, health and medicine, engineering and manufacturing, marketing, ocean exploration and so on [4]-[9]. Because of the delightful appearance of artificial neural networks, a large number of applications have been proposed in recent decades. The back propagation (BP) algorithm that was introduced by Rumelhart [10] is the well-known method for training a multilayer feed-forward artificial neural networks [11]. However, the BP algorithm suffers from two major drawbacks: low convergence rate and instability. They are caused by a possibility of being trapped in a local minimum and prospect of overshooting the minimum of the error surface [12]-[14].

Received September 23rd, 2013, 1st Revision October 1st, 2013, 2nd Revision November 4th, 2013, Accepted for publication November 13th, 2013.

In recent years, many improved learning algorithms have been proposed to overcome the flaws of gradient descent based systems. These algorithms include a direct enhancement method using a poly tope algorithm [14], a global search procedure such as evolutionary programming [15], and genetic algorithm (GA) [16]. The standard gradient-descent BP is not path driven, but population driven. However, the improved learning algorithms have explorative search topographies. Therefore, these approaches are expected to evade local minima often by promoting exploration of the search space. The Stuttgart Neural Network Simulator (SNNS) [17], which was developed in the recent past use many different algorithms including Error Back Propagation [13], Quick prop algorithm [18], Resilient Error Back Propagation [19], Back percolation, Delta-bar-Delta, Cascade Correlation [20], *etc*. All these algorithms are derivatives of steepest gradient search, so ANN training is relatively slow. For fast and efficient training, second order learning algorithms have to be used. The most effective method is Levenberg Marquardt (LM) algorithm [21], which is a derivative of the Newton method. This is quite multifaceted algorithm since not only the gradient but also the Jacobian matrix should be calculated. The LM algorithm was developed only for layer-by-layer ANN topology, which is far from optimal [22]. LM algorithm is ranked as one of the most efficient training algorithms for small and medium sized patterns. LM algorithm is coined as one of the most successful algorithm in increasing the convergence speed of the ANN with MLP architectures [23]. It is a good combination of Newton's method and steepest descent [24]. It Inherits speed from Newton method but it also has the convergence capability of steepest descent method. It suits specially in training neural network in which the performance index is calculated in Mean Squared Error (MSE) [25] but still it is not able to avoid local minimum [26]-[27].

In order to overcome these issues this study proposed a new algorithm that combines Cuckoo Search (CS) and Levenberg Marquardt Back propagation (LMBP) algorithms to train neural network for XOR and OR datasets. The proposed CSLM algorithm reduces the error and improved the performance by escaping from local minima.

The remaining of the paper is organized as follows. Section two describes the Cuckoo Search algorithm. In Section three, the implementation of the proposed CSLM algorithm is elaborated. In Section Four, the performance of the proposed CSLM on some experimental data is discussed. The paper is finally concluded in Section Five.

2 The Cuckoo Search Algorithm

Xin-She Yang [28] proposed a metaheuristic Cuckoo Search (CS) algorithm based on the forceful parasitic behavior of some cuckoo species by laying their eggs in the nests of other bird species. Sometimes the host bird cannot differentiate between its own and cuckoo eggs. But, if an egg is discovered by the host bird as not its own, then it either throw these unknown eggs away or simply leave its nest. Some species in cuckoo are very specialized in the impersonating the color and pattern of the eggs of the host species. This reduces the chances of their eggs being abandoned and thus increases the chances of their survival. The CS algorithm follows the three basic rules:

1. Each cuckoo lays one egg at a time, and put its egg in randomly chosen nest;
2. The best nests with high quality of eggs will carry over to the next generations;
3. The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability pa [0, 1].

In this case, the host species can either throw the egg away or build a completely new nest somewhere else. The last assumption can be approximated by the fraction *Pa* of the *n* nests that are replaced by new nests (with new random solutions). For a maximization problem, the quality or fitness of a solution can simply be proportional to the value of the objective function. In this algorithm, each egg in a nest represents a solution, and a cuckoo egg represents a new solution, the aim is to use the new and potentially better solutions (cuckoos) to replace a not so good solution in the nests. Based on these three rules, the basic steps of the Cuckoo Search (CS) can be summarized as the following pseudo code:

```
Step 1: Generate initial population of N host nest $i = 1...N$
Step 2: While ($f\min < MaxGeneration$) or (stop criterion)
Step 3: Do Get a Cuckoo randomly by Levy flights and evaluate its fitness $F_i$.
Step 4: Choose randomly a nest $j$ among $N$.
Step 5: If $F_i > F_j$ Then
Step 6: Switch $j$ by the new solution, End If
Step 7: A segment ($p_a$) of worse nest are abandoned and new ones are built.
Step 8: Keep the optimal solutions (or nest with quality solutions).
Step 9: Rank the solutions and find the current best.
Step 10: end while
```

When creating new solutions x^{t+1} for a cuckoo i, a Levy flight is performed

$$x^{t+1} = x_i^{\,t} + \alpha \oplus levy(\lambda) \tag{1}$$

where $\alpha > 0$; is the step size which should be related to the scales of the problem of interest. The product $\oplus$ means entry wise multiplications. The random walk via Levy flight is more effective in exploring the search space as its step length is much longer in the long run. The Levy flight essentially provides a random walk while the random step length is drawn from a Levy distribution.

$$Levy \sim u = t^{-\lambda},\ 1 < \lambda \leq 3 \tag{2}$$

This has an infinite variance with an infinite mean. Here the steps essentially construct a random walk process with a power-law step-length distribution with a heavy tail. Some of the new solutions should be generated by Levy walk around the best solution obtained so far, this will speed up the local search. However, a substantial fraction of the new solutions should be generated by far field randomization whose locations should be far enough from the current best solution. This will make sure the system will not be trapped in a local minimum.

3 The Proposed CSLM Algorithm

The proposed method known as Cuckoo Search based Levenberg-Marquardt (CSLM) algorithm is given in Figure 1. Cuckoo Search (CS) is a metaheuristic algorithm that starts with a random initial population. It works with three basic rules i.e. selection of the best source by keeping the best nests or solutions, replacement of host eggs with respect to the quality of the new solutions or cuckoo eggs produced based randomization via Levy flights globally (exploration) and discovering of some cuckoo eggs by the host birds and replacing according to the quality of the local random walks (exploitation) [29]. In the figure, each cycle of the search consists of several steps initialization of the best nest or solution, the number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability, p_a [0, 1].

In this algorithm, each best nest or solution represents a best possible solution (i.e., the weight space and the corresponding biases for NN optimization in this study) to the considered problem and the size of a food source represents the quality of the solution. The initialization of weights was compared with output and the best weight cycle was selected by cuckoo. The cuckoo would continue searching until the last cycle to find the best weights for networks. The solution that was neglected by the cuckoo was replaced with a new best nest. The main idea of this combined algorithm is that CS is used at the beginning stage of searching for the optimum. Then, the training process is continued with the LM algorithm. The LM algorithm incorporates the Newton method and gradient descent method. The flow diagram of CSLM is shown in Figure 1. In the first stage CS algorithm finish its training then LM algorithm starts training with the weights generated by CS algorithm and the LM train the network until the stopped condition is satisfied.

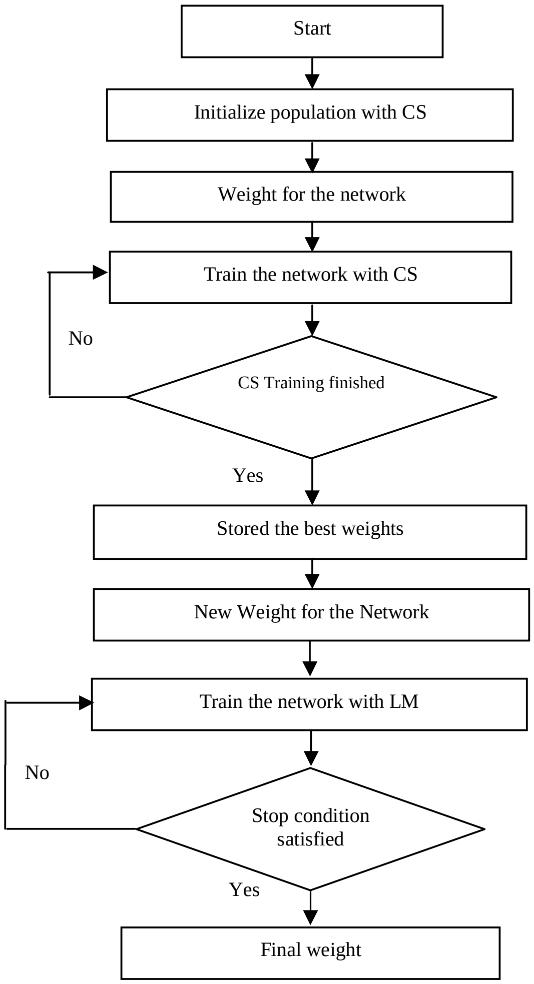


Figure 1 The Proposed CSLM Algorithm.

4 Results and Discussions

4.1 Preliminaries

In order to illustrate the performance of the proposed algorithm, CSLM is trained on 2-bit, 3-bit XOR and 4-bit OR datasets. The simulation experiments were performed on a 1.66 GHz AMD E-450 APU with Radeon and 2 GB RAM using MATLAB 2009b software. The proposed CSLM algorithm is compared with Artificial Bee Colony Levenberg Marquardt algorithm (ABC-LM), Artificial Bee Colony Back Propagation (ABC-BP) algorithm and simple back propagation neural network (BPNN) based on MSE and maximum epochs was set to 1000. The three layer feed forward neural network are used for each problem; i.e. input layer, one hidden layer, and output layers. The number of hidden nodes is formed of five neurons. In the network structure the bias nodes are also used and the log sigmoid activation function is placed as the activation function for the hidden and output layers nodes. For each algorithm, 20 trials are repeated.

4.2 Two Bit Exclusive-OR Problem

The first test problem is the exclusive OR (XOR) which is a Boolean function of two binary input to a single binary output. In the simulation we used 2-5-1, feed forward neural network structure for two bit XOR problem. For all algorithms, the parameters range is set to [5,-5], [5,-5], [5,-5], [1,-1] respectfully. Table 1 shows the CPU time, number of epochs, accuracy, SD and the MSE for the 2 bit XOR test problem with five hidden neurons. Figure 2 shows the MSE of the proposed CSLM algorithm and ABC-LM algorithm for the 2-5-1 network structure. From Table 1, it can be clearly seen that the proposed CSLM algorithm converged to the global minima within 126 epochs, while ABC-LM needs much time and number of epochs to converge. Figure 3 illustrates the average accuracy for the CSLM, ABC-BP, ABC-LM, and conventional BPNN algorithms.

Table 1 CPU time, Epochs and MSE for 2-5-1 NN architecture.

Algorithm	BPNN	ABC-BP	ABC-LM	CSLM
CPUTIME	42.643	172.33	123.95	14.41
EPOCHS	1000	1000	1000	126
MSE	0.220	2.39E-04	0.125	0
SD	0.0105	6.7E-05	1.5E-06	0

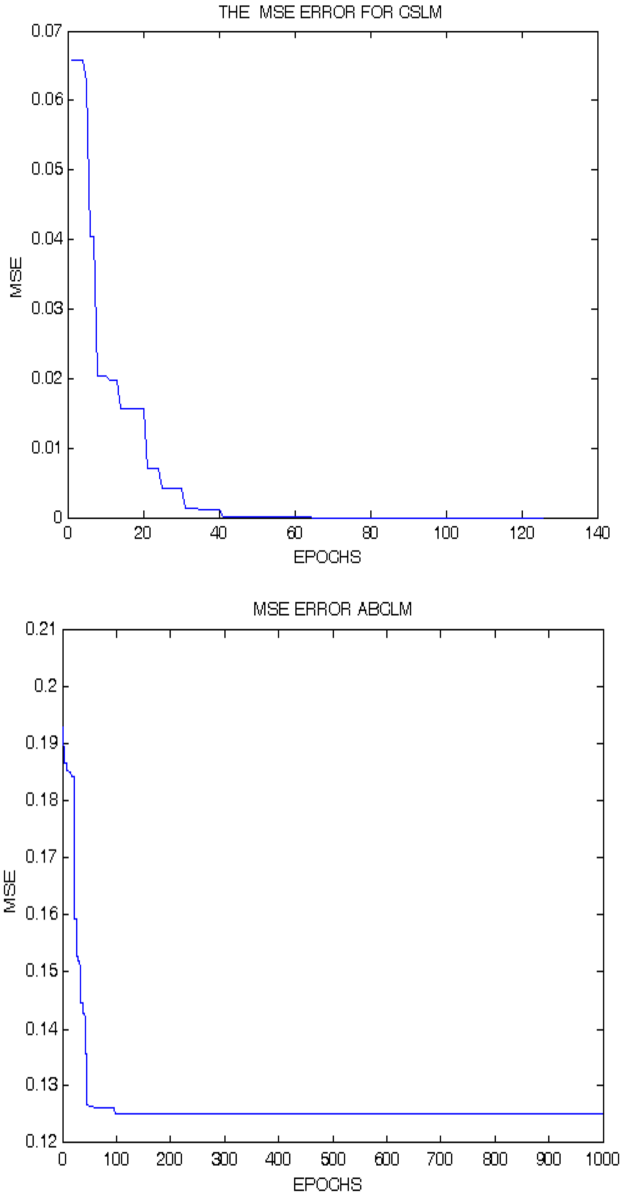


Figure 2 MSE for (a) CSLM and (b) ABC-LM on 2 Bit XOR and 2-5-1 NN architecture.

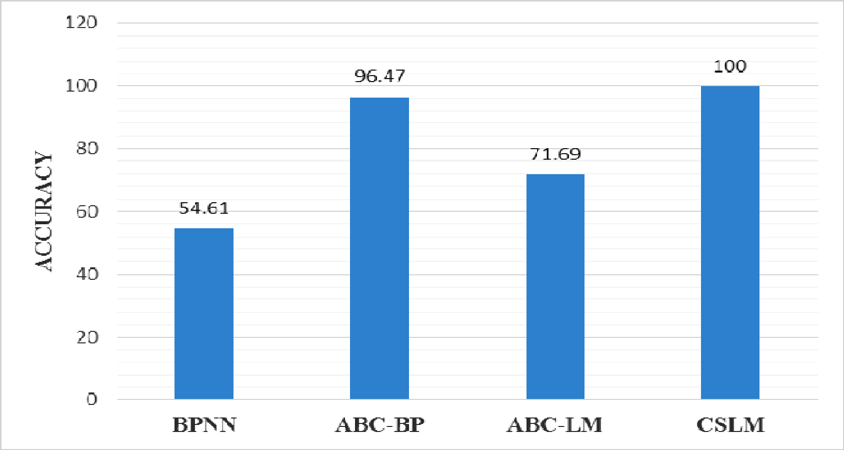


Figure 3 Average accuracy for 2-5-1 NN architecure.

4.3 Three Bit Exclusive-OR Problem

In the second phase, we used 3 bit XOR dataset which comprises of three inputs and a single binary output. The parameter range is same as used for two bit XOR problem. For the 3-5-1 the network, it has twenty connection weights and six biases. Table 2 shows the CPU time, number of epochs, the MSE standard deviation (SD) of the MSE, and accuracy for the 3 bit XOR test problem with 5 hidden neurons. Figure 4, displays the 'MSE performances vs. Epochs' of CSLM, and ABC-LM algorithms for the 3-5-1 network structure. While Figure 5, shows the average accuracy performance of the CSLM, ABC-LM, ABC-BP and BPNN, algorithms. For the three bit XOR, the CSLM algorithm converges to global minima within 671 epoch and 99.9% accuracy. From Table 2, it is clear that the proposed CSLM algorithm has better performance than BPNN, ABC-BP, ABC-LM in terms of MSE, Epochs, CPU time.

Table 2 CPU time, Epochs and MSE for 3-5-1 NN architecture.

Algorithm	BPNN	ABC-BP	ABC-LM	CSLM
CPUTIME	50.03	172.33	123.79	80.36
EPOCHS	1000	1000	1000	671
MSE	0.25	0.08	0.01	7.5E-07
SD	0.00064	0.0187	0.0055	3.14E-06

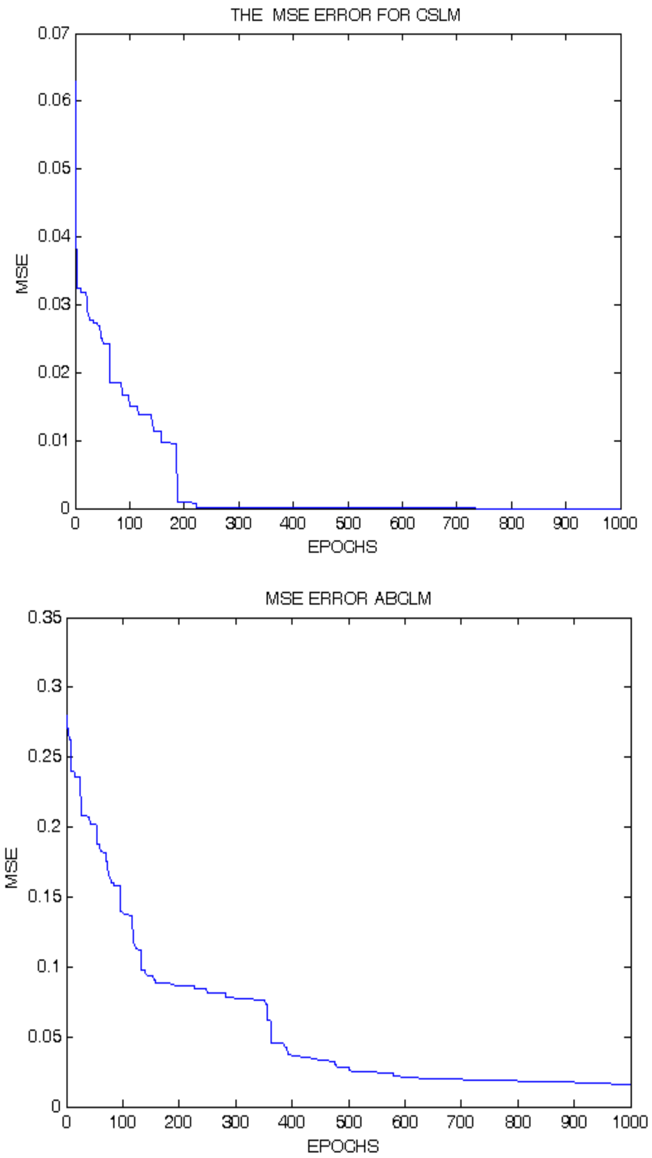


Figure 4 MSE for (a) CSLM and (b) ABC-LM on 3 BIT XOR and 2-5-1 NN architecture.

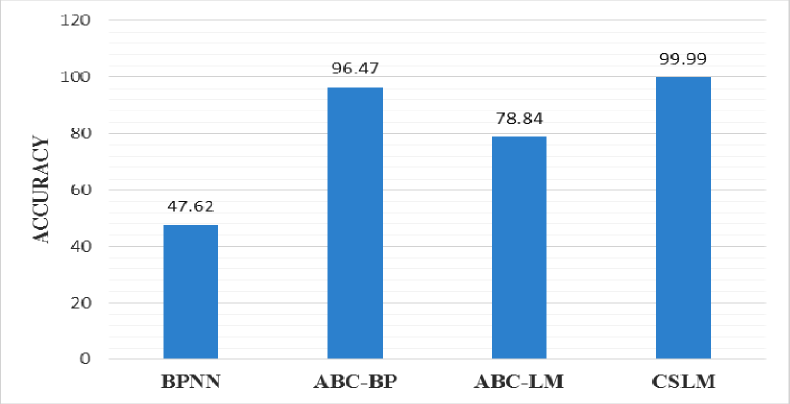


Figure 5 Average accuracy for 3-5-1 NN architecture.

4.4 The 4 Bit OR Problem

The third problem is 4 bit OR which consists of four inputs and a single binary output. The network structure is same as the 2 and 3 bit XOR problem. In 4 bit OR, if the number of inputs all is 0, the output is 0, otherwise the output is 1. Again for the four bit input we apply 4-5-1, feed forward neural network structure. For the 4-5-1 network structure it has twenty five connection weights and six biases. Table 3 illustrates the CPU time, epochs, and MSE performance of the proposed CSLM, ABC-BP, ABC-LM and BPNN algorithms respectively. Figure 6, shows the 'MSE performance vs. Epochs' for the 4-5-1, network structure of the proposed CSLM, and ABC-LM algorithms. From the Table 3, it can be observed that the proposed CSLM algorithm convergence rate is very fast than the other techniques in terms of CPU time and number of epochs. In Figure 7, the average accuracy comparison of the proposed CSLM, and ABC-BP, ABC-LM and BPNN algorithms can be seen. The proposed CSLM can be seen converging to global minima with 100% accuracy in Figure 7.

Table 3 CPU time, Epochs and MSE for 4-5-1 ANN architecture.

