Assessing blood-pressure measurement in tablet-based mHealth apps

Rima Murthy
Department of Computer Science
Dartmouth College

David Kotz
Department of Computer Science
Dartmouth College

***Abstract*—We propose a new method to record contextual information associated with a blood-pressure reading using a tablet's touchscreen and accelerometer. This contextual information can be used to verify that a patient's lower arm remained well-supported and stationary during her blood-pressure measurement. We found that a binary support vector machine classifier could be used to distinguish different types of lower-arm movements from stationary arms with 90% accuracy overall. Predetermined thresholds for the accelerometer readings suffice to determine whether the tablet, and therefore the arm that rested on it, remained supported. Together, these two methods can allow mHealth applications to guide untrained patients (or health workers) in measuring blood pressure correctly.**

I. Introduction

As low-cost smartphones and tablets become commonplace in the developing world, mobile data collection apps are increasingly used for health services in the field and for public health monitoring [1], [2]. Mobile apps that interface with personal medical sensors have proven particularly effective for community health-data collection [3]. Any mHealth data collection system should include contextual information, wherever pertinent, along with the medical readings to enhance fidelity of the health data, as proposed by Prasad et al. in their provenance framework for mHealth [4].

In this paper, we focus on including contextual information associated with a blood-pressure measurement. Research has indicated that health workers are more prone to error in blood-pressure measurement than certified nurses who are themselves more prone to error than physicians [5], [6]. Therefore, it is useful to integrate a correcting component with mHealth data-collection systems to guide non-clinical workers in the field while they measure a patient's blood pressure and to record contextual data about the blood-pressure reading. We present a method to monitor lower-arm support and movement during blood-pressure measurement using an Android tablet's touchscreen interface and built-in accelerometer and without the use of dedicated sensors for monitoring blood-pressure (BP) posture. We chose to work with the Android-based tablets because several mHealth data collection solutions have already been built on this open-source framework [1], [7], [8]. Furthermore, we expect that Android tablets will be deployed widely for health data collection since the form factor of tablets are more conducive to data collection and tablet sales have already surpassed personal computer sales in some developing countries [9], which could further enable large-scale, cost-effective mHealth deployments.

Our method works as follows:

1) A health worker attaches an electronic BP cuff to a patient's arm.
2) The patient's lower arm is lowered onto the tablet touchscreen, with the palm facing upward.
3) Our app initiates the BP measurement using wireless communication to the cuff.
4) Our app monitors the tablet's acceleration along x, y and z axes to determine whether the arm remained supported for the duration of blood-pressure measurement.
5) Our app monitors touch events on the tablet's touchscreen to monitor the position and movement of the lower arm.
6) Our app uses a pre-trained binary support vector machine (SVM) classifier to analyze touchscreen event data to determine whether the lower arm remained stationary during the BP measurement.

We make two contributions in this paper:

- We describe a method for measuring lower-arm stillness and support during a blood-pressure measurement, using an off-the-shelf Android tablet.
- We evaluate our method for measuring lower-arm stillness with a preliminary user study of 12 subjects and found that our method can distinguish stationary arms from different types of lower-arm movement with 90% accuracy.

The remainder of our paper describes our method and our evaluation via the user study.

II. Motivation

Blood pressure is difficult to measure accurately because it can be influenced by behavioral factors such as stress, smoking, caffeine intake, exercise or natural factors like circadian variation. To compound these factors, incorrect techniques of measuring blood pressure can introduce additional errors. Our discussions with doctors in India and our understanding of the American Heart Association guidelines yield the following clinical recommendation for blood-pressure measurement posture [5]:

1) The patient should be seated comfortably with the back supported and the upper arm bared without constrictive clothing.

2) The arm should be supported at heart level, and the bladder of the cuff should encircle at least 80% of the arm circumference.
3) Neither the patient nor the observer should talk during the measurement and the patient must not move her arm while the blood pressure is being taken.

Blood-pressure cuffs available in the market are already equipped with sensors to detect movement. These sensors are also programmed to repeat a reading when movement is detected because arm movements could perturb blood in the veins and thus result in an erroneous reading. However, since the typical cuff and its embedded sensors attach to the upper arm, our tests show that lower-arm movement may go undetected.

III. Method

Our method can be used to verify two lower-arm related posture requirements for blood-pressure measurements: first, it detects a subject's lower-arm movement, and second, it verifies whether the arm is well-supported on a flat surface for the duration of blood-pressure measurement.This method could enable non-clinical workers to measure blood pressure correctly without the use of dedicated sensors, unlike a previous pressure sensor-based approach presented by Smithayer [10].