Algorithm	BPNN	ABC-BP	ABC-LM	CSLM
CPUTIME	63.280	162.494	118.7274	6.16
EPOCHS	1000	1000	1000	43
MSE	0.0527	1.91E-10	1.82E-10	0
SD	0.0084	1.5E-10	2.31E-11	0

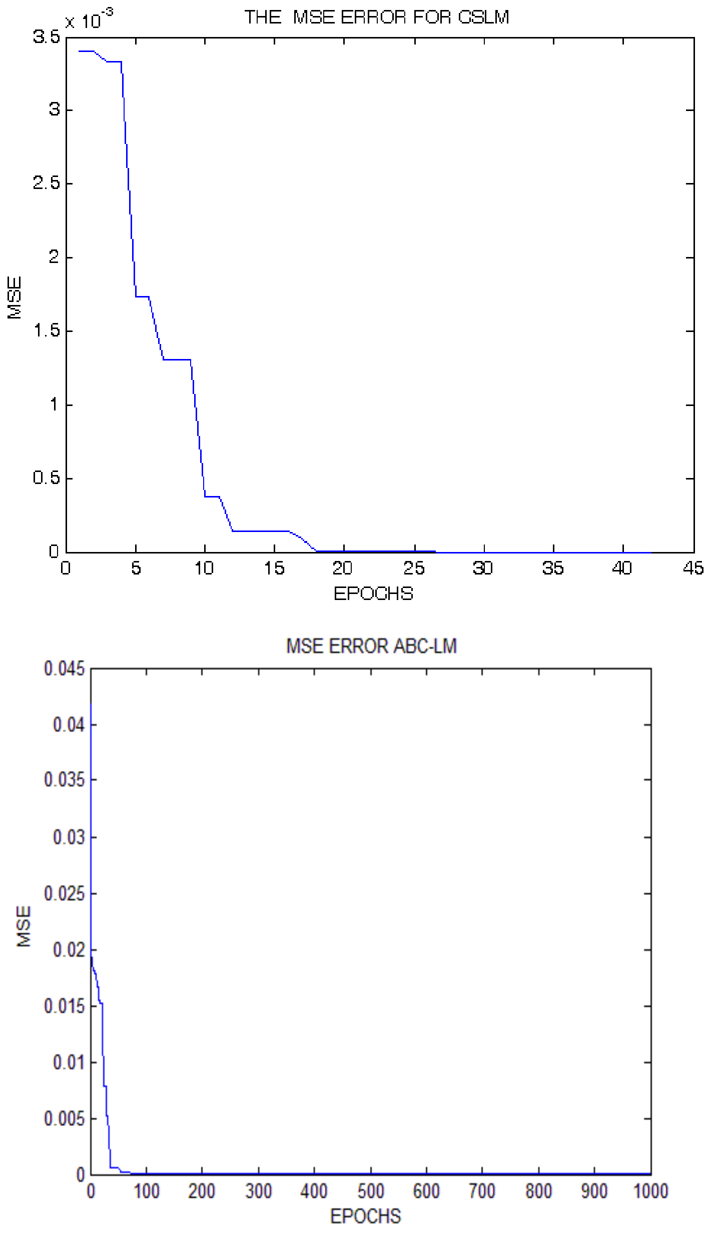


Figure 6 MSE for (a) CSLM and (b) ABC-LM on 4 Bit OR and 2-5-1 NN architecture.

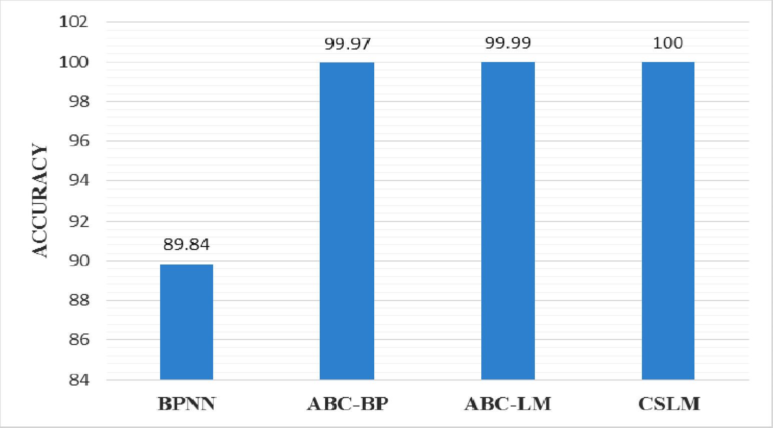


Figure 7 Average accuracy for 3-5-1 NN architecture.

5 Conclusion

In this paper, an improved Cuckoo Search and Levenberg Marquardt back propagation (CSLM) algorithm is proposed. The main idea of CSLM is that the CS is used at the beginning stage for generating the optimal weights, then LM continues the training by inheriting the best weights from CS algorithm [30]. The proposed CSLM algorithm is trained on XOR and OR datasets. The experimental results show that the proposed CSLM algorithm has better performance than ABC-LM, ABC-BP and BPNN algorithms in terms of MSE, number of epochs, (Standard Deviation) SD and accuracy.

Acknowledgements

The authors would like to thank Office of Research, Innovation, Commercialization and Consultancy Office (ORICC), Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of Higher Education (MOHE) Malaysia for financially supporting this Research under Fundamental Research Grant Scheme (FRGS) vote no. 1236.

References

[1] Radhika, Y. & Shashi, M., *Atmospheric Temperature Prediction using Support Vector Machines*, International Journal of Computer Theory and Engineering, **1**(1), pp. 1793-8201. 2009.

[2] Akcayol, M.A. & Cinar, C., *Artificial Neural Network Based Modeling of Heated Catalytic Converter Performance*, Journal of Applied Thermal Engineering, **25**, pp. 2341-2350, 2005.

[3] Shereef, K.I. & Baboo, S.S., *A New Weather Forecasting Technique using Back Propagation Neural Network with Modified Levenberg-Marquardt Algorithm for Learning*, IJCSI International Journal of Computer Science, **8**(6), pp. 1694-0814, 2011.

[4] Kosko, B., *Neural Network and Fuzzy Systems, 1st ed*., Prentice Hall of India, 1994.

[5] Krasnopolsky, V.M. & Chevallier, F., *Some Neural Network application in environmental sciences. Part II: Advancing Computational Efficiency of Environmental Numerical Models*, Neural Networks, **16**(3-4), pp. 335-348, 2003.

[6] Coppin, B., *Artificial Intelligence Illuminated*, Jones and Bartlet Illuminated Series, USA, Chapter 11, pp. 291- 324, 2004.

[7] Basheer, I.A. & Hajmeer, M., *Artificial Neural Networks: Fundamentals, Computing, Design, and Application*, Journal of Microbiological Methods, **43**(1), pp. 03-31, 2000.

[8] Zheng, H., Meng, W. & Gong, B., *Neural Network and its Application on Machinery Fault Diagnosis*, IEEE International Conference on Systems Engineering (ICSYSE), pp. 576-579, 17-19 September, Kobe, Japan, 1992.

[9] Rehman, M.Z. & Nawi, N.M., *Improving the Accuracy of Gradient Descent Back Propagation Algorithm* (*GDAM*) *on Classification Problems*, International Journal of New Computer Architectures and their Applications (IJNCAA), **1**(4), pp.838-847, 2012.

[10] Rumelhart D.E., Hinton G.E. & Williams R.J., *Learning Representations by Back-Propagating Errors*, Nature, **323**, pp. 533-536, 1986.

[11] Lahmiri, S., *A Comparative Study of Backpropagation Algorithms in Financial Prediction*, International Journal of Computer Science, Engineering and Applications (IJCSEA), **1**(4), 2011.

[12] Nawi, N.M., Ransing, R.S. & AbdulHamid, N., *BPGD-AG: A New Improvement of Back-Propagation Neural Network Learning Algorithms with Adaptive Gain*, Journal of Science and Technology, **2**(2), 2011.

[13] Ahmed, W., Saad, E. & Aziz, E., *Modified Back Propagation Algorithm for Learning Artificial Neural Networks*, The 18th National Radio Science Conference (NRSC), pp. 345-352, 27-29 March, Mansoura, Egypt, 2001.

[14] Wen, J. Zhao, J.L., Luo. S.W. & Han, Z., *The Improvements of BP Neural Network Learning Algorithm*, 5th Int. Conf. on Signal Processing WCCC-ICSP, pp.1647-1649, 21-25 August, Beijing, China, 2000.

[15] Salchenberger, L.M., Cinar, E.M. & Lash N.A., *Neural Networks: A New Tool for Predicting Thrift Failures*, Decision Sciences, **23**(2), pp. 899-916, 1992.

[16] Sexton, R.S., Dorsey, R.E. & Johnson, J.D., *Toward Global Optimization of Neural Networks: A Comparison of the Genetic Algorithm and Backpropagation*, Decision Support Systems, **22**, pp. 171–186, 1998.

[17] SNNS (Stuttgart Neural Network Simulator), http://wwwra.informatik. unituebingen.de/SNNS/, 25th January, 2013.

[18] Fahlman, S.E., *Faster-Learning Variations on Back Propagation: An Empirical Study*, 1988 Connectionist Models Summer School by Scott E. Fahlman, pp. 38-51, San Mateo, CA, 1988.

[19] Riedmiller, M. & Braun. H., *A Direct Adaptive Method for Faster Back Propagation Learning: The RPROP Algorithm*, IEEE International Conference on Neural Networks (ICNN93), 28th March-1st April, San Francisco, CA, 1993.

[20] Fahlman, S.E. & Lebiere, C., *The Cascade-Correlation Learning Architecture*, Advances in Neural Information Processing Systems, **2**, pp. 524-532, San Mateo, Calif, 1990.

[21] Hagan, M.T. & Menhaj, M.B., *Training Feed Forward Networks with the Marquardt Algorithm*, IEEE Trans. on Neural Networks, **23**, pp. 899-916, 1994.

[22] Wilamowski, B.M., Cotton, N., Hewlett, J. & Kaynak, O., *Neural Network Trainer with Second Order Learning Algorithms*, 11th International Conference on Intelligent Engineering Systems, Budapest, Hungary, IEEE, 2007.

[23] Hagan, M.T. & Menhaj, M.B., *Training Feed Forward Networks with the Marquardt Algorithm*, IEEE Trans. Neural Network., **5**(6), pp. 989-993, 1994.

[24] Xiao-ping, C., Chang-hua, H., Zhi-qiang, Z. & Ying-jie, L., *Fault Prediction for Inertial Device Based on LMBP Neural Network*, Electronics Optics & Control, **12** (6), pp.38-41, 2005.

[25] Haykin, S., *Neural Networks*, Beijing, China Machine Press, pp. 501-522, 2004.

[26] Nawi, N.M., Khan, A. & Rehman, M.Z., *A New Levenberg Marquardt Based Back Propagation Algorithm Trained with Cuckoo Search*, ICEEI, UKM, 2013.

[27] Yan, J., Cao, H., Wang, J., Liu, Y. & Zhao, H., *Levenberg-Marquardt Algorithm Applied to Forecast the Ice Conditions in Ningmeng Reach of The Yellow River*, 5th International Conference on Natural Computation, pp. 184-188, 14-16 August, Tianjin, China, 2009.

[28] Yang, X. & Deb, S., *Cuckoo Search via Lévy Flights*, Proceedings of World Congress on Nature & Biologically Inspired Computing, India, pp. 210-214, 2009.

[29] Yang, X. & Deb, S., *Cuckoo Search via Levy Flights*, Nature and Biologically Inspired Computing (NaBIC 2009), pp. 210-214, 2009.

[30] Nawi, N.M., Khan, A. & Rehman, M.Z., *A New Levenberg-Marquardt based Back-propagation Algorithm trained with Cuckoo Search*, 4th International Conference on Electrical Engineering and Informatics (ICEEI), 24-25 June, Bangi, Malaysia, 2013.

Keystrokes Inference Attack on Android: A Comparative Evaluation of Sensors and Their Fusion

Ahmed Al-Haiqi, Mahamod Ismail & Rosdiadee Nordin

Department of Electrical, Electronic & Systems Engineering,
Faculty of Engineering & Built Environment, National University of Malaysia,
46300 UKM Bangi, Selangor, Malaysia
Email: ahmadalhaiqi@gmail.com

Abstract. Introducing motion sensors into smartphones contributed to a wide range of applications in human-phone interaction, gaming, and many others. However, built-in sensors that detect subtle motion changes (e.g. accelerometers), might also reveal information about taps on touch screens: the main user input mode. Few researchers have already demonstrated the idea of exploiting motion sensors as side-channels into inferring keystrokes. Taken at most as initial explorations, much research is still needed to analyze the practicality of the new threat and examine various aspects of its implementation. One important aspect affecting directly the attack effectiveness is the selection of the right combination of sensors, to supply inference data. Although other aspects also play crucial role (e.g. the features set), we start in this paper by focusing on the comparison of different available sensors, in terms of the inference accuracy. We consider individual sensors shipped on Android phones, and study few options of preprocessing their raw datasets as well as fusing several sensors' readings. Our results indicate an outstanding performance of the gyroscope, and the potential of sensors data fusion. However, it seems that sensors with magnetometer component or the accelerometer alone have less benefit in the context of the adverted attack.

Keywords: *android; motion sensors; sensor-based attack; side-channel attack; smartphone security; touchscreen taps inference.*

1 Introduction

Key logging attacks have been, and continue to be, a major security concern in traditional computing platforms. PC operating systems like Windows allow system message interception, thereby enabling background applications to capture and log the key strokes of active applications in the foreground. Such Trojan applications are very common in traditional PCs, and although usually called keyloggers, many are capable of capturing more information than mere keyboard input, such as screenshots and mouse clicks. Even when keyloggers are not easy to install or hide, hardware keyboards allow for backdoor channels that could be resorted to in order to guess the actual input. Such channels are

Received September 19th, 2013, Revised October 25th, 2013, Accepted for publication November 25th, 2013.

commonly termed side channels, and include physical phenomena like electromagnetic and acoustics emanations.

Moving to smartphones, the situation is quite different. Smartphone OSes, like Android, restrict the interception of keystrokes to the current view that has the focus, and no direct method can be used by Trojan apps to log user taps (but for a workaround example, see [1]). Moreover, the lack of physical keyboards limits side channel avenues at the disposal of attackers. For instance, the rich heritage of research on electromagnetic and acoustic emanations is not applicable on smartphones. Yet, innovative side channels to sniff on user input have been proposed in the literature, including the analysis of finger smudges left on the touch screen surface [2], employing the dial tone sounds [3] that resembles more traditional acoustic emanations attacks, and even the old-school low-tech shoulder-surfing [4].

Embedding sensors into smartphones had made them an unprecedented platform, combining communications, computing and sensing capabilities. User interface, gaming, and healthcare are but a few domains in which sensors found instant applications [5], and a key idea around which many more applications evolve is context awareness. On the flip side, sensors bring along many serious implications, especially related to user privacy. Researchers have studied the potential threat of more traditional sensors, namely, GPS, camera and recorder, on user privacy and/or security (e.g. [6]). Less traditional sensors, such as motion sensors (e.g. accelerometers and gyroscopes), have received little attention, until recently. The key observation that moved motion sensors into the threat spot is the correlation between user taps on touch screen and vibrations or motion changes to the body of the smartphone itself. Accelerometers, for example, can sense the linear acceleration caused by the force of user taps, while gyroscopes can measure angular velocities around specific axes. Obviously, the original benign purpose of theses sensors are far from such vicious uses and that makes them a surreptitious side channels.

Few authors in academia have demonstrated the feasibility of such a side channel attack, dealing with the task as a classification problem to map sensors reading into key labels. Section 3 presents a brief survey of these works. The availability of several sensors on consumers electronics devices, in particular smartphones, raises a seemingly interesting question of which sensor, or collection of sensors thereof, is of greater potential in the context of the new threat. In this paper, we set off to experiment with the performance of different sensors that are supported by Android operating system, and integrated in most Android-powered phones. Section 2 explains shortly the considered sensors, and how it is possible to capture their data with Android help. For the purpose of the comparison, we actually implemented the attack collecting sensors data through an Android app. Details of the experiment are given in section 4.

It is important to notice that the aim of this paper is not to improve the accuracy of the attack, or evaluate its practicality, but is more about comparing the performance of sensors under the same conditions, which are well-controlled. Our experiment in this light could focus on the relative performance between different datasets on the same setting, rather than looking for the best among several settings. The latter is the focus of most recent works discussed in section 3, while the early works were concerned with the bare feasibility of the attack.

2 Technical Background

Android supports a variety of sensors, of which the relevant to the current attack are motion and position sensors. Table 1 lists the supported motion sensors, as of Android 4.2.2. Besides those sensors, Android also provides a synthetic sensor based on the values from the accelerometer and magnetometer, through a method call in its sensors API (*getOrientation()*). Accelerometer raw data contains both gravity acceleration (due to the earth's gravity force) and linear acceleration (due to the linear motion of the phone body). The former is more of a bias to our experiment as we do not consider here the inclination of the phone with respect to the earth surface, and therefore both raw accelerometer's data as well as gravity sensor's data are excluded, as the linear accelerometer can take their role.

Table 1 Motion sensors that are supported on Android platforms (source [7]).

Sensor	Description	Units of measurement
TYPE_ACCELEROMETER	Acceleration force along the x, y and z axes (including gravity).	m/s^2
TYPE_GRAVITY	Force of gravity along the x, y and z axes	m/s^2
TYPE_LINEAR_ACCELEROMETER	Acceleration force along the x, y and z axes (excluding gravity).	m/s^2
TYPE_GYROSCOPE	Rate of rotation around the x, y and z axes	rad/s
TYPE_ROTATION_VECTOR	Rotation vector component along the x, y and z axes (axis * sin (θ/2)).	Unit-less

Motion sensor readings are expressed with reference to a 3-axis coordinate system. This coordinate system is defined relative to the device's screen when the device is held in its default orientation (Figure 1): the x axis is horizontal and points to the right, the y axis is vertical and points up, and the z axis points toward the outside of the screen face. This coordinate system is not with reference to the earth's frame, and hence does not change irrespective of the device orientation. For example, if the phone is laid horizontally on a table with its screen surface to the sky, a motion towards the left would generate negative readings along the x axis. If the phone is lifted horizontally, the z axis's readings are positive.

Android OS provides a sensors programming framework, composing a set of classes and interfaces that could be used to access the sensors and read their raw data. Basic tasks accomplished by that API include detection of available sensors on a device, determining their capabilities (e.g., maximum range, power requirements, and resolution), reading raw sensor data, defining the minimum rate at which to acquire sensor data, and to register/unregister sensor event listeners that monitor sensor changes.

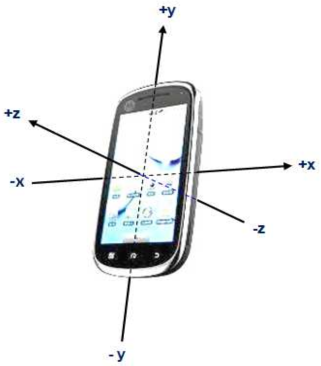


Figure 1 Coordinate system of the motion sensors.

Sensor values are obtained through an object, SensorEvent, in a multi-dimensional array named values[]. Elements of the array correspond to sensor readings along each of the three coordinate axes. For example, values[0] might convey the acceleration force along the x axis while values[1] contains the acceleration force along the y axis, and so forth. Acceleration forces reflect the change in velocity along the axes due to user motion (e.g. walking) or due to direct user input (e.g. tapping on the device). Raw acceleration data, represented by the accelerometer sensor includes the acceleration due to gravity force. Android provides also filtered linear acceleration that can be formulated as

Linear acceleration = raw acceleration – gravity acceleration

Theoretically, accelerometer data could be integrated twice to obtain the distance of motion. However, the result of this process is prone to much error in practice, which leaves very little benefit considering the tiny differences in displacement of different keys on the touch screen. It could also be argued that displacement is proportional to the force by which the user taps on the device, the same as the acceleration itself, and therefore there is really no specific value in taking the additional double integration step. The tilt of the device, however, is another path to consider, utilizing the accelerometer, which we leave for future work.

Gyroscope readings reflect the rate of rotation (i.e. angular velocity) around the three axes, and ideally measure zero when the device is at rest. Again, it is possible to calculate the angle of rotation via integrating the gyroscope data. Similar argument to the case of accelerometers could be made, though we actually consider this option in one of our experiments. Gyroscopes data are known to suffer from a drift in the reported values when integrated, because of the accumulated measurement errors over time (i.e. noise plus initial offset). Typically, data from other sensors are utilized to compensate for such errors. In our case, a keystroke on the device is short enough to justify neglecting accumulated drift and compute the integration of just the few readings during a single keystroke anew with each tap.

The rotation vector represents the orientation of the device as a combination of an angle and an axis, in which the device has rotated through an angle θ around an axis (x, y, or z) [7]. It is a synthetic sensor that maps the device coordinate system to the global coordinate system using the accelerometer, magnetometer, and the gyroscope. In the global coordinate system, the x axis points to the east, the y axis points to the north, while the z axis points to the sky away from the surface of the earth.

Finally, Android also provides a method called getOrientation() that takes as input a rotation matrix. The rotation matrix might be generated using data from both the accelerometer and the magnetometer. The output of the getOrientation() method is the rotation around the x (pitch), y (roll), and z (yaw) axes. The coordinate system of this method is similar to the global coordinate system described in the previous paragraph, except that the x and z axes are inverted. However, it is worthy to mention that nearby magnets can largely affect magnetometer data, leading to less accurate results.

3 Related Work

Authors in [8] suggested first the use of motion sensors to infer keystrokes on touch screens. They developed an Android application, named TouchLogger, to demonstrate the attack. The application used numbers-only soft keypad in the landscape mode. TouchLogger utilized the synthetic Orientation sensor, which relies on accelerometer and magnetometer hardware sensors. Orientation sensor was deprecated in Android 2.2 (API level 8).

Following was another work [9], where an Android application, ACCessory, was built to evaluate a predictive model, trained only on acceleration measurements. ACCessory attempted to infer area zones on the screen as well as character sequences (to construct typed passwords).

The next work [10] adopted an online processing, where the training and classification were performed on the smartphone itself through a Trojan application, TapLogger, to stealthily monitor the movement changes of the device and try to log the number pad passwords and screen lock PINs. Two sensors were used: the accelerometer for taps detection, and the Orientation sensor for tap positions inference.

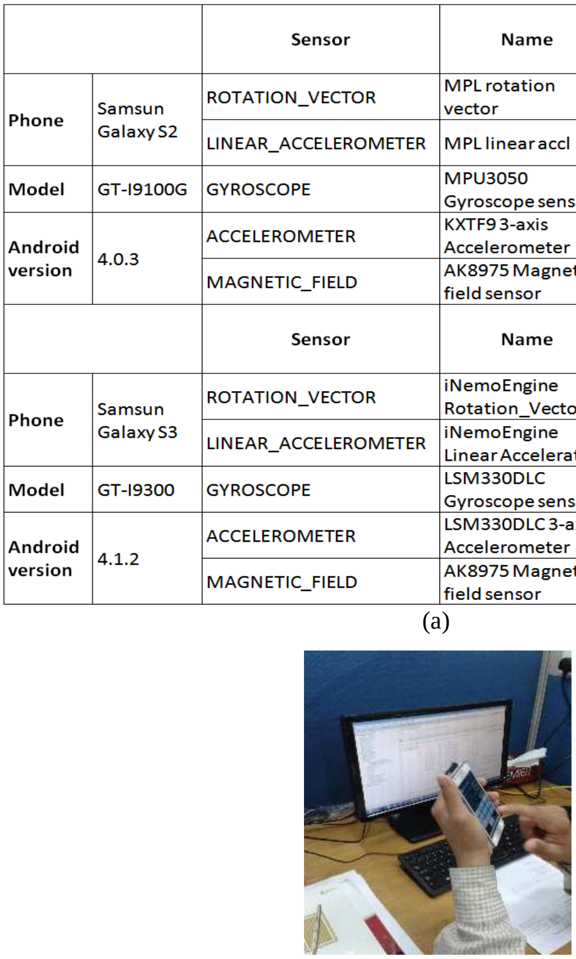
The same authors of TouchLogger published another work again [11]. The purpose of the study was to provide a more thorough investigation on the practicality of such an attack, and to compare the performance of different classification schemes, and the impact of different devices, screen dimensions, keyboard layouts or keyboard types. This paper examined the use of gyroscopes output on mobile devices for the attack, and indicated that inference based on the gyroscope is more accurate than that based on the accelerometer.

TapPrints [12], the framework presented in another paper was evaluated across several platforms including different operating systems (iOS and Android) and form factors (smartphones and tablets). It also showed a combined approach that uses both the accelerometer and gyroscope for achieving better accuracy.
Finally, and most recently, the authors of [13] focused solely on the accelerometer sensor to further investigate the practicality of sensors side channels in inferring Android four-digit PINs and password pattern (swiping). Contrary to previous last two works, they found that accelerometer based techniques perform nearly as well, or better, than gyroscopic based techniques.

To the best of our knowledge, no previous study addressed the relative performance of all relevant Android sensors in conducting the inference attack. In addition, we also consider fusing the data from more than one sensor for that matter. In a preliminary version of our study, we started by evaluating sensors performance on only a single Android smartphone [14]. In this paper, we augment the analysis with sensors from another more recent device, and include more variations in the data preprocessing and fusion options, as well as in the feature set of the sensors data.

4 Methodology

Keystrokes inference can be viewed as a machine learning problem, in particular, a classification task that maps collected patterns of raw sensor signals data into corresponding key classes. Abiding by typical machine learning process sequence, raw data are collected from the source, and pre-processed, and then features are selected. Part of the resulting dataset is labeled with the correct class to form a training examples subset, and the rest is reserved for evaluation purposes as a test subset. In the following subsections we present those steps in more detail.



		Sensor	Name	Min delay (μs)	vendor
Phone	Samsun Galaxy S2	ROTATION_VECTOR	MPL rotation vector	10000	Invensense Technology
		LINEAR_ACCELEROMETER	MPL linear accl	10000	Invensense Technology
Model	GT-I9100G	GYROSCOPE	MPU3050 Gyroscope sensor	10000	Invensense Technology
Android version	4.0.3	ACCELEROMETER	KXTF9 3-axis Accelerometer	10000	Kionix Corporation
		MAGNETIC_FIELD	AK8975 Magnetic field sensor	10000	Asahi Kasei Microdevices
		Sensor	Name	Min delay (μs)	vendor
Phone	Samsun Galaxy S3	ROTATION_VECTOR	iNemoEngine Rotation_Vector	10000	STMicroelectronics
		LINEAR_ACCELEROMETER	iNemoEngine Linear Acceleration	10000	STMicroelectronics
Model	GT-I9300	GYROSCOPE	LSM330DLC Gyroscope sensor	5000	STMicroelectronics
Android version	4.1.2	ACCELEROMETER	LSM330DLC 3-axis Accelerometer	10000	STMicroelectronics
		MAGNETIC_FIELD	AK8975 Magnetic field sensor	10000	Asahi Kasei Microdevices

(a)

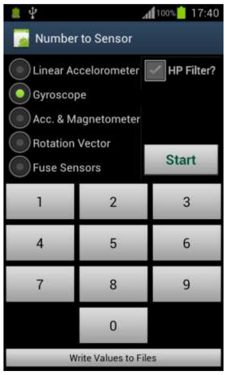
(b)

Figure 2 Experiment settings (a) Hardware specs and (b) user typing profile.

4.1 Data Collection

An application was built for the purpose of acquiring sensors data out of an Android smartphone. Our raw data are the readings of four sensors, two of which are synthetic sensors, derived from a combination of up to three hardware sensors, as noted in section 2. A set of two devices was used in the experiment: a Galaxy S2 and a Galaxy S3, both are of different screen sizes and sensors specs. One user was utilized to type all data sets to ensure consistent typing and holding style factors that could affect the inference performance. Figure 2(a) lists the main hardware specifications of the devices and their built-in sensors, while Figure 2(b) shows the typing profile of the user throughout the experiment.

The UI of the application, depicted in Figure 3, allows for the selection of the sensor in each session, and a layout similar to the dialing soft keypad of Android 4.0.3 is presented to the user, where he was asked to key in almost the same set of around 300 keys in each session. The key set covers uniformly the ten digits of the numbers soft keypad. For each device, two datasets were generated from the same linear accelerometer, but distinct by the use of high pass filtering, an option that the user can choose by selecting a checkbox on the screen. Filtering is one of the common techniques to mitigate noise in sensors data, and in that context, low pass filters are more useful. However, high pass filters can extract the most fluctuating components, in which we are more interested. The interface also includes an option for fusing the data, upon which all four sensors are registered with Android, and the readings from all sensors are recorded in the same session. This is useful to inspect the case when the feature vector of a dataset example comprises components from different sensors together.

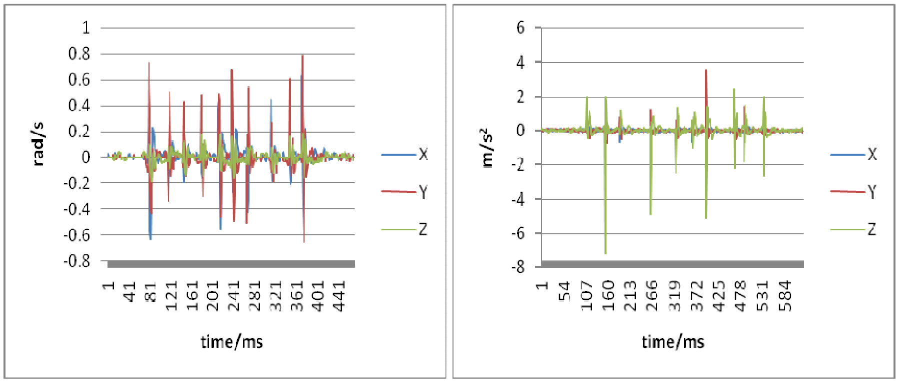


○ Linear Accelorometer
● Gyroscope
○ Acc. & Magnetometer
○ Rotation Vector
○ Fuse Sensors
☑ HP Filter?

Figure 3 User Interface of the Android app used to collect the data.

4.2 Pre-processing

Raw sensors data usually need to be processed before feeding into a learning system. One of the most important steps in the context of the current attack is the detection, and extraction of the signal segments that correspond to key strokes from the continuous stream of sensors reading. This step could form a separate research task, and several approaches might be followed. Some of the previous works on this attack regard this task as a straightforward anomaly detection problem [9], or a simple classification problem [12], while others treat it as a significant part of their whole system [10]. Figure 4 illustrates the signals collected from the gyroscope and the accelerometer along the x, y and z axes while dialing the phone number “0123034880” using the screen shown in Figure 3. It could be seen from the distinct spikes corresponding to the individual key taps, that the task of isolating keystrokes is conceivably doable. Yet, some authors leave this step as a separate undertaking that lies on the shoulder of the attacker, and assumes knowledge of the keystrokes delimitations [13]. We followed the same suit, as the course of extracting this knowledge is independent of the eventual performance, assuming consistency among all datasets. We isolated the keystrokes sensors data by matching their timestamps to the start and end time of each button click event, which we also collect during the experiment using Android-provided motion events, namely the “onTouch” method. This method was implemented as per the onTouchListener interface requirement. It is these events that supply the experiment with the labels necessary for the training examples.



(a) (b)

Figure 4 Readings from the (a) gyroscope and (b) accelerometer while dialing the number “0123034880”.

Other pre-processing techniques are also possible and usually crucial for successful learning, including normalization and calibration. Normalization is needed when different features of input data belong to different scales of several order of magnitude discrepancies. Rescaling might be necessary to ensure that no single feature has influence that may not reflect their real relative importance in deciding the outcome. In our case, all sensors reading in all three axes are of the same or only one order of magnitude variance. Calibration is usually also needed, for example, to remove the projection of Earth gravity from accelerometer data and initial orientation from gyroscope data. Using the linear accelerometer in our experiment ensures the already gravity-less acceleration measurements, while initial bias in other sensors are not of much concern since the whole experiment is conducted under the exact same conditions, including any initial biases.

In our data sets we have encountered no missing values, as we have collected all the data ourselves programmatically. Very few outliers could be seen, probably because of abrupt unintended motion of the user hands. We simply got rid of those outliers, as they play no representative role in the input data. We also did not need any dimensionality reduction techniques, as our data are already of relatively low dimension (18, in most cases, except when applying sensors fusion, where every sensor contributes 18 features, and when integrating gyroscope data, where three more features, the derived rotation angles along the three axes, are added).

Another significant pre-processing we applied, besides keystrokes extraction, is the aggregation of sensors data by each key, as explained in the next section, to create the features. Some works in the literature had to normalize the sampling rate of sensors, termed de-jittering in [11], to compensate for non-uniform sensors sample intervals. However, this is basically needed for the purpose of standard signal analysis methods, whereas our features are mainly simple statistics that use aggregation of few samples per each key, as discussed later, and no de-jittering was applied.

Finally, we also included the option of integrating the gyroscope data before training. The purpose is to experiment with the gyroscopic data as a source on angular displacement rather than angular speed. Taken as a rate of change in angular shift, gyroscope data could be integrated based on basic calculus to obtain the distance traveled, in this case the shift angle. Despite the fact that integration often introduces drift to the gyroscope readings, we neglect this error as the integration is performed for each set of readings associated with one keystroke. In this way, we ensure a sort of reset action per every key tap, and the time intervals to accumulate noise errors are very short.

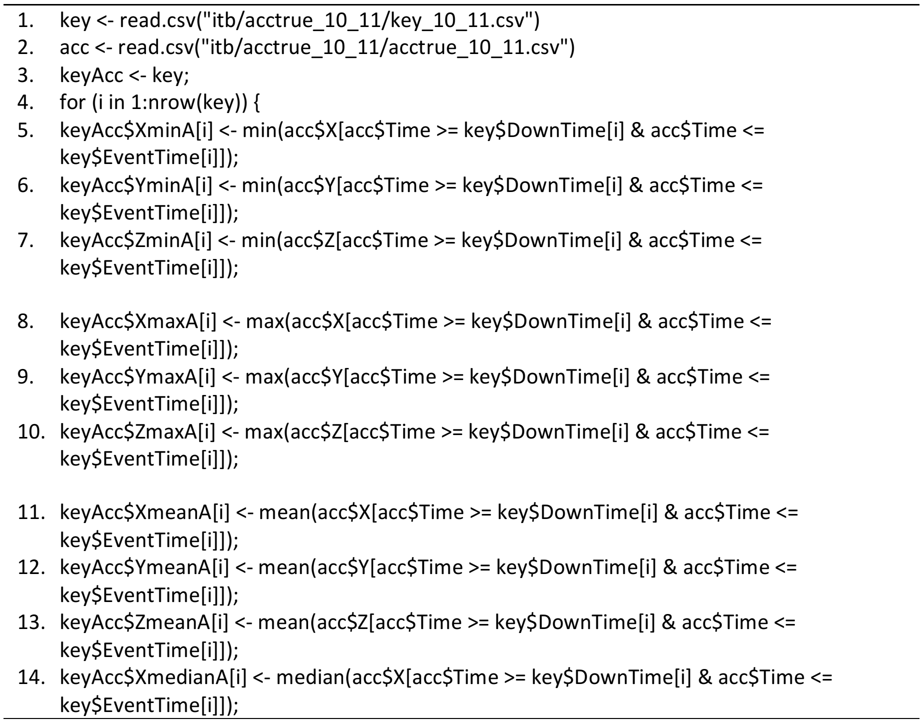
4.3 Feature Selection

For the purpose of classification, the input dataset is a set of examples. Each example is a features vector (collection of features or attributes), which is fed to the classifier and mapped collectively to a pre-set output label (a digit key, in our case). Previous works on the keystrokes inference attack vary greatly in the set of features employed for classification, and there is no obvious evidence of which set is better, indicating an open research area in this direction. For our experiment, we have chosen to apply simple statistics on sensors data from the time domain only, though some authors include also frequency domain signal features.

The output of our application is a set of files, each recording a continuous stream of readings from one sensor, or the combination of sensors in case of data fusion. In addition, one file always contains the touch events information, namely the start time, end time, and the particular tapped key. For each sensor, there is a corresponding output file in every session. This file contains all the collected sensor's readings during the session, a reading per line. Each sample reading is a quadruple, {x, y, z, t} for the sensor's output along the three axes, plus a timestamp in milliseconds. Likewise, in every respective session, the other file records each touch event (a keystroke) with the corresponding interval in milliseconds. One touch event normally spans several sensor samples. In our experiment, a key tap takes on average 80 ms, and the sampling rate of the sensors is, at most, 100 samples per second in theory, except for the gyroscope in Galaxy S3 phone, where the minimum delay is around 5000 microsecond, which represents a frequency of 200 samples per second. This means around 10 samples per key in the best case, excluding the case of the STMicroelectronics gyroscope where the available samples per key are more, and hence the expected performance is better in general. In practice, however, we found that each key corresponds to an average of 5 sensor samples (15 samples for Galaxy S3 gyroscope).

Individual samples are meaningless relative to a key tap, and so we aggregate the samples for each key, i.e. combine all the samples that span the period of a key tap producing simple standard statistics of min, max, mean (μ), median (M), standard deviation (σ) and skewness (3*(μ–M)/σ). In this manner, for a dataset of 300 keys, for instance, we obtain 300 examples, each of which comprises 18 features, plus the class label (the key symbol itself). Each example is written to a line in a CSV file, which is used as the final input dataset for the learning process. All the motion sensors generate three components of the physical quantity they measure; one component along each of the three coordinates axes. Therefore, out of the six statistics the result is 18 different values where each statistic is computed for every axis. For instance, one example in the training dataset would comprise the attributes XminA, YminA, ZminA, XmaxA, YmaxA, ZmaxA, XmeanA, and so forth. The first letter in the attribute name refers to the axis to which the feature value belongs, and the last letter refers to the sensor from which the dataset was driven. In the case of this example, the sensor is the accelerometer. The middle part of the attribute name tells the particular statistic constituting the feature. The last attribute would be the particular keystroke for which the time interval matched the interval of the aggregated sensor data. Note that every keystroke spans several sensor readings. In our dataset, each value is taken to be a feature to characterize a keystroke.

The aggregation could be accomplished using any programming language or computational package (e.g. MATLAB or Octave), though we have written simple scripts in the R language to match sensors and key data, and perform the statistical calculations. Figure 5 shows one of those scripts for the accelerometer data, where the steps of aggregating sensors data and producing the features are detailed.



```
key <- read.csv("itb/acctrue_10_11/key_10_11.csv")
acc <- read.csv("itb/acctrue_10_11/acctrue_10_11.csv")
keyAcc <- key;
for (i in 1:nrow(key)) {
keyAcc$XminA[i] <- min(acc$X[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);
keyAcc$YminA[i] <- min(acc$Y[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);
keyAcc$ZminA[i] <- min(acc$Z[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);

keyAcc$XmaxA[i] <- max(acc$X[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);
keyAcc$YmaxA[i] <- max(acc$Y[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);
keyAcc$ZmaxA[i] <- max(acc$Z[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);

keyAcc$XmeanA[i] <- mean(acc$X[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);
keyAcc$YmeanA[i] <- mean(acc$Y[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);
keyAcc$ZmeanA[i] <- mean(acc$Z[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);
keyAcc$XmedianA[i] <- median(acc$X[acc$Time >= key$DownTime[i] & acc$Time <=
key$EventTime[i]]);
```

Figure 5 An R script that aggregates accelerometer data and produces the features set.

```
keyAcc$YmedianA[i] <- median(acc$Y[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]);
keyAcc$ZmedianA[i] <- median(acc$Z[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]);

keyAcc$XsdA[i] <- sd(acc$X[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]);
keyAcc$YsdA[i] <- sd(acc$Y[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]);
keyAcc$ZsdA[i] <- sd(acc$Z[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]);

keyAcc$XskewA[i] <- 3*(mean(acc$X[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]) - median(acc$X[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]])) / sd(acc$X[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]);
keyAcc$YskewA[i] <- 3*(mean(acc$Y[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]) - median(acc$Y[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]])) / sd(acc$Y[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]);
keyAcc$ZskewA[i] <- 3*(mean(acc$Z[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]) - median(acc$Z[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]])) / sd(acc$Z[acc$Time >= key$DownTime[i] & acc$Time <= key$EventTime[i]]);
}
write.table(keyAcc, "itb/acctrue_10_11/keyAcc_10_11.csv")
```

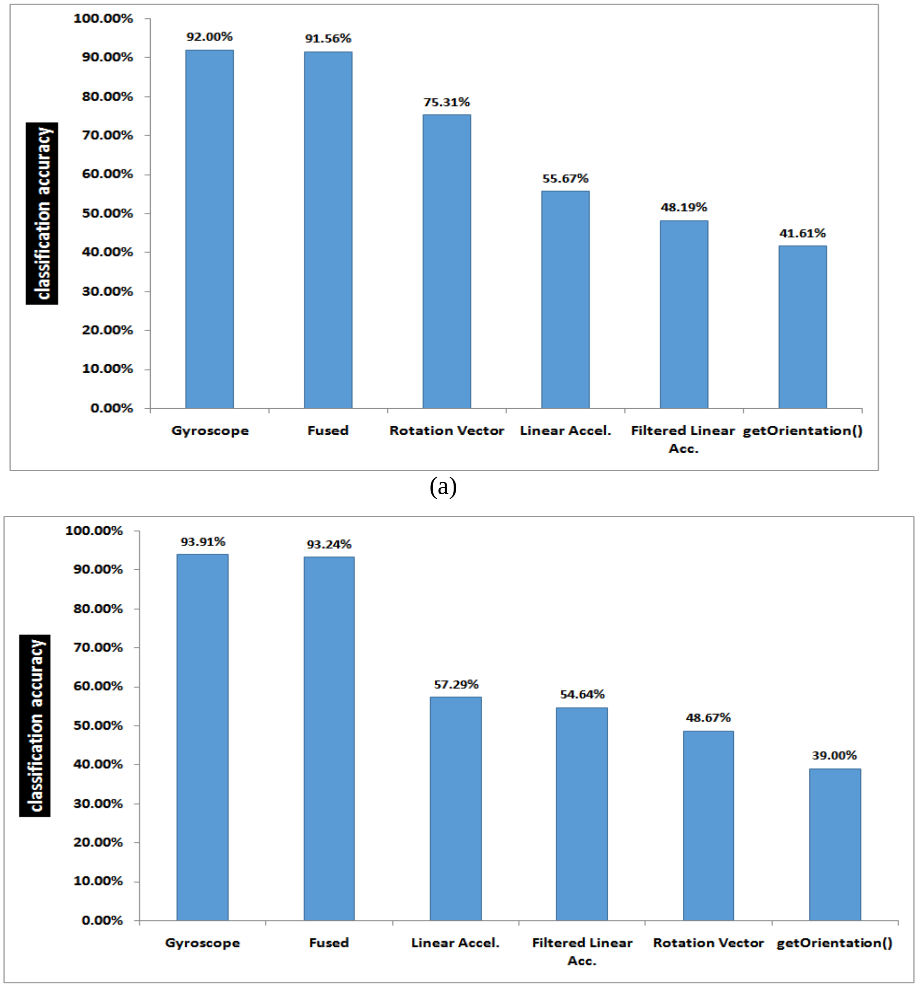
Figure 5 ***Continued.*** An R script that aggregates accelerometer data and produces the features set.

4.4 Classification

The goal of the inference attack is, given a bunch of sensors samples, to map every pattern of readings into an output class, and the percent of correct mapping forms the accuracy of the classifier. We adopted the implementation of classification algorithms in Weka suite of machine learning software [15]. Many classifiers are available in Weka, and choosing a particular method is not critical for our experiment. However, our initial exploration revealed that the ensemble learning can give better results, with implementations in “meta” Weka package. In particular, “Bagging” classifier showed the best performance on average (with FT base model), though we do not aim to venture any claims related to classification algorithms performances. “Bagging” is a general technique for improving the accuracy of a given learning algorithm. As an ensemble learning method, it aggregates multiple learned models of the same type (e.g. decision trees), and uses voting to combine the output of individual models [16].