Regarding the latter challenge, we found that it was straightforward to use the tablet's accelerometer to determine whether the tablet remained well-supported throughout the BP measurement, by ensuring that the x and y (horizontal) accelerations each remained less than $1g$ and the z (vertical) acceleration remained in the range $7.1 - 8.5g$, where g is the Earth's acceleration due to gravity. Although we did not evaluate these thresholds in a field study, they appear to work well from initial experiments that we conducted on different surfaces and surface inclinations.

Thus, in this paper we focus on the challenge of determining whether the subject's lower arm remains still during a blood-pressure measurement, while it rests on the tablet screen. To begin, we describe how we collected a labeled data set for training a machine-learning model.

A. Data collection

We built a labeled data set for stationary and moving arms using an Android tablet's multi-touch touchscreen. We use the term *session* to refer to each 30-second subject session in which a set of touchscreen data is collected. A *moving session* is a session in which the subject was asked to move his/her lower arm, while it rested on the tablet, for 5-10 seconds. A *stationary session* is a session in which the lower arm remained motionless for the 30-second data collection interval. Thus, we create a labeled dataset of moving and stationary sessions.

For this experiment, we collected data from 12 subjects with each subject contributing five stationary and five moving sessions, for a total of 120 sessions in the dataset. The study procedures involving human subjects were approved by the institutional review board (IRB) at Dartmouth College. The data collection exercise was set up as follows:

1) The subject was instructed to rest her elbow on the tablet screen and subsequently lower her arm such that her lower arm was in contact with the touchscreen for 30 seconds, as shown in Figure 1.
2) For 'moving' sessions, the subject was asked to move her arm slightly for approximately 5-10 seconds during the 30-second data collection interval. [1] To capture realistic movements that patients are likely to make per our observations, the primary researcher demonstrated five examples of movement to the subject; the subject was instructed to incorporate these movements during data collection:
    a) the lower arm slides along the x-axis,
    b) the arm slides along the y-axis,
    c) the arm slides along both x and y axes in either up or down direction,
    d) along any of the axes and between any two points on the touch screen, the arm slides in a sudden movement, and finally,

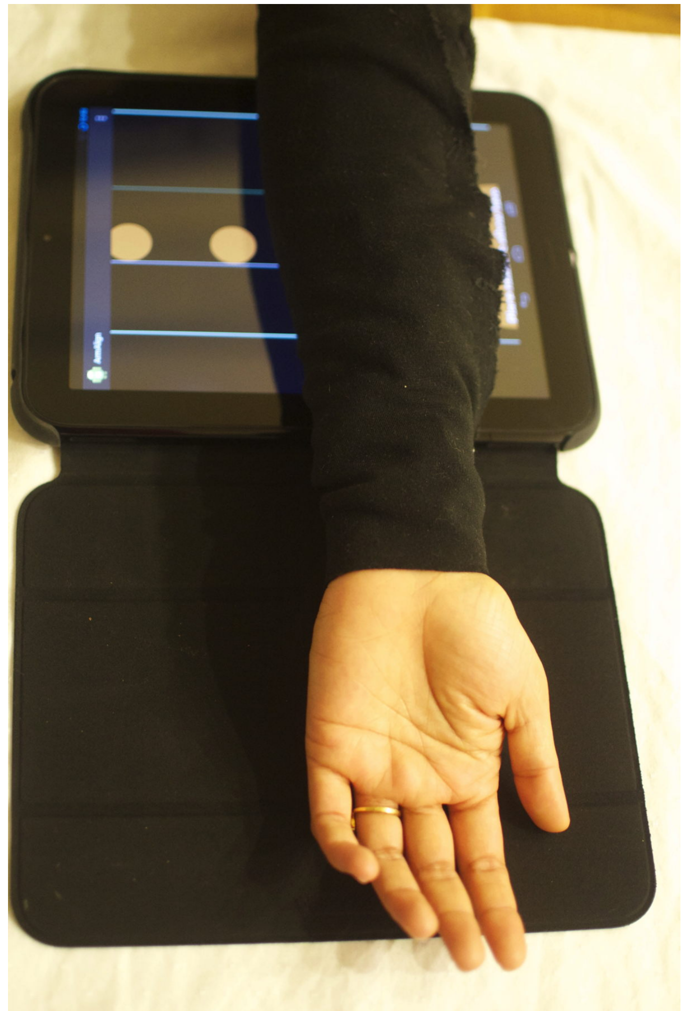

Fig. 1. Using the tablet touchscreen to monitor lower-arm posture

[1]The classification would be trivial if a subject exaggerated her arm movements throughout the entire data collection duration. Indeed, initial experiments suggested that our classifier would detect small datasets limited to such exaggerated movements with 100% accuracy. This increased our confidence that linear classification was sufficient to classify moving arms from stationary ones.

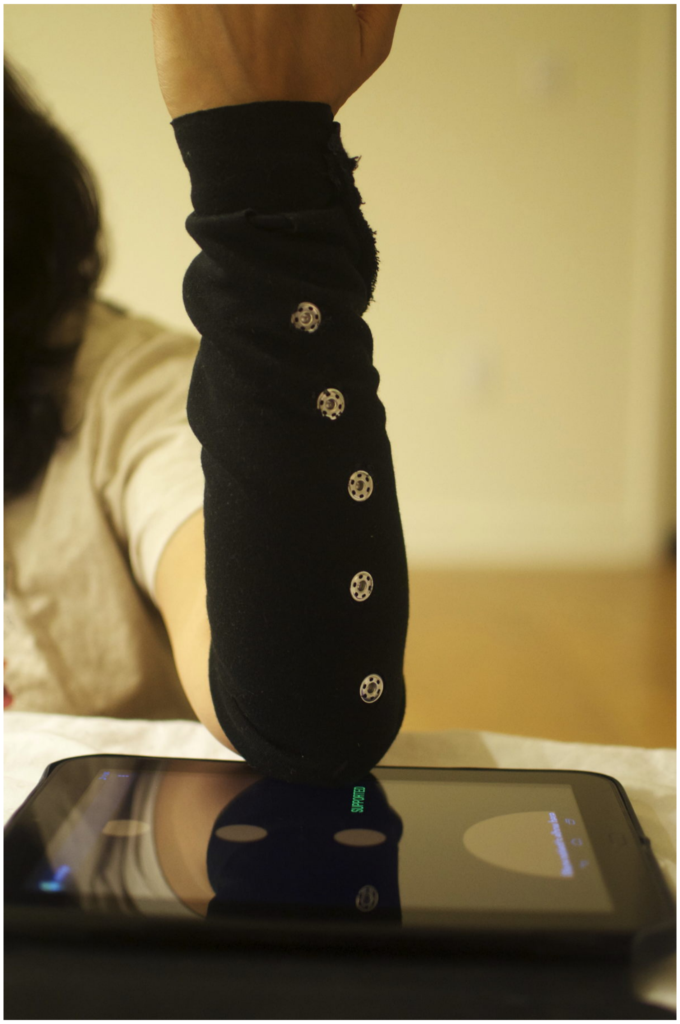

Fig. 2. Protective cover to minimize skin contact with the touchscreen

e) along any of the axes and between any two points on the touch screen, the arm slides gradually from one point on the screen to another.

3) For 'stationary' sessions, the subject was instructed to keep her arm still after the arm was lowered onto the tablet.
4) While the subject rested her arm on the tablet touchscreen, our data-gathering app recorded the touchscreen activity throughout the session. It used standard Android APIs to record the set of *motion events* occurring during the measurement period.

During our preliminary experiments, when the entire skin surface of the lower arm came in direct contact with the tablet's capacitive touchscreen, we found that the touchscreen malfunctioned as a result of being overwhelmed by the sudden and high volume of static discharge. To address this problem, we designed a protective sleeve (with five conductive steel rings embedded) to localize skin contact with the touchscreen around the rings. The protective sleeve, shown in Figure 2, restricts direct skin contact with the touchscreen while allowing us to detect and track touch events that occur around the rings.

The Android framework defines a *motion event* to include the temporal history of one or more *touch events*. For example, a three-fingered swipe is a single motion event that describes three touch events, each describing the path and pressure of one finger as it swipes across the touchscreen.

To detect and track movements on the touchscreen, we describe each *touch event* as a series of four-tuples, each tuple including the timestamp t, the coordinates of the screen position (x, y), and the pressure of the touch p. The change in attributes (x, y) could capture movement along x and y axes and the p attribute could be used to capture movement along z axis. These attributes are commonly used in sample applications provided by the Android framework for gesture detection. In short, then, our approach is to analyze the changes to these four touch-event attributes, across all motion events within a session, to detect lower-arm movement in that session.