5 Evaluation and Discussion

All experiment runs were conducted under 5-folds cross-validation testing option, and almost the same dataset size of around 300 examples. Figure 6 shows the results of classifying the datasets using Bagging ensemble learning, with Functional Trees base model [17]. Figure 6(a) depicts the comparative classification accuracy of all the sensors options in the first device, while Figure 6(b) presents the same results for the second device.



(a)

(b)

Figure 6 Experimental results: classification accuracy comparison among all sensors (a) in Galaxy S2 smartphone (b) in Galaxy S3 smartphone.

It is evident from the two graphs that the relative performances of the sensors in both devices are similar. Because of the larger screen size of Galaxy S3, one would expect better ability to classify different key zones. Nevertheless, its performance is comparable or inferior to that of Galaxy S2 (at least in the many runs of our experiments). One possible reason for this result is the observation that the S3 phone was dressed in a thick leather case, which might contribute to mitigate the impact of the taps force on the surface of the device.

Comparing the gyroscope performance with and without integrating its data reveals no actual value out of integration. The graphs in Figure 7 illustrate the accuracy of two experiments, for each of the two devices. One experiment involves a gyroscope dataset of 18 features as usual. The second experiment adds three more features, corresponding to the integrated sensor values per a coordinate axis. This conclusion seems logical, as the rate of change in distance (velocity) is proportional to the distance, and their indication is the same in the context of our need, namely to make a distinction between different areas on the screen. We have included the analysis of this option since gyroscope data is typically integrated in many applications, where the need is specifically to measure the inclination or tilt of the device over an extended period of usage time, rather than the instant change in angular displacement. It should be noted however that relying on the gyroscopic data for the purpose of classifying different keystrokes bears a strong dependency on the force by which the user taps on the screen, which might seem user or even session-dependent. Nonetheless, what makes this task conceivable is the fact that the angular velocity (the output of the gyroscope) is a vector quantity, and its sign depends on the direction of rotation, making separate areas in the coordinate space distinguishable, at least by the sign of the data.

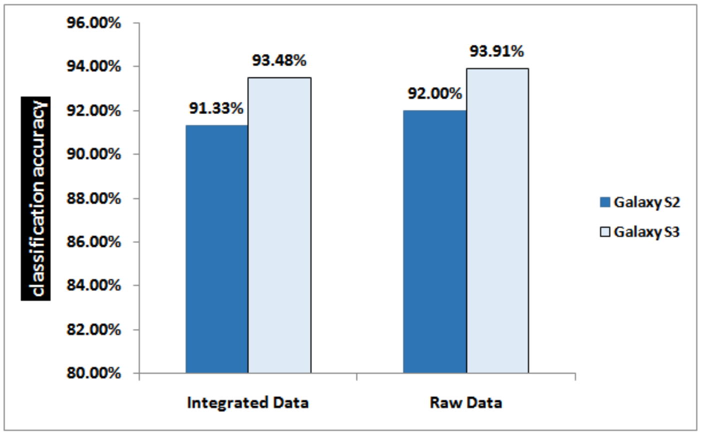


Figure 7 Classification accuracy of the gyroscope with and without integration.

Noticing the marked performance of the gyroscope, it looked appealing to further experiment with the features in its dataset. Trying out separate groups of statistics as well as few combinations of statistical groups resulted in the performance scores shown in Figure 8. For example, in the specific dataset of Figure 8, drawn from Galaxy S3 gyroscope, it appears that the minimum statistic has the most influence in the correct classification of keystrokes.

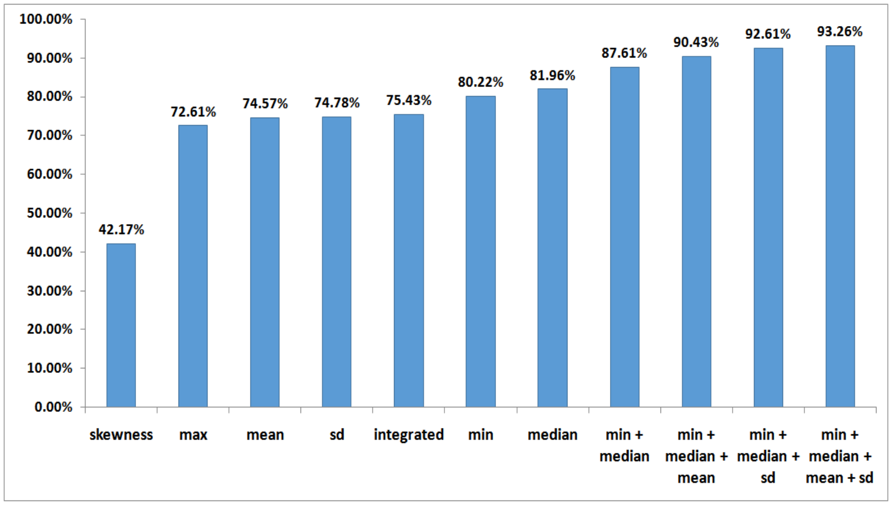


Figure 8 Classification accuracy of few permutations of gyroscopic statistics.

In another set of experiments, we focused on the fused dataset of sensors. In Table 2, we present one instance of those experiments where the relative performance of the individual sensors is shown more manifest, as well as the promising compounds of two sensors. The results are consistent with those in Figure 6, besides the important conclusion that in the context of this particular attack, data from other sensors are more of a noise to the gyroscopic signals. Still, in the absence of a gyroscope, the combination of accelerometer and rotation vector data can be of utility that outperforms their individual performances.

Several concluding notes are in order, based on the obtained results. It is obvious that, confirming to [11]-[12], the gyroscope sensor alone has a superior performance compared to all other sensors. This result is actually consistent with the observation that rotations have more power to distinguish between different keys than shifts. This is also the reason for the inferior accuracy of both accelerometer variations (with and without high pass filtering), and even the rotation vector which relies on the accelerometer as one of the components in deriving its value. Another factor that affects the rotation vector performance is the magnetic field sensor, which is also one of the components in calculating the vector values. Magnetic field sensors are known for their inaccurate and noisy outputs in current commodity smartphones, especially in the presence of nearby metals. What supports this conclusion is that the accuracy of the rotation vector sensor is still much better than the synthetic sensor that is based on the getOrientation method, though both depend on the accelerometer and magnetometer. The difference in accuracy is apparently due to the inclusion of gyroscope into the derivation of the rotation vector.

Table 2 Classification accuracy results of different combinations of sensors.

Sensor	Samsung Galaxy S2	Samsung Galaxy S3
All sensors	91.56%	93.24%
Gyroscope only	95.63%	94.12%
Accelerometer only	56.88%	63.24%
Rotation vector only	75.94%	57.35%
Gyroscope + accelerometer	93.44%	93.53%
Accelerometer + rotation vector	80.63%	71.18%
Gyroscope + rotation vector	92.19%	93.53%

It also seems that sensors fusion is not always the best option, and depends largely on the proper selection of the ingredient sensors. Filtering also depends on the context, and in our case, it appears that the already filtered linear accelerometer has less performance with more filtering; the high pass filter, despite more appropriate to extract occasional motion like keystrokes, could leave the classifier with less than enough information to discern different keys. Finally, it is important to note that all the results related to sensors' performances that we reported in this paper are associated with the application of keystroke inference attack, or similar applications that involve classification of sensors data into discernable zones on the touch screen. This might not be the case in more traditional applications; e.g. gaming or augmented reality.

It is worth mentioning that a key reason which makes this attack an attractive option from a perpetrator perspective is the low requirement for conducting the attack in terms of security permissions. Android does not impose special permissions to access the motion sensors. This means that any Trojan application with a convincing benign purpose has unlimited access to those sensors. This fact has driven many researchers to advocate the enforcement of more restrictions on sensors access through security permissions, similar to those required to access more traditional sensors like the camera and sound recorder.

6 Conclusions

Researchers have already started the study of the potential of integrated sensors on modern smartphones as a privacy threat. Otherwise useful in numerous benign applications, certain sensors could unwantedly reveal sensitive information related to the input of the user. Accelerometers and gyroscopes in particular, among other motion sensors, proved viable in tracking subtle (shakes) resulting from taps or keystrokes on the smartphone touch screen. Selecting the most appropriate sensor or combination of sensors, along with the right features, has much left to be researched. In this paper, we reviewed only a small set of the possible combinations of sensors data and their treatment in the implementation of the new keystrokes inference attack, specifically on Android platforms. The objective was to compare the relative performance of different built-in motion sensors, with respect to this attack.

Four sensors were compared in an experiment that implements the attack, namely, the linear accelerometer, gyroscope, rotation vector sensor, and the combined accelerometer and magnetometer synthetic sensor. Moreover, the option of fusing more than one sensor is examined. To enrich the discussion further, a couple of processing options for the accelerometer (filtering) and the gyroscope (integration) are considered. The results showed a greater benefit of exploiting the gyroscope sensor, or a fusion of several sensors (perhaps excluding the sensors with magnetometer component) from the perspective of an attacker. As discussed in the article, appealing processing in many other applications of sensors could fall short of serving the purpose of the attack.

For a future work, we vision an integrated solution that incorporates more advanced processing of the accelerometer and gyroscope readings to calculate the resulting tilt, rather than rate of change, of the user taps. Fusing the right balance of these two sensors in terms of features and preprocessing has the most promising performance, we believe, based on the current results.

Acknowledgements

This research is supported by the Ministry of Higher Education under research grant LRGS/TD/2011/UKM/ICT/02/02.

References

[1] International Jawsware, Interactive Overlay, available at: http://www.jawsware.mobi/code_OverlayView/ (20 November 2013).

[2] Aviv, A.J., Gibson, K., Mossop, E., Blaze, M. & Smith, J.M., *Smudge Attacks on Smartphone Touch Screens*, in Proceedings of the 4th USENIX Conference on Offensive technologies, USENIX Association Washington, DC, USA, 9 August 2010, pp. 1-7, 2010.

[3] Schlegel, R., Zhang, K., Zhou, X., Intwala, M., Kapadia, A. &Wang, X., *Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones*, in Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS), Internet Society, San Diego, California, 6-9 February 2011, pp. 17-33, 2011.

[4] Maggi, F., Volpatto, A., Gasparini, S., Boracchi, G. & Zanero, S., *A Fast Eavesdropping Attack Against Touchscreens*, in Information Assurance and Security (IAS), 2011 7th International Conference on, Mir Labs, Malacca, Malaysia, 5-8 December 2011, pp. 320-325, 2011.

[5] Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T. & Campbell, A.T., *A Survey of Mobile Phone Sensing*, Communications Magazine, IEEE, **48**, pp. 140-150, 2010.

[6] Cai, L., Machiraju, S. & Chen, H., *Defending Against Sensor-Sniffing Attacks on Mobile Phones*, in Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds, ACM, Barcelona, Spain, 16-21 August 2009, pp. 31-36, 2009.

[7] Motion Sensors, available at: http://developer.android.com/guide/topics/ sensors/sensors_motion.html (20 November 2013).

[8] Cai, L. & Chen, H., *TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion*, in Proceedings of the 6th USENIX Conference on Hot Topics in Security, pp. 9-9, 2011.

[9] Owusu, E., Han, J., Das, S., Perrig, A. & Zhang, J., *Accessory: Password Inference Using Accelerometers on Smartphones*, in Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications, ACM, San Diego, CA, USA, 28-29 February 2012, p. 9, 2012.

[10] Xu, Z., Bai, K. & Zhu, S., *Taplogger: Inferring User Inputs on Smartphone Touchscreens Using On-Board Motion Sensors*, in Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM, Tucson, Arizona, USA, 16-18 April 2012, pp. 113-124, 2012.

[11] Cai, L. & Chen, H., *On the Practicality of Motion Based Keystroke Inference Attack*, in Trust and Trustworthy Computing, ed: Springer, pp. 273-290, 2012.

[12] Miluzzo, E., Varshavsky, A., Balakrishnan, S. & Choudhury, R.R., *Tapprints: Your Finger Taps Have Fingerprints*, in Proceedings of the

10th International Conference on Mobile Systems, Applications, and Services, ACM, Low Wood Bay, Lake District, United Kingdom, 25-19 June 2012, pp. 323-336, 2012.

[13] Aviv, A.J., Sapp, B., Blaze, M. & Smith, J.M., *Practicality of Accelerometer Side Channels on Smartphones*, in Proceedings of the 28th Annual Computer Security Applications Conference, ACM, Orlando, FL, USA, 03-07 December 2012, pp. 41-50, 2012.

[14] Al-Haiqi, A., Ismail, M. & Nordin, R., *On the Best Sensor for Keystrokes Inference Attack on Android*, Procedia Technology, **8**, pp. 947-953, 2013.

[15] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I.H., *The WEKA Data Mining Software: An Update*, ACM SIGKDD Explorations Newsletter, **11**, pp. 10-18, 2009.

[16] Breiman, L., *Bagging Predictors*, Machine Learning, **24**, pp. 123-140, 1996.

[17] Gama, J., *Functional Trees*, Machine Learning, **55**, pp. 219-250, 2004.

Quadrangle Detection Based on A Robust Line Tracker Using Multiple Kalman Models

Hung Kwun Fung & Kin Hong Wong

Department of Computer Science and Engineering,
HSH Engineering Building, The Chinese University of Hong Kong, Shatin, Hong Kong
Email: khwong@cse.cuhk.edu.hk

Abstract. Quadrangle and line tracking are essential for many real world applications of computer vision. In this paper, we propose a computationally efficient line tracker that can robustly and accurately track lines in an image. We use a multiple-model-Kalman filter (MMKF) scheme, which can handle line tracking accurately and robustly. The basic idea is to run N multiple sub-Kalman filters in parallel. Each filter is configured to use a different state transition model. All the filters are updated by the measurement at the same time following the conventional Kalman filter update process. The final prediction is a combination of outputs from all the Kalman filter modules. After lines are detected, we developed a scheme to merge the lines together to become suitable quadrangles. The experimental result shows that the proposed system can track lines and quadrangle robustly in real time. The result is useful in shape detection and should be suitable for building many mobile projector applications.

Keywords: *hough transform; line tracking; multiple kalman filter; quadrangle detection.*

1 Introduction

In many applications, there is often a need to track rectangular and quadrangular objects. For example, in our previous work, there is a need for detecting and tracking quadrangles from images to assist the projector-camera system for displaying virtual objects on a piece of white card board or paper [1]-[2].

In such a system, a camera and a projector are mounted together and facing the same direction.as in Figure 1. A user can move the board freely at a distance away and the camera is capturing and calculating the position of the board relative to the camera-projector pair. Hence, the desired image can be projected correctly onto the board. The user can freely move this very lightweight board as an interface to the computer to view movies, pictures etc. There are plenty of applications for such systems. For example, it can be used for a passenger in a flight to watch movies, so the user does not need to view the fixed TV panel in front of him. Moreover, this can also be used as a teaching tool for medical students, in that the student doctor can examine the internal view of a human

Received September 19th, 2013, 1st Revision October 5th, 2013, 2nd Revision November 21st, 2013, Accepted for publication November 27th, 2013.

body by moving the board up and down to select the correct cross-sectional view of the target object. The application potential for this setup is enormous. To achieve the above goal, we need to detect the board, which appears to be a quadrangle in the image captured by the camera. Therefore, a robust quadrangle detector is required. However, the previous system has a time lag problem, when the board is moving too fast the detection may take up to 0.5 seconds to complete, hence the image may appear to be lagging behind the board. Therefore, a more efficient algorithm for quadrangle detection is a necessity. In previous systems, the Canny edge detector and Hough line transform [1]-[3] are used to obtain the line features. Then an exhausted search over the line combinations is carried out to find four lines which can form a good quadrangle. Finally, the four corners of the detected quadrangle are tracked with a particle filter. Other approaches such as the one by Zhang's [4] is slightly different; it projects the edge map from a Sobel edge detector to the Hough space and carries out the exhausted search directly in the Hough space. Then, when a quadrangle is detected, dynamic programming is employed to track the pose and location of the detected quadrangle. We think that although these detection methods can achieve acceptable results, the performance can certainly be improved by adding Kalman filter based tracking.

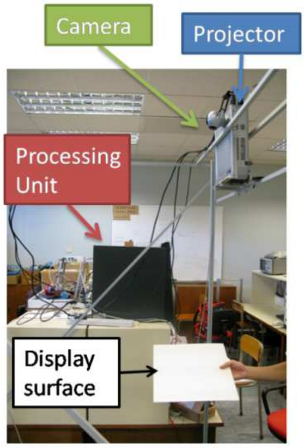


Figure 1 The camera projector system used in our previous work [1]-[2].

We found that line tracking by Kalman filter (KF) [5] and extended Kalman filter (EKF) have been investigated in the work of Mills [6]-[7], the tracking procedure is performed in the Hough space for reliable line tracking even partial occlusion exists. And in [8], Condensation [9] is also employed for tracking the set of lines in the Hough space. However, poor modeling in the Kalman filter may lead to large estimation error [10]. For example, if the line is too close to the origin, Hough-based feature detection (that Kalman filter depends on) would be difficult to function. To address this problem, a mathematical scheme called Multiple Model Adaptive Estimation algorithm (MMAE) is proposed to be used [11] and the Multiple Model Kalman filter (MMKF) is used as a stochastic version of MMAE in [11] and [12]. In this paper, we investigate the use of MMKF and apply it to line tracking. We believe the MMKF is useful to improve the robustness and stability with little sacrifice of speed. The proposed MMKF method is not only useful for tracking lines but may also be extended to handle other shapes such as polygons etc. We believe using this method we can produce a robust line detector for even for low-cost mobile camera-projector systems. Hence, a quadrangle detector based on this line detector can be achieved. In Section 2, the rationale and the design of our method will be discussed. Section 3 is about the details of the implementation. Experimental results are discussed in Section 4 and we conclude the whole paper in Section 5.

2 Theory and Design

2.1 Introduction to Quadrangle Detection and Tracking

In many applications such as a projector-camera system [1], the display content has to be projected onto the correct display rectangular area. To achieve this, the system must able to locate the display area and find out the pose of it. Since the display area can move freely in the 3D camera view frustum and the projection frustum, the rectangular display area is likely to be deformed into a quadrangle by a perspective transformation. Therefore, we need a method to search for quadrangles, which are built from lines. Hence, a robust line detector is needed in such applications.

2.2 Definition of a Quadrangle

As mention in the previous section, the display area we targeted is a rectangle. However, this rectangle is deformed into a quadrangle by perspective transformation. The *i-th* quadrangle $Q_{t,i}$ is considered as a set of 4 straight lines extracted in the image I_t at time t and described as $Q_{t,i} = \{l_p, l_q, l_r, l_s\}_t$,where l_p, l_q, l_r, l_s, are the distinct lines in the image I_t at time t and they are arranged in an anti-clockwise order. A quadrangle is shown in Figure 2(b).

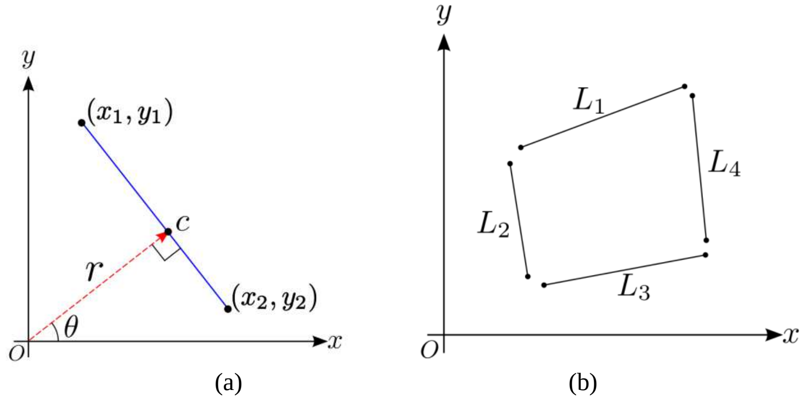


Figure 2 (a) Graphical illustration of the definition of a line. (b) Illustration of the quadrangle defined from lines *Lp*, where *p=1, q=2, r=3, s=4.*

2.3 Line Feature Extraction in Our System

The line feature is extracted by first using Progressive Probabilistic Hough transform (PPHT) [13]. It takes a binary edge image as input and output a set of lines that has votes larger than a threshold. There are two more parameters, the minimum line length and the maximum gap length. The minimum line length not only specifies the type of line the algorithm output but also suppresses false positive output. Due to rounding error, a long line segment may be decomposed into many short line segments. The maximum gap length helps joining those short line segments together. The threshold, the minimum line length and the maximum gap length are specified by the user. However, the values of the parameters are dependent on the application and the scene. Hence, the user has to adjust the values for different applications and scenes. Unlike the original Hough Transform, PPHT has a shorter execution time and, therefore, it is used in our application. On the other hand, PPHT returns the two end points of the lines while the original Hough Transform returns the lines in the form of *r,θ* pair. Although extra steps have to be taken to find the *r,θ* value in PPHT, it is still faster than finding the two end points using the edge map and *r,θ* pair.

2.4 Automatic Quadrangle Detection

If the system is designed for graphic display applications such as that in [1], the quality of the image projected on the surface is important. To facilitate a good projection, the rotation angle of display area should not be too large. Moreover, the distance between the projector-camera pair and the display should be in a proper range. It is because if the projector is too far away, the projected image might not be bright enough for human eyes under bright environment. If they are too close, the whole display area might not be seen by the camera. Therefore, we put a upper limit and a lower limit on the length of l_p, l_q , l_r *and* l_s. The corners of the quadrangle should also be inside the camera view. Because this quadrangle is a deformed rectangle by perspective transformation and the rotation of the display area is small, we observed that there are some useful properties in the quadrangle for detection. With the help of those properties, the system can quickly locate the quadrangle (instead of exhaustively exploring every quadrangle) found in the image.

Therefore, our aim is to develop a set of rules for quadrangle detection. The first observed property is that the opposite sides are almost parallel. The angle between the lines l_p, l_r of the quadrangle is small. Moreover, it is true for the line l_q *and* l_s. Similarly, the second property is that the adjacent sides are almost perpendicular to the each other. That means the angle between the adjacent lines is around 90°. The last property is the ratio between the adjacent sides. These ratios are close to the original rectangle. We derived the detection criteria from all these observations. A candidate of the possible display area is a quadrangle that passes the detection criteria as follows:

1. The opposite sides are almost parallel.
2. The adjacent sides are almost perpendicular to each other.
3. The corners of the quadrangle should be inside the image.
4. The lengths of the four sides are within the lower limit to the upper limit.
5. The ratio between the adjacent sides is close to the display area.

In Figure 3, (a) is the image frame captured, (b) is the edge map produced by using Canny edge detector, (c) is the lines detected using progressive probabilistic Hough transform and (d) is the result of drawing the lines on current frame. Lines are extracted from the image. Let us assume N lines were found in the current image frame I_t. The total number of quadrangles can be formed by all these N lines is

$$\binom{N}{4} = \frac{N!}{4!(N-4)!} = \frac{N(N-1)(N-2)(N-3)}{4!} \tag{1}$$

Since the number of quadrangles grows with $O(N^4)$, it is impractical to examine all the quadrangles. Instead of using all N lines to form quadrangles, the system can select distinctive lines, uses them to construct quadrangles and tests them with the criteria. We observed that a long strong edge may be divided into multiple short lines during the line detection phase. These small segments will induce a lot of duplicated parallel line pairs and perpendicular line pairs. All the duplicated line pairs are referring to the same line pair formed with the same strong edges. Therefore, the time spend on the calculation on the duplicated line pairs is wasted. To overcome this problem, data clustering techniques will first be applied to the set of lines and similar lines are put into the same cluster. Similar lines are determined by calculating the similarity score defined as the following.

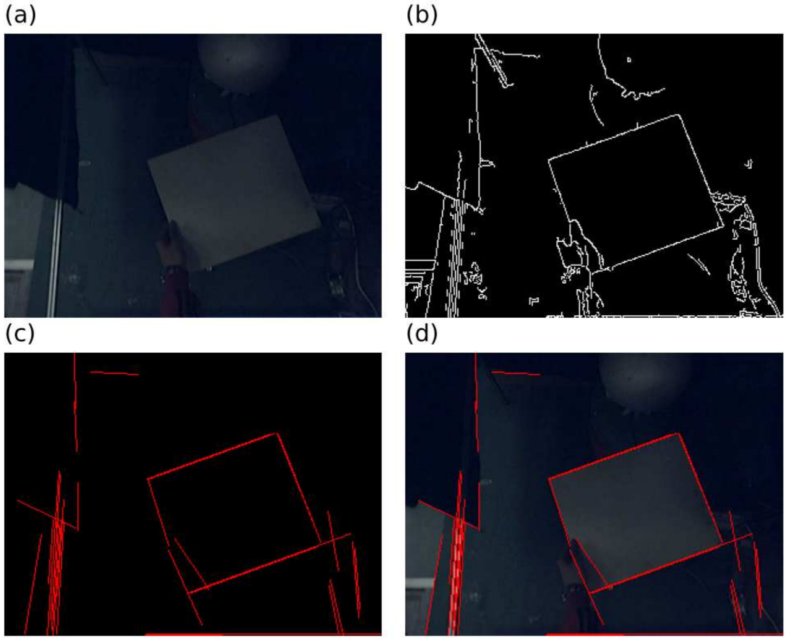


Figure 3 Edge points and lines extracted from a real image.

The similarity measurement *$S(l_i, l_j)$* is the Euclidean distance between the closest point of the two line l_i and l_j . The closest point c_i of the line l_i is

$$\left(r_i \cos(\theta_i), r_i \sin(\theta_i)\right), hence$$

$$S(l_i, l_j) = S(l_j, l_i) = \sqrt{(c_i - c_j)(c_i - c_j))}$$

The lines are similar if and only if $S(l_i, l_j) \leq threshold$

Similarly, we can define lines that are (i) parallel, (ii) perpendicular in a similar fashion. Therefore those lines that satisfy parallel and particular criteria will be grouped together to become quadrangles.

2.5 Line Tracker and the Problem when the Line is Close to the Origin

A line segment can be defined by its two end-points. However, occlusion of these two end-points is unavoidable in many real world situations. In this paper, a line L is represented by using the conventional polar coordinates of *r* and θ [3]. The point *C* is the closest point between the line and the origin (of the coordinate system used), and *r* is the shortest distance between the origin to *C*. In addition, *θ* is the angle between the horizontal axis and the vector from the origin to *C*. The *x*-*y* coordinate of *C* is tracked using the Kalman filter.

A single Kalman filter scheme alone cannot yield good result especially when the target line is near (e.g. *r* is only a few pixel long) or almost passing through the origin. To solve this problem, we propose to use multiple independent Kalman filters to track a line, i.e., each Kalman filter uses a different coordinate system with a different center (origin). To make this effective, we need to select the relative positions of the *N (≥3)* coordinate systems so that the origins are not collinear. The effect is even if a line is close to the origin(s) of one or two coordinate systems it would be far away from the origins of the other coordinate systems. We believe that by using this multiple models scheme, the prediction accuracy and reliability can be improved.

2.6 Kalman Filter Design

The *x*,*y* coordinates of the closest point C of a line L is being tracked using the Kalman filter (KF). We assume the closest point is moving with a constant acceleration, and the motion in both *x*-axis and *y*-axis are independent and linear. The motion of the closest point C can be described using the equation of motion as follows:

$$\begin{aligned}
x_t &= x_{t-1} + \dot{x}_{t-1}\Delta t + \frac{1}{2}\ddot{x}_{t-1}\left(\Delta t\right)^2 \\
\dot{x}_t &= \dot{x}_{t-1} + \ddot{x}_{t-1}\Delta t \\
y_t &= y_{t-1} + \dot{y}_{t-1}\Delta t + \frac{1}{2}\ddot{y}_{t-1}\left(\Delta t\right)^{2,} \\
y_t &= \dot{y}_{t-1} + \ddot{y}_{t-1}\Delta t
\end{aligned} \tag{2}$$

where $x_t, \dot{x}_t$ and $\ddot{x}_t$ are the displacement, velocity and acceleration respectively of the closet point C in frame i in the *x*-direction. Similarity, $y_t, \dot{y}_t$ and $\ddot{y}_t$ are the displacement, velocity and acceleration respectively of the closet point C in frame i in the *y*-direction. Δt is the time interval between two consecutive frames. The state matrix and the measurement matrix of the Kalman filter at time t are:

$$X_t = \begin{bmatrix} x_t & \dot{x}_t & y_t & \dot{y}_t \end{bmatrix}^T, \; Z_t = \begin{bmatrix} x_t & y_t \end{bmatrix}^T \tag{3}$$

Derived from the equations of motion, the state transition model *A* and the observation model H are:

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4}$$

$$Q = \begin{bmatrix} \frac{(\Delta t)^4}{4} & \frac{(\Delta t)^3}{2} & 0 & 0 \\ \frac{(\Delta t)^3}{2} & (\Delta t)^2 & 0 & 0 \\ 0 & 0 & \frac{(\Delta t)^4}{4} & \frac{(\Delta t)^3}{2} \\ 0 & 0 & \frac{(\Delta t)^3}{2} & (\Delta t)^2 \end{bmatrix}, \quad R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \tag{5}$$

Where the noise covariance is *Q* and measurement noise covariance is *R*.

2.7 Our Multiple-Model Kalman filter Line Tracing Algorithm

The multiple-model Kalman filter (MMKF) paradigm is used in our tracker. Figure 4 shows the overview of our line tracker. Z_k is the measurement at time *k*. And it is transformed to the coordinate system of the *i-th* Kalman filter (denoted as KF_i) by the transformation T_i into $Z_{k,i}$. State $X_{k,i}$ is the prediction of the state of the closest point in the next frame by the *i-th* Kalman filter. All the prediction results are transformed back to one common coordinate system by applying the corresponding inverse transformation $(T_i)^{-1}$. Finally, a line fitting function will be applied to find the final prediction of the line.

The advantage is discussed in [12]. In our tracker, each filter uses a different coordinate system. And if the origins of all these coordinate systems are not collinear (as discussed earlier), the line cannot pass through these origins at the same time. This is to ensure that even when *r* in one model is small, the values of *r* in other models are large enough to give a correct result. The prediction of each sub-Kalman filter is the position and velocity of the closest point. By discarding the velocity terms, we can obtain a set of closest (C) points that should lie on a straight line as shown in Figure 3(a). We can use a least squares fitting method to find a best fit line that minimizes the sum of square errors among the line and closest (C) points.

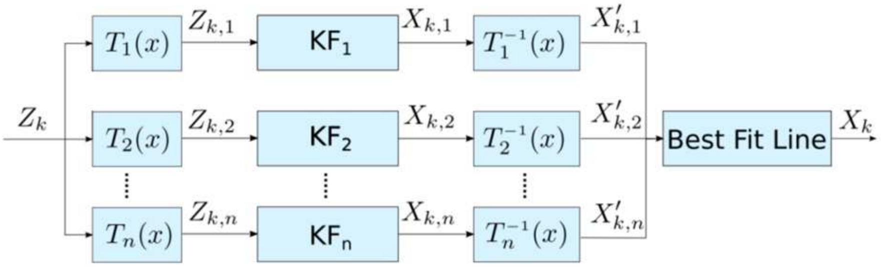


Figure 4 The overview of the line tracker. KFi=the i-th Kalman filter.

3 Experiments

To evaluate the robustness and stability of our proposed trackers, we generate a set of synthetic data to simulate different kinds of movements, e.g. motion with small *r* and complex motions with mixed translation or rotation. In the data set, the two end-points of the lines are given. The length of the line is not fixed for some data set to simulate the lines extracted in real images. The overlap ratio, defined in [12], is used to evaluate how good the line tracker is performing. We first draw the predicted line (ground truth) with n pixels thick (it becomes a ground truth range of a line) and draw the line found by our algorithm on top of it.

The overlap ratio is the percentage of the number of pixels of the line detected inside the ground truth range over the total number of pixels of the line detected. If the line found is completely inside the ground truth range, the overlap ratio is 100% and so on. The width of the ground truth range is to provide a certain tolerance for the results. Because there may be error accumulated in different processes like Canny edge detection, Hough transform and rounding error in the calculation. The width is set to 2 pixels throughout all the experiments conducted, see [12] for details of the definition of overlapping ratio.

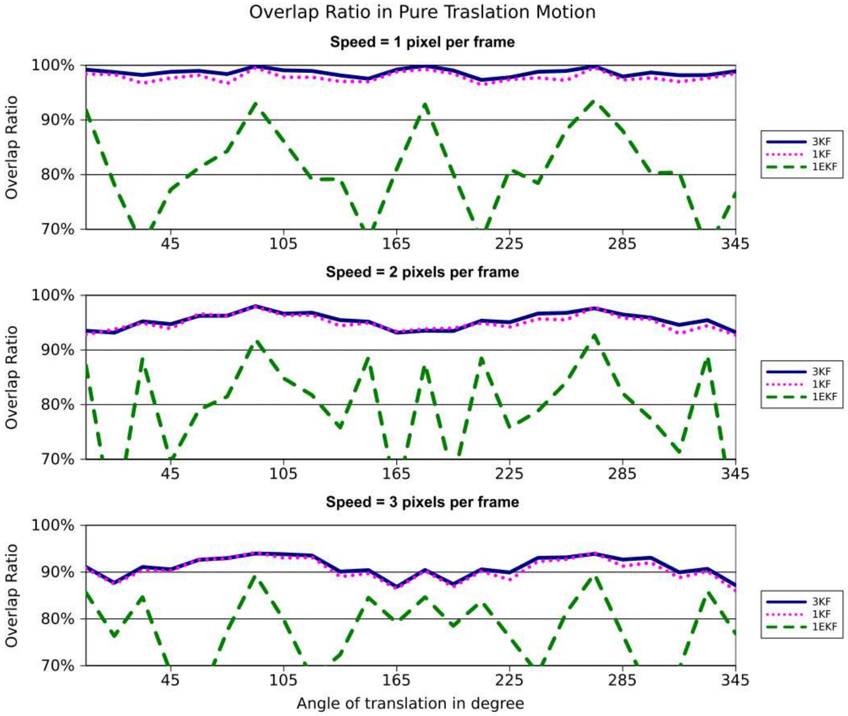


Figure 5 Overlap ratio in Pure translation. Our approach is 3KF.

3.1 Experiment 1: Translation Performance

To compare the performance between trackers, 20 randomly generated line segments with pure translation motion in 24 directions (0^o, $15^{\ o}$, $30^{\ o}$ and so on until $345^{\ o}$) and 3 different speeds (1, 2, 3 pixels per frame) are tested. The trackers are tested in 20 consecutive frames. The average score of all lines in the same directions are shown in Figure 5. We can see that the proposed 3KF can slightly improve the performance in term of accuracy and stability.

3.2 Experiment 2: Rotation Performance

In this experiment, the value of r of the line is fixed to 10, 50 and 100 pixels. For each value of r, the angular velocity is set to 1^o, 2^o, 4^o , 8^o and 16^o degrees per frame. We record the performance of the trackers in 20 frames. The average scores are shown in Figure 6. The score of 3KF drops below 20% when the angular velocity reaches 4 degrees per frame and higher for all tested value or r.

3.3 Experiment 3: Real Video Test

In a real video test, a white cardboard is put on a silver screen so that the cardboard can be easily detected. The camera is moving freely but slowly. Figure 7, in the first row, the left picture shows the snapshot of the system at frame 41 of the testing video, the right picture shows the quadrangle detected. Figure 8 shows the case for frame 42. The system is now using the predictions of the trackers to form the quadrangle.

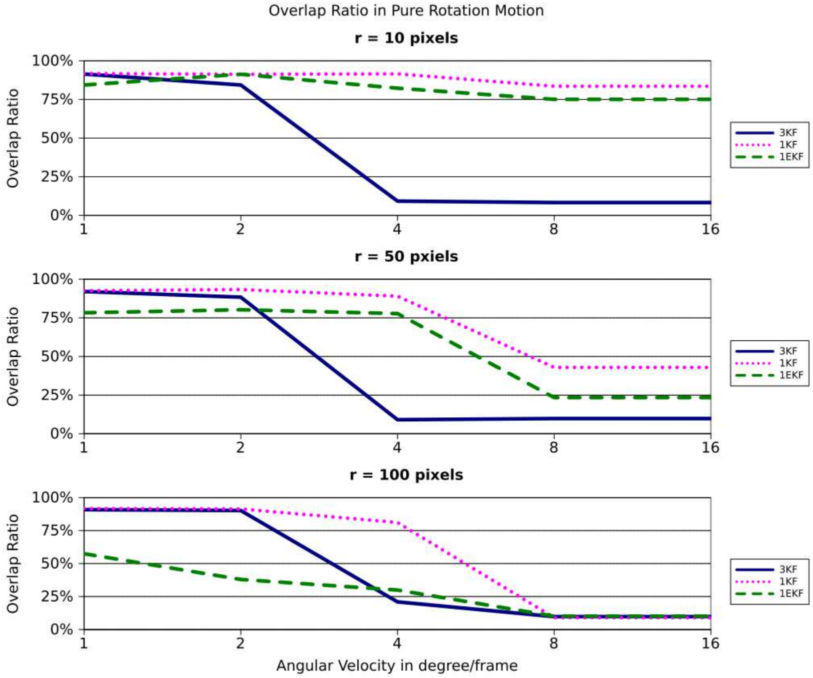


Figure 6 Overlap ratio in pure rotation motion. Our approach is 3KF.

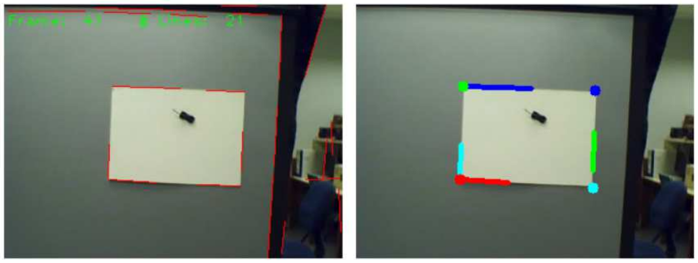


Figure 7 Lines (left) and a quadrangle (right) is detected and shown at frame 41.

Figure 8 shows the case for frame 43 of the sequence. The system failed to extract lines for the top and bottom of the board. However, we can use the prediction ability of the trackers for the top side and bottom side to form a quadrangle using our Real-time Quadrangle Tracking Method mentioned in Section 2.

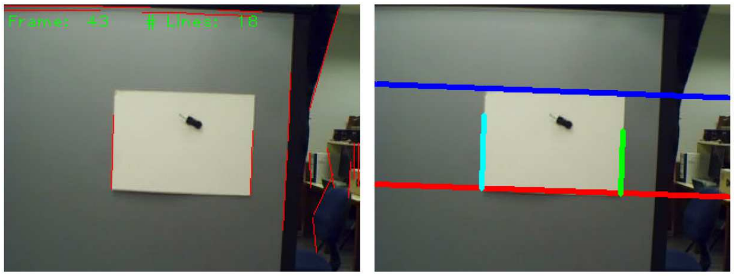


Figure 8 Lines detected and a quadrangle le is tracked at frame 43.

4 Conclusion

The work investigates how we can robustly detect and track a quadrangle in real time. Since a quadrangle composed of four lines, so a robust line tracker is required. A line tracking method adopting the Multiple Model Kalman filter paradigm is proposed in this paper. Each Kalman filter works on a state transition scheme based on a particular coordinate system. By combining all the outputs of these Kalman filter modules, we can produce robust line tracking results. In our tests, we show that our three-Kalman method outperforms the use of a single-Kalman or two-Kalman filter schemes especially when the line is close to the origins of one or two coordinate systems. It is also suggested that the choices of the locations of the coordinate systems may affect the performance, and selection of these coordinates may be a future research direction. The lines detected are then used for quadrangle detection. Our algorithm is also very efficient, and even a small embedded system can process a line at 200 frames per second. We believe using this method can produce a robust line and quadrangle detection module for low-cost mobile applications. The next step is to merge it with our existing camera-project based handheld display system [1] to improve its speed and accuracy. As mentioned in the introduction there is time-lag problem when the board is moving too fast, we hope this efficient quadrangle can solve this problem. Moreover, we are also exploring a hardware approach for line detection and quadrangle detection.. As we have seen, the line detector requires the Canny edge detector which takes up a heavy resource for the calculation of the image gradients. We found that image gradients can be easily calculated by simple subtraction of pixel values direction sent from the camera image bit stream, and it can be implemented by logic gates using an FPGA (field programmed gate array) by a hardware language VHDL or VERILOG. The advantage is not only it is efficient, it is also economical and power saving because expensive and power-hungry math processors can be replaced by simple logic gates. So, our projector-camera system can be easily integrated into mobile phones or other mobile devices.

Acknowledgement

This work is supported by a direct grant (Project Code: 2050455) from the Faculty of Engineering of the Chinese University of Hong Kong.

References

[1] Leung M.L., Lee K.K., Wong K.H. & Chang M.M.Y., *A Projector-based Hand-held Display System*, IEEE CVPR09 , Miami, Florida, United States, pp.1109 – 1114, June 22-25, 2009.

[2] Lee K.K., Leung M.C., Wong K.H. & Chang M.M.Y., *A Hand-held 3D Display System that Facilities Direct Manipulation of 3D Virtual Objects*, VRCAI2009 in Yokohama, pp.65-70, 14-15 Dec, 2009.

[3] Duda R.O. & Hart P.E., *Use of the Hough Transformation to Detect Lines and Curves in Pictures*, Commun. ACM, 15, pp.11–15, January 1972.

[4] Zhang, Z., *Visual Panel: Virtual Mouse Keyboard and 3D Controller with an Ordinary Piece of Paper*, Workshop on Perceptive User Interfaces. ACM Press, pp.1-8, 2001.

[5] Welch, G. & Bishop, G. *An Introduction to the Kalman Filter,* Technical report, Chapel Hill, NC, United States, 1995.

[6] Hills, M., Pridmore, T., & Mills, S., *Object Tracking through A Hough Space*, Proc. Visual Information Engineering, VIE 2003, pp.53-56, 2003.

[7] Mills, S., Pridmore, T. & Hills, M., *Tracking in a Hough Space with the Extended Kalman Filter*, In Proceedings of the British Machine Vision Conference (BMVC2003), pp. 173-182, September 2003.

[8] French, A., Mills, S. & Pridmore, T., *Condensation Tracking through A Hough Space*, ICPR2004, Cambridge, United Kingdom, **4**, pp.195-198, 2004.

[9] Isard, M. & Blake, A., *Condensation: Conditional Density Propagation for Visual Tracking*, International Journal of Computer Vision, **29**(1), Kluwer Academic Publishers, pp. 5-28, 1998.

[10] Fitzgerald, R., *Divergence of the Kalman filter*, IEEE Transactions on Automatic Control, **16**(6), pp. 736-747, 1971.

[11] Aguiar A.P., Hassani V., Pascoal A.M., & Athans M., *Identification and Convergence Analysis of A Class of Continuous-Time Multiple-Model*

Adaptive Estimators, In Proceeding of the 17th IFAC World Congress, Seoul, South Korea, 2008.

[12] Fung, H.K. & Wong, K.-H., *A Robust Line Tracking Method Based on A Multiple Model Kalman Filter Model for Mobile Projector-Camera Systems*, The 4th Int. Conf. on Electrical Engineering and Informatics (ICEEI 2013), Kuala Lumpur, Malaysia, 24-25 June, 2013.