B. Algorithm

Given a set of labeled session data, as described above, we train a machine-learning algorithm to distinguish stationary sessions from moving sessions. First, we need an algorithm to generate a representative feature vector for each BP-measurement session using a subset of touch-event attributes.

Consider a session s. Let M be the set of motion events $m_1, \dots m_i, \dots m_n$ in this session.

For each m_i, there is a set of touch events m_{ij}.

For each m_{ij}, the app has recorded a sequence of attributes t, x, y, p; considered separately, these sequences are denoted t_{ij}, x_{ij}, y_{ij}, p_{ij}. More precisely, x_{ij} is a sequence $x_{ij1}, \dots x_{ijk}, \dots$, for some duration.

For each session s we build a matrix C where each row C_{ij} represents one event m_{ij}. Each such row is a 7-tuple:

$$C_{ij} = [i, j, \text{var}\{x_{ij}\}, \text{var}\{y_{ij}\}, \text{var}\{p_{ij}\}, X_{ij}, Y_{ij}]$$

where i, j serve to label the origin of the data in this row, where

$$\text{var}\{x_{ij}\} = \text{variance of } \forall_k\{x_{ijk}\}$$

and similarly for $\text{var}\{y_{ij}\}, \text{var}\{p_{ij}\}$, and where

$$X_{ij} = \text{range}\{x_{ij}\} = \text{max}_k\{x_{ijk}\} - \text{min}_k\{x_{ijk}\}$$

and similarly for Y_{ij}.

We summarize the C matrices by building the feature matrix F where each row represents one session s, as a 5-tuple that describes the maximum value along each dimension:

$$F_s = [\text{maxvar}\{x\}, \text{maxvar}\{y\}, \text{maxvar}\{p\}, \text{maxX}, \text{maxY}]$$

where maxvar is the maximum variance, that is,

$$\text{maxvar}\{x\} = \text{max}_{ij}\{\text{var}\{x_{ij}\}\}$$

and similarly for $\text{maxvar}\{y\}$, $\text{maxvar}\{p\}$; and where

$$\text{maxX} = \text{max}_{ij}\{X_{ij}\}$$

and similarly for maxY.

We then trained a binary SVM classifier using a subset of the features (columns) in F, and a subset of the labeled data collected earlier.

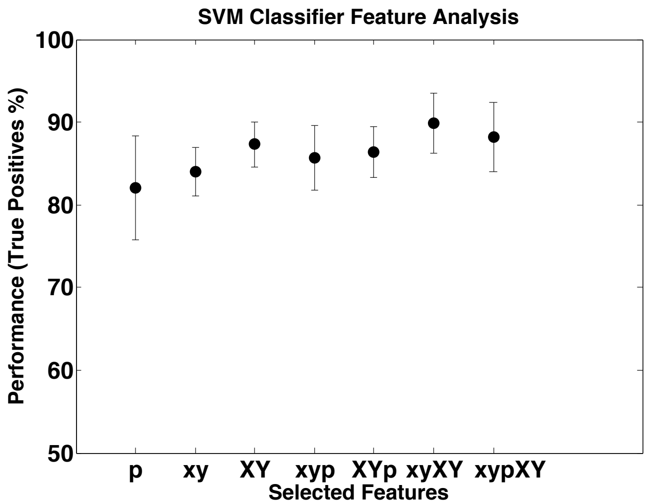

Fig. 3. Feature analysis: Comparing the classifier accuracy across different combinations of features.

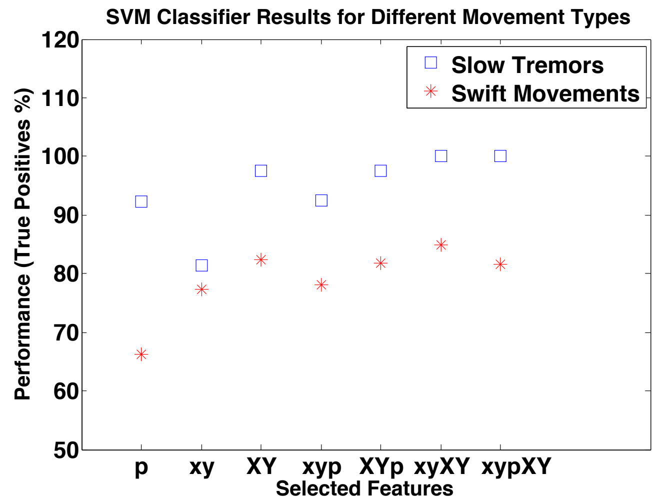

Fig. 4. Prediction accuracy for moving sessions where the movement is known to be either slow tremors or swift movements. From the plot, the method is able to detect slow tremor-like movement better than swift movements.