[13] Matas, J., Galambos, C. & Kittler, J., *Robust Detection of Lines using the Progressive Probabilistic Hough Transform*, Comput. Vis. Image Underst., **78**, pp. 119-137, April 2000.

Wideband and Multiband Antenna Design and Fabrication for Modern Wireless Communications Systems

Adit Kurniawan & Salik Mukhlishin

School of Electrical Engineering and Informatics, Institut Teknologi Bandung,
Jl. Ganesa 10 Bandung 40135, Indonesia
Email: adit@stei.itb.ac.id

Abstract. One of the major challenges due to spectrum scarcity in modern wireless communication is on antenna design that can serve a non-contagious frequency spectrum. In this paper, wideband and multiband design approaches are proposed to produce antennas that can serve various wireless technologies using different frequencies from 2.3 to 6.0 GHz, covering WiFi frequencies at 2.4-2.48 GHz, 5,15-5,35 GHz and 5,725-5,825 GHz, as well as WiMax frequencies at 2.3-2.4 GHz, 2.495-2.695 GHz, 3.3-3.8 GHz, and 5.25-5.85 GHz. The wideband and multiband antenna were implemented on an 0.8 mm thick of FR4 epoxy dielectric substrate with permittivity ε_r=4.3. The return loss of 10 dB can be achieved for 2.3 to 6 GHz in wideband antenna, and a tripe band of 2.3-2.8 GHz, 3.3-3.7 GHz, and above 4.6 GHz in multiband antenna. The gain of both antennas increases almost linearly from 0 dB at 2.3 GHz to around 4.5 dB at 6 GHz, except for the stop band at 2.8-3.3 GHz which has a significant drop of gain, corresponding to the stop frequency band. Antenna radiation pattern is bidirectional at x-y plane and nearly omnidirectional at y-z plane, and shows a similar patern for both wideband and multiband design approaches.

Keywords: *antenna; microstrip; multiband; wideband; wireless communications.*

1 Introduction

Rapid development in wireless communications systems has demanded multiband or wideband antennas to support different technologies and standards. Wireless Local Area Network, now has various standards, i.e IEEE 802.11 b/g as one of WLAN standard operating at frequency ranging from 2.4 GHz to 2.48 GHz, while IEEE 802.11a/g is using frequency from 5.15 GHz to 5.35 GHz or 5.725 GHz to 5.825 GHz. Those WLAN standards are designed for short range of up to approximately one or two hundred meters from the transmitter (WiFi standard). Other standards, such as IEEE 802.16 d/e is designed to obtain wider coverage operating at other frequencies, and is known as WiMax technology. WiMax has various frequency allocations and differs from country to country. In Indonesia frequency bands of 2.3-2.4 GHz and 3.3-3.7 GHz have been assigned to support WiMax technology. Therefore, there is a

Received October 1st, 2013, Revision November 10th, 2013, Accepted for publication November 28th, 2013.
ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2013.7.2.4

need to produce antennas that can accommodate different frequency bands to support different technologies and standards.

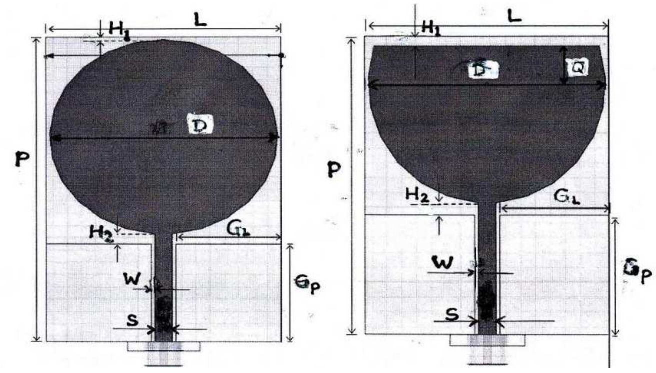
There are several papers on dual band or multi band antennas to comply with the 802.11 a/b/g and IEEE 802.16 d/e standards. Liu, *et al.* [1] proposed inverted-F antenna that can cover the 2.4/5.2 GHz WLAN bands. Raj, *et al.* [2] proposed coplanar antenna printed on FR4 which operates on 2.4/5.2/5.8 GHz bands. Wu, *et al.* [3] reported dual broadband slot antenna, in which two wide resonances were obtained by using a U-shaped strip inset at the center of the slot antenna on a substrate with relative permittivity of 4.7. Another technique to make a compact microstrip antenna can be found in [4], which also present many issues related to recent development in designing compact antenna technologies. Motivated by Chen, et.al [5]-[7] and also by [8], in this paper we propose to design and fabricate an antenna prototype that has wideband and multiband charactersitics.

The rest of the paper is organized as follows. Section 2 presents the wideband and multiband antenna design and simulation. Section 3 describes antenna fabrication and measurement. Section 4 present analysis and discussion, and finally, section 5 summarizes and concludes the entire works.

2 Antenna Design and Simulation

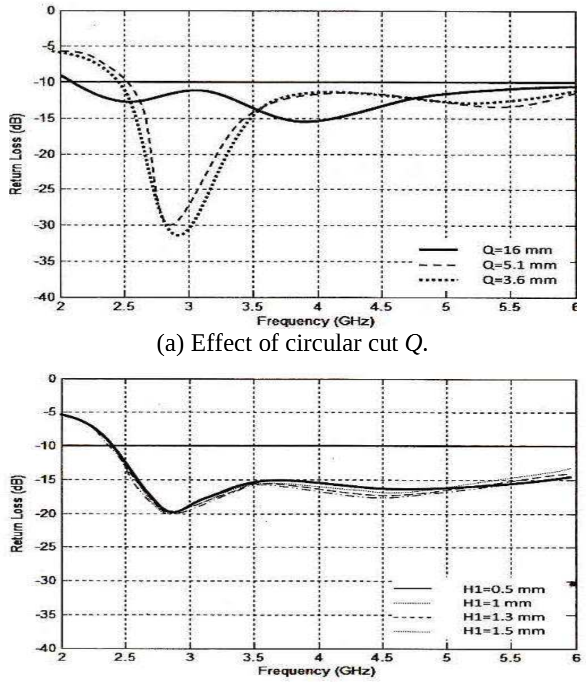
2.1 Wideband Design Approach

Wideband antenna in this proposed work constitutes a circular patch implemented as shown in Figure 1(a), which is then modified by creating a circular cut to produce the desired wideband characteristics as shown in Figure 1(b). The proposed antenna was designed and implemented using FR4 epoxy dielectric substrate with dielectric permitivity ε_r = 4.3 and thickness h = 0.8 mm. Figure 1(a) shows the basic design of wideband antenna [9]. The antenna consist of circular patch and a CPW-fed line [10]-[11]. The CPW fed line consist of a center strip and two finite square patches of the same size that are situated symmetrically on each side of the center strip acting as groundplane. To produce a wideband characteristics between 2.3 – around 6 GHz frequency band, the circular patch has radius of Q = 16 mm or diameter D = 32 mm, a center strip width, S = 2.4 mm and a CPW gap, W = 0.2. The other dimension are P = 50.9 mm, L = 33 mm, G_P = 16 mm, G_L = 15.1 mm, H_1 = 1.3 mm, and H_2 = 1.6 mm. This antena has return loss of < -10 dB form 2.1 – 5.0 GHz showing a wideband characteristic. However in our case, we need to extend the design bandwidth of up to 5.9 GHz in order to cover high band WiFi and WiMax frequencies. To achieve that, we use similar technique with [12]-[13], in that we first truncate the circular patch by decreasing the length of Q as shown in Figure 1(b). It's observed that by cutting the circular patch, the bandwidth becomes wider, but it causes shifting the low frequency to a higher frequency limit. By properly adjusting the length of the cutting edge or decreasing the length *Q*, a new wideband frequency from 2.3 GHz to 6 GHz can be obtained.



(a) (b)

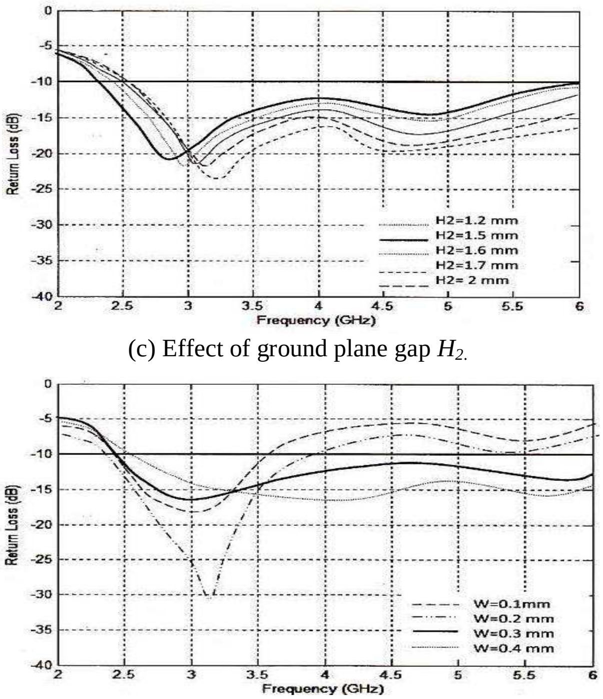
Figure 1 Wideband antenna: (a) Circular patch with 2-square ground plane, (b) Truncated patch to extend antenna bandwidth [13].



(a) Effect of circular cut *Q*.

(b) Effect of circular cut gap $H_{1.}$

Figure 2 Return loss characteristic of truncated circular patch antenna.



(c) Effect of ground plane gap H_2.

(d) Effect of strip line gap *W*.

Figure 2 ***Continued.*** Return loss characteristic of truncated circular patch antenna.

Our next step is to improve impedance matching at high frequency, as we can see from Figure 2 that return loss is still critical at frequencies higher than 3.5 GHz. To do that, we then conducted parametric study by adjusting the gap width, *W*, and the length H_1 and H_2. This process shows that by decreasing *W*, impedance matching at frequency higher than 3.5 GHz can improve, while by increasing H_2 return loss in all frequency can improves, but it is accompanied by shifting the frequency resonance to the higher frequency. In addition, increasing H_1 can deteriorate the return loss and also shifting the resonant frequency to the lower frequency. Optimum result to produce the desired frequency band for return loss characteristic less than 10 dB can be obtained when the dimension of Q = 5.6 mm, H_1 = 0.5 mm, and H_2 = 1.5 mm, W = 0.3 mm. By setting the overall dimension of P =40 mm and L = 33 mm, we obtain the dimension of D = 32 mm, G_P = 16.1 mm, G_L = 15 mm, and S = 2.4 mm. Fig. 2 shows simulation result from this process when we vary the length of Q, H_1, H_2, and W.

From Figure 2 we can see that to cover the lower band of 2.3 GHz, the dimension of Q = 16 mm shows better performance than other dimensions, the dimension of H_1 is rather loose, but the dimension of H_2 = 1.7 mm and W = 0.2 mm exhibit the best lower band characteristics.

2.2 Multiband Design Approach

The multiband antenna design approach was performed by imposing a pair of rectangular slot onto the circular patch in order to create stop band corresponding to the undesired frequency in the spectrum of between 2.3 to 6.0 GHz. Then otimizing the desired antenna bandwidth and center frequency was then performed by varying the dimension of rectangular slot L_s and P_s as shown in Figure 3. However, the gap of CPW fed line and the ground plane, W, the length of the circular cut, H_1, and the gap between circular patch and the ground plane, H_2 were affected and need to be readjusted.

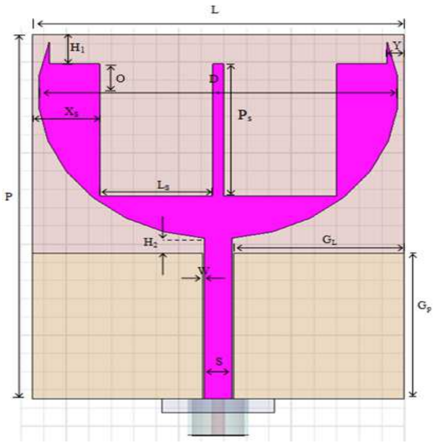
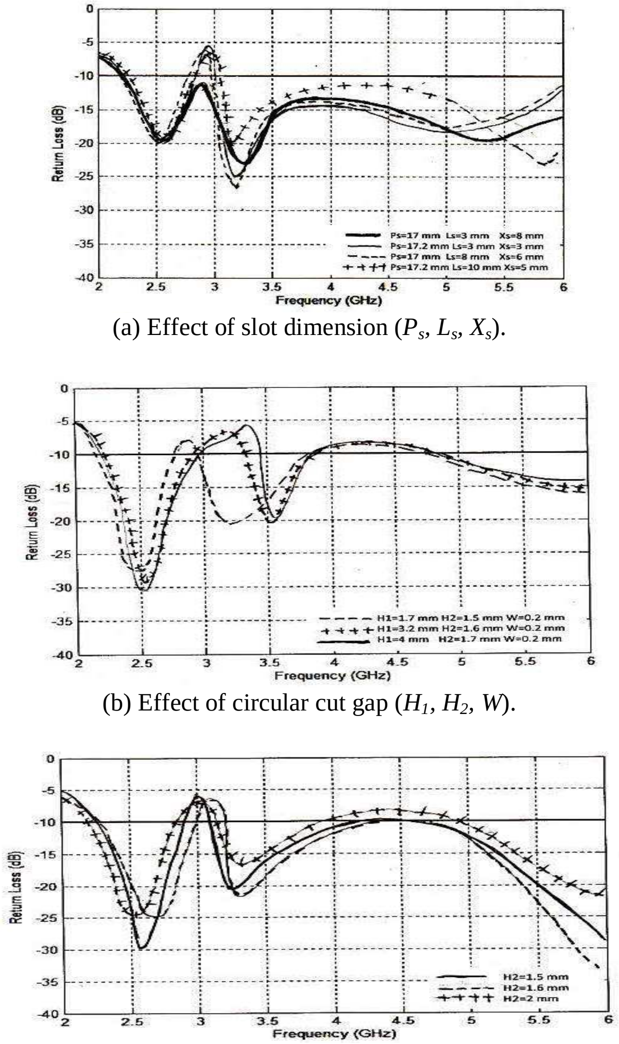


Figure 3 Design of the multiband antenna.

Figure 4 (a) shows the effect of rectangular slot dimension P_s and L_s, as well as the length of X_s on VSWR; while the effect of H_1 and H_2 are shown in Figure 4 (b) and (c) respectively.

From Figure 4 (a) we can see that the rectangular slot dimension to produce stop band characteristics around 2.8 GHz is better when P_s =17-17.2 mm, L_s = 8-10 mm, and X_s = 6 mm. Then the process is followed by readjusting the gap dimension of ground plane to obtain the desired pass bands characteristic. By keeping the dimension of W= 0.2 mm obtained from Figure 2(d), from Figure 4(b) and (c), we can see that the best obtainable pass bands that is most suitable for the desired frequency bands was found when the dimension H_1 = 3.2 mm, and H_2 =1.6 mm to produce tripleband characteristics, i.e. 2.3-2,8 GHz, 3.3-3.8 GHz, and the upper band above 4.6 GHz.



(a) Effect of slot dimension (P_s, L_s, X_s).

(b) Effect of circular cut gap (H_1, H_2, W).

(c) Effect of ground plane $H_{2.}$

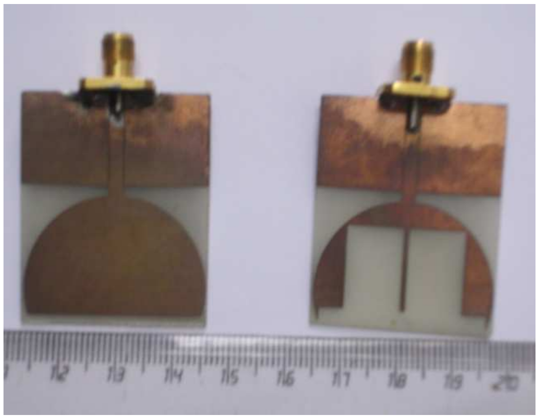
Figure 4 Simulated dimension of the multiband antenna design approach.

As we can see from Figure 4, the multiband antenna characteristics is triple band with the low band of 2.3-2.8 GHz to accomodate for WiFi/WiMax and new generation of cellular technology, the midle band of 3.3-3.7 GHz for

WiMax and extended C band applications, and above 4.6 GHz to provide for various modern wireless spectrum.

3 Antenna Fabrication and Measurement

The antenna prototypes fabricated using multiband and wideband design approach are shown in Figure 5(a) and (b), respectively. Antenna characteristics have been tested by measuring its parameters, particularly VSWR/return loss and antenna gain, to validate the simulation result as well as to verify the antenna design specification. Measurement of VSWR or return loss is most important because our proposed research is to produce wideband and multiband characteristics with sufficient impedance bandwidth requirement.

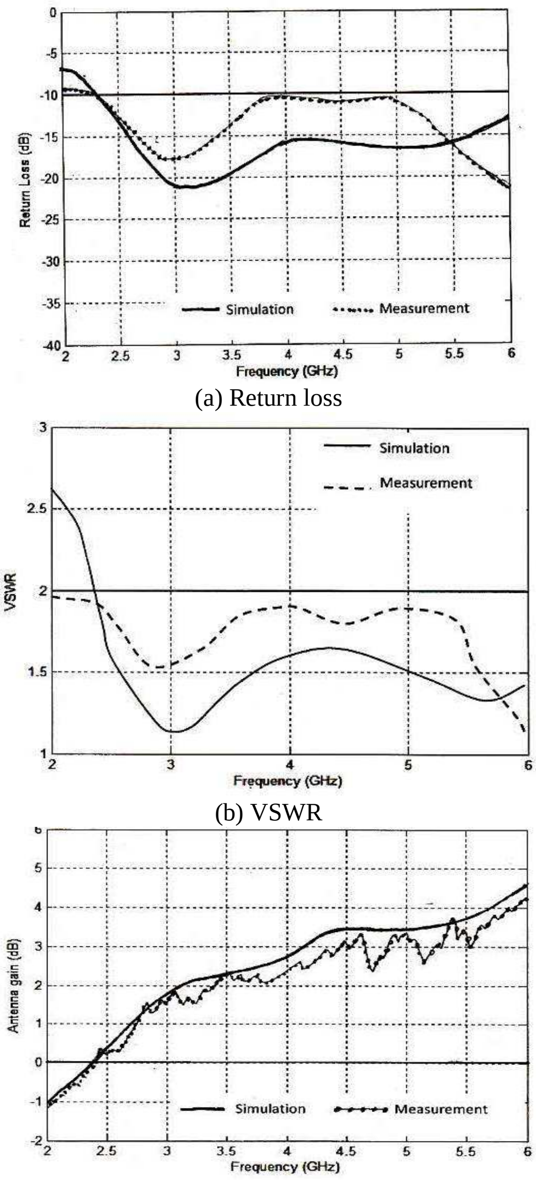


(a) (b)

Figure 5 Antenna fabrication: (a) Wideband, (b) Multiband.

Figure 6 shows the measurement results to validate the simulation of VSWR, return loss, antenna gain, and antenna radiation pattern for wideband type of antenna.

We can see for the wideband antenna shown in Figure 6 that for VSWR = 2 (equal to return loss of - 9.54 dB), wideband characteristic can be achieved for frequency that is higher than 2.3 GHz up to 6 GHz from simulation and can go beyond 6 GHz from measurement results. At frequency between 2.5 to 3.5 GHz VSWR/return loss is better because our basic design is based on that resonant frequency [5]. Comparison of VSWR/return loss between simulation and measurement results exhibit a variation, but both results comply to the required specification. The gain measurement and simulation shows that the gain obtained from measurement shows only slightly lower than that obtained from simulation, which increases almost linearly with frequency from 0 dB at lower band of 2.3 GHz to approximately 4.5 dB at 6 GHz frequency band.

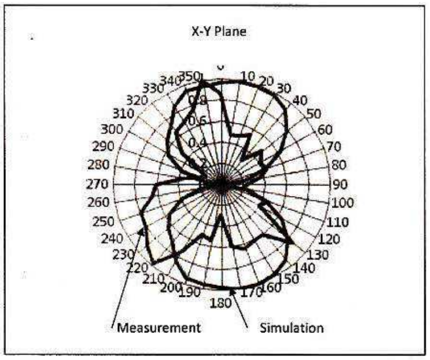


(a) Return loss

(b) VSWR

(c) Gain

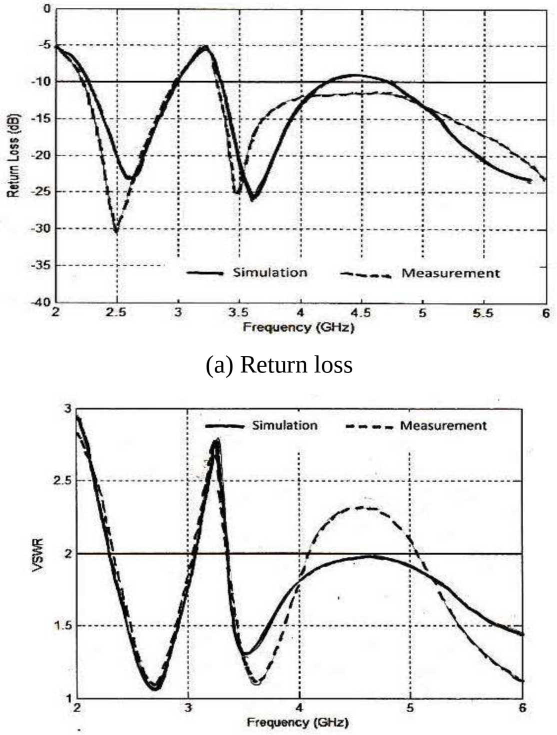
Figure 6 Wideband antenna measurement results.