IV. Results

Given the limited number of moving and stationary sessions data, we conduct a 2-fold cross-validation. We ran ten independent trials; in each trial run,

1) we randomly select moving and stationary rows from the dataset F and split these into non-overlapping subsets of training and test data,
2) we train a model for a linear SVM classifier, using the training subset of F,
3) we apply the model to the test partition of F, noting the correctness of each classification result,
4) we compute the overall classification accuracy as

$$\frac{\text{Total number of correctly classified rows in test set}}{\text{Total number of rows in the test set}}$$

5) we compute classification accuracy for specific moving sessions in the test set that correspond to slow tremor-like movements and swift movements.

We compute the mean accuracy across ten trial runs to report the results presented here.

A. Feature Analysis

Before we review the results of our experiments with this classifier, we need to explain how we chose a suitable subset of features for use in our classifier. Given the feature matrix F, we sought to determine which combination of features (columns of F) would yield a classifier with the best performance. Figure 3 shows this *feature analysis*, comparing the accuracy of the SVM classifier with different feature sets. The features **x**, **y**, **p**, **X** and **Y** in Figure 3 correspond to the maxvar$\{x\}$, maxvar$\{y\}$, maxvar$\{p\}$, maxX and maxY columns of F respectively; hereon, we use this boldface notation to refer to these aggregate features. Simple thresholds for absolute differences in x and y were not enough to distinguish stationary and moving arms reliably; we needed a classifier that captures nuances of movement on the touchscreen.

Figure 3 plots the classifier's accuracy across ten trials; dots indicate the mean and bars indicate the standard deviation. Although the **p** feature alone yields an adequate global performance accuracy, observe that the best feature set excludes the **p** feature. [A more extensive user study is required to understand whether diverse weight of subjects' arms might be expressed by **p** and improve classification.] From these results, we chose the final feature set [**x**, **y**, **X**, **Y**], which resulted in approximately 90% mean accurate classification of stationary and moving sessions in the test data. Table I presents a confusion matrix for this feature set, indicating the classifier's accuracy in predicting stationary or moving sessions. Our classifier could detect stationary arms with 90% accuracy and moving arms with 89.5% accuracy; the false-negative rate for moving sessions was 10.5% and the false negative rate for stationary sessions was 9.6%.

TABLE I. Confusion matrix with feature vector xyXY

	Predicted	
Actual	Stationary	Moving
Stationary	0.904	0.096
Moving	0.105	0.895

B. Classifier performance across different movement types

Figure 4 shows the results for different movement types within the moving class across all feature sets. The accuracy of the classifier is expressed as the rate of true positives. The legends in Figure 4 indicate:

1) *Slow tremors:* the accuracy reported for the set of rows in the test dataset that correspond to slow tremor-like movements.
2) *Swift movements:* the accuracy reported for the set of rows in the test dataset that correspond to swift movements.

Upon examining the underlying data, we found that the range of variance for the values of the **x** and **y** features (that is, $\text{maxvar}\{x\} - \text{minvar}\{x\}$) across small, tremor-like movements was less than the corresponding range for the same features for swift movements. This difference could explain why the linear classifier performed better with slow movements than with swift movements across most feature sets. From Figure 4, we observe that the addition of more features improves classification for swift movements, but less so for slow movements. This difference could be explained by the fact that the dataset contains more examples and variations of swift movements than stationary ones. We can conclude that the classifier can robustly detect different types of movement although the performance appears to be better for slow movements.

V. Related work

Independent researchers and manufacturers of blood-pressure monitors alike have produced several patents that focus on the challenge of enforcing correct blood-pressure posture to increase reading accuracy. All the methods we encountered use additional hardware, such as inclination and pressure sensors [10]–[12]. In this paper, we envision applications wherein the health workers are carrying tablets, anyway, for collecting health data and presenting health information to patients. We present a novel method to enable non-clinical workers, who use Android tablets for mobile health data collection, to measure blood pressure correctly without the use of dedicated external sensors by leveraging the sensors embedded in most tablets. With increasing access to tablets, particularly in the developing world, our method would be easier to integrate with existing mHealth deployments and would be more portable in the field than sensor-based solutions.

VI. Conclusions and future work

Any Android mHealth app could use our method to detect whether an individual's arm is stationary and supported throughout a blood-pressure reading. Furthermore, the app could also provide feedback to alert the health worker if posture requirements were violated by a patient, by analyzing the movement data from the touchscreen in real time.

Indeed this method could be integrated easily into existing mHealth data collection apps built on the Android operating system because the classifier only requires aggregated attributes of the built-in events associated with touchscreen contact. However, since the classifier model is learned entirely from tablet touchscreen data, adjustments to these weights may be required across different Android hardware.

Although the classifier performance at 90% may be sufficient to distinguish most stationary and moving arms, we plan to work with doctors to understand specific types of lower-arm movement that result in erroneous blood-pressure readings. The labels could then be refined so that the classifier recognizes the specific types of lower-arm movements that are medically relevant.

The sleeve used in our experiments could be redesigned to replace the steel rings with conductive fabric to increase patient comfort and usability in the field.

Acknowledgments

This research results from a research program at the Institute for Security, Technology, and Society at Dartmouth College, supported by the National Science Foundation under award numbers 1016823 and 1143548. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We thank Shrirang Mare and Dr. Ashok Chandrashekar, both from Dartmouth College, for their support in the formulation of the methods presented here. We are grateful to the anonymous reviewers for their feedback.

References

[1] Y. Anokwa, C. Hartung, W. Brunette, G. Borriello, and A. Lerer, "Open-source Data Collection in the developing world," *Computer*, vol. 42, no. 10, pp. 97–99, Oct. 2009. DOI 10.1109/MC.2009.328

[2] "The Sana Approach," 2012, http://sana.mit.edu/mission/approach/.

[3] Q. Pan, P. Yang, R. Zhang, C. Lin, S. Gong, L. Li, J. Yan, and G. Ning, "A mobile health system design for home and community use," in *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, January 2012. DOI 10.1109/BHI.2012.6211521

[4] A. Prasad, R. Peterson, S. Mare, J. Sorber, K. Paul, and D. Kotz, "Provenance framework for mHealth," in *Workshop on Networked Healthcare Systems*. IEEE Computer Society Press, January 2013, pp. 1–6. DOI 10.1109/COMSNETS.2013.6465599

[5] T. G. Pickering, J. E. Hall, L. J. Appel, B. E. Falkner, J. Graves, M. N. Hill, D. W. Jones, T. Kurtz, S. G. Sheps, and E. J. Roccella, "Recommendations for blood pressure measurement in humans and experimental animals: A statement for professionals from the subcommittee of Professional and Public Education of the American Heart Association Council on high blood pressure research," *Hypertension*, vol. 45, no. 1, pp. 142–161, 2005. DOI 10.1161/01.HYP.0000150859.47929.8e

[6] I. Villegas, I. C. Arias, A. Botero, and A. Escobar, "Evaluation of the Technique Used by Health-Care Workers for Taking Blood Pressure," *American Heart Association Journals*, vol. 26, 1995. DOI 10.1161/01.hyp.26.6.1204

[7] "What is EMOCHA?" http://main.ccghe.net/content/emocha, online; accessed 01-Feb-2013.

[8] D. Estrin and I. Sim, "Open mHealth Architecture: An Engine for Health Care Innovation," *Science*, vol. 330, no. 6005, pp. 759–760, 2010. DOI 10.1126/science.1196187

[9] "Tablet market in the Middle East and Africa grows 184 per cent in 2013," http://www.idc.com/getdoc.jsp?containerId=prAE24144713, online; accessed 11-Nov-2013.

[10] E. N. Smithayer, "Sensor-based system for verifying blood-pressure measurement position," Dartmouth College, Computer Science, Hanover, NH, Tech. Rep. TR2012-720, January 2012. Available online: http://www.cs.dartmouth.edu/reports/abstracts/TR2012-720/

[11] M. J. Hiroyuki Ota and K. J. Kenji Taniguchi, "Electronic blood pressure meter with posture detector," US Patent US 5 778 879, 1998. Available online: http://www.patentlens.net/patentlens/patent/US_5778879/en/

[12] K. D. Dirk Freund, O. D. Martin Giersiepen, N. D. Brigitte Harttmann, K. D. Ulrich Heck, K. D. Stefan Hollinger, S. D. Frank Kressmann, D. D. Gerrit Ronneberg, and K. D. Fred Schnak, "Blood pressure monitoring device with inclination sensor," US Patent 6 712 769, 2004. Available online: http://www.patentlens.net/patentlens/patent/US_6712769/en/