(d) Radiation pattern

Figure 6 ***Continued.*** Wideband antenna measurement results

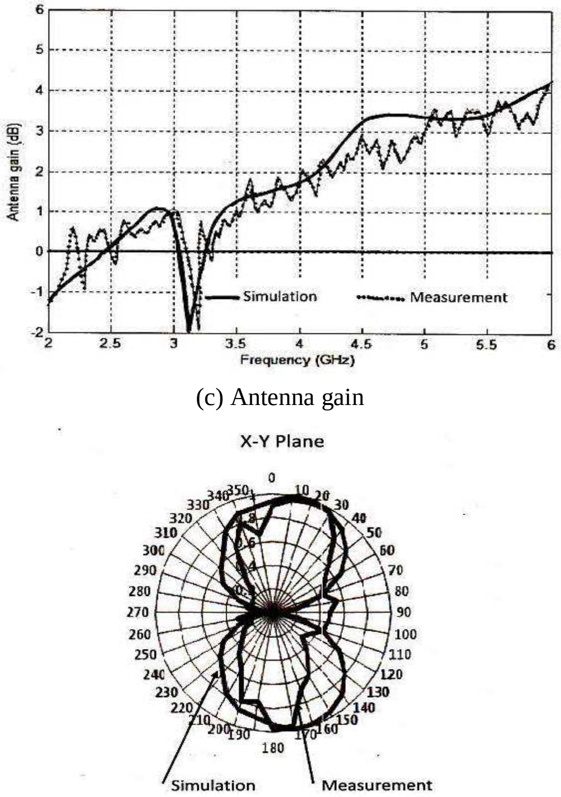
Figure 7 shows the measurement result to validate the simulation of VSWR, return loss, antenna gain, and antenna radiation pattern for multiband antenna.



(a) Return loss

(b) VSWR

Figure 7 Multiband antenna measurement result.



(c) Antenna gain

(d) Radiation pattern

Figure 7 ***Continued.*** Multiband antenna measurement result.

Measurement results for multiband antenna shown in Figure 7 exhibit close agreements with the charactersitics obtained from simulation. We can see from Figure 7 (c) that the gain measurement varies along the frequency band, but its average shows only slightly lower than that obtained from simulation, which increases almost linearly with frequency from 0 dB at lower band of 2.3 GHz to approximately 4.5 dB at 6 GHz frequency band. However, the multiband antenna gain shows a significant drop at frequency between 2.8-3.3 GHz, which corresponds to the stop band between low and midle frequency. Therefore, the antenna gain exhibits a similar behaviour for both wideband and multiband antenna, except at the stop band. We can also see from Figure 6 (d) and 7 (d) that antenna radiation pattern shows a bidirectional horizontal (x-y plane) pattern, and exhibits near omnidirectional vertical (y-z plane) pattern, which is not shown in the picture.

4 Discussion

From extensive simulations of wideband antenna design approach, we have seen the effect of circular truncation, as well as the effect of other dimension, i.e the gap between feed line and rectangular ground plane, the gap between ground plane and circular patch, and dimension of ground plane on the antenna bandwidth. The resonant (centre) frequency seems to be rather insensitive to any changes of circular truncation (Q). This can be explained that basic design of circular patch can be retained. Circular truncation (Q) can change the behavior of higher frequency above 3.5 GHz. The gap between circular cut and the edge of the antenna (H_1) is insensitive to the behavior of center frequency as well as to the antenna bandwidth characteristic of higher frequency band. However the gap between circular patch and the square ground plane (H_2) is sensitive to center frequency as well as to the behavior of higher frequency bandwidth. The gap between feed line and the square ground plane patch is most sensitive to center frequency and higher frequency characteristics.

We also have seen that multiband antenna design approach can be conducted by creating a pair of simetrically rectangular slot on the circular patch to produce undesired stop band along the frequency spectrum that has been obtained in the wideband design. Once the stop band is produced, the desired center frequency and bandwidth can then be obtained by retuning the gap between circular cut and ground plane, the gap between circular cut and the antenna edge, and the gap between the fed line and the ground plane. This can be explained that multiband antenna characteristics can be created by imposing radiaton perturbation to the patch in order to produce stop bands.

5 Conclusion and Further Study

Antenna prototypes that cover all Wi-Fi,WiMAX, and other upcoming modern wireless communication system wich occupy frequency bands of between 2.3 and 6 GHz or above have been designed and implemented on FR4 substrate successfully. The antenna design has been conducted using both wideband and multiband approaches. The wideband design approach has produced an atenna characteristic that covers the frequency of 2.3 – 6 GHz. Wideband characteristic is produced by modifying a basic circular patch with square ground plane by cutting the patch and by adjusting the square ground plane to widen the bandwidth at the higher frequency band.

The multiband design approach has produced a triple band characteristics with the low band of 2.3-2.8 GHz to accomodate for WiFi/WiMax and new generation of cellular technology, the midle band of 3.3-3.7 GHz for WiMax and extended C band applications, and above 4.6 GHz to provide for various upcoming modern wireless applications. The wideband antena gain shows increasing gain with the frequency from 0 dB at 2.3 GHz to around 4.5 dB at 6 GHz; while the multiband antenna gain shows a very similar result with that of the wideband, except for the stop band at 2.8-3.3 GHz which has a significant gain drop, corresponding to the desired stop frequency band. Antenna radiation pattern is bidirectional at x-y plane and nearly omnidirectional at y-z plane, and shows a similar patern for both wideband and multiband design approaches.

Antenna measurement results show good agrement with that obtained from simulation. Slight deviation was found in the measurement of the return loss for wideband antenna,but still meet the specification requirements. The multiband antenna requires more complex design compared to that of the wideband type. However, multiband antennas have an advantage over the wideband antenna in terms of prefiltering capability, in that any undesired frequency spectrum coming from other system or spurious emssion can be rejected at the receiving antenna front end.

In order to obtain more number of frequency bands in the multiband antenna design, our next study would focus on a more challenging design that can produce more thatn triple bands antenna, so that the antenna can provide prefiltering process for more technologies and applications with different frequency spectrum allocation.

Acknowledgements

This research is supported by the Directorate of Higher Education, Republic of Indonesia under the scheme of Decentralization Research Grant 2013, and by the Institute of Technology Bandung under the shceme of Research and Innovation Grant for Research Division. The authors thank for their financial support.

References

[1] Liu, D. & Gaucher, B., *A Branched Inverted-F Antenna for Dual Band WLAN Applications*, in Proceedings of the IEEE Antennas and Propagation Society International Symposium, **3**, pp. 2623-2626, Monterey, CA, June 2004.

[2] Raj, R.K., Joseph, M., Aanandan, C.K., Vasudevan, K. & Mohanan, P., *A New Compact Microstrip-Fed Dual-Band Coplanar Antenna for WLAN Applications*, IEEE Trans. Antennas Propag., **54**(12), pp. 3755-3762, Dec. 2006.

[3] Wu, J.W., Hsiao, H.M., Lu, J.H., & Chang, S.H., *Dual Broadband Design of Rectangular Slot Antenna for 2.4 and 5 Ghz Wireless*, IEE Electron. Lett, **40**(23), Nov. 2004.

[4] Wong, K.-L., *Compact and Broadband Microstrip Antennas*, New York: John Wiley and Sons, 2002.

[5] Chen, W.-S. & Chang, Y.-C., *Novel Design of a Printed Monopole Antenna for WLAN/WiMAX Applications*, Microwave Journal, **50**(10), pp.62-70, 2007.

[6] Chen, W.-S. & Ku, K.-Y., *A Microstrip-fed Monopole Antenna for WLAN USB Applications*, Microwave Journal, **51**(3), p. 104, 2008.

[7] Chen, W.-S. & Kao, B.-H., *A Triple-band Polygonal Slot Antenna for WiMAX Applications*, Microwave Journal, **50**(1), pp. 134-138, 2007.

[8] Kumar, G. & Ray, K.P., *Broadband Microstrip Antenna*, Artech House Inc, ch. 1-2, 2003.

[9] Kurniawan, A. & Mukhlishin, S., *Wideband Antenna Design and Fabrication for Modern Wireless Communications Systems*, in Proceedings of International Conference on Electrical Engineering and Informatics, Kualalumpur, Malaysia, pp. 335-340, 24-25 June 2013.

[10] Simons, R.N., *Coplanar Waveguide Circuits, Components, and Systems*, John Wiley & Sons, Inc., New York, **1-5**, pp. 15-21, 2001.

[11] Wadell, B.C., *Transmission Line Design Handbook*, Artech House, Norwood, **1-5**, pp. 73-75, 1991.

[12] Yoon, J., Jeong, G.T., & Kwak, K.S., *Fabrication and Measurement of Triangular-Slot Antenna for Triple-Band (2.4/5.2/5.8 Ghz) Antenna with Rectangular Tuning Stub*, Microwave and Opt. Technol. Lett., **49**(8), Aug. 2007.

[13] Plessis, M.D. & Cloete, J., *Tuning Stubs for Microstrip Patch Antennas*, IEEE Antennas Propag. Mag, **36**, pp. 52-56, Dec.1994.

Text Normalization Method for Arabic Handwritten Script

Tarik Abu-Ain[1], Siti Norul Huda Sheikh Abdullah[1], Khairuddin Omar[1], Ashraf Abu-Ein[2], Bilal Bataineh[1] & Waleed Abu-Ain[1]

[1]Pattern Recognition Research Group, Center for Artificial Intelligence Technology,
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia,
43600, Bangi, Selangor, Malaysia.
[2]Computer Engineering Department, Al-Balqa' Applied University, Faculty of
Engineering Technology, 15008 Amman, 11134 Jordan.
Email: tabuain@siswa.ukm.edu.my

Abstract. Text normalization is an important technique in document image analysis and recognition. It consists of many preprocessing stages, which include slope correction, text padding, skew correction, and straight the writing line. In this side, text normalization has an important role in many procedures such as text segmentation, feature extraction and characters recognition. In the present article, a new method for text baseline detection, straightening, and slant correction for Arabic handwritten texts is proposed. The method comprises a set of sequential steps: first components segmentation is done followed by components text thinning; then, the direction features of the skeletons are extracted, and the candidate baseline regions are determined. After that, selection of the correct baseline region is done, and finally, the baselines of all components are aligned with the writing line. The experiments are conducted on IFN/ENIT benchmark Arabic dataset. The results show that the proposed method has a promising and encouraging performance.

Keywords: *Arabic handwriting script; baseline detection; handwritten text normalization; preprocessing; slant correction; sub-word extraction.*

1 Introduction

For languages written horizontally such as the Arabic language (see Table 1), the text line virtually splits into three regions: upper, middle and lower regions [1]-[2]. The upper region contains upper dots, upper diacritic, and ascenders. The middle or the baseline region is the main part of the characters, where the loops and connection points between characters lay. Finally, the lower region contains descenders, lower dots and lower diacritic [3]-[4].

In Arabic-text, writing baseline is defined as an unreal horizontal straight line, where all characters lay and join in a specific part of each character [5]. In Arabic printed texts, the baseline is detected ideally by using the horizontal projection histogram. This is done by finding the row that contains the highest

Received September 23rd, 2013, Revision October 9th, 2013, Accepted for publication November 28th, 2013.
ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2013.7.2.5

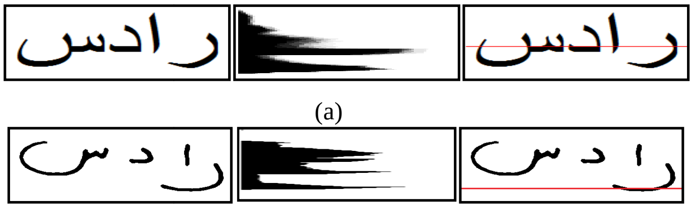
number of foreground pixels as shown in Figure 1(a). Nevertheless, there are problems in using this method on the handwritten text such as the existence of a wide variety of free writing styles and irregularity in sub-words alignment. This problem appears when one of these six letters “د،ذ،ر،ز،و،أ” is located in the beginning or middle of the word, leading to an inaccurate detection of a straight baseline for the text, as shown in Figure 1(b).This contradicts the definition of baseline as stated earlier in this paragraph. Slant problem occurs if the vertical primitives of the text are standing in slant form on the text baseline. The text slant existence affects directly in many processes in document image analysis especially in character segmentation process that use the vertical projection histogram.

Table 1 Arabic Letters and Shapes in the Position of a Text.

Beginning	Middle	End	Isolated
-	-	ـأ	أ
بـ	ـبـ	ـب	ب
تـ	ـتـ	ـت	ت
ثـ	ـثـ	ـث	ث
جـ	ـجـ	ـج	ج
حـ	ـحـ	ـح	ح
خـ	ـخـ	ـخ	خ
-	-	ـد	د
-	-	ـذ	ذ
-	-	ـر	ر
-	-	ـز	ز
سـ	ـسـ	ـس	س
شـ	ـشـ	ـش	ش
صـ	ـصـ	ـص	ص
ضـ	ـضـ	ـض	ض
طـ	ـطـ	ـط	ط
ظـ	ـظـ	ـظ	ظ
عـ	ـعـ	ـع	ع
غـ	ـغـ	ـغ	غ
فـ	ـفـ	ـف	ف
قـ	ـقـ	ـق	ق
كـ	ـكـ	ـك	ك
لـ	ـلـ	ـل	ل
مـ	ـمـ	ـم	م
نـ	ـنـ	ـن	ن
هـ	ـهـ	ـه	ه
-	-	ـو	و
يـ	ـيـ	ـي	ي

In this work, a text normalization method for baseline detection and slant correction in handwritten Arabic-text for character segmentation purposes is proposed. The method consists of several steps, which are: components labeling and segmentation, components text thinning, skeletons direction features extraction, candidate baselines regions determination, correct baseline region selection, components baselines alignment and slant correction. Set of experiments are conducted on a benchmark database for Arabic handwritten text called IFN/ENIT. The result is compared with a set of well-known methods such as Pechwitz [6], Farooq [7], Boukerma [8] and Horizontal Projection Histogram [9]-[11]. The results of the proposed method show a better performance than the previous methods. This paper consists of five sections:

introduction to the study, the relevant literature, the proposed framework, findings of the study, and finally the conclusion and recommendations for future studies.



(a)

(b)

Figure 1 The Result of Using the Horizontal Projection Histogram Method [9]-[11] in Baseline Detection for Arabic Text: (a) Success on a printed text, (b) Failure on a handwritten text.

2 Related Studies

Pechwitz proposed a study on baseline detection method based on polygonally approximated skeleton processing [6]. A study conducted by Farooq proposed another method that uses a two-step linear regression, which is applied on the local minimal points of word contour [7]. This latter method proved to be a better one compared to the previous study done by Pechwitz. However, the results of both of these methods contradict the baseline definition. The problem with these methods is that the linear regression algorithm does not work well on unaligned text.

In another study, Boukerma proposed an algorithm method that uses a piece-wise painting scheme to estimate the baseline by identifying a set of points used in the estimation process [8]. However, this method is defective in cases where large diacritics and small characters exist. On the other hand, Ziaratban *et. al*. proposed a baseline estimation algorithm using a template matching and a polynomial fitting algorithm [12]. However, this method is less effective in texts containing short words, dots and diacritics.

A study conducted by Boubaker proposed a baseline detection method for thinning text [13]. This is done by finding the relation between the text's point of alignment and their trajectory neighbor directions. However, the algorithm is not effective on short words that consist of only isolated characters. Nagabhushan, *et al*. proposed an algorithm method that uses a piece-wise painting scheme to identify points that are used to estimate the baseline [14]. However, this algorithm method is less effective on large diacritics and small characters.

From the above studies, it is obvious that most of these methods are influenced by many factors such as diacritics, isolated characters, long words as well as binding points between the characters.

3 The Proposed Method

The detection process of baseline location is very useful in extracting accurate information such as writing directions, ascenders, descenders, dots and diacritics. Irregularities in Arabic script handwriting style lead to irregularity in the writing of straight text components. A straight baseline detection and text slant correction are crucial steps in the pre-processing stage as a text normalization process.

As discussed above, most methods did not detect the correct baseline in texts consisting of short characters as well as of large diacritics. To overcome this problem, a framework is proposed here to estimate a baseline for each sub-word separately. Then, the estimated baselines are used to estimate a straight writing line for the whole text. Figure 2 depicts the proposed framework.

The process steps of the proposed text normalization are as follows:

Step 1: *Binary the text image.* This is done using Eq. (1) from the method proposed in [15]-[16] (Figure 3(a)) as follows:

$$T_w = m_w - \frac{{m_w}^2 * \sigma_w}{(m_g + \sigma_w)(\sigma_{Adaptive} + \sigma_w)} \quad (1)$$

where T_W is the thresholding value of the binarization window, m_w is the mean value of the pixels in the window; m_g is the mean value of the global image pixels; $\sigma_{Adaptive}$ is the adaptive standard deviation for the window; σ_w is the standard deviation of the window.

Step 2: *Extract the connected components (sub-words) from the text image.* This is carried out using the method proposed in [17] (Figure 3(b)).

Step 3: *Keep the main sub-words body only.* This is important since the presence of noise and dots affect the process of the baseline detection. This step works as the following; for components with the size of less than a threshold value (T_1) when calculated by the Eq. (2), they will be removed.

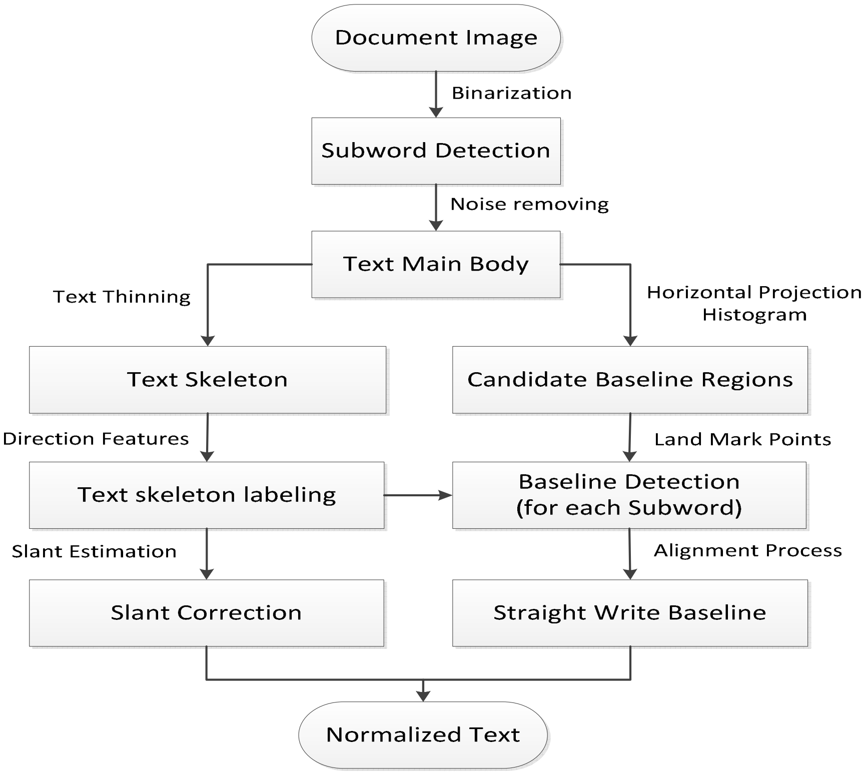


Figure 2 The Proposed Framework for the Normalization of Handwritten Arabic Text.

$$T_1 = \left(\frac{\sum black\ pixels}{number\ of\ connected\ components}\right) / \ V_e\ . \qquad (2)$$

where 3<V_e<4. V_e is a non-constant value that depends on the size of noise, dots and diacritics.

Step 4: *Text skeleton:* A robust text-thinning algorithm [18] is applied to ensure that the size of the width of the text skeleton is exactly one-pixel, as shown in Figure 3(c).

Step 5: *Text skeleton pixels analysis:* Perform a set of direction features [19] on the text skeleton to detect the horizontal and vertical adjacent pixels to draw a closed shape, where each one of them takes a unique label L_H, L_V and $L_{C,}$ respectively as shown in Figure 3(d) as follows:

- Closed shapes are detected using connected component technique [17] and are labelled by L_C.
- The rest of pixels are labelled based on their relationship with the 8-niebour adjacent pixels.
  - All pixels that are adjacent horizontally are labelled with L_H.
  - All pixels that are adjacent vertically are labelled with L_V.
  - Remaining pixels are labelled with L_H or L_V that are labelled based on their neighbour labelled pixels.

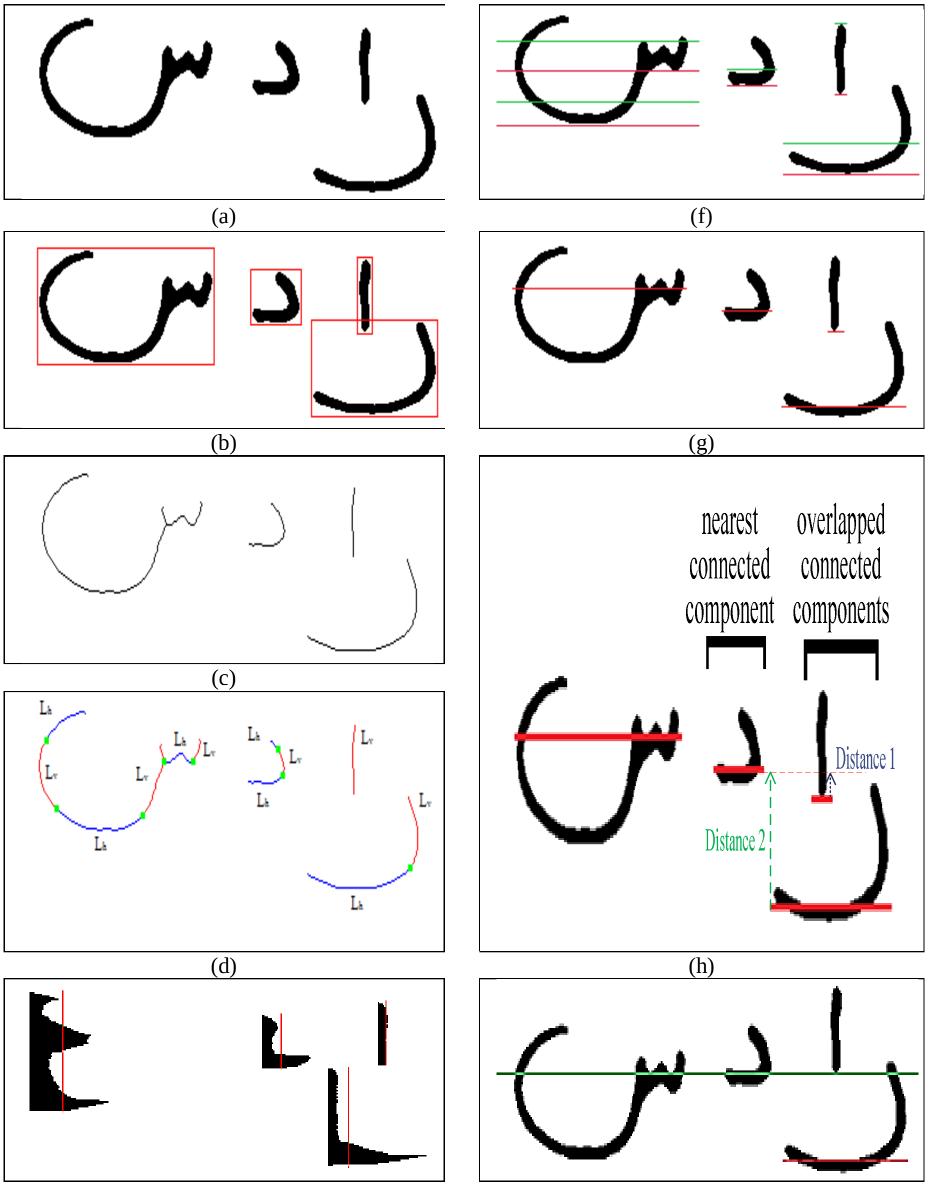
Step 6: *Baseline detection and straightening*

- The steps of baseline detection and straightening is as given below: for each sub-word,
- The horizontal projection histogram [9]-[11] is applied on each sub-word (see Figure 3(e)).
- Calculate the threshold value "T3", which is equal to the mean value of all the black pixels (see the vertical line in Figure 3(e)).
- Detect the candidates' baseline regions where every region is a set of continuous rows of foreground pixels that exceeds the threshold value T3 (see Figure 3(f)).
- Assign the landmark points to the pixels that gather between two different labels (see Figure 3(d)).
- The region that has the most number of landmarks will contain the baseline, which is the row that has the highest number of foreground pixels in the original text image (see Figure 3(g)).
- Because of the existence of many sub-words in the same text line, there will be a baseline for each of them, which are called local baselines. All these local baselines are aligned into one straight line called a global baseline. The exception will occur in cases of overlapping sub-words which are treated as a special case in the following process; measure the vertical distance between each of the overlapped sub-words' baseline and the nearest un-overlapped sub-words' baseline as shown in Figure 3 (h); the sub-word that has the shorter distance is aligned onto the global baseline (see Figure 3(i)).

Step 7: *Slant correction*

Here, we designate how the text skeleton-based representation of an Arabic text provides a considerable role in slant correction proposed algorithm. Figure 4 illuminates the proposed method for slant correction of an Arabic-text.

For each sub-word (see Figure 4(a)), do the following:



(a) (f)

(b) (g)

(c)

(d) (h)

(e) (i)

Figure 3 An Example of a Baseline Detection and Straightening Results of a Handwritten Text Image (رادس) After Completing the First Part of the Proposed Method.

- Look for the pixels labelled as “L_V” that have an only one 8-neighbour foreground pixel.
- For each pixel from the previous step:
  - Trace its 8-neighbour’s pixels until it reaches to none “L_V” labelled pixel or it reaches to the end, re-mark as base-pixel (see Figure 4(b)).
  - Shift all traced pixels labelled as “L_V” straight over the base-pixel (see Figure 4(c)).

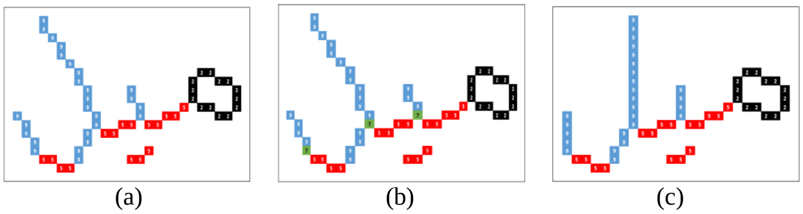


Figure 4 An Example of a Slant Correction Result of a Handwritten Arabic Text Image (ميل) After Completing the Second Part of the Proposed Method.

4 Experiments and Results

The initial results are very promising in solving many important problems that arise in baseline detection for both handwritten and machine printed cursive scripts such as in the case of diacritics, isolated characters and short words as well as the binding points between characters that lay on the baseline.

Due to the unconstrained nature of Arabic handwriting style from person to person, there is no ideal position of the handwritten text baseline (see Figure 5 (f-j)). In contrast with machine printed texts, the baseline is detected ideally using the horizontal projection profile method (see Figure 5 (a-e)).

Many experiments are conducted on the IFN/ENIT dataset [20] to validate the abilities of the proposed framework. To evaluate the proposed method performance, the results of the proposed method are compared visually with the results of Pechwitz [6], Farooq [7], Boukerma [8] (see Figure 6). Based on Figure 6, the proposed algorithm method shows better results. The proposed method is also capable to process machine printed texts as well as handwritten texts without any modification.

This study hints about various crucial problems that arise in baseline detection in handwritten Arabic text and their solutions. The problems of baseline detection for handwritten text have been largely solved in the proposed method such as in the case of diacritics, isolated characters and short words in binding the points between the characters that lay over the baseline accurately.

Moreover, all the sub-words are also intersecting on the straight baseline on the right points.

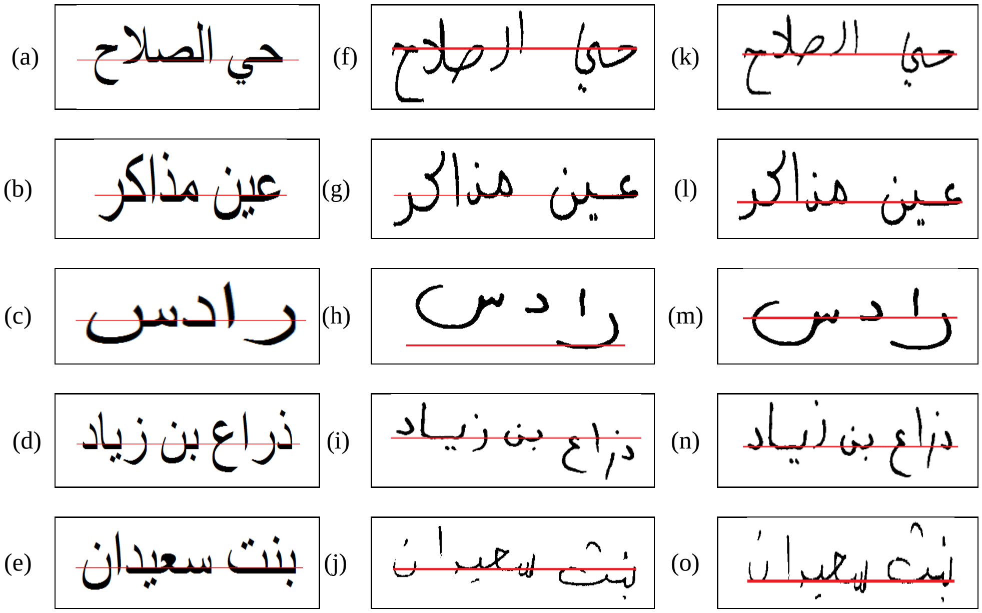
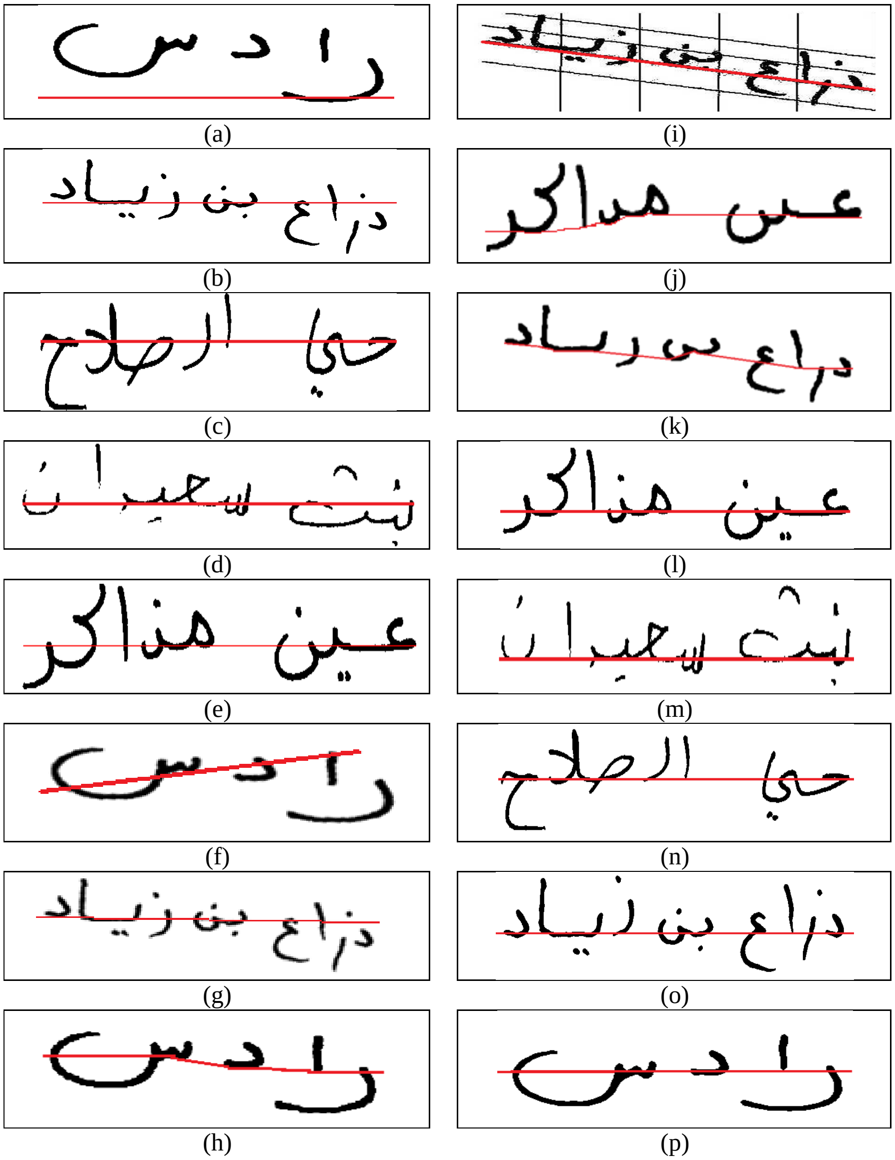


Figure 5 Results of Baseline Detection Using: (a-e) the Ideal Baseline Obtained from the Machine Printed Text, (f-j) the Horizontal Projection Histogram Method [9]-[11] and (k-o) the Proposed Method on Handwritten Text.

5 Conclusion

The objective of this paper is a proposed text normalization method for handwritten Arabic script. The paper discusses seven main stages involved in this method; connected component allocation, candidate baseline regions detection, text skeleton analysis, baseline allocating for each word/sub-word, baseline straightening procedure and lastly text slant correction for segmentation purposes. The visual results prove the remarkable performance of the proposed method on the textual binary images. The results on the IFN/ENIT dataset used in the experiments with the proposed method show a significantly better performance compared to the Pechwitz [6], Farooq [7], Boukerma [8] methods. The proposed method will enhance the performances of several techniques such as, text segmentation, feature extraction and character recognition.



(a) (i) (b) (j) (c) (k) (d) (l) (e) (m) (f) (n) (g) (o) (h) (p)

Figure 6 Results of: (a-e) the Horizontal Projection Histogram Method [16], (f-j) Pechwitz Method [6], (k-l) Farooq Method [7], (m-n) Boukerma Method [8] and (o-p) the Proposed Method.

Acknowledgements

The authors would like to thank the Faculty of Information Science and Technology and Center for Research and Instrumentation Management of the Universiti Kebangsaan Malaysia for providing facilities and financial support under Exploration Research Grant Scheme Project No. ERGS/1/2011/STG/ UKM/01/18 entitled "Calligraphy Recognition in Jawi Manuscripts using Palaeography Concepts Based on Perception Based Model" and Fundamental Research Grant Scheme No. FRGS/1/2012/SG05/UKM/02/8 entitled "Generic Object Localization Algorithm for Image Segmentation”.

References

[1] Grimes, B.F., *Ethnologue: Languages of the world*, fourteenth ed., SIL International, 2000.

[2] Lorigo, M. & Govindaraju, V., *Offline Arabic Handwriting Recognition: A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **28**(5), pp. 712-724, 2006.

[3] Abu-Ain, T.A.H., Abu-Ain, W.A.H., Abdullah, S.N.H.S. & Omar, K., *Off-line Arabic Character-Based Writer Identification – a Survey*, in Proceeding of the International Conference on Advanced Science, Engineering and Information Technology (ICASEIT 2011),Indonesian Students Association-Universiti Kebangsaan Malaysia, Bangi, Malaysia, pp. 161-166, 2011.

[4] Abu-Ain, T., Abdullah, S.N.H.S., Bataineh, B., Abu-Ain, W. & Omar, K., *Text Normalization Framework for Handwritten Cursive Languages by Detection and Straightness the Writing Baseline*, in International Conference on Electrical Engineering and Informatics (ICEEI 2013), UKM, Bangi, Selangor,Malaysia. pp. 654-658, 2013.

[5] Gacek, A., *ArabicManuscripts: A Vademecum for Readers*, BRILL 2009.

[6] Pechwitz, M. & Margner,V., *Baseline Estimation for Arabic Handwritten Words,* in Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR 2002), Niagara-on-the-Lake, Ontario, Canada, pp. 479-484, 2002.

[7] Farooq, F., Govindaraju, V. & Perrone, M., *Pre-Processing Methods for Handwritten Arabic Documents,* in Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR 2005), Seoul, Korea, pp. 267-271, 2005.

[8] Boukerma, H. & Farah, N., *A Novel Arabic Baseline Estimation Algorithm Based on Sub-Words Treatment*, in Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition (ICFHR 2010), Kolkata, India, pp. 335-338, 2010.

[9] Parhami, B. & Taraghi, M., *Automatic Recognition of Printed Farsi Texts*, Pattern Recognition, **14(**1-6), pp. 395-403, 1981.

[10] Saady, Y.E., Rachidi, A., El Yassa, M. & Driss, M., *Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character*, International Journal of Advanced Science and Technology, **33**(17), pp. 33-50, 2011.

[11] Touj, S., Amara, N.B. & Amiri, H., *Arabic Handwritten Words Recognition Based on a Plannar Hidden Markov Model*, The International Arab Journal of Information Technology, **2**(4), pp. 318-325, 2005.

[12] Ziaratban, M. & Faez, K., *A Novel Two-Stage Algorithm for Baseline Estimation and Correction in Farsi and Arabic Handwritten Text Line*, in 19th International Conference on Pattern Recognition (ICPR 2008), Tampa, Florida, USA, pp. 1-5, 2008.

[13] Boubaker, H., Kherallah, M. & Alimi, A.M., *New Algorithm of Straight or Curved Baseline Detection for Short Arabic Handwritten Writing,* in Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR 2009), Barcelona, Spain, pp. 778-782, 2009.

[14] Nagabhushan, P. & Alaei, A., *Tracing and Straightening the Baseline in Handwritten Persian/Arabic Text-line: A New Approach Based on Painting-technique,* International Journal on Computer Science and Engineering, **2**(4), pp. 907-916, July 2010.

[15] Bataineh, B., Abdullah, S.N.H.S. & Omar, K., *An Adaptive Local Binarization Method for Document Images Based on A Novel Thresholding Method and Dynamic Windows*, Pattern Recognition Letters, **32**(14), pp. 1805-1813, 2011.

[16] Bataineh, B., Abdullah, S.N.H.S., Omar, K. & Faidzul, M., *Adaptive Thresholding Methods for Documents Image Binarization*, in Pattern Recognition, ed: Springer Berlin Heidelberg, pp. 230-239, 2011.

[17] Linda, G. & Shapiro, G.C.S., *Computer Vision*: Prentice Hall, p. 608, 2002.

[18] Abu-Ain, W., Abdullah, S.N.H.S., Bataineh, B., Abu-Ain, T. & Omar, K., *Skeletonization Algorithm for Binary Images*, in International Conference on Electrical Engineering and Informatics (ICEEI 2013), UKM, Bangi, Selangor, Malaysia. pp. 690-694, 2013.

[19] Al Hamad, H. & Abu Zitar, R., *Development of an Efficient Neural-Based Segmentation Technique for Arabic Handwriting Recognition*. Pattern Recognition, **43**(8), pp. 2773–2798, 2010.

[20] Pechwitz, M., Maddouri, S.S., Märgner, V., Ellouze, N. & Amiri, H., IFN/ENIT - Database of Arabic Handwritten words, in Proceedings of Colloque international francophone surl'écrit et le document (CIFED 2002), Hammamet, Tunisie, pp. 129-136, 2002.

Journal of ICT Research and Applications
Guidelines for Author

1. Standard of reporting

Authors should present an accurate account of the work performed as well as an objective discussion of its significance. Underlying data should be represented accurately in the paper. A paper should contain sufficient detail and references to permit others to replicate the work. Fraudulent or knowingly inaccurate statements constitute unethical behavior are unacceptable. Professional publication articles should also be accurate and objective, and editorial `opinion' works should be clearly identified.

2. Exclusivity of work

The authors should ensure that they have written entirely original works, and if the authors have used the work and/or words of others this should be appropriately cited or quoted. Plagiarism takes many forms, from `passing off' another's paper as the author's own paper to copying or paraphrasing substantial parts of another's paper (without attribution), to claiming results from research conducted by others. Plagiarism in all its forms constitutes unethical publishing behavior and is unacceptable. An author should not in general publish manuscripts describing essentially the same research in more than one journal or primary publication. Submitting the same manuscript to more than one journal concurrently constitutes unethical publishing behaviour and is unacceptable. In general, an author should not submit for consideration in another journal a previously published paper. We consider for publication from conference paper if it is only an extended version of conference paper with at least 30% of new material.

3. Hazards and Human or Animal Subjects

If the work involves chemicals, procedures or equipment that have any unusual hazards inherent in their use, the author must clearly identify these in the manuscript. If the work involves the use of animal or human subjects, the author should ensure that the manuscript contains a statement that all procedures were performed in compliance with relevant laws and institutional guidelines and that the appropriate institutional committee(s) has approved them. Authors should include a statement in the manuscript that the informed consent was obtained for experimentation with human subjects. The privacy rights of human subjects must always be observed.

4. Authorship of the Paper and Copyright

Authorship should be limited to those who have made a significant contribution to the conception, design, execution, or interpretation of the reported work. All those who have made significant contributions should be listed as co-authors. Whilst those who have participated in certain substantive aspects of the research project, they should be acknowledged or listed as contributors. The corresponding author should ensure that all appropriate and inappropriate co-authors are included on the paper, and that all co-authors have seen and approved the final version of the paper and have agreed to its submission for publication. No manuscript can be published unless accompanied by a signed publication agreement, which serves as a transfer of copyright from author to publisher. A copy of that agreement is required after the paper is accepted.

5. Acknowledgement

Proper acknowledgment of the work of others must always be given. Authors should cite publications that have been influential in determining the nature of the reported work. Information obtained privately, as in conversation, correspondence or discussion with third parties, must not be used or reported without explicit, written permission from the source. Information obtained in the course of confidential services, such as refereeing manuscripts or grant applications, must not be used without the explicit written permission of the author of the work involved in these services.

6. Disclosure Requirements

Author when submitting a manuscript, must disclose any meaningful affiliation or involvement, either direct or indirect, with any organization or entity with a direct financial interest in the subject matter or materials discussed (for example, employment, consultancies, stock ownership, grants, patents received or pending, royalties, honoraria, expert testimony). These kinds of financial involvement are fairly common, unavoidable, and generally do not constitute a basis for rejecting a manuscript. Specifics of the disclosure will remain confidential. If deemed appropriate by the Scientific Editor, a general statement regarding disclosure will be included in the Acknowledgment section of the manuscript.

7. Errors in Published Works

When an author discovers a significant error or inaccuracy in his/her own published work, it is the author's obligation to promptly notify the journal editor or publisher and cooperate with the editor to retract or correct the paper. If the editor or the publisher learns from a third party that a published work contains a significant error, it is the obligation of the author to promptly retract or correct the paper or provide evidence to the editor of the correctness of the original paper.

8. Disclaimer

Opinions expressed in articles published in the *Journal of ICT Research and Applications* are those of the author(s) and do not necessarily represent opinions of the Bandung Institute of Technology (ITB). The *Journal of ICT Research and Applications* does not guarantee the appropriateness for any purpose of any method, product, process, or device described or identified in an article. Trade names, when used, are only for identification and do not constitute endorsement by *Journal of ICT Research and Applications*.